UNIVERSIDAD POLITÉCNICA DE MADRID

FACULTAD DE INFORMÁTICA



Arquitectura Híbrida para Sistemas Cognitivos Artificiales con Comportamiento Emergente, Adaptativo y Auto-organizado

Tesis Doctoral

Óscar Javier Romero López

Ingeniero de Sistemas

Madrid, Marzo de 2.011

Tesis Doctoral

Arquitectura Híbrida para Sistemas Cognitivos Artificiales con Comportamiento Emergente, Adaptativo y Auto-organizado

DEPARTAMENTO DE LENGUAJES Y SISTEMAS INFORMÁTICOS
E INGENIERÍA DE SOFTWARE

presentada en la

FACULTAD DE INFORMÁTICA

de la

UNIVERSIDAD POLITÉCNICA DE MADRID

como requisito para la obtención del

GRADO DE DOCTOR EN INFORMÁTICA

Autor: **Óscar Javier Romero López**
Ingeniero de Sistemas por la
Universidad Distrital de Colombia

Directora: **Angélica de Antonio Jiménez**
Doctora en Informática por la
Universidad Politécnica de Madrid

Madrid, Marzo de 2.011

A Celia, Diego y Sara

Agradecimientos

Inicialmente, me gustaría agradecer la colaboración financiera de aquellas instituciones que creyeron que mis ideas podrían significar un aporte interesante en el campo de las ciencias tecnológicas y la ingeniería. En primer lugar, agradezco la beca otorgada por el Programa Alβan (Programa de Becas de Alto Nivel de la Unión Europea para América Latina, beca n^{o} E05D056455CO), el cual sufragó parcialmente los costes relacionados con los tres primeros años de desarrollo de mi doctorado. En segundo lugar, quiero agradecer el apoyo de COLFUTURO (Programa Colombiano de Crédito-Beca para Posgrados en el exterior) quien cubrió, a través de un préstamo educativo, los costes complementarios asociados al desarrollo de los dos primeros años de mi doctorado. Por último, quiero agradecer la ayuda financiera del Consejo Social de la Universidad Politécnica de Madrid, mediante la cual realicé una estancia investigativa de seis meses en el Imperial College London durante mi cuarto año de doctorado.

Por otro lado, quiero expresar mi más sincero agradecimiento a aquellas personas que influyeron de forma positiva y directa en el desarrollo de la presente tesis. Primero que todo, agradecer a mi tutora, Ph.D. Angélica de Antonio, por sus comentarios críticos y muy objetivos que ayudaron a reencaminar muchas de las ideas que inicialmente no encajaban armónicamente entre sí. Agradezco también toda la confianza que depositó en mi investigación y en mi capacidad para trabajar autónomamente en ella. Quiero además reconocer la orientación técnica de Ph.D. Ricardo Imbert, miembro del laboratorio Decoroso Crespo, cuya tesis sirvió de inspiración en las etapas iniciales de mi investigación. Quiero dar las gracias también a todos los integrantes del laboratorio de robótica del Imperial College London, grupo de investigación en Sistemas Inteligentes y Redes (*Intelligent Systems and Networks*); a su director, Ph.D. Yiannis Demiris, quien me acogió como uno más de sus estudiantes de doctorado; y en en especial, a Murilo Fernandes, quien dedicó parte de su tiempo en explicarme las generalidades del funcionamiento de la plataforma robótica sobre la cual realice la fase de experimentación de mi tesis.

Por último, aunque no menos importante, quiero agradecer a todas aquellas personas que influyeron indirectamente en el desarrollo de esta investigación. Sin ellas, este trabajo no podría haber sido posible. Gracias a Celia, Diego, Sara y Marco, por escucharme hablar largas horas sobre un tema que es complejo de entender, aunque apasionante de descubrir. Por sus consejos, por su paciencia, por sus voces de aliento, por su comprensión, y por su amor... gracias mil !!!

Resumen

En el campo de estudio de los Agentes Autónomos Inteligentes, uno de los retos más complejos de lograr es la simulación de los procesos cognitivos. Generalmente, la meta de integrar diversas micro-teorías en un solo modelo cognitivo que sea capaz de procesar ingentes cantidades de variables, las cuales cambian continuamente con una complejidad incremental, resulta ser una tarea intrincada cuyos frutos no siempre suelen reflejar las expectativas iniciales de sus diseñadores.

En el área de la *Cognición Artificial* es posible distinguir tres grandes aproximaciones: la Cognitivista, basada en sistemas de representación simbólica y mecanismos de aprendizaje explícito; la Emergente, fundamentada en sistemas de representación subsimbólica y procesos cognitivos de naturaleza implícita; y la Híbrida, que busca integrar lo mejor de cada aproximación. Esta última, a pesar de ser un enfoque conciliador que pretende compensar las debilidades y potenciar las fortalezas de cada aproximación, es un área de investigación muy reciente cuyos resultados son aún incipientes.

En la presente tesis, se propone un modelo cognitivo híbrido y original, denominado METÁFORA, cuyas aportaciones principales son: (1) la integración de teorías multidisciplinares – psicocognitivas, neurológicas, biológicas, de inteligencia artificial, etc. – sobre la cognición; (2) un diseño arquitectónico basado en bloques de construcción cibernéticos, que adapta lo mejor de los modelos cognitivos más representativos – diseño por capas, modular, multi-componente, descentralizado, y distribuido; (3) la fusión de los principios propuestos tanto por la aproximación cognitivista como por la aproximación emergente, y la definición de los mecanismos de integración y sincronización necesarios para comunicar estos dos niveles; y (4) un diseño innovador que articula tres niveles de auto-organización inspirados en principios de la biología evolutiva y del desarrollo: la epigénesis, la ontogénesis, y la filogénesis, a partir de los cuales emergen las diversas estructuras cognitivas de la arquitectura.

En el nivel epigenético se define una capa cognitiva controlada por dos clases de máquinas autopoiéticas (i.e., los Sistemas Inmunes Artificiales y las Redes Neuronales Artificiales), cuya responsabilidad es crear las estructuras cognitivas de bajo nivel y dotar a la arquitectura con un grado significativo de plasticidad. En el nivel ontogenético, por un lado, se precisa un mecanismo atencional adaptativo el cual es capaz de orquestar los diversos comportamientos y procesos conscientes y subconscientes del agente, empleando para ello un modelo evolutivo de dispersión de energía de activación y, por otro, se define un mecanismo que evoluciona las estructuras cognitivas de orden superior. Por último, en el nivel filogenético, se propone un mecanismo basado en la ingeniería memética, encargado de co-evolucionar y trasmitir las estructuras cognitivas a través de la sociedad de agentes.

Abstract

In the field of study of Autonomous Intelligent Systems, one of the most complex challenge to achieve is the cognitive processes simulation. Typically, integrating different kinds of micro-theories of cognition into a unique cognitive model that is able to process a huge amount of variables, which are changing continuously with increasing complexity, is a quite intricate task whose results do not use to reflect the initial expectations of its designers.

In the *Artificial Cognition* field there are three main paradigms: Cognitivist approach, based on symbolic systems and explicit learning mechanisms; Emergent approach, ground-ed in sub-symbolic systems and implicit cognitive processes; and Hybrid approach, which integrates the best of both approaches. The latter, in spite of being a conciliatory outlook that intends both to compensate the weaknesses and to enhance the strengths of each approach, is a very recent research field whose results are still incipient.

In this thesis, it is proposed a hybrid and original cognitive model, called METÁFORA, whose main contributions are: (1) the integration of multi-disciplinary theories about cognition – cognitive psychology, neurology, biology, artificial intelligence, etc.; (2) an architectonic design based on cybernetic building blocks, which adapts the best of the most representative cognitive models – a layered, modular, multi-component, decentralized and distributed design; (3) the fusion of the main features proposed by both cognitivist and emergent approaches, and the definition of the integration and synchronization mechanisms necessary to communicate these two levels; and (4) an innovative design which articulates three levels of self-organization, inspired on principles about evolutionary and developmental bio-logy from which diverse cognitive structures can emerge: epigenesis, ontogenesis, and phylogenesis.

At the epigenetic level of the architecture, there is a micro-level cognitive layer controlled by two kind of autopoietic machines (i.e., Artificial Immune Systems and Artificial Neural Networks), whose responsibility is building the low-level cognitive structures and giving the architecture to a significant degree of plasticity. At the ontogenetic level, on the one hand, an adaptive attentional mechanism which is able to orchestrate all the behaviors and conscious and unconscious processes of the agent, is proposed and, on the other hand, a mechanism to evolve high-order cognitive structures is defined. Finally, at the phylogenetic level, a mechanism based on memetic engineering is proposed, which is in charge of co-evolving and transmitting the cognitive structures through the agent society.

Abreviaturas

AE	Algoritmo Evolutivo
AEX	Árbol de Expresión
AG	Algoritmos Genéticos
ANB	Activación de Nivel Base
AM	Algoritmos Meméticos
ATG	Área de Trabajo Global
AV	Agregación por Votos
CAS	Sistema Complejo Adaptativo
CHC	Cross generation elitist selection, Heterogeneous recombination, and Cataclism mutation
CN	Consecuencias Negativas
CP	Consecuencias Positivas
CP	Combinación de Pesos (según el contexto)
EOF	Epigenético, Ontogenético, Filogenético
EP	Emparejamiento Positivo
EN	Emparejamiento Negativo
EM	Estructura de Metas
FDA	Funciones Definidas Automáticamente
GI	Ganancia de Información
GFA	Gestor de Foco de Atención
HTN	Redes Jerárquicas de Tareas (Hierarchical Task Networks)
IA	Inteligencia Artificial
MA	Máquina Autopoiética
MAT	Máquina de Atención
MD	Máquina Darwiniana
ME	Memoria Episódica
MEA	Memoria Episódica Abstracta
MLA	Marcos de Lectura Abiertos
MS	Memoria Semántica
MT	Memoria de Trabajo
NC	Nivel Cognitivista
NE	Nivel Emergente
NFL	Teorema de No-Free-Lunch
PAA	Proceso de Aprendizaje Anticipatorio
PEA	Proceso Evolutivo Artificial
PEG	Programación de Expresiones Genéticas
PG	Programación Genética
PMR	Perceptrón Multicapa con Retropropagación

RC	Red de Comportamientos
RNA	Red Neuronal Artificial
RMA	Red de Memoria Asociativa
RTJ	Redes de Tareas Jerárquicas
RFE	Red de Frecuencia de Estado
RES	Regla de Extracción Sub-simbólica
RP	Regla Pre-determinada
SA	Sistema Clasificador Anticipatorio
SC	Sistema Clasificador de Aprendizaje
SCA	Sistema Cognitivo Artificial
SI	Secuencia de Inserción (PEG)
SIA	Sistema Inmune Artificial
SPA	Sistema de Prospección y Anticipación
STI	Sistema Tutor Inteligente
SNC	Sistema Nervioso Central
SNS	Sistema Nervioso Sintético
SMA	Sistema Multi-Agente
TR	Tiempo de Respuesta

Índice general

I Concepción de la Idea **1**

1. Introducción **3**

1.1. Motivación 3

1.2. Naturaleza del Problema 5

1.3. Estructura del Documento 8

2. Fundamentos Teóricos Preliminares **11**

2.1. Modelos Biológicos 11

2.1.1. Autopoiesis 11

2.1.2. Epigénesis, Ontogénesis y Filogénesis 12

2.1.3. Sistema Inmunológico Natural 14

2.2. Técnicas Computacionales 15

2.2.1. Inteligencia Computacional 15

2.2.2. Agentes Inteligentes 16

2.2.3. Redes Neuronales Artificiales 18

2.2.4. Proceso Evolutivo Artificial 20

2.2.5. Redes de Comportamientos 31

3. Estado del Arte **33**

3.1. Paradigmas sobre Cognición Artificial 33

3.1.1. La Aproximación Cognitivista 35

3.1.2. La Aproximación Emergente 37

3.1.3. La Aproximación Híbrida 39
3.2. Arquitecturas Cognitivas Artificiales 40
3.2.1. Arquitecturas de Aproximación Cognitivista 41
3.2.2. Arquitecturas de Aproximación Emergente 65
3.2.3. Arquitecturas de Aproximación Híbrida 71

4. Naturaleza de la Solución **87**
4.1. Introducción 87
4.2. Un modelo Híbrido de Cognición Artificial 89
4.3. Propiedades de Modularidad, Especialización y Plasticidad 91
4.4. La clave está en el Comportamiento Emergente 93
4.4.1. Fenomenología de la Persistencia 94
4.4.2. Auto-organización, Autopoiesis y Complejidad 95
4.4.3. Comportamiento Adaptativo Multinivel 102
4.5. Bloques de construcción 117
4.6. Una Arquitectura Dirigida por Metáforas 123
4.6.1. Metáforas sobre la Consciencia 126
4.7. Evolución de la Arquitectura Propuesta 128
4.8. Metodología de Diseño 130

5. Hipótesis y requisitos del Sistema **133**
5.1. Relativos al Modelo Cognitivo Integrado 133
5.2. Relativos a las propiedades de Modularidad y Especialización 135
5.3. Relativos a la propiedad de Persistencia del Comportamiento Emergente . 136
5.4. Relativos a las características de auto-organización y paralelismo del Comportamiento Emergente 136
5.5. Relativos al diseño de Máquinas Autopoiéticas 137
5.6. Relativos al diseño de la arquitectura como un Sistema Complejo Adaptativo 137
5.7. Relativos a los bucles de Retroalimentación de la Arquitectura 138
5.8. Relativos a las Máquinas Darwinianas 139
5.9. Relativos a la simulación de la Filogénesis, Ontogénesis y Epigénesis . . 140
5.10. Relativos al diseño de los Bloques de Construcción 142

5.11. Relativos al empleo de Metáforas para el modelado de la Arquitectura . . 142
5.12. Aportaciones Científicas y Tecnológicas 143

II Definición de la Arquitectura 145

6. Diseño Conceptual de la Arquitectura Cognitiva 147
6.1. Introducción 147
6.2. Arquitectura Modular de Alto Nivel 147
6.3. Substrato Computacional de Inspiración Biológica 154
6.3.1. Niveles Jerárquicos Auto-organizativos de la Arquitectura 154
6.3.2. Niveles Adaptativos de la Arquitectura 164
6.3.3. Dinámicas del Sistema Nervioso Sintético 177
6.4. Aproximación Cibernética al Diseño de la Arquitectura 180
6.4.1. Diseño de bloques de construcción para el SIA 181
6.4.2. Diseño de bloques de construcción para la RNA 184
6.4.3. Diseño de bloques de construcción para las RC 186
6.4.4. Diseño de bloques de construcción para el SPA 188
6.4.5. Diseño de bloques de construcción para el SMA 191
6.5. Ciclo Cognitivo de Alto Nivel 192
6.6. Resumen 200
6.7. Discusión 201

7. Toma de Decisiones Guiada por Comportamientos 205
7.1. Inspiración del Modelo Comportamental 205
7.2. Estructura de un Comportamiento 208
7.3. Módulo Conexionista (PMR) 210
7.3.1. Representación 211
7.3.2. Aprendizaje 218
7.3.3. Máquina de Estados para la red PMR 223
7.4. Módulo Inmunológico (SIA) 224
7.4.1. Arquitectura Híbrida del SIA 225
7.4.2. Algoritmo de Asignación de Recompensas 228

7.4.3. Proceso Epigenético para el Sistema Inmune Artificial 231
7.4.4. Modelo de Interacción 247
7.5. Módulo de Producciones 247
7.5.1. Representación 248
7.5.2. Reglas de Extracción Sub-simbólicas (RES) 252
7.5.3. Reglas Predeterminadas (RP) 259
7.5.4. Soporte, Utilidad y Activación de Nivel Base de las Reglas 260
7.5.5. Mecanismo de Razonamiento 263
7.5.6. Modelo de Interacción 265
7.6. Conversión Representacional 266
7.7. Mecanismos de Modulación e Integración de Acciones 272
7.7.1. Selección Estocástica 273
7.7.2. Combinación Basada en Pesos 275
7.7.3. Agregación por Votación 278
7.8. Aprendizaje Vertical 283
7.9. Discusión 285

8. Ejecución de los Procesos Subconscientes 291

8.1. Construcción de Perceptos 291
8.2. Recuperación y Almacenamiento del Conocimiento Declarativo 294
8.2.1. Inspiración del Modelo Declarativo 294
8.2.2. Conocimiento Semántico 295
8.2.3. Conocimiento Autobiográfico 301
8.2.4. Integración de los dos Niveles de Representación 307
8.2.5. Aprendizaje Vertical 308
8.3. Activación de Comportamientos Automáticos 308
8.4. Regulación Homeostática de los Motivadores 309
8.4.1. Inspiración del Modelo basado en Motivadores 309
8.4.2. Mecanismo de Regulación de un Motivador 310
8.5. Generación de la Estructura de Metas 315
8.5.1. Inspiración del Modelo de Metas 315
8.5.2. La Estructura de Metas 316

8.6. Generación del Estado Anticipado 318
8.6.1. Inspiración del Modelo Anticipatorio 318
8.6.2. Sistema Clasificador Anticipatorio 319
8.6.3. Proceso Epigenético para el Mecanismo de Predicción 322
8.7. Administración de la Memoria de Trabajo 336
8.7.1. Inspiración del Modelo 336
8.7.2. Modelo Computacional de la MT 338
8.8. Discusión 341

9. Orquestación Adaptativa de la Experiencia Consciente 345

9.1. Inspiración del Modelo de Consciencia Sintética 346
9.2. Modulación Adaptativa del Foco de Atención 350
9.2.1. Descripción de la Maquinaria Atencional 350
9.2.2. Mecanismo para la Orientación del Foco de Atención 353
9.2.3. Modelo Matemático Subyacente a la Dispersión de Activación . . 358
9.2.4. Caso de Estudio: Orquestación Adaptativa de Comportamientos en un Animat 361
9.2.5. Discusión: Fortalezas y Debilidades del Modelo de Dispersión de Activación 367
9.3. Ontogénesis del Modelo de Consciencia Sintética 375
9.3.1. Máquina Darwiniana para el Proceso Ontogenético 376
9.3.2. Selección de la Máquina de Atención Apropiada 398
9.4. Modelo Dinámico 413
9.5. Discusión 415

10. Proceso de Meta-Cognición 421

10.1. Motivación 421
10.2. Control y Regulación Homeostática: Retroalimentación 424
10.3. Confección y Validación de Planes 428
10.3.1. Descomposición Jerárquica de Tareas 431
10.3.2. Máquina Darwiniana para la Generación de Planes 433
10.3.3. Mecanismo de Simulación Interna 437
10.3.4. Selección y Ejecución del Mejor Plan 445

10.3.5. Modelo Dinámico 451
10.4. Construcción y Propagación de Unidades de Conocimiento 451
10.4.1. Interacción entre la Epigénesis, la Ontogénesis y la Filogénesis . . 455
10.4.2. Anatomía del Algoritmo Memético 460
10.4.3. Ciclo de Vida de los Agentes Cognitivos 476
10.5. Tiempos de Respuesta 478
10.5.1. Tiempos de Respuesta para el Módulo Comportamental 479
10.5.2. Tiempos de Respuesta para el Módulo Declarativo 482
10.5.3. Tiempos de Respuesta para el Módulo de Consciencia 483
10.5.4. Una Nota General sobre la Estimación de los TRs 484
10.6. Discusión 487

III Resultados **491**

11. Concreción Contextual de la Arquitectura **493**
11.1. Definición de la Plataforma Robótica 493
11.2. Modelo Cliente/Servidor para la concreción contextual 495
11.3. Vista de Implementación y de Despliegue 498
11.4. Construcción de Perceptos y Formateo de Acciones 502
11.4.1. Detección de Obstáculos a través del Proxy Sonar 502
11.4.2. Detección de Objetos a través de los Proxies Laser y Fiducial . . 506
11.4.3. Manipulación de la Pinza a través del Proxy Gripper 510
11.4.4. Determinación del Desplazamiento y Localización del Robot a través del Proxy Position2dProxy 513

12. Evaluación de la Arquitectura Cognitiva **517**
12.1. Motivación y Alcance de la Evaluación 517
12.2. Marco Experimental 519
12.2.1. Selección de Métodos de Análisis Estadístico 519
12.2.2. Caracterización de los Experimentos 524
12.3. Experimentos para el Nivel Epigenético 527
12.3.1. Experimento EX01 527

12.3.2. Experimento EX02 538
12.3.3. Experimento EX03 545
12.3.4. Experimento EX04 561
12.4. Experimentos para el Nivel Ontogenético 567
12.4.1. Experimento EX05 568
12.4.2. Experimento EX06 575
12.4.3. Experimento EX07 582
12.4.4. Experimento EX08 586
12.4.5. Experimento EX09 593
12.4.6. Experimento EX10 597
12.4.7. Experimento EX11 606
12.5. Experimentos para el Nivel Filogenético 608
12.5.1. Experimento EX12 608
12.5.2. Experimento EX13 610
12.5.3. Experimento EX14 613

13. Resultados y Conclusiones **615**
13.1. Estudio Comparativo 615
13.2. Evaluación de requisitos para sistemas inteligentes 622
13.3. Contribuciones originales relativas a la propuesta arquitectónica 627

14. Trabajo Futuro **631**

IV Apéndices **641**

16. Análisis Comparativo **643**
16.1. Estudio de las Características Arquitectónicas Generales 644
16.2. Análisis por Tipo de Arquitectura 646
16.3. Análisis por Propiedades del Ambiente Percibido 649
16.4. Análisis de las Capacidades del Agente 651
16.5. Análisis de las Propiedades Cognitivas del Agente 659
16.6. Resultados Generales del Análisis Comparativo 665

17. Histogramas de Frecuencia y Gráficos Q-Q **667**

V Bibliografía **673**

Bibliografía **675**

Índice de figuras

2.1. Estructura de una neurona de procesamiento 19
2.2. Perceptrón Multicapa. 20
2.3. Esquema de interacción del agente con el medio ambiente 21
2.4. Estructura del medio ambiente 21
2.5. Interacción del medio ambiente con el agente 22

3.1. Arquitectura SOAR 42
3.2. Arquitectura EPIC 44
3.3. Arquitectura ACT-R 45
3.4. Arquitectura ICARUS 47
3.5. Arquitectura BDI 50
3.6. Arquitectura Atlantis 51
3.7. Arquitectura COGNITIVA 53
3.8. Arquitectura DCA 55
3.9. Arquitectura ERE 56
3.10. Arquitectura HOMER 57
3.11. Arquitectura MAX 58
3.12. Arquitectura PRODIGY 60
3.13. Arquitectura RALPH-MEA 62
3.14. Arquitectura Theo-Agent 64
3.15. Arquitectura de Subsumción 66
3.16. Arquitectura de Área de Trabajo Global 67

3.17. Arquitectura SASE 69
3.18. Arquitectura HUMANOID 72
3.19. Arquitectura KISMET 76
3.20. Arquitectura CLARION 78
3.21. Arquitectura LIDA 79
3.22. Arquitectura 4CAPS 81
3.23. Arquitectura Novamente 83
3.24. Arquitectura DUAL 84

4.1. Sistema Complejo Adaptativo 101
4.2. Resonancia de pautas espaciotemporales 111
4.3. Sistema Autopoiético 113
4.4. Co-desarrollo en Sistemas Autopoiéticos 115
4.5. Bloques de Construcción de primer orden 123
4.6. Proceso Unificado 131

6.1. Representación caricaturizada de la Arquitectura Modular 148
6.2. Arquitectura Modular 149
6.3. Máquinas Autopoiéticas de Primer Orden 157
6.4. Máquinas Autopoiéticas de Segundo Orden 159
6.5. Máquinas Autopoiéticas de Tercer Orden 160
6.6. Sistemas Autónomos 162
6.7. Modelo Ascendente de la Arquitectura 174
6.8. Algoritmo Memético 175
6.9. Esquema de un proceso computacional básico 177
6.10. Diagrama de bloques para el SIA 182
6.11. Diagrama de bloques para la RNA 185
6.12. Diagrama de bloques para la RC 187
6.13. Diagrama de bloques para el SPA 189
6.14. Diagrama de bloques para el SMA 192
6.15. Ciclo Cognitivo de Alto Nivel. 195

7.1. Representación de una red PMR. 212

7.2. Diagrama de Transición de Estados para la red PMR. 223
7.3. Arquitectura del Sistema Inmune Artificial. 226
7.4. Representación de Anticuerpos y Antígenos. 227
7.5. Medida de la Afinidad en SIA 232
7.6. Dinámica de selección Clonal 232
7.7. Dinámica de la Maduración de Aptitud 233
7.8. Cruce HUX. 237
7.9. Mecanismo de Reiniciliazación en CHC. 238
7.10. Activación en cadena de múltiples Anticuerpos. 240
7.11. Ejemplo de una estructura jerárquica de Contraejemplos activos. 246
7.12. Diagrama de secuencia para el SIA (sin refuerzo) 248
7.13. Diagrama de secuencia 2 para el SIA (con refuerzo) 249
7.14. Correspondencia entre representaciones simbólicas y sub-simbólicas. . . 250
7.15. Extracción de RES 255
7.16. Ejemplo de Extracción, Especialización y Generalización de Reglas Sub-simbólicas. 257
7.17. Ejemplo del comportamiento de la ANB para un chunk o una regla. . . . 264
7.18. Modelo de Secuencia para las reglas RES. 267
7.19. Conversión de Reglas Simbólicas $\leftrightarrow$ Sub-simbólicas. 270
7.20. Fachada y Adaptadores para los Componentes del *Comportamiento*. . . . 271
7.21. Seleción Estocástica de Acciones. 274
7.22. Combinación basada en Pesos. 277
7.23. Mecanismo de Agregación 281
7.24. Sistema de Lógica Difusa. 289

8.1. Mecanismo de Razonamiento Semántico. 301
8.2. Construcción de Componentes Episódicos. 304
8.3. Memoria Episódica Abstracta. 305
8.4. Modelo de Regulación Homeostática de un Motivador. 311
8.5. Proceso de Aprendizaje Anticipatorio. 319
8.6. Estructura de un clasificador en el SA. 321
8.7. Sistema Clasificador Anticipatorio. 325

8.8. Procesos de Generalización y Especialización en el SA 332
8.9. Fases de operación del SPA 336
8.10. Diagrama de Interacción para el Sistema Anticipatorio. 337
8.11. Interacción con la Memoria de Trabajo. 339

9.1. Metáfora del Teatro 349
9.2. Estructura de una Máquina de Atención. 351
9.3. Dispersión de Activación en las Redes de Comportamientos del Animat. . 364
9.4. Representación Fenotípica y Genotípica de una RC. 378
9.5. Representación Fenotípica del Comportamiento C1. 380
9.6. Representación Fenotípica de C2 382
9.7. Representación Fenotípica del Comportamiento C2 a partir del MLA 9.4. 384
9.8. Cromosoma Multigénico 386
9.9. Representación Fenotípica de la Expresión 9.13. 394
9.10. Representación Fenotípica de la Expresión 9.14. 395
9.11. Fases de Selección de una MAT. 409
9.12. Diagrama de Actividades para el Proceso de Consciencia Sintética. 414

10.1. Comportamiento de los Bucles de Retroalimentación 428
10.2. Plan Jerárquico 432
10.3. Cromosoma con Gen Homeótico 435
10.4. Actuación Real vs. Actuación Simulada 440
10.5. Ciclo Cognitivo Simplificado 441
10.6. Simulación Interna 442
10.7. Diagrama de Actividades para el Mecanismo de Planificación. 452
10.8. Diagrama de Actividades que muestra el flujo de control entre los procesos deliberativos vs. los procesos reactivos. 453
10.9. Incremento Continuo de Información. 457
10.10Integración EOF 458
10.11Proceso Evolutivo para los diferentes Acervos de Memes. 463
10.12Comportamiento de los operadores genéticos basado en la varianza de la población. 471
10.13Diagrama de Actividades para el Proceso Filogenético. 473

10.14Ciclo de Vida de un Agente Cognitivo 477

11.1. Estructura de Control del Servidor Player. a) Cuando se emplean robots reales. b) Cuando se emplea el simulador Stage. Pueden existir múltiples proxies conectados al servidor en cualquier momento. 496

11.2. Modelo Cliente/Servidor de la Plataforma Robótica Player/Stage. 497

11.3. Diagrama de Componentes para el Modelo Cognitivo. 499

11.4. Diagrama de Despliegue. 500

11.5. Sensores Ultrasonido. a) Disposición angular de los sensores alrededor del robot (enumerados del 0 al 15). b) Ejemplo de las lecturas de los sensores del Sonar en Stage. 503

11.6. Sensores Laser y Fiducial. a) Lecturas del escáner láser y sus posiciones en el robot (181 lecturas). b) Representación de los sensores Laser y Fiducial del robot en Stage. La región azul corresponde al área de detección barrida por el escáner láser, y el cono de percepción al área de detección barrida por los sensores fiducial. 506

11.7. Ubicación relativa de un Objeto con respecto a la orientación del robot. . 508

11.8. Proxy Gripper. a) Ubicación de la pinza en el robot y representación de los haces de luz infrarroja. b) Simulación en Stage del robot levantando una caja. 510

12.1. Flujo de Ejecución de las Pruebas Estadísticas. 525

12.2. Estructura Taxonómica de los Experimentos. 526

12.3. Módulos de METÁFORA activos durante los experimentos del Nivel Epigenético. 527

12.4. Lectura de los elementos en las casillas circundantes al robot. 529

12.5. Escenarios para el Experimento 1. a) Distribución diagonal de cajas. b) Inclusión de Muros en el escenario. c) Distribución de cajas en forma de "V". 530

12.6. Curva de aprendizaje para el SIA (Escenario 1). 532

12.7. Curva de aprendizaje para el SIA - Escenario 2. a) Aprendizaje sin conocimiento previo. b) Aprendizaje a partir del conocimiento previo adquirido en el escenario 1. 533

12.8. Curva de aprendizaje para el SIA - Escenario 3. a) Aprendizaje sin conocimiento previo. b) Aprendizaje a partir del conocimiento previo adquirido en los escenarios 1 y 2. 533

12.9. Comparativo entre las curvas de aprendizaje del SIA 534

12.10Diagrama de Clases para el SIA 536

12.11Reporte JProbe para el SIA 537

12.12Escenario de Aprendizaje 4 (recorrido óptimo: 10 desplazamientos). . . . 539

12.13Curvas de aprendizaje para los sistemas SIA, LCS, XCS, y ACS. 541

12.14Escenarios de prueba para el Experimento EX03. a) Escenario 1. b) Escenario 2. 546

12.15Rutas de Navegación empleando diferentes configuraciones de aprendizaje. CP: Combinación de Pesos. AV: Agregación por Votación. 550

12.16Curvas de aprendizaje para los componentes de ⟨Evadir-Obstáculos⟩ 552

12.17Progresión de los Factores de Peso de cada componente en la prueba de Tolerancia a Fallos. 558

12.18Tiempos de Respuesta para cada componente por separado e integrado . . 559

12.19Tiempos de Respuesta de la salida integrada: Simulación vs. Robots Reales 560

12.20Escenario original para la identificación de objetos. 563

12.21Escenario Modificado 1 para la identificación de objetos. 564

12.22Escenario Modificado 2 para la identificación de objetos. 564

12.23Curvas de aprendizaje para los escenarios original y modificados. 565

12.24Objeto de doce vértices. a) objeto original. b) objeto distorsionado del escenario modificado 1. c) objeto distorsionado del escenario modificado 2. 566

12.25Módulos de METÁFORA activos durante los experimentos del Nivel Ontogenético. 568

12.26Topología de la Red de Comportamientos (RC) objetivo. 570

12.27Escenario para la meta Recolectar-Todas-Cajas 570

12.28Progresión del Mejor Nivel de Aptitud y de la Aptitud Promedio de la Población. 571

12.29Escenarios para el experimento EX06 576

12.30Escenario 1 para el experimento EX06 577

12.31Progresión de los niveles de activación para cada *Comportamiento* en el Escenario 1 577

12.32Escenario 2 para el experimento EX06 579

12.33Progresión de los niveles de activación para cada Comportamiento en el Escenario 2 579

12.34Progresión del Mejor Nivel de Aptitud y de la Aptitud Promedio de la Población - EX06. 580

12.35Escenario para el experimento EX07. 582

12.36Conducta aprendida por el robot - EX07 583

12.37Progresión de los niveles de activación para cada *Comportamiento*. BC-TA: ⟨`Buscar-Caja-Tipo-A`⟩. RCTA: ⟨`Recoger-Caja-Tipo-A`⟩. ACTA: ⟨`Almacenar-Caja-Tipo-A`⟩. EO: ⟨`Evadir-Obstáculos`⟩.584

12.38Progresión del Mejor Nivel de Aptitud y de la Aptitud Promedio de la Población - EX07. 585

12.39Conducta aprendida por el robot Esc1-EX08 587

12.40Progresión de los niveles de activación para cada *Comportamiento* Esc1-EX08 588

12.41Conducta aprendida por el robot Esc2-EX08 589

12.42Progresión de los niveles de activación para cada *Comportamiento* Esc2-EX08 589

12.43Conducta aprendida por el robot Esc3-EX08 590

12.44Progresión de los niveles de activación para cada *Comportamiento* Esc3-EX08 591

12.45Progresión del Mejor Nivel de Aptitud y de la Aptitud Promedio de la Población - EX08. 592

12.46Anomalía de Sussman. a) estado inicial y estado final del apilamiento de bloques. b) La meta M1 impide el logro de la meta M2. c) La meta M2 impide el logro de la meta M1. 593

12.47Progresión de los niveles de activación para cada *Comportamiento*-EX09 595

12.48Progresión de los niveles de activación para cada *Comportamiento*-EX09-1 596

12.49Progresión del Mejor Nivel de Aptitud y de la Aptitud Promedio de la Población - EX09. 596

12.50Escenario para el experimento EX10 599

12.51Árbol de Expresión para el Mejor Plan - EX10 602

12.52Secuencia de tareas del Mejor Plan - EX10 604

12.53Convergencia Evolutiva EX10 605

12.54Tiempos de Respuesta EX11 607

12.55Escenario EX12 609

12.56Curvas de Aprendizaje - EX12 610

12.57Conducta aprendida por los robots - EX13 612

12.58Curvas de Tendencia - EX13 612

12.59Curvas de Tendencia - EX14 614

17.1. Escenario 1 - SIA: Histograma y Gráfico Q-Q 667
17.2. Escenario 1 - LCS: Histograma y Gráfico Q-Q 668
17.3. Escenario 1 - XCS: Histograma y Gráfico Q-Q 668
17.4. Escenario 1 - ACS: Histograma y Gráfico Q-Q 668
17.5. Escenario 2 - SIA: Histograma y Gráfico Q-Q 669
17.6. Escenario 2 - LCS: Histograma y Gráfico Q-Q 669
17.7. Escenario 2 - XCS: Histograma y Gráfico Q-Q 669
17.8. Escenario 2 - ACS: Histograma y Gráfico Q-Q 670
17.9. Escenario 3 - SIA: Histograma y Gráfico Q-Q 670
17.10Escenario 3 - LCS: Histograma y Gráfico Q-Q 670
17.11Escenario 3 - XCS: Histograma y Gráfico Q-Q 671
17.12Escenario 3 - ACS: Histograma y Gráfico Q-Q 671
17.13Escenario 4 - SIA: Histograma y Gráfico Q-Q 671
17.14Escenario 4 - LCS: Histograma y Gráfico Q-Q 672
17.15Escenario 4 - XCS: Histograma y Gráfico Q-Q 672
17.16Escenario 4 - ACS: Histograma y Gráfico Q-Q 672

Índice de Tablas

3.1. Comparación entre los paradigmas de cognición Cognitivista y Emergente 36

6.1. Comparativo entre técnicas bio-inspiradas 153

7.1. Sistema de aspersión 244

7.2. Correspondencia entre los sensores y las acciones que idealmente deberían enviarse al actuador. Nótese que las salidas de los contraejemplos son los inversos de los valores deseados para el actuador. 244

7.3. Representacion Binaria de la *propiedad* simbólica "color". 268

7.4. Acciones para el comportamiento desplazarse 279

7.5. Acciones para el comportamiento evadir obstáculos 280

7.6. Conteo de Votos para cada salida de los Componentes. 281

8.1. Regulación homeostática del motivador Hambre. 314

10.1. Síntesis de las Estrategias de Búsqueda Local y Global empleadas para la Construcción de Conocimiento 459

10.2. Ejemplo de Sustitución por Hacinamiento. 475

11.1. Lectura del Sonar en un instante de tiempo t. 504

11.2. Vector conformado por las lecturas mínimas del Sonar. 504

11.3. Representación Sub-simbólica de los Perceptos correspondientes a la Detección de Obstáculos. 504

11.4. Ejemplo de la información detectada por los sensores fiducial para un objeto identificable. 508

11.5. Percepto sub-simbólico Fiducial 509

11.6. Ejemplo de un percepto sub-simbólico construido a partir de la información detectada por los sensores de la pinza. 512

11.7. Ejemplo de un percepto sub-simbólico construido a partir de la información detectada por el proxy P2d. 514

11.8. Ejemplo de formato de una acción que controla los actuadores de desplazamiento del robot. 516

12.1. Prueba de Normalidad para EX01 531

12.2. Prueba de heterocedasticidad para EX01 531

12.3. Relación de Desempeño del SIA 533

12.4. Relación de Desempeño del SIA con contraejemplos 534

12.5. Error Cuadrático Medio 535

12.6. Configuración paramétrica de los Sistemas de Aprendizaje 540

12.7. Relación de Desempeño del SIA vs. los otros sistemas de aprendizaje. . . 541

12.8. Prueba de Normalidad Shapiro-Wilk SIA 542

12.9. Prueba de Heterocedasticidad Levene SIA 542

12.10Prueba de Friedman SIA 542

12.11Ranking Friedman EX02 543

12.12Pruebas Posteriori EX02 543

12.13Configuración NN EX03 547

12.14Estadísticas EX03 552

12.15Prueba de Normalidad EX03 553

12.16Prueba de Heterocedasticidad EX03 553

12.17Prueba de Friedman EX03 553

12.18Ranking Friedman EX03 553

12.19Pruebas Posteriori EX03 554

12.20Variación de los Factores de Peso durante el aprendizaje 555

12.21Reglas Nuevas construidas por cada componente después del aprendizaje 556

12.22Configuración de Hardware y S.O. 558

12.23Estadísticas para los Tiempos de Respuesta - EX03 560

12.24Reglas de clasificación aprendidas para el Objeto D1 566

12.25Configuración Paramétrica EX05 569

12.26Configuración de Parámetros Globales - EX06 581

12.27Configuración de Parámetros Globales - EX07 585
12.28Configuración de Parámetros Globales - EX08 592
12.29Configuración de Parámetros Globales - EX09 597
12.30Configuración Paramétrica EX10 598
12.31Configuración de Parámetros Globales - EX10 601
12.32Estadísticas para los Tiempos de Respuesta - EX10 607
12.33Configuración Paramétrica del Proceso Filogenético - EX12 609
12.34Estadísticas de las Curvas de Aprendizaje - EX12 610
12.35Estadísticas de las Curvas de Aprendizaje - EX13 611
12.36Estadísticas de las Curvas de Aprendizaje - EX14 614

13.1. Correlación METÁFORAteorías neuro-psicológicas 627

16.1. Comparativo de las Características Arquitectónicas Generales 644
16.2. Frecuencias de las Características Generales 645
16.3. Análisis Comparativo por Tipo de Arquitectura 648
16.4. Frecuencias del Tipo de Arquitectura 649
16.5. Análisis Comparativo por Propiedades del Ambiente 652
16.6. Frecuencias Propiedades del Ambiente 653
16.7. Análisis Comparativo por Capacidades del Agente 657
16.8. Frecuencias de las Capacidades del Agente 658
16.9. Análisis comparativo por Propiedades del Agente 663
16.10Frecuencias de las propiedades del Agente 664

Parte I

Concepción de la Idea

Capítulo 1

Introducción

"The last thing one knows when writing a book is what to put first.".
– Blaise Pascal.

1.1. Motivación

Uno de los campos de investigación más interesante y prometedor que ha venido atrayendo una gran cantidad de público multi-disciplinar, en áreas de estudio tales como la psicología, la neurofisiología, la lingüística, la biología, la física, la filosofía, y la inteligencia artificial, es aquél que se conoce con el nombre de *Ciencias Cognitivas*. Una línea de investigación en este área, cuyo propósito principal es el estudio de la fenomenología de los procesos cognitivos y la manera en que éstos pueden ser modelados computacionalmente, es designada como *Cognición Artificial*.

La ciencia e ingeniería relativa a la construcción de sistemas cognitivos artificiales que intentan emular las capacidades mentales humanas, tiene una larga historia que se remonta a más de 60 años. El término *mental*, por lo menos en el alcance de la presente tesis, no pretende denotar ningún dualismo relacionado con la mente y el cuerpo; por el contrario, el término será empleado para distinguir el *desarrollo mental* del *crecimiento físico* en los organismos inteligentes. Así pues, las facultades mentales suponen todos los aspectos relacionados con el comportamiento robusto, como por ejemplo, la percepción, la actuación, la deliberación, la motivación, el aprendizaje, la planificación, entre muchos otros. Así pues, a lo largo de esta tesis, el término *cognición* será empleado para indicar la manera en que interactúan dichas facultades mentales.

En términos generales, la *cognición* implica la habilidad para comprender cómo podrían ser las cosas, no sólo en el presente sino en algún escenario alternativo o en algún estado futuro, y poder determinar, en consecuencia, cómo debería actuar el sistema. La capacidad de recordar lo que ha sucedido en algún punto en el pasado ayuda a la posterior anticipación de eventos futuros, por tanto, la memoria es un elemento necesario: emplea el pasado para predecir el futuro y luego asimila lo que sucede actualmente con el fin de

adaptar y mejorar la habilidad anticipatoria del sistema en un proceso continuo de percepción-actuación. Se podría decir entonces, que la cognición se "libera" sutilmente del presente a fin de permitir al sistema actuar efectivamente, adaptarse, y mejorar de forma progresiva.

Pero, ¿qué hace que una acción sea la más apropiada para su selección?, ¿qué tipo de comportamiento hace posible la cognición?... estas preguntas despliegan otras dimensiones del problema: ¿qué motiva la cognición?, ¿cómo es guiada la percepción?, ¿cómo deben ser seleccionadas las acciones?, ¿qué se debe aprender y qué se debe olvidar? Un sistema inteligente puede mejorar sus destrezas cognitivas, pero ¿qué se necesita como punto de partida?, ¿qué dirige el proceso de desarrollo de dichas destrezas?, en otras palabras, además de la percepción autónoma, la actuación, la anticipación, la asimilación, y la adaptación, existen otras motivaciones subyacentes al proceso de cognición que deben ser consideradas. Estas motivaciones guían la atención perceptual, la toma de decisiones, y el desarrollo del sistema, lo que resulta a largo plazo en el comportamiento robusto e inteligente que se espera evidenciar en un Sistema Cognitivo Artificial (SCA).

Desde esta perspectiva, un SCA exhibe un comportamiento efectivo a través de facultades tales como la percepción, la actuación, la deliberación, la comunicación, e incluso, a través de interacciones individuales y/o sociales con el entorno. Así pues, el sello distintivo de un sistema cognitivo es su capacidad para funcionar efectivamente en circunstancias para las cuales no fueron diseñados explícitamente. Esto implica, que un sistema cognitivo debe poseer cierto grado de plasticidad y flexibilidad ante las situaciones imprevistas.

Algunos autores en el área de los SCAs han ido incluso un poco más allá. Por ejemplo, Brachman (Brachman, 2002) define un *Agente Cognitivo* como un sistema tal que, además de tener la capacidad para razonar, aprender de sus propias experiencias, mejorar su desempeño con el paso del tiempo, y responder inteligentemente a las situaciones que nunca ha enfrentado antes, sería también capaz de explicar qué es lo que está haciendo y por qué lo está haciendo. Estas habilidades le permitirían identificar los problemas potenciales asociados con la realización de una tarea o determinar cuándo requiere de nueva información a fin de completarla. Hollnagel (Hollnagel y Woods, 1999) sugiere que un SCA debe ser capaz de visualizar un problema desde diferentes perspectivas, y emplear conocimiento sobre sí mismo y sobre el entorno de manera que pueda planificar y modificar sus acciones en base a tal conocimiento. Por consiguiente, la cognición también supone un cierto sentido de auto-reflexión, adicional a las características de adaptación y anticipación.

De esta manera, la cognición artificial puede ser vista como el proceso mediante el cual el sistema logra un comportamiento robusto, adaptativo, anticipatorio, y autónomo. Esta perspectiva refleja la posición de Freeman y Núñez quienes, en su libro *Reclaiming Cognition* (Freeman y Núñez, 1999), reafirman la primacía de la actuación, la intención, y la emoción en el proceso cognitivo. Por otro lado, este punto de vista contrasta con la perspectiva obsoleta de aquellos autores quienes ven la cognición sólo como un sub-sistema del cerebro (e.g., un módulo de la mente), que está relacionado con la planificación y el razonamiento, que actúa sobre las representaciones construidas por el sistema perceptual, y que decide qué acciones deberían ser ejecutadas a continuación.

Una vez definido aquello que se quiere modelar, resulta necesario determinar el cómo. En general, la idea de una arquitectura resulta útil porque permite factorizar los aspectos comunes del amplio rango de comportamientos que caracterizan cierto contenido. Una arquitectura particular, esto es, un conjunto específico de mecanismos y estructuras, representa una teoría de lo qué resulta común entre los elementos de un nivel superior. En este sentido, se puede definir el término *Arquitectura Cognitiva* como una teoría de las estructuras y mecanismos específicos que subyacen la cognición humana. La factorización de qué es común entre los procesos cognitivos, a través del fenómeno explicado por las micro-teorías, parece ser un avance significativo hacia la producción de una teoría unificada de la cognición.

Por tanto, una *arquitectura cognitiva* se puede definir simplemente como una parte de un sistema que provee y administra los recursos primarios de un agente quien exhibe un comportamiento inteligente. Para muchas de las arquitecturas cognitivas, estos recursos definen el substrato sobre el cual un sistema simbólico/físico está construido. Otra definición interesante para el término de *arquitectura cognitiva* es la propuesta por Manuel de la Herrán: *en la ciencia cognitiva, el concepto "Arquitectura" se refiere a la estructura no flexible subyacente al dominio flexible del proceso cognitivo, es decir, a la estructura que soporta los procesos cognitivos superiores (de la Herrán, 2005).*

En este orden de ideas, el diseño eficiente de una arquitectura cognitiva artificial sirve como soporte para la simulación de los procesos cognitivos de orden superior, de cuya interacción puede emerger el comportamiento inteligente. Así como el término cognición, la inteligencia es un concepto que puede generar debate debido a la perspectiva desde la que se le defina. Personalmente, me gusta lo que decía Jean Piaget: *"la inteligencia es lo que uno utiliza cuando no sabe qué hacer"*. Esta definición capta el elemento de novedad, la capacidad de estar a la altura de la situación y de buscar salidas, tan necesaria cuando no existe una *respuesta correcta*, cuando lo más probable es que no baste con actuar como de costumbre, esto es, *improvisación inteligente*. Así pues, la inteligencia tiene que ver con el proceso de improvisar y perfeccionar en la escala temporal del pensamiento y la acción.

El objetivo final que intenta reflejar la presente tesis, es la propuesta de una arquitectura cognitiva artificial que define múltiples niveles auto-organizativos donde operan, a diferentes escalas temporales, diversos procesos evolutivos basados en la teoría del darwinismo universal, los cuales, por un lado, coordinan la construcción y auto-ensamblado de estructuras cognitivas de complejidad variada y, por otro, permiten la emergencia del comportamiento y la improvisación inteligente. Tomando como base esta arquitectura, será posible construir agentes inteligentes con comportamiento robusto, autónomo, y adaptativo... y desde una perspectiva más ambiciosa, aunque fuera del alcance de esta tesis, será posible simular los procesos de creatividad.

1.2. Naturaleza del Problema: ¿por qué se requiere definir otra arquitectura cognitiva?

Existen muchas posiciones sobre la *cognición artificial*, donde cada una adopta una postura significativamente diferente sobre la naturaleza de la cognición, sobre lo que debe

estar en capacidad de hacer un Sistema Cognitivo Artificial (SCA), y sobre aquello que debe ser considerado para analizar y sintetizar su diseño. Sin embargo, entre todas estas posiciones se pueden identificar dos vastas clases: la *Aproximación Cognitivista* basada en sistemas representacionales que procesan información simbólica, y la *Aproximación de los Sistemas Emergentes*, que abarca los *sistemas conexionistas*, los *sistemas dinámicos*, y los *sistemas enactivos*, estos tres últimos basados en mayor o menor proporción en principios de auto-organización. Existe una tercera aproximación, la *Híbrida*, la cual intenta combinar lo mejor de las aproximaciones cognitivista y emergente.

Grosso modo, en la aproximación cognitivista se simulan directamente las características inteligentes que se pretenden conseguir mediante el procesamiento de información simbólica. Por otro lado, la aproximación emergente se orienta a la simulación de los elementos de más bajo nivel que componen o intervienen en los procesos inteligentes, con la esperanza de que a partir de su combinación emerja espontáneamente el comportamiento inteligente. Los sistemas de inspiración emergente, argumentan en contra de los sistemas de inspiración cognitivista quienes ven la cognición como un proceso "simbólico, racional, encapsulado, estructurado y algorítmico", y defienden la posición que trata la cognición como un proceso emergente, auto-organizado, y dinámico (Thelen y Smith, 1994).

Cada uno de estos dos paradigmas ha realizado contribuciones importantes en el modelado computacional de los procesos cognitivos, no obstante, siguen siendo aportes concretos y aislados sobre áreas muy especializadas del proceso de cognición. La aproximación híbrida parece ser entonces una opción conciliadora e integradora entre estos dos enfoques. Sin embargo, debido a que ésta es un área reciente de estudio, los principios, fundamentos e incluso los resultados obtenidos, se encuentran aún en una fase embrionaria de desarrollo, donde todavía siguen sin cubrirse muchos aspectos necesarios para la simulación del proceso cognitivo (como un todo) y para la emergencia del comportamiento inteligente.

Esta situación es la principal justificación que responde a la cuestión planteada por la naturaleza del problema: ¿por qué es necesario definir una aproximación arquitectónica diferente a las ya propuestas? A continuación, se enuncian razones más específicas que ponen de manifiesto la necesidad de proponer una nueva arquitectura cognitiva, necesidad que dirige el propósito de la presente tesis[1]:

- **Integración:** el propósito final de muchas de las aproximaciones de cognición artificial es proponer una *teoría unificada* a través de la cual pueda modelarse el proceso global de cognición; probablemente los diseñadores de estas arquitecturas sientan celos de los físicos, quienes esperan encontrar una solución elegante para sus fórmulas matemáticas, es decir, una solución que cumpla con las propiedades esenciales de ser unificadora y tener la capacidad de explicar una gran cantidad de fenómenos naturales mediante expresiones concisas.

  Sin embargo, el problema esencial de las visiones unificadoras lo expresa de manera apropiada el siguiente refrán: *"no porque tengas un martillo, todos tus problemas serán clavos"*. No es posible atender todos los aspectos relacionados con la

[1] Estas necesidades fueron identificadas a partir del estado del arte revisado al inicio de la investigación, el cual se presenta documentado en el capítulo 3, y cuyo análisis comparativo se presenta en el anexo A

cognición artificial desde un sólo punto de vista, ni desde una sola teoría cognitiva, psicológica, o neurológica, ni empleando un único modelo computacional. Como consecuencia, el diseño de un sistema cognitivo artificial debe implicar la integración de diversas micro-teorías sobre la cognición, propuestas desde diferentes perspectivas (e.g., la neurociencia, la psicología cognitiva, la biología, la inteligencia artificial, etc.), y debe ser una aproximación conciliadora que compense las debilidades y potencie las fortalezas de los diferentes paradigmas sobre cognición artificial.

- **Adaptabilidad:** un sistema cognitivo que no cumpla con ciertos criterios de *adaptabilidad*, no puede considerarse un sistema *autónomo*. Por ejemplo, un agente cognitivo que encapsula una gran cantidad de conocimiento que ha pre-programado un experto humano, podrá desempeñarse eficientemente en la resolución de un conjunto específico de tareas, sin embargo, si el agente es trasladado a otro contexto donde dicho conocimiento inicial ya no es válido, y éste no posee la capacidad para adaptarse a nuevas situaciones (mediante la modificación del conocimiento previo y el aprendizaje de nuevo conocimiento), dependerá siempre de dicha programación inicial y su autonomía estará restringida a ello.

  La adaptabilidad viene dada en una gran variedad de formas, operando sobre diferentes niveles organizativos de la arquitectura, y afectando diversos componentes y procesos de la misma. Sin embargo, con frecuencia, la adaptabilidad es considerada como una característica aislada que deben cumplir ciertos componentes o módulos del sistema, y no como un criterio que subyace al diseño de toda la arquitectura. Por ejemplo, usualmente en los distintos paradigmas de cognición, la adaptabilidad es concebida únicamente como la capacidad de "modificar" y "aprender" nuevas unidades de conocimiento (e.g., aprender nuevas reglas simbólicas en los sistemas de enfoque cognitivista, y modificar los pesos sinápticos como consecuencia de una señal de refuerzo en los sistemas de enfoque emergente).

  No obstante, la adaptabilidad es mucho más que sólo la capacidad de aprender nuevo conocimiento. El *aprendizaje*, visto como la modificación de los estados internos del agente en respuesta a los cambios en el ambiente producidos por la ejecución de las acciones, no es sino uno de los niveles más básicos de adaptabilidad del proceso cognitivo. La capacidad para aprender las relaciones de causa y efecto a fin de anticiparse a los eventos que sucederán en el futuro inmediato, corresponde a un siguiente nivel de adaptabilidad: la capacidad *predictiva*. Si esta capacidad de predecir los eventos futuros puede ser ampliada a escalas temporales más prolongadas, donde sean considerados los efectos de la interacción de redes anidadas de relaciones causales, se habla entonces de la capacidad adaptativa del agente para *planificar*. Estas capacidades que le permiten al agente, no sólo actuar anticipadamente ante los cambios dinámicos del ambiente, sino predecir las necesidades de futuros cambios de dirección y crear planes en función de ello, es lo que se denomina *inteligencia*. El comportamiento inteligente es esencialmente adaptativo, en la medida en que éste representa modos de responder a las demandas de un entorno cambiante.

  El punto que se requiere resaltar acá, es que los diferentes niveles de adaptabilidad no deben ser considerados como características independientes y aisladas en

el diseño de la arquitectura, sino que pertenecen a un todo global e integrado denominado "comportamiento adaptativo" del agente. En términos generales, el comportamiento adaptativo es un tipo de conducta que se emplea para ajustar otro tipo de conducta de nivel inferior, lo que permite al agente convertir un comportamiento disruptivo o caótico en algo constructivo y ordenado a través de los distintos niveles de adaptabilidad.

- **Auto-organización:** la capacidad de organizar la estructura interna de un sistema a través de un proceso incremental que aumenta la complejidad sin ser guiado por ningún factor externo, es una característica típica de los sistemas de enfoque emergente. No obstante, en los sistemas de enfoque cognitivista esta característica está ausente, y en los sistemas de enfoque híbrido es apenas considerada. Similar al problema de la adaptabilidad, la auto-organización es concebida por los distintos sistemas cognitivos artificiales como una característica que opera en un nivel concreto de organización (e.g., las dinámicas de una red neuronal auto-organizada – mapas de Kohonen – o de un atomata celular). No obstante, la auto-organización resulta útil si se considera como un proceso que opera a diferentes escalas, donde a partir de la auto-organización de los elementos de un nivel inferior emerge un nuevo nivel de organización más complejo, y así sucesivamente.

  Existe escasa evidencia (o por lo menos no lo suficientemente divulgada a través de medios científicos) de que las múltiples arquitecturas cognitivas de los diferentes paradigmas de cognición argumenten que los procesos cognitivos de orden superior emergen como consecuencia de la auto-organización incremental de estructuras cognitivas más simples[2]. Por ejemplo, no existe una aproximación tal que proponga que el ensamblaje auto-organizado de estructuras cognitivas, de las más básicas a las mas complejas (e.g., reglas $\rightarrow$ comportamientos $\rightarrow$ esquemas de resolución de problemas $\rightarrow$ planes ...), soporte la emergencia de procesos cognitivos de orden superior, como la planificación, la deliberación, la creatividad, etc.

1.3. Estructura del Documento

La presente tesis se estructura en doce capítulos, además de este primero introductorio y de dos anexos, donde se expone el argumento teórico, el diseño, y la evaluación de la aproximación arquitectónica propuesta.

Capitulo 2: este capítulo, de naturaleza también introductoria, presenta los fundamentos teóricos necesarios para la comprensión de la terminología empleada en los capítulos posteriores.

Capitulo 3: en este capítulo se presenta una revisión del estado del arte sobre los principios básicos que exponen los diferentes paradigmas de cognición artificial, y sobre las arquitecturas cognitivas más relevantes propuestas para cada caso. Este capítulo, junto

[2] Se ha llegado a esta conclusión después de haber realizado una amplia revisión de la literatura relacionada y de un análisis comparativo llevado a cabo sobre las 30 arquitecturas cognitivas más relevantes (citadas en artículos de journals con un alto factor de impacto), como se documenta en el capítulo 3 sobre el estado del arte.

con el anexo A, establecen el estado del conocimiento actual sobre el cual se identificarán las necesidades que atiende la presente tesis.

Capitulo 4: este capítulo expone las ideas fundamentales que subyacen al diseño de la arquitectura cognitiva. Estas ideas se orientan hacia la solución del problema descrito en la introducción, y hacia la compensación de las debilidades comunes de los sistemas cognitivos de enfoque híbrido identificadas en el análisis comparativo del anexo A.

Capitulo 5: las hipótesis y requisitos funcionales sobre los cuales se fundamenta el diseño de la arquitectura propuesta, son sintetizados en este capítulo.

Capitulo 6: en este capítulo se presenta el diseño conceptual de la arquitectura, el cual se divide en tres grandes partes: un diseño modular donde son determinados todos los módulos y componentes que conforman la arquitectura, un diseño de inspiración biológica donde son identificados los niveles auto-organizativos y adaptativos de la aproximación arquitectónica, y un diseño de inspiración cibernética donde son modelados los componentes principales de la arquitectura como ciclos retroalimentados de causa y efecto. Finalmente, se propone un ciclo cognitivo de alto nivel donde es posible apreciar la interacción entre los distintos componentes de la arquitectura.

Capitulo 7: en este capítulo se describen los módulos multi-propósito (y todos sus componentes asociados) encargados de la toma de decisiones del agente: los *Comportamientos*. Concretamente, en este capítulo se describen dos componentes de naturaleza cognitivista (i.e., las reglas pre-determinadas y las reglas de extracción sub-simbólica) y dos componentes de naturaleza emergente (i.e., las Redes Neuronales Artificiales y los Sistemas Inmunes Artificiales). Adicionalmente, se detallan los diversos mecanismos para la modulación e integración de acciones, así como los mecanismos relativos al aprendizaje de nuevas pautas cognitivas.

Capitulo 8: los diversos procesos que actúan de forma sub-consciente soportando el procesamiento consciente del agente, son detallados en este capítulo. Grosso modo, este capítulo describe el mantenimiento de las estructuras y los mecanismos relativos a: el estado pasado, el estado actual, y estado futuro del agente; y la manera en que estos interactúan de forma dinámica en la creación de los contenidos conscientes del sistema cognitivo.

Capitulo 9: este capítulo describe un modelo atencional basado en un mecanismo de dispersión de energía de activación, capaz de dirigir adaptativamente el foco de atención del agente. Se propone además, un modelo evolutivo basado en programación genética, encargado del desarrollo de las estructuras cognitivas de orden superior (i.e., las redes de comportamientos) encargadas de orquestar los procesos conscientes y subconscientes del agente.

Capitulo 10: en este capítulo, de naturaleza miscelánea, son pormenorizados los mecanismos relacionados con el proceso de meta-cognición. En concreto, se describen los mecanismos de regulación de las señales de retroalimentación del sistema, la construcción adaptativa de planes como mecanismo ensamblador de estructuras cognitivas, los procesos de simulación interna, la trasmisión y evolución de unidades de conocimiento (memes), y las restricciones de los tiempos de respuesta del sistema.

Capitulo 11: a pesar de que la arquitectura cognitiva propuesta es de propósito general, en este capítulo se describe una concreción contextual para un ambiente robótico a fin de validar la eficiencia y desempeño de la arquitectura. Así pues, son detallados la plataforma robótica, el modelo de implementación, y la caracterización tanto de las percepciones como de las actuaciones del agente.

Capitulo 12: en este capítulo se presenta la evaluación de la arquitectura empleando para ello la concreción contextual propuesta. Concretamente, se plantean diversos experimentos para cada uno de los niveles organizativos de la arquitectura. Adicionalmente, se describe el análisis estadístico que soporta las conclusiones obtenidas.

Capitulo 13: este capítulo presenta las conclusiones generales sobre la investigación, resumiendo las principales contribuciones realizadas.

Capitulo 14: este capítulo expone las principales líneas futuras de trabajo que derivan de la presente investigación.

Apéndice A: este anexo presenta el análisis comparativo realizado para las 30 arquitecturas cognitivas descritas en el capítulo 3 sobre el estado del arte.

La extensión de la presente tesis se debe, por un lado, a la complejidad misma del tema y, por otro, a que integra una gran variedad de argumentos (algunos de ellos aparentemente inconexos entre sí), con lo cual se requiere definir un marco conceptual que sirva de apoyo para el desarrollo de las ideas. No obstante, si el lector desea simplificar la lectura del libro, lo puede hacer siguiendo el siguiente orden: anexo A $\rightarrow$ capítulo 5 $\rightarrow$ capítulo 6 $\rightarrow$ capítulo 7 $\rightarrow$ capítulo 8 $\rightarrow$ capítulo 9 $\rightarrow$ capítulo 10 $\rightarrow$ capítulo 11 $\rightarrow$ capítulo 12 $\rightarrow$ capítulo 13 $\rightarrow$ capítulo 14. Como se puede observar, la lectura general de la tesis resulta ser claramente articulable, y depende del lector el decidir qué leer con detenimiento o simplemente omitir. Bienvenidos!!

Capítulo 2

Fundamentos Teóricos Preliminares

"Todas las teorías son legítimas y ninguna tiene importancia. Lo que importa es lo que se hace con ellas".
– Jorge Luis Borges.

Como fundamento al diseño de la aproximación que se desarrollará en los capítulos siguientes, se presenta a continuación la terminología necesaria relacionada, por un lado, con los modelos biológicos que sirven de substrato a la propuesta y, por otro, con las técnicas de inteligencia computacional empleadas.

2.1. Modelos Biológicos

2.1.1. Autopoiesis

La *Autopoiesis* (del griego, $\alpha\upsilon\tau o$ auto, "sí mismo"; y $\pi o\iota\eta\sigma\iota\varsigma$, poiesis, "creación" o "producción"), es un neologismo propuesto por los biólogos Humberto Maturana y Francisco Varela (Maturana y Varela, 1980) para designar la organización de los sistemas vivos.

El término Autopoiesis denota los sistemas que presentan una red de procesos u operaciones (que los define como tal y los hace distinguibles de los demás sistemas), mediante la cual pueden crear o destruir elementos del mismo sistema, como respuesta a las perturbaciones del medio. Aunque el sistema cambie estructuralmente, dicha red permanece invariante durante toda su existencia, manteniendo la identidad de éste. Un ejemplo claro de esto son los seres vivos, sistemas autopoiéticos que están vivos sólo mientras permanecen en Autopoiesis.

Aunque un sistema autopoiético se mantenga en desequilibrio puede conservar una "persistencia" estructural absorbiendo la energía de su medio permanentemente. Al igual que la célula y los seres vivos, los sistemas autopoiéticos tienen la capacidad de conservar la unión de sus partes e interactuar entre ellas. Los sistemas autopoiéticos son autónomos,

lo que los hace sistemas cerrados que se auto-regulan continuamente. Otros ejemplos de sistemas autopoiéticos son la "consciencia", un organismo, etc. Estos sistemas emergen de una red de procesos que logran transformar los componentes internos, siendo el mismo sistema quien articula su identidad en relación con el entorno. Así pues, la autopoiesis designa la manera en que los sistemas mantienen su identidad gracias a procesos internos que auto-reproducen sus propios componentes.

Estos sistemas están abiertos a su medio porque intercambian materia y energía, aunque al mismo tiempo se mantienen cerrados operacionalmente pues son sus funciones quienes los distinguen del entorno. No obstante, son "autónomos" en sus operaciones debido a la capacidad que tienen de reaccionar a los estímulos del medio que los rodea.

La célula es un modelo paradigmático de un sistema autopoiético de primer orden, lo cual induce a pensar sobre la existencia de sistemas autopoiéticos de orden superior. Los organismos multi-celulares pueden definirse como sociedades de células diferenciadas por tipo, involucradas en una red compleja de procesos coordinados con un nivel muy sofisticado de organización. Si se considera que las células individuales son los componentes de los sistemas autopoiéticos de segundo orden, resulta evidente que un gran número de estos componentes desaparecen y son renovados constantemente cada día. Este es el caso, en particular, de las células epidérmicas.

Existen algunos autores (Maturana y Varela, 1980; Varela, 1992a; Beer, 2004; Bourgine y Stewart, 2004; Sarmiento, 2009) quienes proponen que la *cognición*, la *consciencia individual*, y el *comportamiento adaptativo*, desde la perspectiva que sugiere que los organismos son sistemas que auto-producen y auto-regulan la organización interna de sus componentes como resultado de las interacciones que tienen con su entorno, pueden ser considerados sistemas autopoiéticos de tercer orden. Asimismo, es posible pensar en sistemas autopoiéticos de orden superior: la sociedad, la ecosfera, etc. como lo menciona Bourgine (Bourgine y Stewart, 2004).

2.1.2. Epigénesis, Ontogénesis y Filogénesis

Epigénesis: en Biología, la *Epigénesis* (del griego *epi*: por encima de y *genesis*: origen, generación, procesos de formación) es el proceso de desarrollo de un individuo, a través del cual su estructura se diferencia y hace más compleja; el caso paradigmático es el del crecimiento, en el que a partir de un cigoto se desarrolla una compleja estructura celular y orgánica. Por extensión, en teoría de sistemas se incluyen los mecanismos de *aprendizaje* que permiten a un determinado individuo modificar ciertos aspectos de su estructura interna o externa como resultado de la interacción con su entorno inmediato (Jaenisch y Bird, 2003).

Por tanto, la Epigénesis representa el proceso de "sintonización" final a través del cual cada individuo se adapta de forma eficiente a su entorno (mediante diversos mecanismos de aprendizaje) partiendo del conjunto de capacidades contenidas en su código genético. Los ejemplos más evidentes de sistemas con capacidad de *aprendizaje* según la teoría epigenética, son el sistema nervioso central y el sistema inmunitario. En el caso del sistema nervioso central esta capacidad de aprendizaje (dada por la gran plasticidad

neuronal) resulta de vital importancia, pues el número estimado de conexiones sinápticas en un cerebro humano supera con creces el número de nucleótidos contenidos en el genoma humano (Maturana y Varela, 1987).

Ontogénesis: la *Ontogénesis* (del griego *ontos*: ser y *genesis*: proceso de formación), es la historia del cambio estructural de una unidad sin que ésta pierda su organización. Este continuo cambio estructural se da en la unidad, en cada momento, o como un cambio desencadenado por interacciones provenientes del medio donde se encuentre o como resultado de su dinámica interna. El *desarrollo* animal u ontogenia cumple dos funciones principales (Maturana y Varela, 1987): (1) Genera diversidad celular (diferenciación) a partir del huevo fecundado (cigoto) y organiza los diversos tipos celulares en tejidos y órganos (morfogénesis y crecimiento), y (2) asegura la continuidad de la vida de una generación a la siguiente (reproducción).

Filogénesis: la *Filogénesis* (del griego *phyle/phylon*: tribu o raza, y *genesis*: proceso de formación) es la parte de la biología que estudia la evolución de las especies de forma global, en contraposición a la ontogenia, que estudia la evolución del individuo (Cavalli-Sforza, 1967). Podría decirse que la Filogénesis es la sucesión de formas orgánicas emparentadas secuencialmente por relaciones reproductivas y que sus cambios constituyen el cambio filogenético o evolutivo. La filogénesis, engloba todos aquellos mecanismos que, dirigidos por la selección natural, han dado lugar a la gran variedad de especies que se pueden observar actualmente. Los mecanismos evolutivos operan a nivel de poblaciones, y su resultado principal es la obtención de un código genético que permita a los individuos de una población concreta adaptarse de forma óptima al entorno en el cual llevan a cabo sus funciones.

En el proceso evolutivo de la naturaleza viva, la filogenia y la ontogenia se hallan indisolublemente ligadas, esto es, se condicionan recíprocamente (ley biogenética): la ontogenia es el resultado del desarrollo histórico, es decir, es un resultado de la filogenia; por otra parte, la filogenia se forma sobre la base de los cambios individuales, es decir, sobre la base de la ontogenia. En esta interacción se refleja la unidad de la parte (el individuo) con el todo (el género), de lo singular con lo general; la dialéctica del proceso, en forma de espiral de desarrollo, en cada una de cuyas etapas se reproducen, en forma superada, los saltos cualitativos realizados en las etapas precedentes. Darwin expuso la idea de la unidad entre filogenia y ontogenia, idea que fue desarrollada, entre otros, por Haeckel y Michurin (Haeckel, 1883).

Así, la epigénesis y la ontogénesis se consideran generalmente procesos generales de desarrollo hacia un estado adulto, mediante los cuales se alcanzan ciertas formas estructurales que le permiten al organismo desempeñar ciertas funciones en concordancia con el plan innato que lo delimita con respecto al medio circundante. Por otra parte, se considera la filogénesis como una historia de transformaciones adaptativas a través de procesos reproductivos, tendiente a llevar a cabo el plan de la especie con una total subordinación del individuo a ese fin (Varela y Maturana, 1973).

Estos tres modelos biológicos juegan también un rol importante en el desarrollo cognitivo tanto de individuos como de sociedades. Biológicamente, la interacción de los tres permite la creación de nuevos ensamblados de circuitos neuronales, el refinamiento de destrezas cognitivas, y la generación de conductas innatas y aprendidas. Según Maturana

(Maturana y Varela, 1987), el ámbito de conductas posibles de un organismo está determinado por su estructura. Cuando se desarrollan estructuras cognitivas independientemente de su historia, se dice que dan origen a conductas instintivas o innatas (filogenéticas). Por el contrario, si las estructuras provienen de historias particulares se dice que las estructuras son ontogénicas y las conductas aprendidas.

En el alcance de la presente tesis, las teorías biológicas de la epigénesis, la ontogénesis, y la filogénesis servirán como metáforas epistemológicas a través de las cuales se pueden modelar, respectivamente, los procesos de *aprendizaje*, *desarrollo*, y *evolución* de estructuras cognitivas que actúan como bloques de construcción en la simulación de facultades cognitivas de orden superior, tema que se discutirá en capítulos posteriores.

2.1.3. Sistema Inmunológico Natural

La Inmunología puede definirse como el estudio de los mecanismos que proveen resistencia en contra de las enfermedades. Así, el Sistema Inmune tiene como función principal proteger a un ser vivo de los ataques constantes de los microorganismos externos. El sistema inmune reconoce específicamente y elimina selectivamente a los invasores externos por un proceso denominado respuesta inmune. El sistema inmune posee mecanismos de defensa natural, rápidos y efectivos en contra de las infecciones. Consiste en una línea de defensa de dos niveles, conocidos como sistema inmune innato y sistema inmune adaptativo. Ambos sistemas dependen de la actividad de los glóbulos blancos o leucocitos, donde la inmunidad innata es mediada principalmente por los granulocitos y los macrófagos, y la inmunidad adaptativa es mediada por los linfocitos.

El sistema inmune innato es la primera línea de defensa en contra de diversos tipos de microorganismos, y es considerado esencial para el control de infecciones bacterianas comunes. Las células del sistema inmune innato tiene un rol crucial en el proceso de iniciación y control de la respuesta inmune adaptativa. Por otro lado, los linfocitos son las células más importantes del sistema inmune adaptativo. Ellos sólo se encuentran presentes en los vertebrados, quienes han evolucionado a un sistema que proporciona un mecanismo de defensa más eficiente y versátil en contra de futuras infecciones, comparado con los mecanismos del sistema inmune innato.

Los linfocitos controlan la respuesta inmune adaptativa y son los responsables del reconocimiento y eliminación de los agentes patógenos. Estos agentes, a su vez, proporcionan una memoria inmune que se forma después de la exposición a la enfermedad o vacunación. Los linfocitos usualmente se activan cuando existe algún tipo de interacción con un estímulo antigénico, lo cual lleva a la activación y proliferación de ellos (N. de Castro y Timmis, 2002). Existen dos tipos principales de linfocitos: los Linfocitos B, conocidos también como células B; y los Linfocitos T, conocidos también como células T.

El sistema inmunitario de los animales vertebrados reacciona ante los antígenos ajenos produciendo una respuesta inmune específica que desencadena la destrucción del antígeno; esto es, reacciona ante los propios antígenos incorporándolos en el mecanismo regulador de una red auto-organizada. Stewart (Stewart, 1993) argumenta que esta conducta se ajusta a los requisitos de la definición del antígeno y, por tanto, que el sistema

inmunitario es una *entidad cognitiva* capaz de adaptarse, aprender, y crear memoria. Las consecuencias de esta perspectiva son: las moléculas y células de un organismo multicelular no son, por sí mismas, cognitivas; una plena articulación de la neurofisiología como ciencia cognitiva requiere un estudio de la conducta emergente en el nivel de las interacciones significativas entre un sistema y su medio; y los objetos de la cognición, ya sea con o sin neuronas, se materializan al confluir las percepciones y acciones del propio sistema cognitivo.

Adicionalmente, Stewart (Stewart, 1993) sostiene que un sistema inmune es un sistema cognitivo puesto que es capaz de: percibir, esto es, que las interacciones entre el sistema y el ambiente resultan en modificaciones en el estado interno del sistema; y dirigir las acciones en función de las percepciones, puesto que la respuesta inmune desencadena una serie de procesos inmunológicos (e.g., la selección clonal, la maduración clonal, las meta-dinámicas, etc.) encargados de la destrucción del antígeno que la provocó.

En el contexto de la teoría autopoiética, el trabajo de Varela aporta una perspectiva diferente sobre el sistema inmune que extiende la teoría de las redes idiotípicas[1]. El sistema inmune ya no es considerado un sistema dirigido por la presencia de los antígenos. Por el contrario, el sistema inmune es una red operacionalmente cerrada que reacciona de forma autónoma a fin de definir y preservar la identidad del organismo, proceso que es denominado auto-afirmación (self-assertion) (Varela y Coutinho, 1991). La perspectiva autopoiética del sistema inmune acentúa la auto-reactividad y, en contraste con (Matzinger, 2002), evita una distinción de los estímulos externos entre peligrosos o no, lo que causaría solamente dos respuestas inmunes diferentes. Por el contrario, el sistema inmune se comporta de una sola forma, aunque produce diferentes respuestas dependiendo de los estímulos externos y del estado actual (Bersini, 2002; Nanas y Roeck, 2009). Así pues, debido a que el sistema inmune cambia con el transcurso del tiempo (e.g., durante el desarrollo), el mismo estímulo puede disparar diferentes respuestas inmunes.

2.2. Técnicas Computacionales

2.2.1. Inteligencia Computacional

La inteligencia es un concepto que aún no ha podido ser bien definido, ya que por lo general se le otorga exclusividad humana (García, 2002). Lo que si es cierto es que la inteligencia está relacionada con el aprendizaje, pues cuando un organismo aprende a responder a los cambios de su medio ambiente, puede sobrevivir durante un mayor tiempo.

El grado de inteligencia no puede ser medido en un único medio ambiente, pues éste depende de las condiciones locales del lugar con el cual interactúa el organismo, ya que sus adaptaciones han sido desarrolladas para amoldarse a dicho espacio. Así, los delfines no serían tan inteligentes en tierra como el hombre, e igualmente el hombre no tendría comparación con un delfín en el mar.

[1]La teoría de la red idiotípica (Jerne, 1973) se basa en dos hipótesis:(1) algunas porciones variables del anticuerpo (idiotipos) pueden comportarse como determinantes antigénicos, y (2) los anticuerpos anti-idiotipo actúan de reguladores de la actividad del idiotipo y de los linfocitos T.

La Inteligencia Artificial (IA) es un campo de estudio que combina las ciencias de la computación, la filosofía, las matemáticas, la biología, entre otras, con el fin de dotar a las máquinas con la capacidad de "pensar" y presentar conductas "inteligentes" (Charniak y McDermott, 1985). El campo de la IA es amplio y su concepto teórico es aún muy debatido, por lo que no se entrará a profundizar sobre el tema.

2.2.2. Agentes Inteligentes

El término **agente** proviene del latín *agere*, que significa hacer. Agente deriva del participio *agens*, y expresa la capacidad de acción o actuación de una entidad. Tras esta introducción etimológica, es importante resaltar que la mayoría de la literatura relacionada con los agentes no conserva una definición común del concepto de agente. Sin embargo, y al estar hablando de productos de ingeniería, se pueden establecer unos criterios que permitan distinguir lo que es un agente de lo que no, ofreciendo un modelo razonable de sus características y de su comportamiento.

Según Tolosa (G. y F., 1999) la definición de agente está dada según la línea de investigación en la cual se enmarque el uso del término (Inteligencia Artificial, Ingeniería de Software, Sistemas Autónomos, etc.). Así, se puede encontrar que para Wooldridge y Jennings (Wooldridge et al., 2000), un agente es un programa auto-contenido capaz de controlar su proceso de toma de decisiones, basándose para ello en la percepción que tiene de su ambiente, y en la intención de lograr uno o varios de sus objetivos. Según esta definición, un agente es un sistema que reúne las siguientes características:

- *Autonomía*: un agente encapsula un estado (no accesible a otros agentes), y toma decisiones sobre qué hacer basándose en su estado y experiencia, sin la intervención directa de otros agentes o de humanos. Sin embargo, una noción menos exigente de autonomía propone que un agente es autónomo si es capaz de tomar sus propias decisiones.
- *Reactividad*: un agente está situado dentro de un entorno, y es capaz de realizar percepciones de ese entorno y responder en consecuencia a los cambios que en él se producen.
- *Pro-actividad*: un agente no actúa simplemente en respuesta a los cambios que se producen en su entorno, sino que también tiene la capacidad de tomar la iniciativa en lo que a ejecutar acciones se refiere. En otras palabras, además de poder exhibir un comportamiento reactivo ante los cambios en el ambiente, los agentes deben ser deliberativos, es decir, deben tener la capacidad de planificar qué hacer antes de ejecutar una acción sobre el ambiente en el que se encuentran inmersos.
- *Racionalidad*: el comportamiento de un agente depende del conjunto de percepciones que procesa en un momento dado. Debido a que estas percepciones del ambiente son limitadas o incompletas, el agente sólo puede tener una representación parcial del mismo. Al enfrentarse a esta limitación, el agente debe aplicar su racionalidad para determinar la acción que va a llevar a cabo ante una percepción dada. Para lograrlo, realiza un proceso de "cartografiado", el cual consiste en seleccionar la acción más apropiada que puede desempeñar en respuesta a determinados conjuntos de percepciones.

- *Habilidad social*: un agente interactúa con otros agentes (y posiblemente también con personas) a través de algún tipo de lenguaje de comunicación de agentes (y en algunos casos, mediante lenguaje natural). A través del lenguaje, un agente puede adquirir la habilidad necesaria para involucrarse en actividades sociales (como por ejemplo la resolución de problemas de forma cooperativa).

Pattie Maes (Maes, 1994), del AI Lab del MIT define a los agentes autónomos como sistemas computacionales que habitan en algún ambiente dinámico y complejo, donde perciben su estado y actúan autónomamente, llevando a cabo una serie de objetivos o tareas para los cuales fueron diseñados. Por otro lado, para Ferber (Ferber, 1999), un agente es una entidad física o virtual tal que:

- Se encuentra en un ambiente que percibe y representa parcialmente, sobre el cual actúa autónoma y proactivamente.
- Cuenta con un conjunto de metas u objetivos que trata de optimizar y los cuales tiende a satisfacer por medio de su conducta.
- Posee habilidades y está en capacidad de ofrecer servicios.
- Posee recursos propios.

Existen diversas taxonomías que establecen criterios para clasificar los agentes desde distintos puntos de vista, desde sus características individuales hasta su utilidad, pasando por su comportamiento externo. Una de las propuestas de clasificación más destacadas es el modelo de clasificación llamado de las vocales, propuesto por Yves Demazeau (Demazeau y Tavares da Silva, 2002). La **A** (de agente) caracteriza sus rasgos individuales: arquitectura, funcionamiento interno, complejidad, etc. La visión externa se desglosa en varios apartados: la **E** (de entorno) caracteriza los requisitos computacionales para que el agente funcione correctamente; la **I** (de interacción) considera las capacidades de comunicación del agente; la **O** (de organización) considera el papel del agente en el conjunto del sistema; y, por último, la **U** (de utilidad) hace referencia a la aplicación de la que forma parte el agente.

Según sus características individuales (A), los agentes pueden clasificarse en agentes *reactivos* y agentes *cognitivos* (Franklin y Graesser, 1996). Los agentes reactivos realizan tareas sencillas y su modelo computacional está basado en un ciclo de *percepción/reacción*. La reacción consiste en la ejecución de procedimientos basándose estrictamente en aquello que percibe el agente por medio de sus sensores. Además, un agente reactivo no realiza procesos de razonamiento, ni tiene ningún mecanismo explícito de representación del conocimiento.

Por otra parte, los agentes cognitivos realizan tareas complejas que requieren de algún tipo de representación explícita del conocimiento. Para realizar estas tareas, los agentes cognitivos necesitan llevar a cabo procesos cognitivos, deliberativos y de razonamiento, como por ejemplo la planificación y el aprendizaje. El modelo computacional de un agente cognitivo está basado en un ciclo de *percepción-asimilación-razonamiento-actuación*. Franklin también propone que los agentes cognitivos son intencionales, es decir, tienen objetivos y planes explícitos que les permiten alcanzar sus metas.

Otra característica importante que acentúa la diferencia entre los agentes cognitivos y los reactivos, es que los cognitivos hacen una planificación deliberativa de su actuación, por medio de la cual siguen una serie de pasos hasta conseguir el objetivo deseado. Los reactivos, en cambio, responden a los estímulos sin seguir ningún plan previo. A fin de compensar las debilidades de cada tipo de agente, es posible concebir sistemas heterogéneos cuyo comportamiento se derive de los dos modelos antes descritos. Es decir, es posible dotar a los agentes cognitivos o deliberativos con capacidades de reacción a los eventos; a tales agentes se les suele llamar Agentes Híbridos.

Según el entorno en el que funcionan (E), existen agentes que requieren un entorno especial, o una plataforma software específica, y agentes que se ejecutan en las plataformas computacionales existentes. Un ejemplo de los primeros son los agentes móviles, donde cada plataforma proporciona los mecanismos para gestionar su ciclo de vida y para desplazarse de un nodo a otro. Los últimos se crean mediante los recursos del sistema operativo y siguen el ciclo de vida de cualquier aplicación.

Según el modo de interacción (I), se tienen los agentes especializados en la interacción con el usuario u otros agentes de interfaz, y agentes especializados en la interacción con los recursos. Según el modo de organización (O), se puede considerar agentes individuales, que no tienen capacidad de cooperación y que realizan sus tareas solos; y agentes cooperativos, que pueden realizar tareas solos o colaborando con otros agentes. Según la utilidad (U), se puede clasificar a los agentes de acuerdo con la finalidad o el propósito con el que fueron construidos. Para ello se pueden seguir dos criterios: el dominio de aplicación y el tipo de tarea que realizan dentro de él.

Para mayor información acerca de los Agentes Autónomos y los Sistemas Multi-Agente, el lector puede consultar a (Wooldridge, 2002) y (Wooldridge y Jennings, 1994).

2.2.3. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (RNA) (Haykin, 1999) pretenden imitar el funcionamiento del cerebro humano, compuesto por neuronas biológicas. El desarrollo de las RNA tiene un claro trasfondo neurofisiológico pues no en vano se busca simular las funciones biológicas del pensamiento humano. Una red neuronal artificial es una colección de nodos interconectados por medio de caminos que permiten la propagación de una señal. Estas redes están basadas en la biología del cerebro donde células especiales, las neuronas, se encuentran conectadas entre sí y transmiten señales eléctricas a través de dichas conexiones. En la figura 2.1, se observa una red neuronal artificial simplificada, donde cada nodo representa una neurona y las líneas conectoras, las conexiones sinápticas.

Así pues, cada neurona de la red recibe una serie de entradas a través de las interconexiones sinápticas, y emite una salida que viene dada por tres funciones:

1. Una función de propagación (también conocida como función de excitación), que por lo general consiste en la sumatoria de cada entrada multiplicada por el peso de su interconexión (valor neto). Si el peso es positivo, la conexión se denomina excitativa; si es negativo, se denomina inhibitoria.

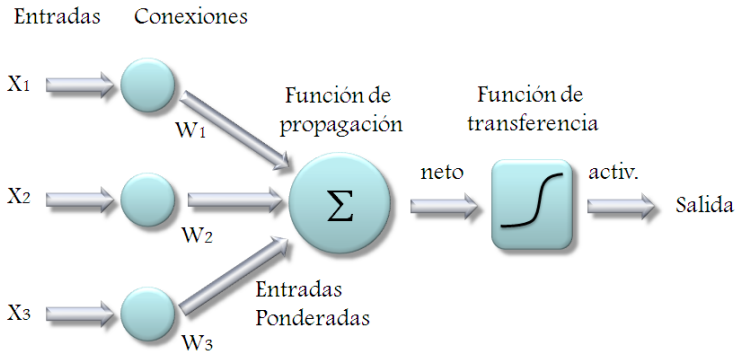


Figura 2.1: Estructura de una neurona de procesamiento. Cada neurona de una red, recibe las entradas ($X_1..X_n$) que contienen la información externa, y cada entrada a su vez tiene asociado un peso ponderado ($W_1..W_n$). Estas entradas multiplicadas por su peso ponderado son sumadas y utilizadas en una función de activación que determina la respuesta neuronal.

2. Una función de activación, que modifica a la anterior. Puede no existir, siendo en este caso la salida la misma función de propagación.
3. Una función de transferencia, que se aplica al valor devuelto por la función de activación. Se utiliza para acotar la salida de la neurona y generalmente viene dada por la interpretación que se desea dar a dichas salidas. Algunas de las más utilizadas son la función sigmoidea y la tangente hiperbólica.

Algunas de las ventajas principales que proveen los modelos de redes neuronales artificiales (RNA) se enuncian a continuación:

- *Aprendizaje*: las RNA tienen la habilidad de aprender mediante una fase que se llama "entrenamiento de la red". Esta fase consiste en proporcionar a la RNA, conjuntos de datos de entrada relacionados con una salida (respuesta) esperada.
- *Auto-organización*: una RNA crea su propia representación de la información en su interior, mediante el reajuste de los pesos de las conexiones sinápticas.
- Tolerancia a fallos: debido a que una RNA almacena la información de forma redundante, ésta puede seguir respondiendo de manera aceptable aún si se daña parcialmente.
- Flexibilidad: una RNA puede manejar cambios no importantes en la información de entrada, como señales con ruido o información incompleta.
- Tiempo real: las RNAs operan en paralelo, con lo cuál se pueden obtener respuestas en tiempos críticos.

Existen diferentes modelos de RNAs (e.g., Perceptrón, Adaline, Memorias Asociativas, Red de Elman, Red de Hopfield, Redes de Kohonen, entre otras), no obstante, aquél que más interesará durante el desarrollo de la presente tesis es el Perceptrón Multicapa (Rosenblatt, 1961; Rumelhart et al., 1986). El perceptrón multicapa es una RNA formada

por múltiples capas diseñada para resolver problemas que no son linealmente separables. Las capas pueden clasificarse en tres tipos (obsérvese la figura 2.2):

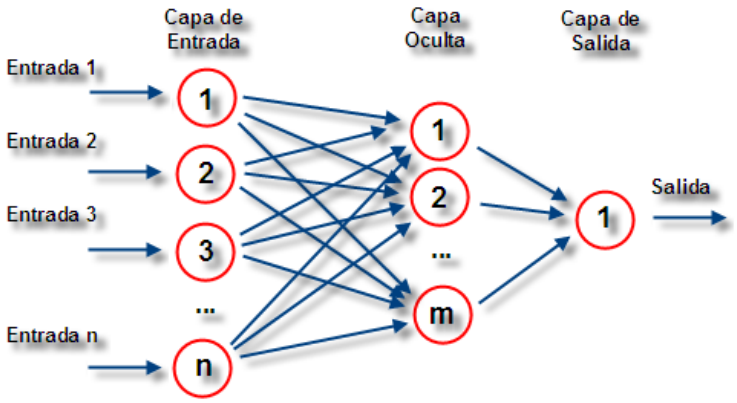


Figura 2.2: Perceptrón Multicapa.

1. *Capa de entrada*: constituida por aquellas neuronas que introducen los patrones de entrada en la red. En estas neuronas no se produce procesamiento.
2. *Capas ocultas*: formadas por aquellas neuronas cuyas entradas provienen de capas anteriores y cuyas salidas pasan a neuronas de capas posteriores.
3. *Capa de salida*: neuronas cuyos valores de salida se corresponden con las salidas de toda la red.

La propagación hacia atrás (también conocida como retropropagación del error o regla delta generalizada), es un algoritmo utilizado en el entrenamiento de estas redes, por ello, el perceptrón multicapa también es denominada red de retropropagación. Para profundizar acerca del tema de redes neuronales artificiales, y en particular sobre el modelo perceptrón multicapa, el lector puede consultar la literatura especializada (Haykin, 1999; Anderson, 1993a).

2.2.4. Proceso Evolutivo Artificial

La meta del Proceso Evolutivo Artificial (PEA) – y en general de cualquier proceso evolutivo, es dar origen a un organismo (agente) altamente adaptado al medio ambiente en que habita, es decir, todo proceso evolutivo tiene como fin, generar una población de agentes mejor adaptados, donde la inteligencia es considerada una variante del proceso adaptativo. Dicho nivel de inteligencia puede ser logrado aplicando determinados procesos de adaptación y aprendizaje sobre una población inicial de agentes.

En la implementación del proceso evolutivo son fundamentales dos elementos: el agente y el medio ambiente. No puede existir proceso evolutivo si falta alguno de estos dos elementos. El ambiente moldea al agente durante su proceso de adaptación, y éste sobrevive siguiendo las reglas que el medio le impone. Por tanto, agente y medio ambiente

poseen propiedades y elementos que son comunes a todo proceso evolutivo (Futuyma, 1986).

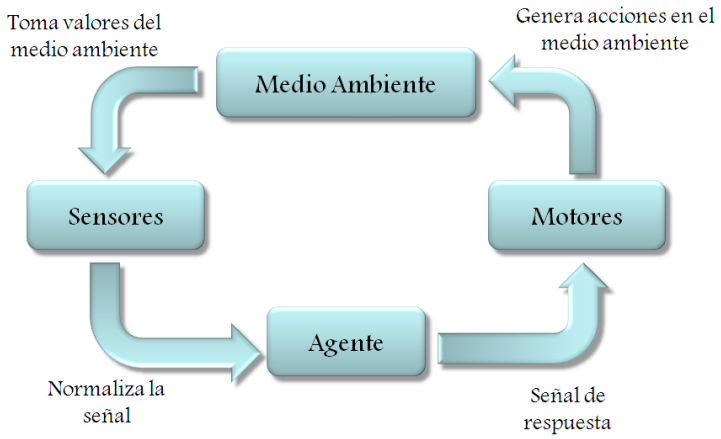


Figura 2.3: Esquema de interacción del agente con el medio ambiente (Futuyma, 1986).

Los agentes son los elementos fundamentales del proceso evolutivo, ya que buscan adaptar su comportamiento a fin de poder actuar de manera inteligente ante las condiciones del medio ambiente. Dichas actuaciones sólo pueden tener lugar gracias al intercambio de información entre el agente y el medio ambiente, puesto que para reaccionar a un estímulo es necesario percibirlo primero y luego provocar un cambio físico o cognitivo (en este último caso, el cambio se refleja como una modificación del sistema nervioso central – SNC– del agente). La interacción del agente con el medio se da gracias a los sensores, que se encargan de percibir la información externa, y a los efectores o motores, que se encargan de ejecutar la acción de respuesta al estímulo percibido por el agente.

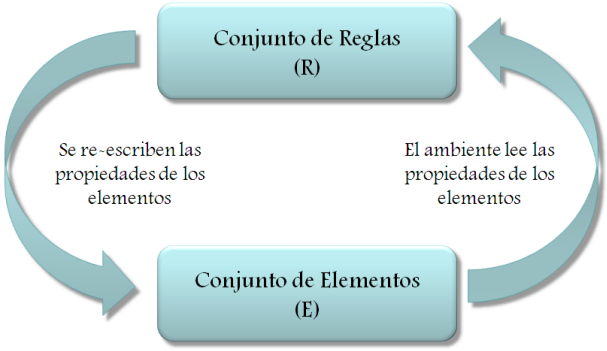


Figura 2.4: Estructura del medio ambiente (Futuyma, 1986).

El medio ambiente es el segundo elemento importante del PEA. Las ciencias biológicas han demostrado que las adaptaciones y los cambios que sufre una población de organismos con el paso del tiempo son producto del proceso evolutivo y dependen enteramente de las condiciones del medio ambiente en el que habitan sus integrantes (Eiben y Smith, 2003). En el PEA, el medio ambiente artificial está conformado por dos conjuntos: los elementos físicos (conocidos de esta forma porque poseen propiedades físicas,

e.g., los objetos, las fuentes de energía, etc.), y un conjunto de reglas (también llamadas leyes físicas).

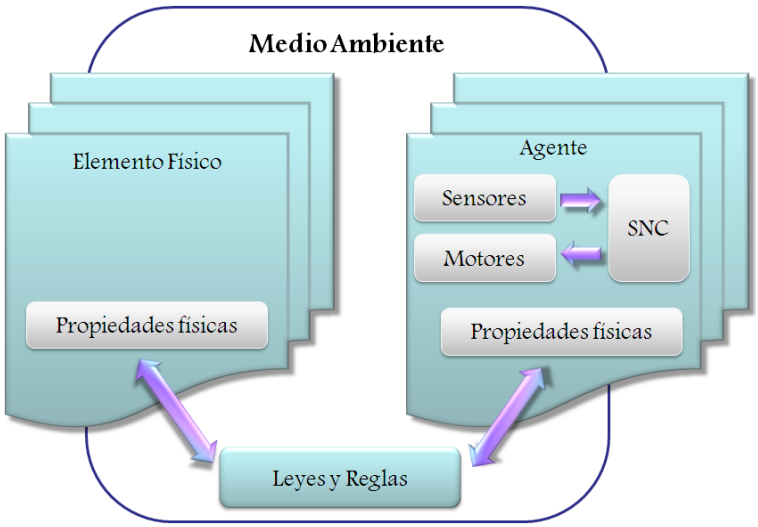


Figura 2.5: Interacción del medio ambiente con el agente (Futuyma, 1986). SNC: Sistema Nervioso Central.

El diseño del medio ambiente es fundamental a la hora de crear un simulador de procesos evolutivos, ya que los resultados finales dependen de su configuración. Así pues, el diseño de las reglas debe siempre tener en cuentas las posibles interacciones entre los elementos y los límites de acción de los mismos (como se observa en la figura 2.5).

2.2.4.1. Algoritmos Genéticos

Los Algoritmos Genéticos (AG) fueron introducidos por John Holland (Holland et al., 1989) y D. Goldberg (Goldberg, 1989) con el fin de imitar algunos de los mecanismos que se observan en la evolución de las especies. Esta técnica se basa en los mecanismos de selección que utiliza la naturaleza, de acuerdo con los cuales los individuos más aptos de una población son los que sobreviven, al adaptarse más fácilmente a los cambios que se producen en su entorno. Hoy en día se sabe que estos cambios se efectúan en los genes de un individuo (unidad básica de codificación de cada uno de los atributos de un ser vivo), y que sus atributos más deseables (i.e., los que le permiten adaptarse mejor a su entorno) se transmiten a sus descendientes cuando éste se reproduce sexualmente.

Los AGs son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin. Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas. Un AG consiste en una función matemática o una rutina de software

que toma como entradas a los ejemplares y retorna como salidas cuáles de ellos deben generar descendencia para la nueva generación.

Un AG define una *función de aptitud*, que no es más que la función objetivo del problema de optimización. El algoritmo genético únicamente maximiza, pero la minimización puede realizarse fácilmente utilizando el recíproco de la función maximizante. Una característica principal de esta función es que es capaz de "penalizar" a las malas soluciones, y de "premiar" a las buenas, de forma que sean estas últimas las que se propaguen con mayor rapidez.

La codificación más común de las soluciones es a través de cadenas binarias, aunque se han utilizado también números reales y letras. El primero de estos esquemas ha gozado de mucha popularidad debido a que es el que propuso originalmente Holland, y además porque resulta muy sencillo de implementar. Por otro lado, para tratar de imitar el comportamiento evolutivo observado en los cromosomas de los organismos vivos, los AG utilizan herramientas lógicas conocidas como operadores genéticos, que buscan generar un comportamiento adaptativo. Algunos de estos operadores son:

- *Cruce*: las características de los padres se recombinan para dar paso a una nueva descendencia.
- *Selección*: las características de un padre se conservan de una generación a otra.
- *Mutación*: se altera el código genético del progenitor para dar paso a una nueva especie.

A continuación se enuncian las ventajas y desventajas más relevantes de este tipo de algoritmos (Poli et al., 2008):

- No necesitan conocimientos específicos sobre el problema que intentan resolver.
- Operan de forma simultánea con varias soluciones, en vez de trabajar de forma secuencial como las técnicas tradicionales de búsqueda y optimización.
- Cuando se usan para problemas de optimización (e.g., maximizar una función objetivo) resultan menos afectados por los máximos locales (falsas soluciones) que las técnicas tradicionales.
- Resulta sumamente fácil ejecutarlos en las modernas arquitecturas masivamente paralelas.
- Usan operadores probabilísticos, en vez de los típicos operadores deterministas de las otras técnicas.
- Pueden tardar mucho en converger, o no converger en absoluto, dependiendo en cierta medida de los parámetros que se utilicen (e.g., tamaño de la población, número de generaciones, etc.) o del tipo de codificación empleada.
- Pueden converger prematuramente debido a una serie de problemas de diversa índole.

2.2.4.2. Programación Genética

La Programación Genética (PG), inventada por Cramer en 1985 (Cramer, 1985) y desarrollada posteriormente por Koza en 1992 (Koza, 1992), fue propuesta con el fin de resolver el problema de las soluciones de longitud fija presente en los AG, a través de entidades no lineales con diferentes tamaños y formas. El alfabeto usado para crear estas entidades resulta ser también muy variado, creando así un sistema de representación más versátil.

La programación genética aplica un proceso evolutivo a una población de individuos (programas), que se representan mediante expresiones simbólicas empleando generalmente la sintaxis del lenguaje LISP. La estructura de almacenamiento de los individuos es un árbol sintáctico el que los nodos son símbolos que representan funciones y las hojas representan símbolos terminales. Todos los programas en la población inicial han de ser sintácticamente correctos (ejecutables) y los operadores genéticos (cruce, reproducción, mutación, etc.) también han de producir programas sintácticamente correctos.

En el argot de la teoría evolutiva, las entidades de la PG son simples replicadores que sobreviven por virtud de sus propias características. Las entidades no lineales (árboles sintácticos) de la PG se asemejan a las moléculas de proteínas en que emplean un alfabeto más enriquecido y en que utilizan una representación compleja y jerárquica. Sin embargo, estas entidades resultan ser difíciles de reproducir debido a que las alteraciones genéticas son hechas directamente en el propio árbol sintáctico. En consecuencia, la mayoría de alteraciones generan estructuras imposibles o no válidas sintácticamente. En comparación con la naturaleza, la expresión de cualquier gen proteínico resulta siempre en una estructura proteínica válida (e.g., en la naturaleza, no existe algo tal como una proteína estructuralmente incorrecta).

De esta manera, en la PG los operadores genéticos actúan directamente sobre el árbol sintáctico, y aunque a primera vista esto pueda parecer ventajoso, limita enormemente esta técnica (e.g., es imposible hacer que un árbol de naranjas produzca mangos sólo con injertar y podar las ramas). Además, la paleta disponible de operadores genéticos es muy reducida, debido a que la mayoría de ellos resultan en árboles sintácticos inválidos. Por consiguiente, la PG emplea exclusivamente una clase especial de recombinación que opera en el nivel de los árboles sintácticos. En este operador de cruce genético específico de la PG, las ramas seleccionadas son intercambiadas entre dos árboles padres con el fin de crear los descendientes. La idea detrás de la implementación es intercambiar bloques pequeños y sintácticamente concisos con el propósito de evolucionarlos en soluciones jerárquicas más complejas.

El operador de mutación en la PG también opera diferente con el fin de garantizar la creación de programas sintácticamente válidos. De esta manera, el operador de mutación selecciona un nodo del árbol y reemplaza la rama que se encuentra debajo de dicho nodo por una rama generada aleatoriamente. La permutación es el tercer operador usado en la PG y, de igual forma que los operadores de recombinación y mutación, está fuertemente restringido: dos nodos estructuralmente equivalentes son escogidos y sus posiciones son intercambiadas. Aunque Koza describió estos tres operadores como los operadores básicos de la PG, el cruce es prácticamente el único operador genético en la mayoría de las implementaciones de PG (Poli et al., 2008).

Una línea de investigación que se ha derivado de la PG, es la Programación de Expresiones Genéticas (PEG) propuesta por Ferreira (Ferreira, 2001). La PEG es un modelo computacional que hereda características tanto de los Algoritmos Genéticos (AG) como de la Programación Genética (PG). Estos dos últimos modelos están condenados a tener una de dos limitaciones: los AG son fáciles de manipular genéticamente debido a que se emplean representaciones lineales (*cromosomas*), aunque este tipo de representaciones no ofrece la suficiente complejidad funcional requerida en la mayoría de los casos; mientras que las representaciones no lineales (*árboles de expresión*) de la PG exhiben mayor complejidad funcional aunque resultan ser extremadamente difíciles de reproducir con modificación. Por su parte, la PEG emplea estos dos tipos de representación y define un mecanismo de traducción que convierte los cromosomas (genotipo) en árboles de expresión sintácticamente bien formados (fenotipo). Este tipo de representación híbrida y su mecanismo de traducción bidireccional es lo que le permite a la PEG exhibir un comportamiento más robusto, complejo y flexible que sus análogos los AG y la PG (Ferreira, 2001; Ferreira, 2006).

Así pues, la diferencia fundamental entre los tres algoritmos (AG, PG y PEG) reside en la naturaleza de los individuos: en los AGs los individuos son cadenas lineales de longitud fija (los cromosomas); en la PG los individuos son entidades no lineales de diferentes tamaños y formas (los árboles sintácticos); y en la PEG los individuos están codificados como cadenas lineales de longitud fija (el genoma o los cromosomas) los cuales son expresados posteriormente como entidades no-lineales de diferentes tamaños y formas (e.g., las representaciones de diagramas simples o los árboles de expresión). Adicionalmente, la PEG define un conjunto más amplio y enriquecido de operadores genéticos. Para mayor información sobre PEG, el lector puede consultar a (Ferreira, 2001).

2.2.4.3. Sistemas Clasificadores de Aprendizaje

Un Sistema Clasificador (SC) es un sistema de aprendizaje de máquina, propuesto por Holland (Holland et al., 1989; Booker et al., 1989), que aprende sintácticamente cadenas de reglas simples llamadas clasificadores. Estos clasificadores guían el desempeño del sistema en un ambiente arbitrario. Un SC deriva su nombre de su habilidad para aprender a clasificar los mensajes ambientales dentro de categorías generales, cumpliendo para ello con funciones similares a las de un sistema de control. Un SC consta de tres partes principales: un sistema de reglas y mensajes, un algoritmo de asignación de recompensas, y un algoritmo genético. Los SC poseen grandes ventajas en comparación con los clásicos sistemas basados en reglas de producción –SBRP– (e.g., los Sistemas Expertos), a saberse:

1. Los SBRP requieren una representación rígida y exhaustiva del conocimiento a través de un conjunto de reglas simbólicas, mientras que el SC emplea un módulo de aprendizaje (guiado por un AG) mediante el cual descubre progresivamente nuevas reglas (clasificadores) que se adaptan mejor a las necesidades tanto ambientales como internas.
2. Las reglas de un SBRP deben ser provistas por humanos, mientras que las reglas del SC son aprendidas por el propio agente.

3. Cuando una regla de un SBRP no cumple ninguna aplicabilidad, ésta no es retirada de la base de reglas, mientras que el SC define un mecanismo que reemplaza aquellos clasificadores más obsoletos por otros potencialmente mejores.
4. Cuando en un SBRP existen varias reglas aplicables a una situación, el sistema debe recurrir a mecanismos complejos para el tratamiento de la incertidumbre, mientras que en el SC se facilita esta tarea por medio de un algoritmo de asignación y distribución de recompensas, el cual recompensa aquellos clasificadores más aptos, y penaliza aquellos menos aptos.
5. Una de las mayores ventajas del SC es su capacidad de aprender en diversos escenarios, trabajar con información incompleta, y clasificar en jerarquías la información que proviene del ambiente.

Los clasificadores del SC son reglas sub-simbólicas de la forma <condición>::<acción>, las cuales se componen de cadenas de datos de longitud fija, cuyos elementos pertenecen a un alfabeto ternario permisible (i.e., cada elemento $\in \{0, 1, \#\}$, donde el símbolo "#" es un comodín que puede tomar los valores de 0 o 1). Dado un estímulo ambiental, los clasificadores compiten (de acuerdo con su nivel de ganancia o aptitud) por el derecho a responder a éste con su acción. Mientras que los clasificadores compiten entre sí por obtener la máxima ganancia, el AG se encarga de evolucionar aquellos mejor adaptados, cumpliéndose el principio que los más aptos sobreviven y los menos aptos son eliminados.

Los mensajes capturan la información percibida por el sistema, y la estructuran en cadenas binarias del tipo <mensaje> (donde cada elemento $\in \{0, 1\}$). Estos mensajes son almacenados en una cola de mensajes, donde el sistema los va procesando uno a uno, verificando si coinciden con la parte <condición> de alguno de los clasificadores. Por ejemplo, el mensaje <10010> podría activar los clasificadores: <10#1#>::<11> y <1#010>::<01>, de los cuales ganaría aquél que tuviese una mayor ganancia. Obsérvese que gracias al símbolo comodín (#), un clasificador funciona como una plantilla que puede generalizar múltiples entradas sensoriales, es decir, la condición del primer clasificador (<10#1#>) podría ser activada por los siguientes mensajes ambientales: <10010>, <10011>, <10110>, y <10111>. El clasificador ganador envía a la lista de mensajes su parte <acción>, la cual puede activar nuevamente la acción de otro clasificador, o puede enviar una señal motora a los efectores del agente en el caso de no existir más coincidencias.

El algoritmo presentado por Holland para la asignación y distribución de recompensas del SC, se llama *Bucket Brigade*. Este algoritmo consta de dos etapas: en la primera, se subasta el derecho a enviar una nueva acción a la lista de mensajes entre los clasificadores activados por el mensaje actual, quienes hacen apuestas en función de su nivel de ganancia. Para elegir al clasificador ganador se emplea un método estocástico (e.g., el método de la ruleta), donde los clasificadores que han acumulado una mayor ganancia tendrán mayor probabilidad de ser escogidos. Posteriormente, se procesa la etapa de recaudación de apuestas, en la cual, el clasificador ganador recoge un porcentaje de la apuesta de los clasificadores que apostaron pero no ganaron, los cuales son además penalizados por el sistema.

El último componente de este sistema, el AG, es utilizado para introducir nuevos clasificadores potencialmente mejores que los ya existentes en el sistema. Este AG, se invoca cada cierto número de pasos, seleccionando para su replicación, aquellos clasificadores que han acumulado una mayor aptitud (en función de su ganancia). La única diferencia con un AG tradicional, es que el operador de mutación puede incluir un nuevo símbolo en la parte condición de los clasificadores: el comodín #.

2.2.4.4. Sistemas Inmunes Artificiales

Los sistemas inmunes artificiales (SIA) (N. de Castro y Timmis, 2002) son sistemas adaptativos, inspirados en principios y modelos de la teoría inmunológica, los cuales son aplicados a la resolución de problemas. Existen varias propiedades inmunológicas que son altamente atractivas desde una perspectiva computacional. Es difícil encontrar otro sistema biológico que tenga un conjunto de características tan poderosas y diversas (Burnet, 1978). Entre las características más importantes se pueden destacar:

- El Reconocimiento de Patrones: las células y moléculas del sistema inmune definen múltiples mecanismo para el reconocimiento de patrones. Por ejemplo, existen moléculas simples que se pueden unir a un antígeno específico, o reconocer señales moleculares (e.g., las linfoquinas) y, por otro lado, existen moléculas intracelulares (e.g., MHC) que se unen a proteínas específicas y las exhiben a través de la superficie de la célula a otras células inmunes.
- La Unicidad: cada individuo posee su propio sistema inmune, con sus particulares vulnerabilidades y capacidades.
- La Auto-Identidad: la unicidad del sistema inmune pone de manifiesto el hecho de que cada célula, molécula y tejido que no es nativo del cuerpo, puede ser reconocido y eliminado por el sistema inmune.
- La Diversidad: existen variados tipos de elementos (células, moléculas, proteínas, etc.) que juntos, cumplen con la función de identificar las propiedades del cuerpo para protegerlo contra invasores dañinos y células que presenten un mal funcionamiento. Adicionalmente, existen diferentes líneas de defensa, como la inmunidad innata y la adaptativa (sección 2.1.3).
- La Robustez: ninguna célula o molécula es por sí sola esencial para el funcionamiento del sistema inmune.
- La Autonomía: no existe un elemento central que controle al sistema inmune. No requiere de intervención externa o de mantenimiento.
- La Arquitectura Multicapa: múltiples capas de distintos mecanismos que actúan de forma cooperativa y competitiva, se combinan para proporcionar una alta seguridad global.

La estructura principal de un Sistema Inmune Artificial consta de los siguientes mecanismos (J. Timmis y Neal, 2001):

1. *Representación de componentes inmunes*: considerando los conceptos inmunes revisados en la sección 2.1.3, se tiene que tanto el sistema inmune innato como el sistema inmune adaptativo están formados por leucocitos que poseen moléculas receptoras capaces de reconocer e identificar patrones de moléculas. La porción de una molécula receptora superficial de un leucocito que reconoce un patrón molecular específico se denomina anticuerpo, y el patrón molecular reconocido por un anticuerpo se denomina antígeno. Finalmente, la capacidad que tiene un anticuerpo de reconocer un antígeno se denomina su afinidad (matching level) (N. de Castro y Timmis, 2002).
2. *Funciones de Afinidad*: matemáticamente, se puede representar la forma generalizada de cualquier molécula m en un espacio de formas S, como una cadena de atributos (conjunto de coordenadas) de longitud L. Esta cadena puede estar compuesta por cualquier tipo de atributo, como valores reales, enteros, bits, y símbolos. Estos atributos generalmente se encuentran determinados por el dominio del problema del SIA y son importantes para la definición de las medidas que serán utilizadas para cuantificar sus interacciones. El tipo de atributo definirá el tipo de espacio de formas a ser usado, así:

   - Espacio de Formas Reales: las cadenas de atributos son vectores de valores reales.
   - Espacio de Formas de Hamming: compuesto de cadenas de atributos construidas a partir de un alfabeto finito de longitud k.
   - Espacio de Formas Entero: las cadenas de atributos se componen de valores enteros.
   - Espacio de Formas Simbólico: normalmente, compuesto por diferentes tipos de cadenas de atributos, donde al menos una de ellas es simbólica, por ejemplo, un nombre, un color, etc.

3. *Maduración de Afinidad*: el proceso de maduración de la afinidad juega un rol preponderante en la respuesta inmune adaptativa. En base a él, se realiza la selección y variación genética. Al usar los espacios de formas propuestos para modelar las moléculas del anticuerpo, la representación de un componente inmune resulta similar a la representación utilizada para los cromosomas de un algoritmo evolutivo (AE). De esta manera, se pueden utilizar los mismos mecanismos de selección y mutación para ambas estrategias. Los procedimientos de selección son generales y aplicables a cualquier tipo de espacio de formas. Sin embargo, los operadores de mutación se clasifican de acuerdo con el espacio de formas bajo estudio.
4. *Selección Clonal*: el principio o teoría de selección clonal, plantea una explicación de cómo hace el sistema inmune para describir las características básicas de una respuesta inmune a un estímulo antigénico. Este principio establece la idea de que sólo aquellas células que reconocen a los antígenos proliferan; de esta manera son seleccionadas aquellas que tienen la capacidad de reconocerlos. La Selección Clonal opera en las células B y las células T. Las principales características de la teoría de selección clonal son (Burnet, 1978):

   - Hipermutación Somática: las nuevas células son copiadas (clonadas) de sus padres, y son sometidas luego a nuevos cambios genéticos aleatorios, que

posteriormente son expresados como patrones diversos de anticuerpos, como una forma de mutación somática acelerada.
- Selección Negativa: eliminación de linfocitos recientemente diferenciados, reactivos a patrones antigénicos portados por componentes propios del organismo, llamados antígenos propios.
- Expansión Clonal: proliferación y diferenciación cuando ocurre el contacto entre linfocitos maduros y antígenos extraños al organismo.
- Mono-especificidad: restricción fenotípica de un patrón a una célula diferenciada y retención de este patrón por descendientes clonales.
- Auto-inmunidad: El concepto de un clon prohibido resistente a eliminación temprana por antígenos propios como la base de las enfermedades auto-inmunes.

5. *Red Inmune*: Jerne (Jerne, 1973) propone que la estructura de una red inmune describe los tipos de interacciones entre sus componentes moleculares y celulares sin referirse a las consecuencias funcionales que pudieran tener. La estructura corresponde entonces a la forma en que los elementos de los sistemas inmunes se estructuran en torno a la red idiotípica de interacciones, es decir, sus patrones de conectividad. Esto podría representarse, por ejemplo, bajo la forma de una matriz de conectividad. La estructura de la red se encuentra fuertemente relacionada con la memoria inmune, ya que ésta refleja los resultados de las interacciones de los elementos activos del sistema inmune, tanto entre ellos como con el entorno en que se encuentran.
6. *Dinámica*: la dinámica de la red inmune explica las variaciones en el tiempo de la concentración y afinidad de las células y moléculas que componen el sistema inmune. Por consiguiente, la dinámica de la red inmune describe cómo la red se adapta a sí misma y al entorno, tomando en consideración las interacciones con otras células y moléculas inmunes, y las interacciones con antígenos extraños. La dinámica mantiene uno de los elementos responsables del *aprendizaje* y de la *memoria* de la red.
7. *Meta-dinámica*: una propiedad única del sistema inmune que va más allá de la dinámica de la red es la continua producción y reclutamiento de nuevas células y moléculas. La meta-dinámica de la red inmune, también conocida como mecanismo de reclutamiento inmune, permite la incorporación de nuevos elementos a la red, de acuerdo con el estado global del sistema, o sea, de acuerdo con la sensibilidad de la red con respecto a este nuevo elemento. Esta sensibilidad generalmente se mide mediante la afinidad de este elemento con los elementos que ya se encuentran presentes en la red inmune. La meta-dinámica también es importante para que la red cubra nuevas porciones del espacio de antígenos y, por consiguiente, sea capaz de adaptarse por sí misma al amplio rango de invasores desconocidos.

2.2.4.5. Algoritmos Meméticos

Los algoritmos meméticos (AMs), son técnicas de optimización que combinan sinérgicamente conceptos tomados de otras meta-heurísticas, tales como la búsqueda basada en

poblaciones (como en los algoritmos evolutivos), y la mejora local (como en las técnicas de seguimiento del gradiente). La denominación "memético" surge del término inglés "meme", acuñado por Richard Dawkins (Dawkins, 1992), como el análogo del gen en el contexto de la evolución cultural. En su libro, Dawkins sostenía que el meme es una unidad de información residente en el cerebro y el replicador mutante en la evolución cultural humana. Es un patrón que puede influir sobre sus alrededores y es capaz de propagarse. La siguiente cita ilustra el concepto:

> *"Ejemplos de "memes" son melodías, ideas, frases hechas, modas en la vestimenta, formas de hacer vasijas, o de construir bóvedas. Del mismo modo que los genes se propagan en el acervo genético a través de gametos, los "memes" se propagan en el acervo memético saltando de cerebro a cerebro en un proceso que, en un amplio sentido, puede denominarse imitación".*
> – (Dawkins, 1992).

Los memes se copian por imitación, enseñanza u otros métodos, y compiten por espacio en nuestros recuerdos y por la oportunidad de ser copiados de nuevo. Grandes grupos de memes que se copian y transmiten juntos se llaman memes complejos co-adaptados, o memeplexes (del inglés meme complexes). Según la definición de Blackmore (Blackmore, 2003), la forma en la que un meme se replica es la imitación. Esto requiere capacidad cerebral para imitar general o selectivamente un modelo. Dado que el proceso de aprendizaje social cambia de una persona (agente) a otra, no puede decirse que el proceso de imitación sea completamente imitado. La igualdad de una idea puede ser expresada con diferentes memes de apoyo. Esto quiere decir que la tasa de mutación en la evolución memética es extremadamente alta, y que las mutaciones son incluso posibles dentro de todas y cada una de las interacciones del proceso de imitación. Esto resulta muy interesante cuando se advierte que existe un sistema social compuesto de una compleja red de micro-interacciones, pero a un nivel superior surge un orden para crear la cultura (Blackmore, 2003).

Tras esta definición subyace la idea central de los AMs: mejoras individuales de las soluciones en cada uno de los agentes junto con procesos de cooperación y competición de tipo poblacional. Un AM mantiene en todo momento una población de diversas soluciones al problema considerado. Cada una de estas soluciones es llamada agente. Esta denominación es una extensión del término "individuo" tan comúnmente empleado en el contexto de los Algoritmos Evolutivos, y permite capturar elementos distintivos de los AMs (e.g., un agente puede contener más de una solución al problema considerado (Berretta et al., 2003)). Estos agentes se relacionan entre sí en un marco de competición y de cooperación, de manera muy semejante a lo que ocurre en la naturaleza entre los individuos de una misma especie (los seres humanos, sin ir más lejos). Cuando se considera la población de agentes en su conjunto, esta interacción puede ser estructurada en una sucesión de grandes pasos temporales denominados generaciones.

Cada generación consiste en la actualización de la población de agentes, usando para tal fin una nueva población obtenida mediante la recombinación de las características de algunos agentes seleccionados. Precisamente, este componente de selección es, junto con el paso final de actualización, el responsable de forzar la competición entre agentes. Más concretamente, la selección se encarga de elegir una muestra de los mejores agentes

contenidos en la población actual. Esto se realiza mediante el empleo de una función guía F_g encargada de cuantificar cuán bueno es cada uno de los agentes en la resolución del problema abordado. Por su parte, el reemplazo o actualización incide en el aspecto competitivo, encargándose de la importante tarea de limitar el tamaño de la población, esto es, eliminar algunos agentes para permitir la entrada de otros nuevos y así enfocar la tarea de búsqueda. En este proceso también puede emplearse la información proporcionada por la función guía para seleccionar los agentes que se eliminarían.

En un contexto más diverso, los AMs son empleados bajo diferentes nombres incluyendo: Algoritmos Evolutivos Híbridos (Crina y Ajith, 2007), Algoritmos Evolutivos Baldwinianos (Dill y Perkowski, 2001), Algoritmos Evolutivos Lamarckianos (Wuhong et al., 2005), Algoritmos Culturales (Reynolds, 1994), y Búsqueda Local Genética (Freisleben y Merz, 1996). Desde un punto de vista computacional, los AMs constituyen un paradigma de optimización basado en la explotación sistemática de conocimiento acerca del problema que se desea resolver, y de la combinación de ideas tomadas de diferentes meta-heurísticas, basadas tanto en técnicas poblacionales como en técnicas de búsqueda local. De la utilidad práctica de los AMs da cuenta la plétora de problemas de optimización complejos que han podido atacarse con éxito gracias a estas técnicas (Moscato y Cotta, 2003).

2.2.5. Redes de Comportamientos

Las Redes de Comportamientos (RC) son una aproximación al problema de la selección de acciones en agentes autónomos, propuesta por Pattie Maes (Maes, 1989). Para Maes, un agente se concibe como una colección de módulos de destrezas cognitivas (comportamientos) cuyo mecanismo de selección de acciones es un proceso que emerge de un conjunto de dinámicas de activación/inhibición entre dichos módulos. Maes, propone un modelo de red de comportamientos no jerárquico, que combina características de los planificadores tradicionales y los sistemas reactivos. Las entradas de la red la constituyen los sensores (que obtienen los estímulos externos producidos en el entorno), y los objetivos/motivaciones (derivados de los estímulos generados internamente). Por otra parte, cada nodo o comportamiento de la red, define un conjunto de proposiciones lógicas sobre el ambiente, las cuales deben ser satisfechas por el agente para que el comportamiento pueda ser ejecutable.

Las Redes de Comportamientos definen un mecanismo de selección de acciones que no es completamente "racional" u óptimo, pero que resulta ser, sin embargo, robusto, veloz, y lo "suficientemente apropiado" para la toma de decisiones (Maes, 1989). Así pues, las características de las RCs demuestran que:

- Favorecen los comportamientos que están orientados hacia las metas, en particular, aquellos que contribuyen a la satisfacción de varias metas de forma inmediata.
- Favorecen los comportamientos que son relevantes a la situación actual, en particular, explotan las oportunidades y son altamente adaptativas a las situaciones impredecibles.
- Favorecen los comportamientos que contribuyen al plan en ejecución (a menos que otro comportamiento lo haga de una mejor manera), por ejemplo, se aferran al

logro de una meta en particular a menos que exista una mejor razón para empezar a trabajar en algo diferente.

- Predicen (o planifican), en particular para evitar situaciones peligrosas y gestionar efectivamente las metas conflictivas.
- Son robustas (nunca colapsan por completo), incluso cuando algunos componentes fallan o no están presentes.
- y son reactivas y rápidas.

Capítulo 3

Estado del Arte

"I do believe that there are some universal cognitive tasks that are deep and profound – indeed, so deep and profound that it is worthwhile to understand them in order to design our displays in accord with those tasks".
– Edward Tufte.

3.1. Paradigmas sobre Cognición Artificial

Como se ha comentado ya en la naturaleza del problema, existen tres grandes paradigmas que intentan modelar y simular, desde distintos puntos de vista, los procesos relativos a la cognición humana. Estos paradigmas son: el Cognitivista, el Emergente, y el Híbrido. Los sistemas cognitivos pertenecientes a cada tipo de paradigma exhiben características comunes entre ellos, las cuales pueden ser organizadas en categorías, como se propone en (Vernon et al., 2007). Estas categorías se describen brevemente a continuación:

1. **Operación Computacional:** los sistemas cognitivistas procesan y manipulan información simbólica, generalmente de forma secuencial, mediante mecanismos basados en reglas (e.g., los sistemas expertos). Por otro lado, los sistemas emergentes explotan los procesos de auto-organización, auto-producción, auto-mantenimiento, y auto-desarrollo de las unidades de conocimiento, mediante la interacción concurrente de redes de componentes distribuidos (e.g., los sistemas conexionistas basados en redes neuronales).
2. **Esquema Representacional:** los sistemas cognitivistas emplean patrones simbólicos que hacen referencia a eventos del mundo externo. Estos patrones son el producto descriptivo de un diseñador humano, proceso que usualmente (aunque no necesariamente) se realiza de forma puntual y local. Por el contrario, las representaciones de los sistemas emergentes son estados globales del sistema codificados dinámicamente en una red distribuida de componentes.

3. **Bagaje Semántico:** las representaciones conceptuales de los sistemas cognitivistas se fundamenten en la identificación de símbolos perceptuales y son construidas bien por un diseñador o mediante aprendizaje inductivo. Estas representaciones pueden ser interpretadas de forma directa por un humano. Por otro lado, las representaciones conceptuales de los sistemas emergentes se fundamentan en la conservación de los estados anticipados del mundo externo y en la construcción adaptativa de destrezas. Estas representaciones sólo tienen significado para el agente en la medida que contribuyen a la viabilidad continuada del sistema, y son inaccesibles de forma directa por la interpretación humana.
4. **Restricciones Temporales:** los procesos de los sistemas cognitivistas no son dirigidos necesariamente por eventos del mundo externo, con lo cual no suelen estar constreñidos por tiempos de respuesta específicos. Por otro lado, los sistemas emergentes suelen operar con eventos del ambiente de forma sincrónica y en tiempo real.
5. **Epistemología Inter-Agente:** para los sistemas cognitivistas, una epistemología compartida y absoluta entre los agentes cognitivos se garantiza en virtud de su punto de vista sobre la realidad: cada agente es embebido dentro de un ambiente, cuya estructura y semántica son independientes del sistema de cognición. Por otro lado, la epistemología de los sistemas emergentes es el resultado subjetivo de un conjunto de experiencias compartidas y consensuadas entre un grupo de agentes compatibles filogenéticamente.
6. **Personificación:** en principio, los sistemas cognitivistas no necesitan personificar al agente en virtud de sus orígenes en el funcionalismo (el cual argumenta que la cognición es independiente de la plataforma física en la cual está implementado el sistema (Freeman y Núñez, 1999)). Por otra parte, los sistemas emergentes están intrínsecamente personificados, donde la implementación física juega un rol constitutivo y esencial en el proceso cognitivo (Vernon, 2006).
7. **Percepción:** en los sistemas cognitivistas la percepción provee una interfaz entre el mundo externo y la representación simbólica de ese mundo. Mediante la información sensorial, la percepción abstrae representaciones espacio-temporales fieles del mundo externo. Por otro lado, en los sistemas emergentes, la percepción es un cambio en el estado interno del sistema que surge como respuesta a las perturbaciones ambientales y cuyo propósito es mantener la estabilidad del sistema.
8. **Actuación:** en los sistemas cognitivistas, las acciones son consecuencias causales derivadas del procesamiento simbólico de las representaciones internas. Por otra parte, en los sistemas emergentes, las acciones son perturbaciones del ambiente ocasionadas por el propio agente.
9. **Anticipación:** es la capacidad del agente cognitivo para predecir estados futuros (por lo general inmediatos) del ambiente, y de actuar en consecuencia. Comúnmente, en los sistemas cognitivistas, la anticipación toma la forma de planificación mediante alguna forma de razonamiento procedimental o probabilístico soportado por un modelo a priori. Por otro lado, la anticipación en el paradigma emergente requiere que el sistema explore un número de estados específicos dentro de su espacio de estados auto-construidos del tipo percepción-acción, sin comprometer las acciones previamente asociadas.

10. **Adaptación:** es la capacidad que tiene el agente cognitivo para acomodar y refinar sus procesos y estructuras en función de los cambios ambientales e internos. Para el enfoque cognitivista, la adaptación usualmente implica la adquisición de nuevo conocimiento, mientras que, en los sistemas emergentes, la adaptación implica una alteración o re-organización estructural interna a fin de efectuar un conjunto nuevo de dinámicas (Schöner, 2006).
11. **Motivación:** es el conjunto de estímulos que impulsan a un agente a realizar determinadas acciones y a persistir en ellas para su culminación. Las motivaciones afectan: la percepción (a través de la atención), la actuación (a través de la selección de acciones), y la adaptación (a través de factores que gobiernan los cambios), de tal manera que en los sistemas cognitivistas, la motivación resuelve los callejones sin salida del sistema (*impasses*), y en los sistemas emergentes, amplia el espacio de interacciones disponible (von Hofsten, 2003).
12. **Relevancia de la Autonomía:** es la capacidad del agente cognitivo para tomar decisiones propias sin estar bajo la influencia directa de factores externos. En el paradigma cognitivista el grado de autonomía depende de la forma en que han sido programados previamente (por un humano) todos los procesos cognitivos, con lo cual no es una autonomía total, mientras que, en el paradigma emergente, la autonomía es un aspecto que resulta de los procesos auto-organizativos del propio agente, y es una característica esencial de la cognición debido a que es el proceso por el cual los agentes autónomos se transforman en sistemas viables y efectivos.

La Tabla 3.1 resume las características principales antes descritas, para cada uno de los paradigmas Cognitivista y Emergente. En las secciones siguientes, se discutirán aspectos más detallados sobre cada paradigma, y posteriormente se describirán brevemente algunas de las aproximaciones arquitectónicas más divulgadas en cada caso, incluyendo las de enfoque híbrido. Adicionalmente, en el anexo A, se presenta un análisis comparativo que pone en evidencia los avances y las limitaciones más importantes en el diseño de Sistemas Cognitivos Artificiales.

En las siguientes secciones se exponen las características más relevantes de cada paradiga de cognición artificial.

3.1.1. La Aproximación Cognitivista

Para los sistemas cognitivistas, la cognición es considerada un aspecto representacional desde un sentido particular: ésta implica la manipulación de representaciones simbólicas explícitas acerca del estado y del comportamiento del mundo externo, lo que facilita, por un lado, la interacción apropiada, adaptativa, anticipatoria, y efectiva con el entorno y con otros agentes; y por otro, el almacenamiento del conocimiento adquirido a través de la experiencia a fin de razonar, incluso de manera más efectiva, en las situaciones futuras (Hollnagel y Woods, 1999). Paralelamente, la percepción está relacionada con la abstracción de representaciones espacio-temporales del mundo exterior extraídas a partir de los datos sensoriales. En cuanto al razonamiento, éste es simbólico, esto es, un proceso procedimental según el cual las representaciones de un mundo externo pueden ser manipuladas a fin de inferir los cambios que pueden ocurrir en la configuración del mundo

Característica	Paradigma Cognitivista	Paradigma Emergente
Operación Computacional	Manipulación sintáctica de símbolos	auto-organización concurrente de una red
Esquema Representacional	Patrones de Símbolos	Estados del sistema global
Bagaje Semántico	Asociaciones del tipo percepto-símbolo	construcción de destrezas
Restricciones Temporales	Comportamiento no constreñido a los tiempos de respuesta	comportamiento dirigido sincrónicamente en tiempo real
Epistemología Inter-Agente	Independiente del agente	Dependiente del agente
Personificación	No necesaria	La cognición implica personificación
Percepción	Representaciones simbólicas abstractas	Respuestas a perturbaciones
Actuación	Consecuencia causal de la manipulación de símbolos	Perturbación del ambiente efectuada por el sistema
Anticipación	Razonamiento procedimental o probabilístico usando modelos a priori	Recorrido auto-efectuado del espacio de estados de percepción-acción
Adaptación	Aprende nuevo conocimiento	Desarrolla nuevas dinámicas auto-organizativas
Motivación	Resuelve los callejones sin salida	Incrementa el espacio de interacción
Autonomía	No necesariamente implicada	La cognición implica autonomía

Tabla 3.1: Comparación entre los paradigmas de cognición Cognitivista y Emergente

(mediante una percepción encargada de dicha configuración alterada) como resultado de las acciones causales.

En la mayoría de aproximaciones cognitivistas relacionadas con la creación de SCAs, las representaciones simbólicas (o esquemas representacionales, en el caso de aquellos sistemas que tienen la capacidad de aprender) son el producto descriptivo de un diseñador humano. Esto es significativo debido, por un lado, a que estas representaciones pueden ser accedidas e interpretadas directamente por un humano y, por otro, a que el conocimiento semántico puede ser integrado y extraído directamente del sistema. Sin embargo, se ha argumentado que éste es también el principal factor limitante de los sistemas cognitivistas: estas representaciones que dependen de un programador, predisponen efectivamente al sistema (o lo ciegan (Winograd y Flores, 1986)), y lo constriñen a una descripción idealizada que depende y es consecuencia de los requisitos cognitivos de la actividad humana. Esta aproximación funciona apropiadamente siempre y cuando el sistema no tenga

que desviarse mucho de las condiciones bajo las cuales dichas descripciones fueron formuladas.

Por último, el paradigma cognitivista propone el supuesto que "el mundo que percibimos es isomórfico en relación con nuestras percepciones sobre éste como un ambiente geométrico" (Brachman, 2002). El objetivo de la cognición es pues, para los sistemas cognitivistas, el poder razonar simbólicamente sobre sus propias representaciones con el fin de efectuar comportamientos robustos, anticipatorios, y orientados a las metas. Usualmente, esta visión sobre la cognición despliega un arsenal de técnicas relacionadas con el *aprendizaje de máquina* y el *modelado probabilístico*, entre muchas otras, en un intento por tratar con la incertidumbre inherente, la variación de tiempos, y la naturaleza incompleta de los datos sensoriales que han sido empleados para dirigir este contexto representacional. Sin embargo, esto no altera el hecho de que la estructura representacional de los sistemas cognitivistas se basa aún en las descripciones de sus diseñadores.

3.1.2. La Aproximación Emergente

La aproximación emergente tiene un punto de vista diferente sobre la cognición. Para esta aproximación, la cognición es el proceso por el cual un sistema autónomo se transforma en un sistema viable y efectivo dentro de su entorno. Por tanto, la cognición es un proceso auto-organizativo mediante el cual el sistema está continuamente reconstruyéndose a sí mismo en tiempo real a fin de mantener su identidad operacional como resultado de la moderación de interacciones mutuas entre el sistema y el ambiente, y de la *co-determinación* (Maturana y Varela, 1987).

El proceso de co-determinación supone que el agente cognitivo es especificado por su ambiente y, que al mismo tiempo, es el proceso cognitivo quien determina que es real y significativo para el agente. En cierto sentido, la co-determinación significa que el agente construye su realidad (i.e., su propio mundo) como resultado de su interacción con el ambiente. En este contexto, el comportamiento cognitivo se define algunas veces como la inducción automática de una ontología, la cual será inherentemente específica a la personificación del agente y dependerá de la historia de interacciones del sistema (e.g., sus experiencias). Por tanto, para las aproximaciones emergentes, la percepción está relacionada con la adquisición de datos sensoriales a fin de facilitar la actuación efectiva y depende de la riqueza de la interfaz con los actuadores (Maturana y Varela, 1987). Así pues, la cognición no es considerada como el proceso por el cual la estructura de un ambiente externo es abstraída y representada en mayor o menor medida, en términos isomórficos.

En (Vernon et al., 2007) se argumenta que la cognición es un proceso complementario a la percepción. Por tanto, la percepción se ocupa de las situaciones inmediatas, mientras que la cognición se ocupa de situaciones que ocurren en períodos de tiempo más largos. Así pues, la cognición refleja el mecanismo por el cual un agente compensa la naturaleza inmediata de la percepción y puede, como consecuencia, adaptarse y anticiparse a las acciones ambientales que suceden en escalas de tiempo mucho más prolongadas. Esto es, que la cognición está intrínsecamente conectada con la habilidad de un agente para actuar con prospección: el agente puede operar en el futuro y ocuparse de aquello que pueda llegar ser, no solamente de lo que actualmente es.

En contraste con la aproximación cognitivista, la mayoría de los sistemas emergente consideran que el modelo primario para el aprendizaje cognitivo es la construcción de habilidades anticipatorias en contraposición con la adquisición de conocimiento, y que los procesos que guían la actuación y mejoran dicha habilidad mientras lo hacen, son postulados como características esenciales de los sistemas inteligentes (Christensen y Hooker, 2000).

A continuación, se describen las tres grandes ramas que derivan de los sistemas de inspiración emergente:

Los Modelos Conexionistas: estos sistemas se basan en el procesamiento en paralelo y distribuido de patrones de activación no simbólicos (en lugar de reglas lógicas), a fin de procesar la información de entrada y lograr un comportamiento efectivo como resultado (Medler, 1998). En este sentido, las redes neuronales, las cuales intentan simular de forma abstracta los sistemas nerviosos biológicos, son un buen ejemplo de los modelos conexionistas, y que en términos de (Smolensky, 1996) podrían considerarse como: "sistemas dinámicos que capturan de mejor forma las regularidades estadísticas en los datos de entrenamiento". Una revisión más completa sobre este tema se encuentra fuera del alcance de la presente tesis, no obstante, el lector puede dirigirse a la sección 2.2.3 donde se explican los principios básicos de las redes neuronales, o consultar la literatura especializada (e.g., (Anderson, 1993a)).

Los Modelos de Sistemas Dinámicos: en general, son sistemas abiertos, disipativos, no lineales, y que no se encuentran en equilibrio constante (Thelen y Smith, 1994). Son sistemas en el sentido que agrupan un vasto número de componentes interoperantes determinados por una gran cantidad de grados de libertad; son disipativos en el sentido que propagan energía de algún tipo; y no están en equilibrio en el sentido que son incapaces de mantener la estructura o la función sin emplear fuentes externas de energía, material, e información (y, de allí, se consideran abiertos). La propiedad de no linealidad es crucial: advierte que la disipación no es uniforme y que sólo un pequeño número de los grados de libertad del sistema determinan su comportamiento complejo. Las implicaciones de los modelos dinámicos son muchas, como se sugiere en (Thelen y Smith, 1994): "la cog-nición no es simbólica, ni representacional, y toda la actividad mental es emergente, situada, histórica, y personificada". También es socialmente construida, esto quiere decir que ciertos niveles de la cognición emergen a partir de las interacciones dinámicas entre los agentes cognitivos.

Los Modelos Enactivos: estos modelos llevan el paradigma emergente aún mucho más allá. En contraposición con el paradigma cognitivista, el cual posee un punto de vista sobre la cognición que requiere de la representación a priori de un objetivo (Varela, 1992b), los modelos enactivos afirman que la cognición es un proceso por el cual los requisitos necesarios para mantener la existencia continuada de una entidad cognitiva son manifestados o exhibidos por la propia entidad (Winograd y Flores, 1986; Maturana y Varela, 1987; Maturana y Varela, 1980), es decir, que existe co-determinación entre la entidad (el agente) que interactúa con el ambiente, y el propio ambiente en el cual ésta está inmersa. Por tanto, nada es "pre-configurado", y de allí que no son necesarias las representaciones simbólicas. En su lugar, existe una interpretación enactiva: el contexto ambiental es el encargado de determinar el nivel de relevancia de las representaciones que construye el propio agente.

El objetivo de la investigación en el área de los sistemas enactivos es el tratamiento total de la naturaleza y emergencia de los sistemas autónomos, cognitivos, y sociales. Esto se fundamenta principalmente en el concepto de Maturana y Varela (Maturana y Varela, 1980) sobre *autopoiesis*[1] – literalmente *auto-reproducción* – el cual argumenta que un sistema emerge como una entidad sistémica coherente, distinta del entorno, y como consecuencia de ciertos procesos de auto-organización. Sin embargo, los sistemas enactivos definen diferentes grados de autopoiesis y sistemas de diferentes ordenes: los de primer orden que corresponden a entidades celulares que logran una identidad física a través del acoplamiento estructural con el ambiente, los de segundo orden que poseen un sistema nervioso que permite la asociación de múltiples estados internos, y los de tercer orden que exhiben cierto nivel acoplamiento entre los sistemas o agentes cognitivos de segundo orden (Varela, 1992b). Una descripción más detallada sobre los ordenes autopoiéticos y cómo pueden ser estos aplicados en el modelado de los procesos de cognición, será presentada en secciones posteriores de esta tesis.

3.1.3. La Aproximación Híbrida

Una vez expuestas las relativas fortalezas de los paradigmas cognitivista y emergente, resulta evidente que la combinación de los dos ofrecería un lugar prometedor para la gestación de nuevos esquemas más completos donde poder simular la cognición (Duch et al., 2002). Los sistemas cognitivistas son capaces de procesar la información y realizar funciones cognitivas de alto nivel, tales como la planificación y el razonamiento deliberativo, en una forma que se asemeja a la experticia humana. Sin embargo, las cuestiones principales que pretende atender esta aproximación son la formulación de entidades simbólicas extraídas a partir de información de bajo nivel, así como también el tratamiento de grandes cantidades de información y la incertidumbre presente en la misma.

Por otro lado, los sistemas emergentes son más apropiados para la captura de la especificidad contextual de la actuación humana, y la manipulación simultánea de múltiples piezas de información de bajo nivel. Pese a esto sus principales deficiencias recaen en la dificultad para realizar funciones cognitivas de alto nivel. Por consiguiente, el beneficio potencial de una aproximación combinada o híbrida es que cada método pueda atender las limitaciones del otro, permitiendo así la creación de un sistema cognitivo más complejo y completo que cubre todos los niveles de procesamiento, desde los estímulos hasta la cognición de alto nivel.

La investigación en este área ha conducido al desarrollo de muchas propuestas de sistemas cognitivos híbridos, los cuales pueden ser divididos (en términos generales) en dos grandes clases, según el tipo de memoria de los módulos constitutivos: local-distribuida y simbólica-conexionista (Sun y Alexandre, 1997). La primera clase de sistemas híbridos abarca una combinación de módulos locales (donde cada concepto es especificado por un nodo de procesamiento diferente), y módulos distribuidos (donde cada concepto es representado por un conjunto de nodos imbricados). En comparación, la segunda clase de sistemas híbridos concierne una combinación de módulos simbólicos (e.g., una regla

[1] La autopoiesis es un tipo especial de auto-organización: un sistema autopoiético es un sistema homeostático (e.g., un sistema auto-regulado) en el cual el proceso de regulación abarca no sólo algunos parámetros del sistema sino toda la organización del sistema en sí (Maturana y Varela, 1987).

de producción o una memoria basada en grafos), y módulos conexionistas (de tipo local o distribuido).

Según este último criterio, los sistemas híbridos pueden ser categorizados en dos clases principales, de acuerdo con la dirección en la que se produce el aprendizaje: aprendizaje descendente (top-down) y aprendizaje ascendente (bottom-up) (Sun y Zhang, 2004). La primera categoría considera la transición de conocimiento desde un nivel conceptual explícito (accesible) hasta un nivel sub-conceptual implícito (inaccesible), mientras que la segunda indica un aprendizaje desde el nivel sub-conceptual hasta el nivel conceptual. El aprendizaje descendente puede ser logrado mediante la pre-codificación de un conjunto de reglas expertas en el nivel superior (i.e., el módulo local/simbólico) y permitiendo que el nivel inferior (e.g., un modelo distribuido de redes neuronales artificiales) aprenda a través de la observación de acciones, proceso que es guiado por el nivel superior (Sun y Zhang, 2004). Por el contrario, el aprendizaje ascendente puede ser llevado a cabo mediante la extracción o traducción del conocimiento implícito, que está codificado en el nivel inferior, dentro de un conjunto de reglas conceptuales en el nivel superior (Duch et al., 2001; Sun et al., 2001).

Así pues, las aproximaciones híbridas parecen extraer lo mejor de los dos paradigmas (Vernon et al., 2007): la adaptabilidad de los sistemas emergentes – ya que los sistemas híbridos incrementan sus marcos representacionales a través del aprendizaje y la experiencia – más la ventaja que proporciona el punto de inicio avanzado de los sistemas cognitivistas – debido a que la definición a priori de las estructuras cognitivas reduce el coste adicional relacionado con su aprendizaje –.

3.2. Arquitecturas Cognitivas Artificiales

Las *arquitecturas cognitivas* en general poseen características propias que las hace divergir respecto a las demás. Por ejemplo, algunas arquitecturas utilizan una representación uniforme del conocimiento, algunas otras una representación heterogénea, mientras que otras no tienen ninguna representación explícita. Así pues, la configuración del conjunto de características es lo que ofrece el soporte a las capacidades cognitivas específicas de cada arquitectura. Comúnmente, la selección de las características se lleva a cabo mediante el seguimiento de algunos supuestos metodológicos, que por lo general están bajo la influencia de los mismos dominios y entornos en los cuales será contextualizada la arquitectura. Por tanto, la gran diversidad de arquitecturas cognitivas existentes en la actualidad depende de la enorme variedad de decisiones que se deben tomar para sus diseños.

En su mayoría, las arquitecturas cognitivas cumplen con un principio de validez neuro-biológica o psicológica, donde existe un intercambio sinérgico entre los estudios realizados sobre la inteligencia biológica y la inteligencia artificial, teniendo siempre en cuenta las restricciones que impone la cognición humana. Sin embargo, muchas arquitecturas ignoran a propósito las restricciones que limitan la cognición humana. Generalmente, porque están enfocadas al desarrollo de agentes que se comportan de manera efectiva en un determinado ambiente, y donde es posible estudiar las interacciones entre estos dos

(e.g., la actuación del agente en ambientes estáticos, o en situaciones específicas de resolución de problemas, o en entornos puramente reactivos, etc.). En este sentido, el término arquitectura cognitiva es un poco confuso. Un mejor término quizás para estos casos, sea el de *Arquitectura para Agentes*, término que podría incluir tanto los sistemas que hacen un intento explícito por modelar la psicología humana (e.g., las arquitecturas cognitivas) como también aquellos que simplemente exploran algunos aspectos del comportamiento inteligente en general (Duch et al., 2002).

En las secciones siguientes se describen brevemente algunas de las propuestas arquitectónicas más relevantes para Sistemas Cognitivos Artificiales, según el tipo de paradigma cognitivo: cognitivista, emergente, o híbrido. En el anexo A se presenta además un Análisis Comparativo donde se contrastan las diferentes características y criterios de diseño de las arquitecturas descritas a continuación.

3.2.1. Arquitecturas de Aproximación Cognitivista

3.2.1.1. Arquitectura SOAR: Estado, Operador y Resultado

El sistema SOAR (por su acrónimo en inglés de *State, Operator And Result*), es uno de los candidatos de Newell (Newell et al., 1993) para sustentar su Teoría Unificada sobre la Cognición (Newell, 1990). SOAR es un sistema basado en reglas de producción[2] que funciona de manera cíclica, empleando un bucle de producción y otro de ejecución.

Grosso modo, SOAR opera de la siguiente forma: primero, en el ciclo de producción, se disparan todas las reglas que coinciden con el contenido de la memoria de trabajo; luego, por cada regla que se dispara, es posible alterar el estado de la memoria declarativa y causar que otras reglas se disparen a su vez. En este punto, el ciclo de decisión se inicia con la selección (basada en preferencias) de una única acción del conjunto de posibles acciones, generando de esta manera múltiples ciclos de producción por cada ciclo de decisión. Cabe mencionar que, las reglas que emplea SOAR son de bajo nivel, es decir, el conocimiento es encapsulado en un nivel de abstracción muy atómico y auto-contenido. A continuación se presenta una breve descripción de los mecanismos y componentes de SOAR (ver la figura 3.1), y el correspondiente soporte teórico subyacente (Lehman et al., 2006).

- **Estados y Operadores:** son las estructuras básicas soportadas por la arquitectura. Los estados contienen toda la información acerca de la situación actual, incluyendo la percepción (en el estado superior), y las descripciones de las metas actuales y el espacio de estados del problema; mientras que los operadores son los medios para avanzar entre dichos estados.
- **Memoria de Trabajo (MT):** contiene la información perceptual y la jerarquía de estados con sus operadores asociados. Los contenidos de la MT activan la recuperación de conceptos de la Memoria de Largo Plazo y controlan la ejecución de las acciones motoras.

[2]Una regla de producción es un par condición-acción del tipo IF-THEN. Un sistema de producción está compuesto por un conjunto de reglas y un motor de inferencia que interpreta y ejecuta dichas reglas.

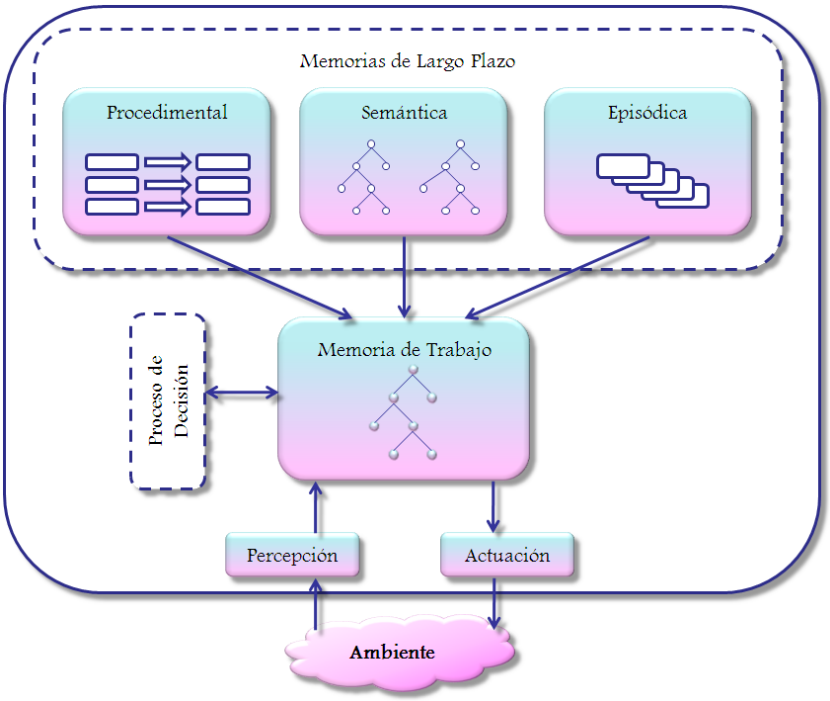


Figura 3.1: Arquitectura SOAR - Distribución de Componentes. Adaptada de (Newell, 1990).

- **Memoria de Largo Plazo (MLP):** almacena el conocimiento del dominio que es procesado por la arquitectura. La MLP soporta tres tipos de representaciones: *conocimiento procedimental* codificado como reglas (e.g., conocimiento sobre cómo y cuándo hacer las cosas), *conocimiento semántico* sobre los hechos del mundo (e.g., cosas que se consideran ciertas) codificado en forma de estructuras declarativas, y *conocimiento episódico* (e.g., los recuerdos y situaciones experimentadas por el agente) codificado como episodios. La memoria procedimental es accedida automáticamente durante el ciclo de decisión, mientras que las memorias semánticas y episódicas son accedidas deliberadamente a través de la creación de entradas específicas en la MT. Todas las MLPs de SOAR son herméticas, esto significa que SOAR no puede examinar los contenidos de sus MPLs directamente, sólo los puede ver a través de los cambios que surgen en la MT.
- **Interfaz Perceptual/Motriz:** este mecanismo cumple con dos funciones: define las correspondencias entre el mundo externo y las representaciones internas en la MT, y ejecuta las acciones determinadas por las representaciones internas en el mundo externo. A través de esta interfaz, la percepción y la actuación son procesos que pueden ocurrir en paralelo con la cognición.
- **El ciclo de decisión:** es el proceso arquitectónico básico que soporta la cognición. El ciclo de decisión está compuesto de tres fases. La *fase de elaboración* implica el acceso en paralelo a la MLP a fin de elaborar el estado, sugerir nuevos operadores, y evaluar dichos operadores. Durante la *fase de decisión* se interpreta el lenguaje (independiente del dominio) de los operadores. El resultado del proceso

de decisión puede ser un cambio en el operador seleccionado, o la generación de un punto muerto (*impasse*) en el caso de que las preferencias estén incompletas o en conflicto. Luego, en la *fase de aplicación*, se disparan las reglas que modifican el estado. La selección de un único operador por ciclo de decisión supone un *cuello de botella cognitivo* en la arquitectura, que representa el límite de cuánto trabajo cognitivo puede realizarse en un instante.

- **Impasses:** advierten una ausencia de conocimiento, y además, una oportunidad para aprender. Un *impasse* ocurre automáticamente en el momento en que se determine que el conocimiento actual no es suficiente para llevar a cabo el proceso de decisión. El lenguaje de los *impasses*, como el lenguaje de las preferencias, es definido independientemente del dominio. Cuando se crea un *impasse*, la arquitectura también inicia automáticamente la creación de un nuevo sub-estado cuya meta es resolver el *impasse*.
- **Mecanismos de aprendizaje:** SOAR define cuatro mecanismos de aprendizaje. El proceso de *chunking*[3] crea automáticamente nuevas reglas en la MLP en el mismo momento en que un impasse es generado. Esto mejora el desempeño y desplaza el conocimiento deliberativo (que ha sido recuperado en un sub-estado superior) hacia un estado donde puede ser empleado de forma reactiva. Las nuevas reglas convierten los *pre-impasses* en cambios pendientes que pueden prevenir el *impasse* en situaciones futuras similares. Adicionalmente, define un mecanismo de aprendizaje por refuerzo que ajusta los valores de las preferencias de los operadores, un mecanismo de aprendizaje episódico que almacena una histórico de las experiencias, y un mecanismo de aprendizaje semántico que captura nuevos conceptos declarativos abstractos.

3.2.1.2. Arquitectura EPIC: Control Interactivo de Procesos

EPIC (Lehman et al., 1997) (por su acrónimo en inglés de *Executive Process Interactive Control*), es una arquitectura que integra diferentes modelos de percepción de alta fidelidad, mecanismos motrices, y un sistema basado en reglas de producción. El proceso de decisión de EPIC emplea el conocimiento encapsulado en las reglas de producción junto con los parámetros perceptuales-motrices del agente.

Existen dos tipos de parámetros en EPIC: un conjunto de *parámetros del sistema*, los cuales se definen a priori para todas las tareas (e.g., la duración de un ciclo de decisión corresponde a 50 ms), y los *parámetros ajustables* a los cuales se les asigna un valor inicial, pero que puede variar dependiendo de la tarea (e.g., el tiempo requerido para efectuar el reconocimiento de formas a través del procesador visual corresponde a 250 ms).

EPIC se compone de un procesador cognitivo (con un interprete de reglas de producción y una memoria de trabajo), y de múltiples procesadores perceptuales: auditivo,

[3] Su traducción corresponde a la agrupación de contenido en unidades pequeñas y manejables. En psicología hace referencia al mecanismo de memoria de corto plazo y las técnicas relacionadas para explotarlo. Podría decirse entonces que un chunk es un trozo manipulable de conocimiento.

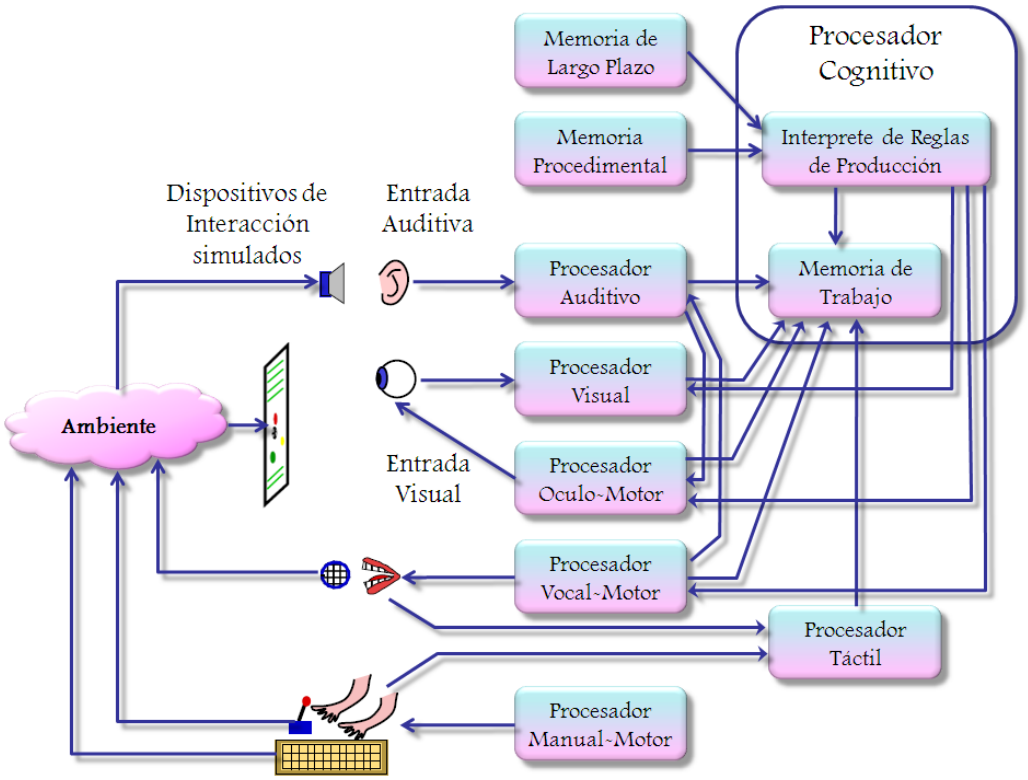


Figura 3.2: Arquitectura EPIC. Adaptada de (Lehman et al., 1997).

visual, oculomotor, vocal, táctil, y motor-manual. Los procesadores perceptuales se ejecutan en paralelo y únicamente modelan los aspectos temporales de la percepción, es decir, no desempeñan ningún procesamiento perceptual adicional.

Similar a la arquitectura SOAR, el procesador cognitivo de EPIC es un sistema de producción en el cual pueden dispararse múltiples reglas por cada ciclo de decisión. Sin embargo, las reglas en EPIC son de nivel mucho más alto que las reglas en SOAR, lo cual significa que no posee un conocimiento detallado sobre el mundo.

EPIC es una arquitectura bastante limitada que no define mecanismos de aprendizaje ni de adaptación, cuyos principales aportes se reducen a la incorporación de los múltiples procesadores que dispone (Duch et al., 2002). Integrada con el motor de SOAR, EPIC se ha aplicado principalmente en simulaciones de control de trafico aéreo (Rosbe et al., 2001).

3.2.1.3. Arquitectura ACT-R: Control Adaptativo del Pensamiento Racional

La arquitectura ACT-R (por su acrónimo en inglés de *Adaptive Control of Thought-Rational*), propuesta por Anderson (Anderson y Lebiere, 1998; Anderson y Qin, 2004), es otra aproximación a una teoría unificada de la cognición. ACT-R propone una descomposición modular inspirada en el funcionamiento de ciertas áreas del cerebro, y define además una teoría sobre cómo estos módulos deben ser integrados para producir una

cognición coherente. La arquitectura comprende cinco módulos especializados, cada uno dedicado a procesar un tipo diferente de información (como se observa en la figura 3.3).

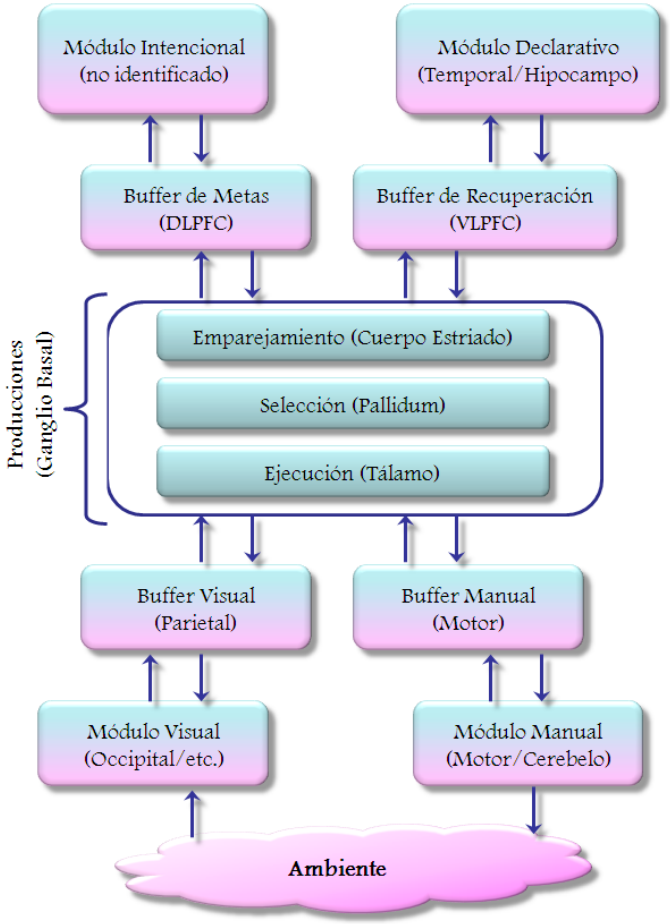


Figura 3.3: Arquitectura ACT-R. Componentes funcionales. Adaptada de (Anderson et al., 1998).

ACT-R propone: un módulo de visión que determina la identidad y la posición de los objetos en el campo visual, un módulo manual que controla las manos, un módulo declarativo que recupera la información de largo plazo, un módulo de metas que actualiza el estado interno del agente, y un sistema basado en reglas de producción que coordina la operación de los otros cuatro módulos.

En ACT-R pueden producirse dos tipos de cuello de botella: el primero se debe a que el contenido de cualquier espacio de memoria (buffer) está limitado a sólo a un tipo de unidad de conocimiento declarativa, llamada *chunk*. El segundo, se debe a que ACT-R sólo puede disparar la acción de una única regla en cada ciclo cognitivo. Esto último contrasta con las arquitecturas SOAR y EPIC que permiten la activación simultánea de múltiples reglas.

En cuanto al módulo de metas, ACT-R posee un mecanismo que supervisa las intenciones del agente a fin de dirigir el comportamiento del sistema hacia el logro de una

meta específica. En efecto, esto garantiza que la operación del sistema sea consistente en la solución de un problema dado. Por otro lado, la información almacenada en la memoria declarativa soporta a largo plazo la coherencia cultural y personal del agente y, junto con el sistema de reglas de producción, forman el núcleo del sistema cognitivo de ACT-R.

La memoria procedimental es encapsulada en el módulo de reglas de producción, y se encarga de coordinar la operación global de la arquitectura. Este módulo posee un mecanismo de resolución de conflictos que se ocupa de calificar las múltiples reglas que se activan en cada ciclo, y de seleccionar la más apropiada conforme al nivel de utilidad más alto. La utilidad de una regla es un factor que está en función de varios parámetros: la probabilidad estimada de que la meta actual será lograda si la regla es seleccionada, el valor de activación de la meta actual, y la estimación del coste que implicaría seleccionar la regla (generalmente, en proporción al tiempo), todos estos aprendidos en experiencias pasadas donde ha sito utilizada dicha regla. En este sentido, ACT-R puede adaptarse a nuevas circunstancias (Byrne, 2003).

El conocimiento declarativo codifica de manera efectiva los elementos de su ambiente, mientras que el conocimiento procedimental codifica las transformaciones observadas; por tanto, la cognición compleja resulta de la interacción entre el conocimiento declarativo y el procedimental (Anderson, 1996). Una característica esencial de ACT-R es que los dos tipos de conocimiento pueden ser re-configurados para cada dominio de aplicación, mediante el ajuste de las estadísticas y parámetros respectivos. Similar a la arquitectura SOAR, ACT-R ha evolucionado significativamente en los últimos años (Qin et al., 2007; Taatgen y Anderson, 2008).

3.2.1.4. Arquitectura ICARUS: Un sistema físico simbólico

ICARUS (Langley, 2005) sigue la tradición de otras arquitecturas de enfoque cognitivista, tales como ACT-R, SOAR y EPIC: explota las representaciones simbólicas, identifica patrones que le permiten seleccionar elementos relevantes del conocimiento, opera en un ciclo continuo de *percepción-actuación*, y emplea un mecanismo incremental de aprendizaje.

Por otro lado, existen algunas diferencias importantes que aporta ICARUS respecto a las demás arquitecturas de enfoque cognitivista. ICARUS distingue entre los *conceptos* y las *destrezas* del agente, dedicando a cada uno de ellos diferentes tipos de representación y de memoria. La *memoria conceptual* codifica el conocimiento general acerca de las clases de objetos y las relaciones entre los mismos, mientras que la *memoria para las destrezas* codifica el conocimiento acerca de la forma de actuar y alcanzar las metas. Adicionalmente, ICARUS adopta una estricta organización jerárquica de su memoria de largo plazo mediante dos estrategias: una estrategia ascendente (bottom-up) que extrae nuevo conocimiento conceptual a partir de la inferencia de datos perceptuales, y una estrategia descendente (top-down) que se encarga de seleccionar las acciones del agente a partir de la meta actual y del conocimiento almacenado en la memoria de destrezas.

ICARUS representa todo el conocimiento de la memoria de largo plazo como una jerarquía de conceptos con valores de certeza asociados. Esta jerarquía es encapsulada en

un módulo independiente que opera de forma asíncrona, llamado *Labrinth*. La arquitectura incluye además otros tres módulos responsables de la percepción, la planificación, y la actuación, conocidos como *Argus*, *Daedalus*, y *Meander* respectivamente. En la figura 3.4 se pueden observar los módulos y procesos funcionales, así como las memorias que emplea ICARUS.

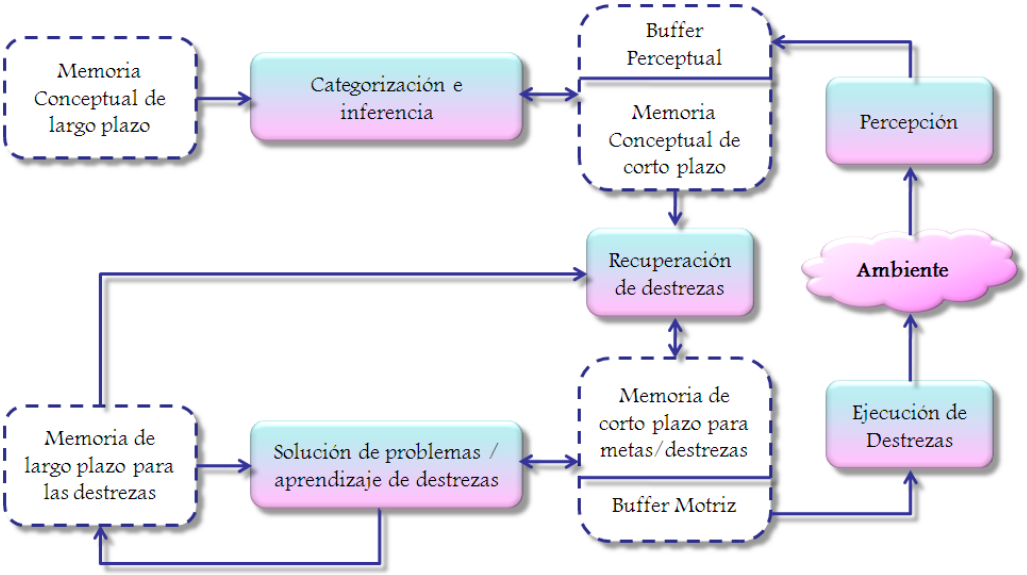


Figura 3.4: Arquitectura ICARUS. Procesos funcionales y conexiones de las memorias. Adaptada de (Langley, 2005).

Labrinth juega un rol importante en la arquitectura. No sólo es el responsable de mantener el conocimiento a largo plazo, sino que actúa además como un mecanismo de comunicación centralizado que coordina los otros módulos. *Labrinth* contiene un área conocida como la memoria activa (o memoria de corto plazo), a través de la cual cada módulo puede enviar solicitudes de extracción de conocimiento a la memoria de largo plazo, para luego poder leer y operar sobre los resultados.

Argus, el componente perceptual, traduce las señales ambientales en estados cualitativos los cuales son enviados a la memoria activa de *Labrinth*. *Labrinth* está continuamente monitorizando la memoria activa a fin de clasificar todas las percepciones nuevas. Si la clasificación resultante es asociada con un problema específico, entonces *Labrinth* expresa el problema como un estado cualitativo al cual le asigna una prioridad y luego le retorna nuevamente a la memoria activa.

Paralelamente, *Daedalus* puede estar trabajando en la solución de un problema, aunque también está continuamente monitorizando la memoria activa. Si un problema nuevo tiene asociada una prioridad más alta que el problema actual que trata *Daedalus*, éste interrumpe su planificación y empieza a trabajar con el nuevo problema, publicando todos los resultados en la memoria activa. Una vez se obtiene una solución completa, *Daedalus* puede continuar con el procesamiento del problema anterior.

Meander también opera en paralelo, monitorizando constantemente la memoria activa. Este módulo tiene la capacidad para iniciar la ejecución de planes parciales si

considera que la probabilidad que establece *Daedalus* para el problema en cuestión no será reducida. Por su parte, *Argus* tiene la responsabilidad de completar el ciclo cognitivo mediante la monitorización de aquellos aspectos del ambiente sobre los que *Meander* está operando, y si los estados percibidos no coinciden de forma adecuada con los estados esperados, se postula una nueva observación la cual *Labrinth* intenta clasificar a fin de crear un nuevo problema para que *Daedalus* se ocupe de él.

3.2.1.5. Arquitectura ADAPT: Percepción Activa y Adaptación

ADAPT (Benjamin et al., 2004), es una arquitectura que ha surgido en respuesta a los múltiples requisitos que exigen las aplicaciones robóticas (e.g., las dinámicas adaptativas, la percepción activa, la reactividad motriz, etc.), los cuales no son soportados generalmente por las arquitecturas clásicas del paradigma cognitivista.

Por lo general, los sistemas robóticos incorporan múltiples comportamientos que operan de forma concurrente y distribuida, los cuales se comunican entre sí en tiempo real. Estas características no suelen ser atendidas por las arquitecturas cognitivistas clásicas (e.g., SOAR, ACT-R, EPIC, etc.) puesto que su principal objetivo es "la búsqueda y selección secuencial de acciones", esto es, que sus mecanismos de aprendizaje se orientan hacia la composición secuencial de acciones en lugar de emplear un enfoque concurrente, y que su organización estructural suele ser jerárquica en vez de distribuida. Los estudios de Benjamin no sugieren que no sea posible conducir tales resultados con las arquitecturas mencionadas, pero sí argumenta que no son sus características esenciales.

Por tanto, Benjamin (Benjamin et al., 2004; Benjamin et al., 2008) propone una arquitectura cognitiva llamada ADAPT (por su acrónimo en inglés de *Adaptive Dynamics and Active Perception of Thought*), la cual se basa esencialmente en los principios de SOAR, aunque también adopta características de ACT-R (e.g., la memoria declarativa de largo plazo en la cual se almacenan los esquemas sensoriomotrices encargados del control de la percepción y la actuación), y de EPIC (e.g., todos los procesos perceptuales son activados en paralelo), pero con la diferencia de que los datos sensoriales de bajo nivel son almacenados también en la memoria de trabajo donde son procesados luego por el mecanismo cognitivo.

Dentro de las características más relevantes que incorpora ADAPT, se puede mencionar la distinción que hace entre dos tipos de metas: las *metas asociadas a tareas* (e.g., "encontrar el objeto azul") y las *metas asociadas a la arquitectura* (e.g., "iniciar un esquema para escanear el escenario"). Debido a esto, ADAPT también posee dos tipos de acciones: las *acciones asociadas a tareas* (e.g., "recoger el objeto azul") y *acciones asociadas a la arquitectura* (e.g., "iniciar un esquema de comprensión del problema"). Mientras que la parte arquitectónica es restringida a la activación de sólo una meta o acción por instante de tiempo, la parte relacionada con las tareas no tiene tales restricciones y por tanto se pueden procesar múltiples metas y acciones – o esquemas – simultáneamente. Las metas asociadas con la arquitectura se representan de forma procedimental, es decir, como reglas de producción, mientras que las metas asociadas a tareas se representan de forma declarativa en la memoria de trabajo.

3.2.1.6. Arquitectura BDI: Creencias, Deseos e Intenciones

De los modelos basados en propiedades antropomórficas para el diseño de agentes intencionales, también denominados sistemas de razonamiento práctico, el más conocido y estudiado, sin duda, ha sido el modelo de *creencia-deseo-intención*, más comúnmente conocido como modelo BDI (Dennett, 1987) (por su acrónimo en inglés de *Belief-Desire-Intention*).

Las *creencias* representan la información que posee el agente acerca del estado "más probable" del entorno[4]. Las creencias son esenciales debido, por un lado, a que el mundo percibido por el agente es dinámico y por tanto los eventos pasados deben ser recordados y, por otro, a que la visión que tiene el agente sobre el mundo es local, con lo cual debe inferir aquello que sucede fuera de su esfera de percepción (Rao y Georgeff, 1991).

Por otro lado, los *deseos* representan un estado final anhelado, esto es, las motivaciones propias del sistema. La diferencia fundamental entre los deseos y las metas radica en que los deseos pueden ser inconsistentes entre sí, mientras que a las metas se les presupone consistencia. No obstante, aunque tradicionalmente se habla de deseos, los propios autores del modelo prefieren trabajar directamente con metas para evitar la inconsistencia (Rao y Georgeff, 1991).

En cuanto a las *intenciones*, éstas reflejan el subconjunto de objetivos que el agente persigue en el momento presente. Estos objetivos se ejecutan, generalmente, como hilos concurrentes que representan los diferentes cursos de acción del agente. Las intenciones surgen de la inconveniencia relacionada con la continua re-planificación que debe hacer el agente en entornos muy cambiantes. En la figura 3.5 se pueden observar los componentes de la arquitectura BDI.

Algunos modelos extendidos de esta arquitectura (Winikoff, 2005; Huber, 2007) incluyen elementos adicionales como los objetivos y los planes del agente. Aunque ya inicialmente en (Bratman et al., 1988) se distinguían dos tipos de planes (i.e., las intenciones estructuradas en planes, y las recetas o librerías de planes), en la arquitectura BDI original sólo se habla del primer tipo. Sin embargo, debido a la habitual limitación de recursos, Georgeff (Georgeff et al., 1999) considera conveniente almacenar los planes genéricos más frecuentes a fin de ahorrar tiempo en la re-planificación.

El mecanismo de funcionamiento de la arquitectura BDI es bastante intuitivo. En el modelo original, frente a cambios en el entorno (reactividad) o frente a nuevas necesidades (proactividad), el sistema genera nuevas opciones (sub-planes). Todas las opciones generadas son filtradas, y aquellas que sobreviven al filtrado son sometidas a un proceso de deliberación que las pondera y produce como resultado las intenciones, las cuales se incorporan a los planes existentes del agente...

Como debilidades principales de la arquitectura se puede mencionar que: (1) el proceso de deliberación de un agente BDI debe ser ajustado en función de la variabilidad del

[4]Se prefiere hablar de creencias en lugar de conocimiento porque se asume que las creencias que un agente posee pueden contener cierto grado de imprecisión, mientras que el conocimiento se supone debiera ser correcto. Además, los agentes tienen una información limitada sobre el entorno: no tienen ni presencia ni omnisciencia, con lo cual su conocimiento sobre el mundo intenta aproximarse al estado más probable del mismo.

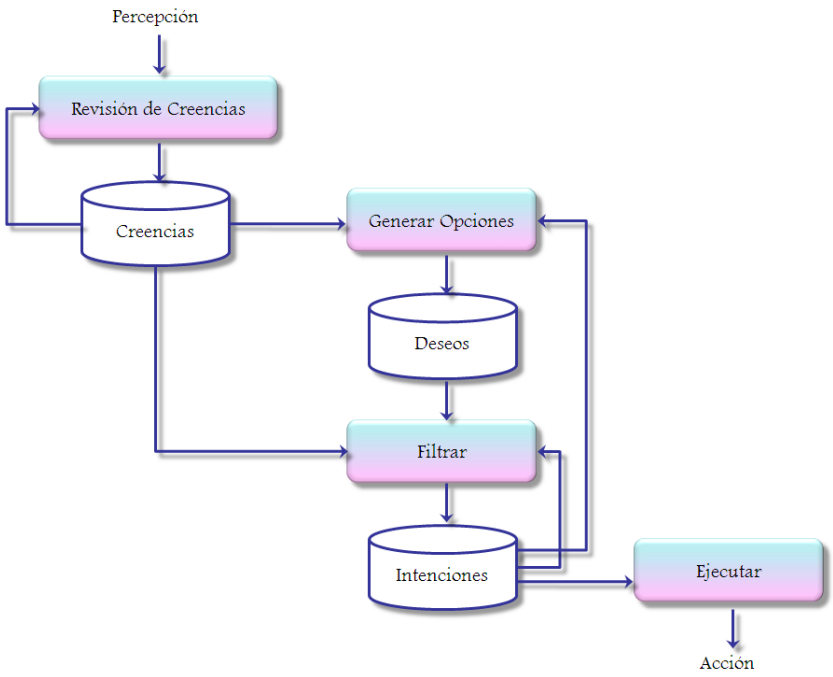


Figura 3.5: Arquitectura BDI. Adaptada de (Dennett, 1987).

entorno, con lo cual no existe una "adaptabilidad" del agente al mismo; (2) la mayoría de las implementaciones de la arquitectura no consideran una representación explícita de las metas (Alexander et al., 2005); (3) los agentes BDI carecen usualmente de mecanismos que les permita aprender de las experiencias pasadas a fin de adaptarse en situaciones futuras (Toan et al., 2005), ni tampoco soportan un esquema de aprendizaje social en entornos multi-agente (Guerra et al., 2004), entre otras.

3.2.1.7. Arquitectura ATLANTIS: Una aproximación reactiva - deliberativa

La arquitectura ATLANTIS se propone como un mecanismo de control interno para sistemas robóticos autónomos, desarrollado en el Instituto Tecnológico de California. En respuesta a las debilidades patentes identificadas comúnmente en las dos tendencias clásicas de diseño de sistemas robóticos (i.e., la aproximación reactiva vs. la deliberativa), Gat (Gat, 1997) plantea como alternativa plausible una arquitectura híbrida de tres capas[5].

La arquitectura propuesta coincide, en cuanto a su estructura, con otras soluciones coetáneas que atienden el mismo tipo de problemas, mediante la definición de: un mecanismo de control reactivo de respuesta inmediata, un planificador deliberativo con tiempos de respuesta más prolongados, y un mecanismo secuenciador que interconecta los otros dos. Sin embargo, a diferencia de otras implementaciones, ATLANTIS emplea en el nivel intermedio los denominados "Paquetes de Acción Reactivos" (*Reactive Action Packages - RAPs*), una solución basada en la propuesta de Firby (Firby, 1989) que se

[5]Cabe resaltar que es una arquitectura híbrida en el sentido que integra niveles de reactividad y deliberación, aunque sigue perteneciendo al grupo de arquitecturas de enfoque cognitivista.

encarga de completar las acciones abstractas propuestas por los planes del nivel deliberativo con acciones más concretas del nivel reactivo. La arquitectura de ATLANTIS se presenta en la figura3.6.

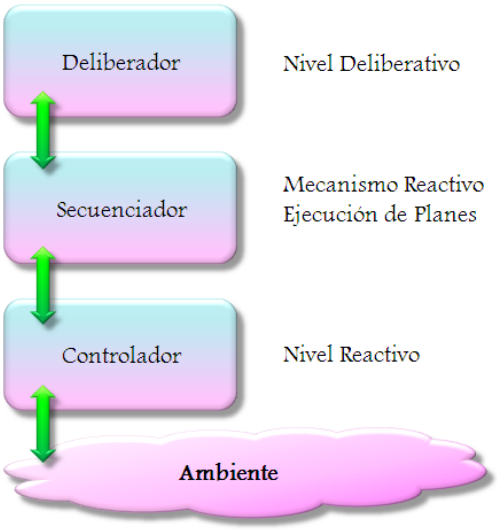


Figura 3.6: Arquitectura Atlantis. Adaptada de (Gat, 1997).

Básicamente, la arquitectura consta de un nivel reactivo, denominado *controlador* que, por medio de una función de control (función de transferencia) comunica los sensores con los efectores y viceversa. Este primer nivel está especialmente diseñado para dar respuesta a los comportamientos primitivos el agente, los cuales imponen intrínsecamente una serie de restricciones: por ejemplo, el tiempo de ejecución no debe superar el límite establecido por una constante lo suficientemente pequeña como para terminar cada ciclo cognitivo de forma estable[6], sus algoritmos deben estar diseñados para detectar (y no evitar) los fallos presentados en la implementación de su función de control, entre otras.

El segundo nivel, llamado *secuenciador*, sirve como nexo entre el nivel inferior reactivo y el nivel superior deliberativo. Su objetivo es seleccionar el comportamiento primitivo que el *controlador* debería usar en un momento dado, y proporcionar los parámetros adecuados para dicho comportamiento. Este nivel también está constreñido a un tiempo de ejecución: el *secuenciador* no debe realizar cómputos que superen determinados umbrales de tiempo.

En el nivel superior, denominado *deliberador*, es donde se realizan las operaciones que requieren de un alto consumo de tiempo (e.g., el razonamiento y la planificación). Este nivel parte de la premisa de que el entorno puede haber cambiado desde el momento en que se invocó el algoritmo de planificación hasta el momento en que se produjo el resultado, aspecto que se resuelve mediante el empleo de heurísticas que facilitan la replanificación. La ejecución de los planes es interrumpible, y por tanto puede ceder el control a aquellas acciones con prioridad más alta. El *secuenciador* emplea un mecanismo

[6]Esta restricción hace referencia a la limitación que posee la propiedad de racionalidad planificada (Gat, 1997).

de "liberación" que asegura la disponibilidad de todos los recursos cuando se produce una interrupción.

Como debilidades principales del modelo se pueden mencionar: (1) la ausencia de un nivel social que permita la cooperación multi-agente; (2) la carencia de mecanismos que reflejen las motivaciones del agente; (3) la inexistencia de un mecanismo de meta-cognición que se encargue de orquestar los diferentes niveles de la arquitectura, entre otros. A fin de abordar algunas de esta debilidades, en (Qureshi et al., 2004) se propone una aproximación híbrida que integra ATLANTIS con BDI.

3.2.1.8. Arquitectura COGNITIVA: Una arquitectura influida por emociones

Internamente, COGNITIVA (Imbert, 2005) puede ser considerada como una arquitectura híbrida, en el sentido que combina capacidades reactivas, deliberativas y sociales. Posee tres niveles cuasi horizontales, a saber: un *nivel reactivo* que proporciona respuestas inmediatas a los cambios percibidos en el entorno; un *nivel deliberativo* que proporciona al agente comportamientos guiados por metas, desde el punto de vista de sus capacidades individuales; y un *nivel social* que dota al agente de comportamientos en los que se considere la existencia de otros y la interacción con ellos. La figura 3.7 resume el ciclo de funcionamiento de la arquitectura genérica.

Para la manipulación del estado actual del agente y sus creencias, COGNITIVA define una taxonomía que diferencia, en un primer nivel, el objeto de la creencia: recintos –físicos, conceptuales o virtuales–, objetos, individuos, y la situación actual. A su vez, las creencias del agente relacionadas con recintos, objetos e individuos se clasifican en: creencias acerca de las características definitorias (DC) de recintos, objetos, o individuos; creencias acerca estados transitorios (TS); creencias relativas a las actitudes; y creencias acerca de la situación actual.

De entre todas las creencias, cobran especial importancia para una arquitectura emocional aquellas que constituyen lo que se ha denominado el *modelo personal del agente*. El modelo personal recoge las creencias que el agente posee acerca de sí mismo. Las principales son los *rasgos de personalidad*, DCs cuyos valores determinan el comportamiento coherente y estable del agente; las *emociones*, TSs que identifican el estado de ánimo de un individuo en un instante concreto; y los *estados físicos*, TSs que expresan el estado del"cuerpo" de un individuo en un instante concreto.

Para la actualización eficiente de creencias e historia pasada, COGNITIVA incorpora expectativas, inspiradas en la propuesta de (Seif El-Nasr et al., 2000a), adaptadas, a su vez, del modelo OCC (Ortony et al., 1988). Las expectativas capturan la predisposición del agente hacia los acontecimientos que tienen o pueden tener lugar. A partir de la consideración de las expectativas acerca de un acontecimiento, la confirmación de su ocurrencia o su no ocurrencia desembocará en la producción de un conjunto rico de emociones.

El mecanismo que incorpora COGNITIVA para controlar el comportamiento reactivo es el de los *intereses*, que amplían el concepto de los *motivadores* de los modelos de emoción motivacionales (Cañamero, 1997). Los intereses, mantenidos por los niveles deliberativo y social, aunque accedidos por los tres niveles, representan el rango de

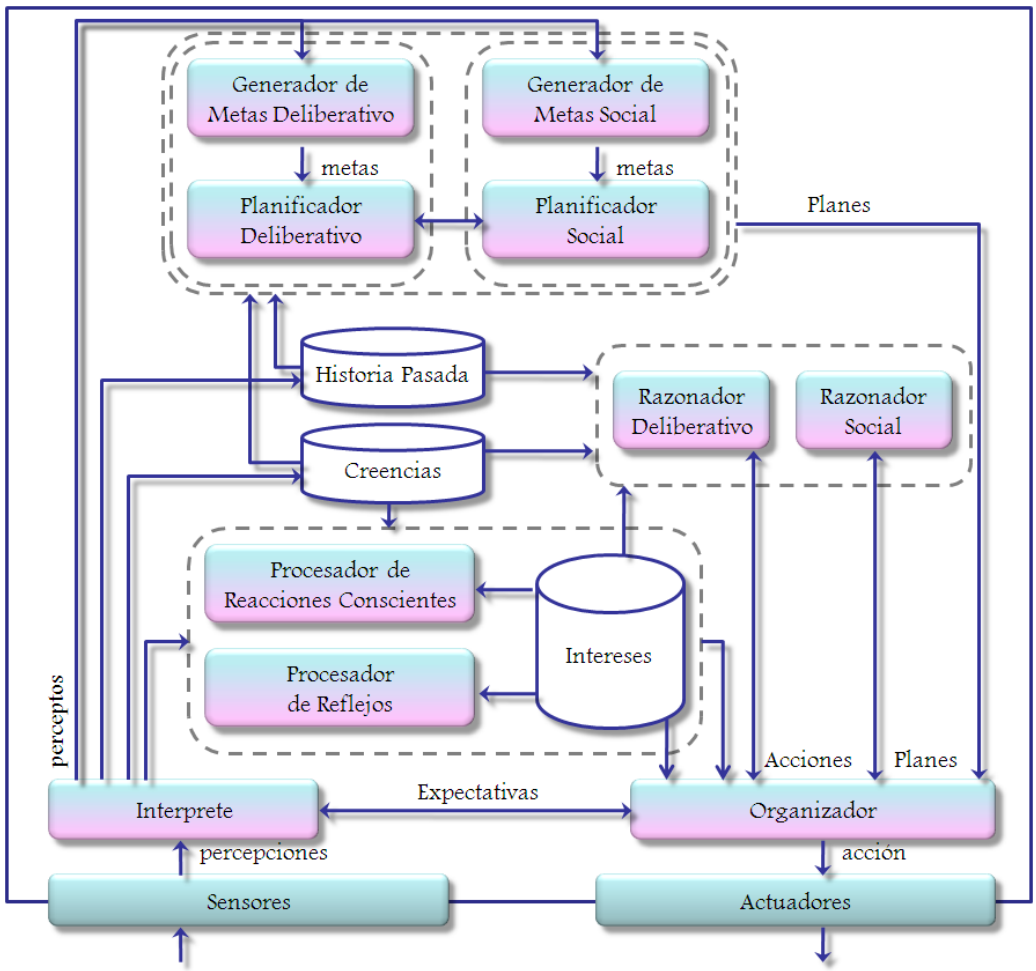


Figura 3.7: Arquitectura COGNITIVA. Ciclo de operación interno. Adaptada de (Imbert, 2005).

valores deseables para cada uno de los estados transitorios del agente, en un instante determinado.

Por otro lado, la arquitectura COGNITIVA proporciona dos mecanismos para afrontar los comportamientos reactivos: Procesamiento de reflejos y Procesamiento de reacciones conscientes. Adicionalmente, COGNITIVA propone, por un lado, la participación de un *planificador deliberativo*, que aporta acciones a los planes desde la perspectiva de las capacidades individuales del agente, y por otro lado, la de un *planificador social*, que se apoya en las posibilidades de interacción con otros individuos para proponer acciones para los mismos planes. Ambos planificadores cuidarán de mantener la coherencia entre los comportamientos deliberativos-sociales y los reactivos. Cada vez que un planificador proponga una acción deberá ajustar debidamente el valor de los umbrales de los intereses del agente, si fuera preciso, para asegurar dicha coherencia.

3.2.1.9. Arquitectura DCA: Arquitectura de Control Dinámico

La arquitectura de control dinámico DCA (por su acrónimo en inglés de *Dynamic Control Architecture*) propuesta por (Hayes-Roth et al., 1989; Hayes et al., 1995), está compuesta por un conjunto de sistemas independientes para la cognición, la percepción, y la actuación. Todos los sub-sistemas operan de forma concurrente y asíncrona, y se comunican a través de una Interfaz de comunicaciones globalmente accesible (IC). Su estructura modular subyacente divide el espacio del problema en sub-espacios, con lo cual se aceleran los tiempos de respuesta y se reduce la complejidad general que debe tratar cada sub-sistema.

Los módulos de entrada y salida de la arquitectura poseen espacios de memoria (buffers) de tamaño limitado, y filtros perceptuales que son modificables dinámicamente, ambos determinados por el componente de cognición. Los buffers de entrada, localizados en la IC, son alimentados con información variable dependiendo del filtro perceptivo, lo que implica que cuanto más relevante sea la información, mayor cantidad de atención recibirá del agente. Sin embargo, debido a que la IC tiene un buffer de tamaño limitado, el sistema cognitivo no enfoca su atención con mucha frecuencia sobre la información del buffer, con lo cual muchos eventos pueden suceder en el mundo sin ser percibidos por el agente.

Por otro lado, el módulo cognitivo puede dividirse en dos sub-sistemas. El primero es una base de conocimiento que contiene todo tipo de representaciones conceptuales, incluyendo conocimiento factual, procedimental, y de resolución de problemas. El segundo es un ciclo de razonamiento que está compuesto por tres componentes: un *administrador de agenda* responsable de identificar y dar prioridad a las tareas, un *planificador* que puede interrumpir al administrador de agenda cuando éste está "listo" a fin de programar la siguiente mejor acción sobre el tercer componente, el *Ejecutor*, encargado de procesar las acciones. La arquitectura se ilustra en la figura 3.8.

Las operaciones que lleva a cabo DCA pueden tener múltiples efectos, entre los cuales se puede mencionar: la modificación de los filtros perceptuales y de las acciones deseadas, la incorporación de nuevas conclusiones durante el razonamiento en curso, etc. Adicionalmente, DCA puede aplicar múltiples métodos de razonamiento sobre el mismo problema y trabajar sobre múltiples problemas simultáneamente.

El agente define un plan de control de acuerdo con sus metas, el cual almacena en la base de conocimiento y lo recupera nuevamente cuando necesita enfocar el ciclo de razonamiento sobre la ejecución de una tarea. Este proceso implica la determinación de qué acciones tienen mayor prioridad en la agenda, cuándo puede ser interrumpido el planificador, y qué información pueden contener los filtros perceptuales. Los únicos cambios que se pueden efectuar en el plan de control son aquellos que realiza el agente tomando en cuenta las condiciones ambientales para una situación específica. Esto le permite al agente mantener cierta coherencia global en cuanto a su razonamiento, su percepción, y su actuación, mientras que se mantiene activa la capacidad reactiva del agente.

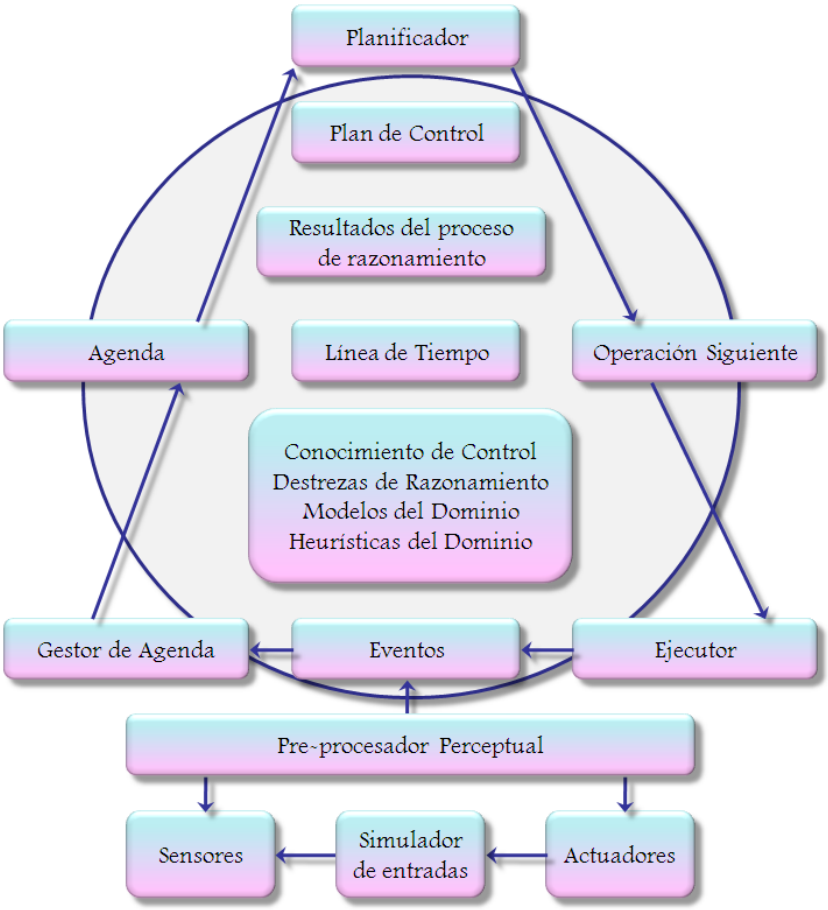


Figura 3.8: Arquitectura DCA. Adaptada de (Hayes et al., 1995)

3.2.1.10. Arquitectura ERE: Integración de Planes, Agenda, y Control

La Arquitectura ERE (Drummond et al., 1991) (por su acrónimo en inglés de *Entropy Reduction Engine*), está conformada por tres componentes principales (como se observa en la figura 3.9):

- **Componente Reductor**: este componente sintetiza eficientemente un conjunto de estrategias para la resolución de problemas. Adicionalmente, emplea un conjunto de "restricciones de comportamiento", definidas por el usuario, a partir de las cuales es posible descomponer el problema en unidades más pequeñas que procesa luego el Proyector.
- **Componente Proyector:** emplea las estrategias del componente Reductor para filtrar la búsqueda y programar en la agenda las acciones más apropiadas. Para esto, el Proyector emplea la descomposición del problema generada por el Reductor, y la "teoría casual" que provee el usuario. La teoría casual está compuesta por un conjunto de acciones, eventos exógenos, y restricciones del dominio. Internamente, el Proyector construye un grafo con los posibles comportamientos futuros, luego

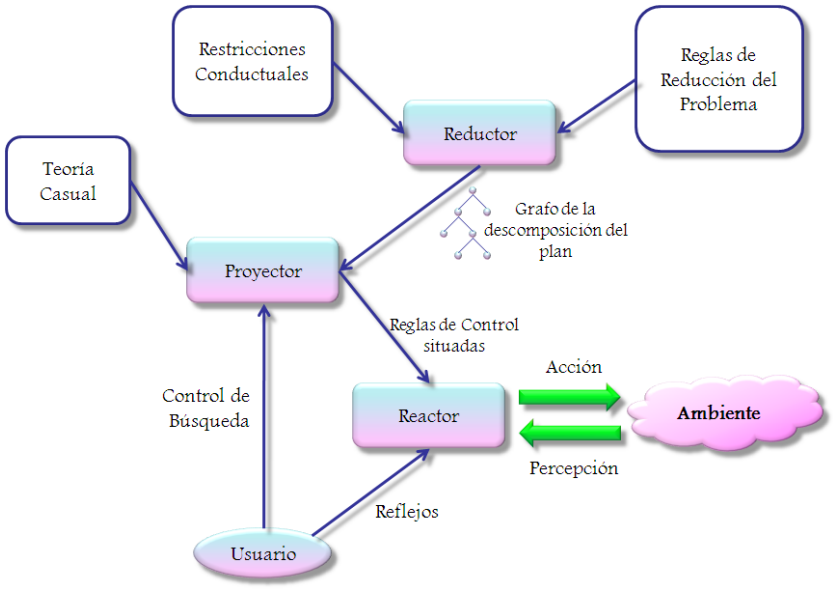


Figura 3.9: Arquitectura ERE. Adaptada de (Drummond et al., 1991).

las acciones son evaluadas con respecto al grafo, y por último el grafo se va acotando progresivamente mediante las restricciones. Una vez el grafo se encuentra constreñido de forma conveniente, es compilado por el Reactor.

- **Componente Reactor:** ejecuta una serie de reglas de control derivadas de los planes del componente Proyector. El reactor está acoplado con los sensores y efectores del agente, lo cual permite comparar los datos sensoriales con las condiciones de las *reglas de control*. Si la condición de una regla de control es activada, entonces las acciones de dicha regla son ejecutadas por los efectores. Las reglas de control pueden definidas tanto por el usuario como por el Proyector.

Los componentes Reductor, Proyector, y Reactor se ejecutan asíncronamente y su actividad es coordinada por un controlador centralizado. La importancia de ERE no recae en la construcción de un plan básico, sino en la habilidad del sistema para gestionar predictivamente la ejecución de los planes a la luz de una retroalimentación continua del ambiente y de la capacidad para re-planificar cuando ocurren fallos.

3.2.1.11. Arquitectura HOMER: Integración de facultades cognitivas

La filosofía subyacente de la Arquitectura HOMER, es la "integración". Vere y Bickmore (Vere y Bickmore, 1990) argumentan que la investigación aislada de las facultades cognitivas (e.g., planificación, aprendizaje, etc.) ha progresado lo suficientemente como para permitir la creación de una arquitectura cognitiva integrada con características interesantes.

HOMER es una arquitectura modular, que se compone de una memoria, un planificador, un procesador de lenguaje natural, unos cuantos mecanismos de monitorización,

y un ejecutor de planes. Este diseño modular se debe a su intención de integrar múltiples áreas de investigación en el campo de la Inteligencia Artificial, en un solo agente cognitivo. En HOMER no existe realmente un módulo central de control, aunque se podría argumentar que el planificador cumple con gran parte de esta función.

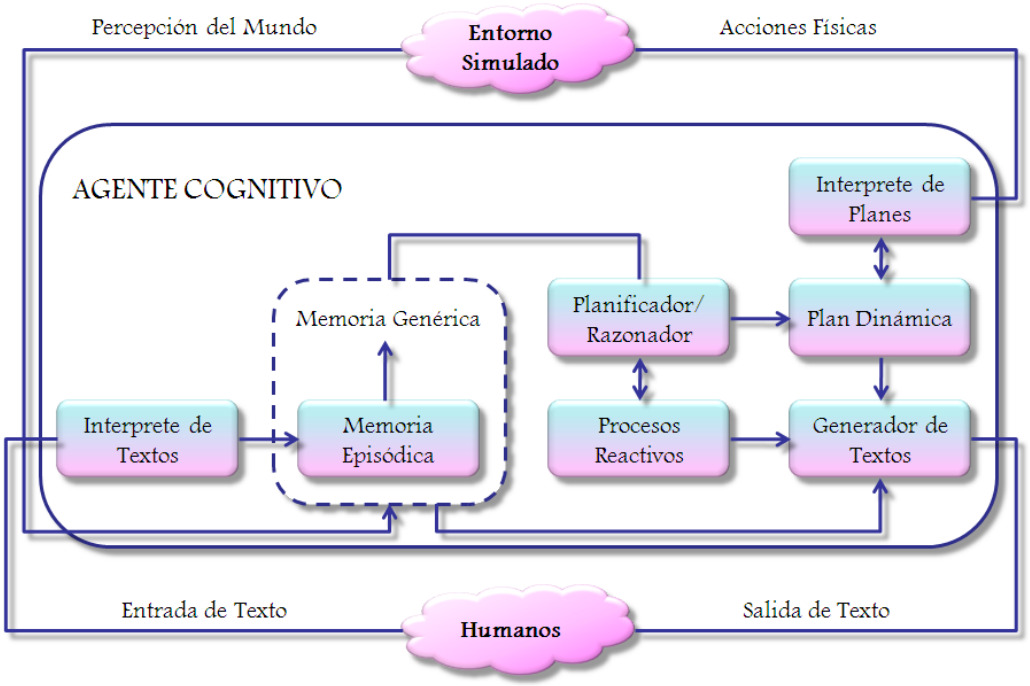


Figura 3.10: Arquitectura HOMER. Adaptada de (Vere et al., 1990).

Como se puede observar en la figura 3.10, la memoria puede dividirse conceptualmente en tres sub-componentes: el *Conocimiento General* que incluye conocimiento del mundo y del dominio del problema, el *Conocimiento Léxico*, y el *Conocimiento Episódico*. El conocimiento léxico no es contenido dentro de la memoria general sino que se encuentra en un repositorio separado similar a una enciclopedia léxica.

Tanto el módulo planificador como el módulo de razonamiento se derivan de la arquitectura DEVISER V (Vere, 1991). El Ejecutor de Planes funciona en común acuerdo con el proceso Monitor (o proceso reflexivo) a fin de llevar a cabo la ejecución satisfactoria de un plan. Luego de esto, el proceso Monitor puede proveer información de retroalimentación al Ejecutor de Planes en caso de que sea necesaria una re-planificación como consecuencia de los cambios ambientales.

3.2.1.12. Arquitectura MAX: Una arquitectura basada en marcos lógicos y reglas

En la arquitectura MAX (Kuokka, 1991) todo el conocimiento es almacenado de forma declarativa en estructuras denominadas marcos lógicos (*lframes*). Cada marco denota un posible "estado" conformado por un conjunto de premisas lógicas. Estos marcos pueden estar compuestos de otros marcos, e incluso pueden definir variables locales. Asociadas

a los marcos se encuentran las *reglas*, quienes indican al planificador MEA[7] el momento más apropiado para activarse.

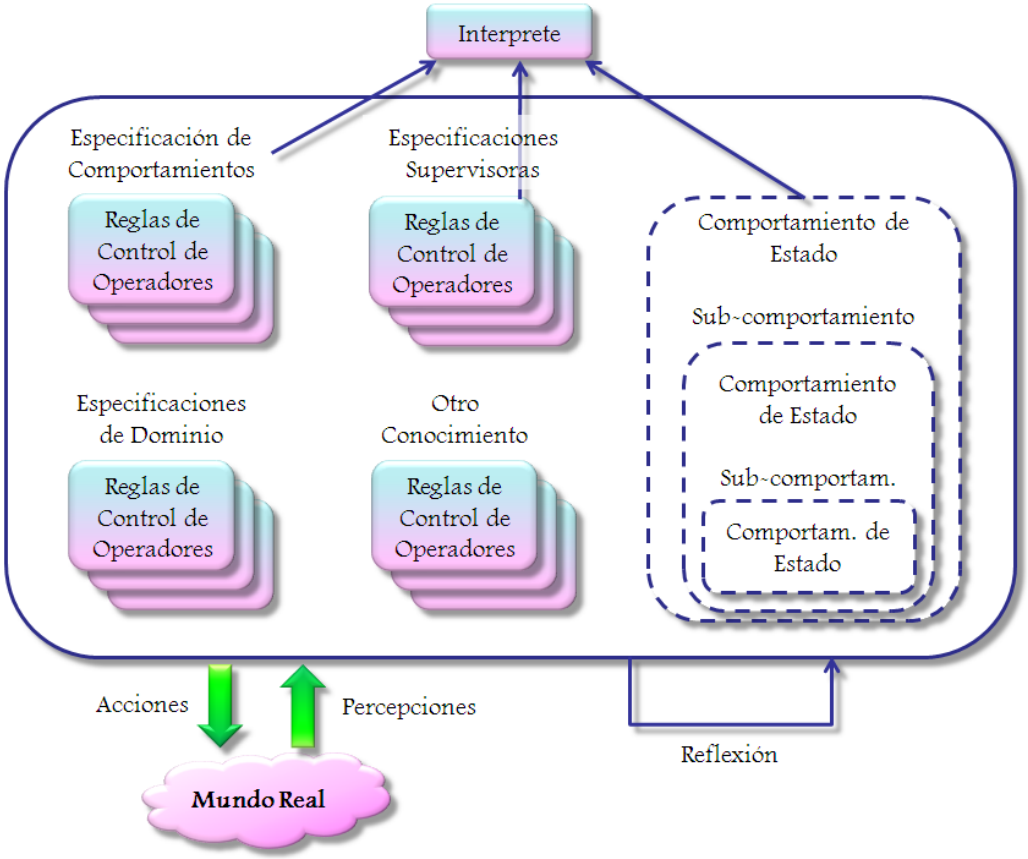


Figura 3.11: Arquitectura MAX. Adaptada de (Kuokka, 1991).

MAX define un sistema de producción cuyo motor de inferencia empareja las condiciones de las reglas con las premisas que existen en la memoria de trabajo a fin de inducir nuevas conclusiones. En términos generales, las estructuras de conocimiento del sistema de producción se organizan de acuerdo con su propósito, esto es, se definen bases de conocimiento diferentes para: los comportamientos, el dominio, las especificaciones de monitorización, la percepción, y la actuación. El motor de inferencia de MAX, denominado el *Interprete*, se encarga de controlar el proceso de razonamiento mediante el encadenamiento hacia adelante de las reglas, obteniendo como resultado la activación de un comportamiento.

Todas las facultades cognitivas que exhibe MAX son procesadas por comportamientos específicos, motivo por el cual sólo puede estar activo un comportamiento en cada instante. Cada comportamiento pueden invocar a otro recursivamente empleando para

[7]El Análisis de Medios-Fines MEA (por su acrónimo en inglés de *Means-Ends Analysis*), es una estrategia de resolución de problemas introducida inicialmente por (Newell y Simon, 1995) en su Solucionador General de Problemas. En MEA el proceso de búsqueda sobre el espacio del problema es desempeñado de dos formas: un encadenamiento hacia atrás cuando se requiere operar sobre las sub-metas, y un encadenamiento hacia adelante cuando se aplican los operadores.

ello una pila de comportamientos activos. Por otro lado, los Monitores son módulos similares a los comportamientos con la diferencia de que son invocados automáticamente cuando se alcanza alguna condición ambiental pre-definida. Estos Monitores son empleados por el agente a fin de responder de manera oportuna a las exigencias del ambiente dinámico.

Kuokka (Kuokka, 1991) afirma que la arquitectura posee módulos pre-fabricados (e.g., los comportamientos y los monitores, como se observa en la figura 3.11) que se encargan de tareas tales como: enfocar la atención del agente, definir múltiples estrategias de resolución de problemas, monitorizar la ejecución de las acciones, explorar nuevas metas, coordinar el aprendizaje basado en explicaciones, depurar los comportamientos, interrumpir los procesos, y reanudar de forma inteligente los procesos interrumpidos.

Una de las aplicaciones más conocida de esta arquitectura ha sido una simulación robótica, donde el agente debía alcanzar múltiples metas (algunas de ellas conflictivas). Kuokka demostró que el robot era capaz de enfocarse en tareas con mayor importancia, y luego continuar con las tareas pendientes menos relevantes (Kuokka, 1991). Una de las principales debilidades de esta arquitectura es la ausencia de mecanismos que actualicen los módulos de aprendizaje y de razonamiento, ya que éstos deben ser definidos a priori por el programador.

3.2.1.13. Arquitectura PRODIGY: integración de mecanismos de aprendizaje

La arquitectura PRODIGY (Carbonell et al., 1991; Veloso et al., 1996) fue diseñada inicialmente como el núcleo de un solucionador de problemas el cual estaba estrechamente interconectado con múltiples módulos de aprendizaje. Esta característica le permite a PRODIGY actuar como una plataforma de pruebas donde es posible validar la eficiencia de diversos mecanismos de aprendizaje. Cabe resaltar que el propósito de PRODIGY no es intentar emular la cognición o el razonamiento humano, sino idealizar los procesos de la cognición en general. Los módulos principales de PRODIGY se ilustran en la figura 3.12, y se describen brevemente a continuación:

- PRODIGY emplea un lenguaje llamado PDL (*Prodigy Description Language*), basado en lógicas de primer orden, a fin de expresar todo el conocimiento del sistema mediante una base representacional homogénea y declarativa, lo cual facilita la comunicación entre los distintos módulos de la arquitectura.
- La arquitectura emplea un conjunto de reglas de producción para representar el conocimiento de control (o meta-conocimiento). Estas reglas de control pueden ser aprendidas por el agente, o predeterminadas por el diseñador.
- PRODIGY es una arquitectura modular que permite la escalabilidad del sistema. Además, todos los módulos comparten el mismo solucionador central y la misma estructura de datos.
- La arquitectura define un solucionador de problemas que es guiado por las reglas de control, las cuales le permiten conservar la independencia del dominio. En ausencia de conocimiento de control, PRODIGY emplea el Análisis de Medios-Fines (MEA) implementado mediante una estrategia de búsqueda del *"primero en profundidad"*.

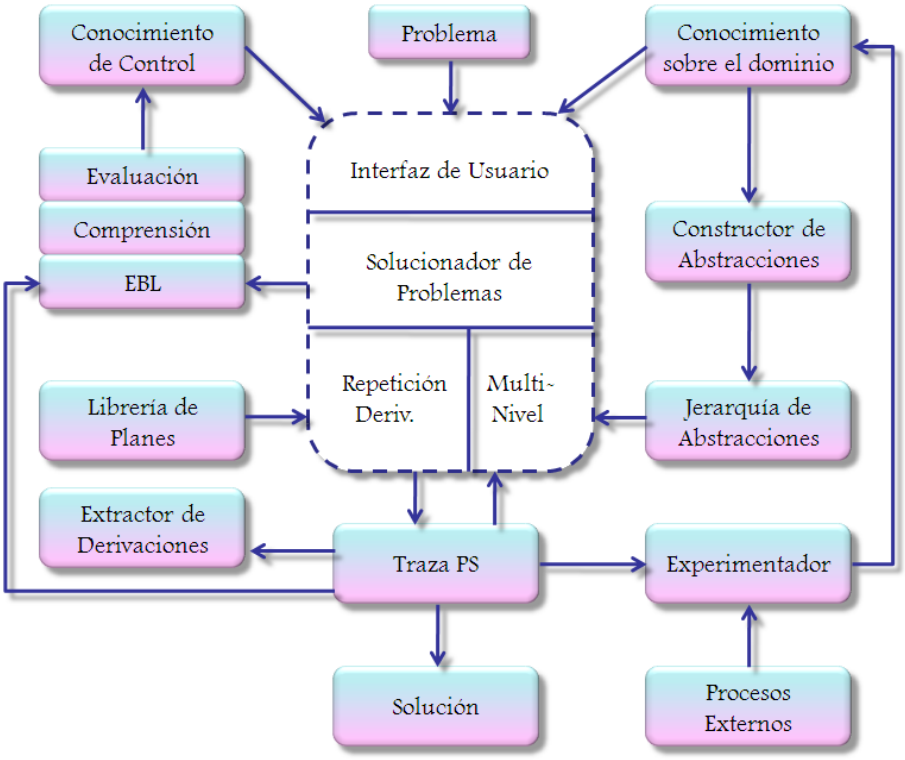


Figura 3.12: Arquitectura PRODIGY. Adaptada de (Carbonell et al., 1991).

- Los múltiples métodos de aprendizaje que define la arquitectura son:
  - *EBL*: el Aprendizaje Basado en Explicaciones adquiere nuevas reglas de control a partir de la propia traza generada durante la solución del problema.
  - *Analogía derivacional y Aprendizaje Basado en Casos:* en este tipo de aprendizaje, las soluciones a problemas pasados son usadas para resolver nuevos problemas mediante analogía.
  - *Análisis estático:* este módulo de aprendizaje adquiere nuevo conocimiento de control, pero a diferencia de EBL no requiere un entrenamiento previo de los conceptos objetivo. Las reglas se basan en una inspección directa que se realiza sobre el conocimiento del dominio.
  - *Aprendizaje por experimentación:* el mecanismo de experimentación es activado cuando el proceso de ejecución de planes detecta una discrepancia entre las expectativas internas y las observaciones externas.
  - *Aprendizaje a través de la interfaz de usuario:* emplea un módulo de procesamiento de lenguaje natural que le permite al usuario dar consejos a PRODIGY sobre la solución de algún problema en particular, como también evaluar su desempeño.
  - *Generador de abstracciones:* este módulo es usado para la planificación jerárquica. Inicialmente, el mecanismo encuentra soluciones generales a los problemas en un espacio abstracto, y posteriormente, guía la búsqueda hacia soluciones específicas en un espacio más acotado.

3.2.1.14. Arquitectura RALPH-MEA: Integración multi-arquitectura

La arquitectura RALPH-MEA (Ogasawara y Russell, 1994) (por su acrónimo en inglés de *Rational Agents with Limited Performance Hardware-Multiple Execution Architecture*), es un sistema que integra las fortalezas de múltiples enfoques arquitectónicos. Concretamente, RALPH-MEA integra cuatro *Arquitecturas de Ejecución (AE)* que operan en paralelo sobre el mismo problema. Cada AE emplea diferentes tipos de representación del conocimiento y diferente lógica de procesamiento:

- **Arquitectura para las Decisiones Teóricas**: esta arquitectura emplea conocimiento sobre el estado actual, los efectos de las acciones, y las utilidades de los estados proyectados, a fin de predecir las situaciones futuras y producir planes optimizados como consecuencia. RALPH-MEA emplea esta arquitectura en situaciones donde existe cierta holgura en los tiempos de respuesta, y por ende puede generar planes más extensivos.
- **Arquitectura Basada en Metas**: esta arquitectura emplea conocimiento sobre el estado actual, los efectos de las acciones, y la mejor acción del estado actual a fin de lograr la meta vigente. Esta arquitectura puede también predecir situaciones futuras, y RALPH-MEA la utiliza en situaciones donde se cumplen los siguientes criterios: el agente debe alcanzar una única meta final, existe conocimiento sobre la mejor acción a ejecutar en cada paso, y existe tiempo suficiente para el proceso de planificación.
- **Arquitectura de Utilidad de Actuación**: esta arquitectura emplea conocimiento sobre el estado actual y las métricas de utilidad. RALPH-MEA utiliza esta arquitectura cuando dispone de estadísticas apropiadas sobre la utilidad de cada unidad de conocimiento, lo que le permite tomar decisiones de manera más rápida en comparación con las otras arquitecturas enfocadas a la planificación
- **Arquitectura Condición-Acción**: esta arquitectura emplea reglas del tipo "si (estado) entonces (acción)" a fin de proveer al agente con un conjunto de decisiones rápidas que se ejecutan en una cantidad mínima de tiempo. Esto resulta ser útil en situaciones que demandan una alta reactividad.

Por otro lado, RALPH-MEA emplea un módulo de meta-razonamiento que integra los resultados de cada AE y produce una decisión final. Cabe destacar que todas las AEs comparten la misma base de conocimiento, por tanto, para que la información pueda ser interpretada por cada tipo de AE se requiere de un proceso de compilación para cada caso, ejecutado por el módulo de meta-razonamiento. Por ejemplo, los planes que genera la AE basada en Metas pueden ser compilados en reglas del tipo *si-entonces* que interpretará luego la AE de Condición-Acción. Esta compilación podría verse como una forma de modular los procesos de aprendizaje dentro de la arquitectura. Los componentes de la arquitectura RALPH-MEA se pueden observar en la figura 3.13.

Por último, RALPH-MEA posee un módulo de planificación que monitoriza la ejecución de los planes producidos por las AEs. El *Ejecutor de Planes* dirige los actuadores del agente, mientras que el *Proceso Monitor* supervisa el ambiente. El módulo de *Análisis del Impacto en el Plan* analiza los efectos del plan producidos en el ambiente y puede activar el proceso de re-planificación si lo considera necesario.

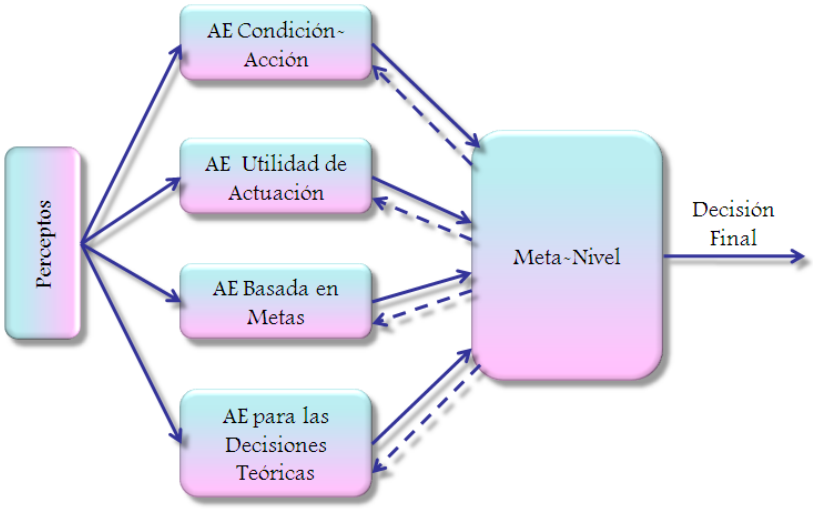


Figura 3.13: Arquitectura RALPH-MEA. Adaptada de (Ogasawara et al., 1994).

3.2.1.15. Arquitectura TETON: integración de múltiples memorias

La arquitectura TETON (VanLenh y Ball, 1991; Nokes et al., 2008) está compuesta por tres partes: (1) la *Memoria de Largo Plazo* encargada de almacenar los procedimientos (operadores) y las reglas de control; (2) la *Memoria de Trabajo* que se ocupa del almacenamiento de los resultados actuales del ciclo de ejecución, los cuales se traducen luego en metas; y (3) el *Ciclo de Ejecución* durante el cual el mecanismo de razonamiento aplica las reglas de control sobre la memoria de largo plazo a fin de seleccionar un operador. El operador seleccionado es luego interpretado, y sus resultados parciales son almacenados en la memoria de trabajo.

La memoria de largo plazo es una base de conocimiento que almacena los operadores, las reglas de control, y los hechos generales acerca del mundo. Cada operador define ciertos criterios de selección y un conjunto de pre-condiciones. Si las pre-condiciones se cumplen, entonces se interpreta el resto del operador y se decide si se deben almacenar elementos en la memoria de trabajo o si se debe crear deliberadamente una sub-meta. Cuando una meta ha sido "olvidada" por el agente, el operador correspondiente puede informar su estado al Interprete, quien re-dirigirá nuevamente el foco de atención hacia la solución de dicha meta.

Por otro lado, las reglas de control pueden ser de tres tipos: *Reglas de Consideración*, *Reglas de Rechazo*, y *Reglas de Preferencia*. Estas reglas definen, como sus nombres lo sugieren, un conjunto de pautas para: considerar aspectos específicos sobre el mundo, rechazar o descartar información irrelevante, y crear un orden jerárquico de preferencia sobre el resto de reglas.

TETON almacena las metas del agente en la memoria de trabajo como estructuras conformadas por ranuras. Existe una ranura para indicar el estado temporal de la meta (e.g., una meta pasada, presente, o futura), otra ranura para indicar el estado que debe ser alcanzado por la meta, y una última ranura donde se guarda un enlace al estado en el

cual fue creada dicha meta. Además de las metas, la memoria de trabajo almacena otros tres tipos de información, como sigue: una *memoria situacional* donde se almacena información relativa al ambiente, una *memoria de capacidad limitada* donde se almacenan los eventos importantes y recientes, y una *memoria temporal* empleada para dirigir la búsqueda de acciones durante la planificación.

Durante el ciclo de ejecución, lo primero que hace TETON es interpretar las reglas de control y ejecutar el cuerpo del operador seleccionado. En aquellas situaciones donde no se encuentra ningún operador aplicable, el sistema crea un *impasse*. Los *impasses* son resueltos mediante múltiples métodos, incluyendo algunos basados en fundamentos psicológicos tales como omitir el *impasse* o escoger aleatoriamente un operador (VanLenh y Ball, 1991). Una propiedad importante de este método es que no permite la selección LIFO[8], lo que implica que cualquier meta, y no solamente la última creada, puede ser seleccionada para ser la meta actual del sistema.

Por último, el ciclo de ejecución puede verificar además si la meta actual tiene alguna "condición de atajo". La presencia de una condición de atajo advierte que la meta ya ha sido alcanzada y que el ejecutor está trabajando sobre lo que se conoce como "reconstrucción de metas", situación que ocurre cuando una o más metas realizadas previamente se deshacen. En este caso, la meta alcanzada no necesita ser ejecutada y simplemente se etiqueta como completada.

3.2.1.16. Arquitectura Theo-Agent: Aprendiendo a partir de las explicaciones

La arquitectura Theo-Agent (Mitchell et al., 1991) fue diseñada como una plataforma para sistemas "auto-modificables" que integra la resolución de problemas, el aprendizaje de nuevas experiencias, y la representación del conocimiento. Los objetivos principales de esta arquitectura apuntan hacia un diseño escalable en el cual pueden ser agregados fácilmente nuevos módulos de aprendizaje, métodos eficientes para representar el conocimiento, y diversos mecanismos de razonamiento. El flujo de información básico de Theo-Agent se presenta en la figura 3.14 y sus características principales se resumen a continuación:

- **Representación del conocimiento empleando Marcos:** Theo-Agent representa todo el conocimiento, incluyendo las creencias y las estrategias de resolución de problemas, mediante un sistema homogéneo de marcos. Así pues, la solución de un problema se reduce a la búsqueda de un valor para una ranura en particular del marco, empleando para ello el conocimiento disponible.
- **Mecanismo de Inferencia**: debido a que el proceso deliberativo del sistema no es activado a menos que se requiera encontrar un valor para una ranura, se podría decir entonces que Theo-Agent es una arquitectura dirigida por *impasses*. La base de conocimiento sobre la cual razona Theo-Agent es aumentada progresivamente mediante un aprendizaje basado en explicaciones que refina el conocimiento previo y mejora la velocidad de inferencia del sistema.

[8] Acrónimo en inglés de *Last In - First Out* - Último en Entrar Primero en Salir. En teoría de colas, guarda la analogía con una pila de elementos.

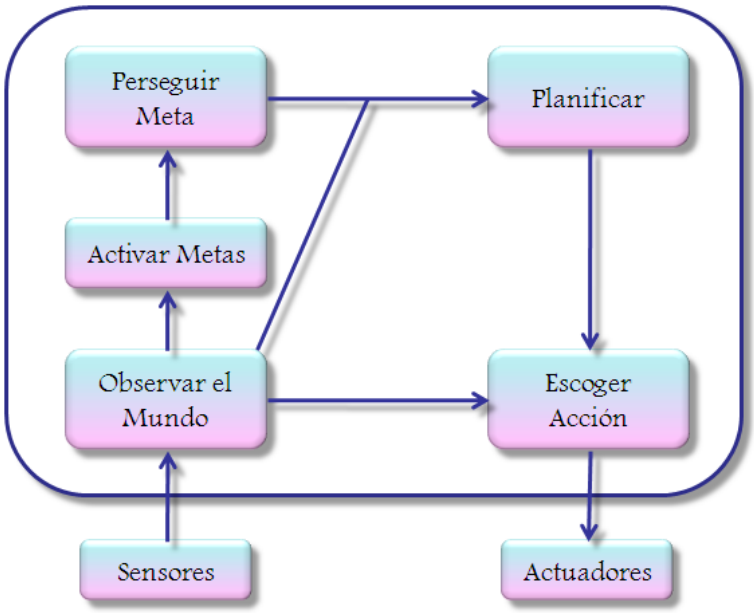


Figura 3.14: Arquitectura Theo-Agent. Adaptada de (Mitchell et al., 1991).

- **Representación Indexada**: el conocimiento sobre la resolución de problemas es indexado de acuerdo con la descripción del problema. Esto permite un fácil acceso al conocimiento más relevante.
- **Sistema de Mantenimiento de la Verdad:** cuando Theo-Agent rellena una ranura de un marco a través del proceso de inferencia, es almacenada además una explicación de la misma, lo cual ayuda a mantener cierta consistencia en el conocimiento adquirido. Adicionalmente, cuando el estado del mundo cambia, Theo-Agent actualiza las dependencias entre los marcos respectivos, conservando de esta manera la consistencia del modelo interno.
- **Aprendizaje basado en explicaciones**: Theo-Agent no sólo puede explicar la cadena inferencial realizada por el sistema, sino que además emplea estas inferencias como un método de aprendizaje basado en explicaciones. Las explicaciones del proceso de inferencia son generalizadas a través de un sistema especial llamado TMAC. Una vez ha sido generalizada una explicación, ésta actúa como un macro-operador que puede ser aplicado en procesos inferenciales posteriores.
- **Ordenamiento de Operadores**: para cualquier problema dado (i.e., un valor de una ranura sin rellenar), existen varios operadores aplicables a través de los cuales se puede inferir el valor de la ranura. Theo-Agent posee un mecanismo de aprendizaje inductivo que conserva un conjunto de estadísticas sobre el éxito de cada operador, sus fallos, y el tiempo de procesamiento que requiere. Basándose en estas estadísticas, Theo-Agent ordena los operadores para minimizar el coste de inferencia esperado.
- **Reglas estímulo-respuesta**: Theo-Agent emplea reglas del tipo estímulo-respuesta a fin de controlar la reactividad del sistema en entornos dinámicos. La ejecución de

estas reglas reactivas pueden ser inhibida por el control de un nivel superior (e.g., el mecanismo de inferencia).

3.2.2. Arquitecturas de Aproximación Emergente

3.2.2.1. Arquitectura AAR y Arquitectura de Subsumción

La arquitectura ARR (Vernon et al., 2007) (por su acrónimo en inglés de *Autonomous Agent Robotics*), así como los sistemas cognitivos basados en comportamientos, representan una alternativa emergente a las aproximaciones cognitivistas. En lugar de definir una descomposición funcional de los procesos de la arquitectura (e.g., la representación de conocimiento, la formación de conceptos, el razonamiento, etc.), la arquitectura ARR se basa en la interacción de múltiples sistemas independientes. Inicialmente, la arquitectura define una estructura que le permite al agente actuar efectivamente en situaciones simples; luego, son agregadas incrementalmente nuevas capas que incorporan sistemas más sofisticados, cada una con la capacidad de subsumir las capas inferiores. Este principio de diseño se basa en la arquitectura de subsumción propuesta por Brooks (Brooks, 1986).

De acuerdo con Brooks, la Arquitectura de subsumción está construida por capas de comportamientos débilmente acoplados entre sí, que operan de forma local y asíncrona, donde las capas de niveles superiores se apoyan en las inferiores para crear comportamientos más complejos: el comportamiento del sistema se contempla como un todo que emerge de la interacción de múltiples comportamientos simples. Por otro lado, Brooks propone (Brooks, 1991) que cada capa o comportamiento de la arquitectura es controlada por una Máquina de Estado Finita Aumentada (MEFA) (véase la figura 3.15). Cada MEFA define un conjunto de señales entrada y de salida que se conectan directamente con los respectivos sensores y actuadores del agente. Cuando la entrada de una MEFA excede un umbral predeterminado, el comportamiento de esa MEFA es activado. Adicionalmente, a través de señales de supresión o de inhibición, una MEFA puede subsumir la acción de otras a fin de que el sistema exhiba un comportamiento coherente. Dentro de las debilidades principales de la arquitectura de subsumción se pueden mencionar:

- La Arquitectura presenta un alto acoplamiento entre la percepción y la actuación, lo que permite un comportamiento reactivo, aunque una capacidad limitada de deliberación.
- Debido a la ausencia de una representación interna del mundo y a la necesidad de definir a priori la conexión entre los comportamientos, la arquitectura tiene capacidades muy limitadas de aprendizaje y de adaptabilidad a nuevos entornos.
- Varias extensiones del modelo de subsumción han intentado incorporar ciertas características de representación interna y de arbitraje de comportamientos, como la Arquitectura de Control Basada en Comportamientos (Mataric, 1992) y los Sistemas de Activación Hormonal (Brooks, 1991), sin embargo, la necesidad de pre-conectar los comportamientos y la ausencia de aprendizaje se mantienen aún persistentes, constriñendo la arquitectura a solamente entornos específicos y preconfigurados, donde priman las conductas reactivas en lugar de las deliberativas.

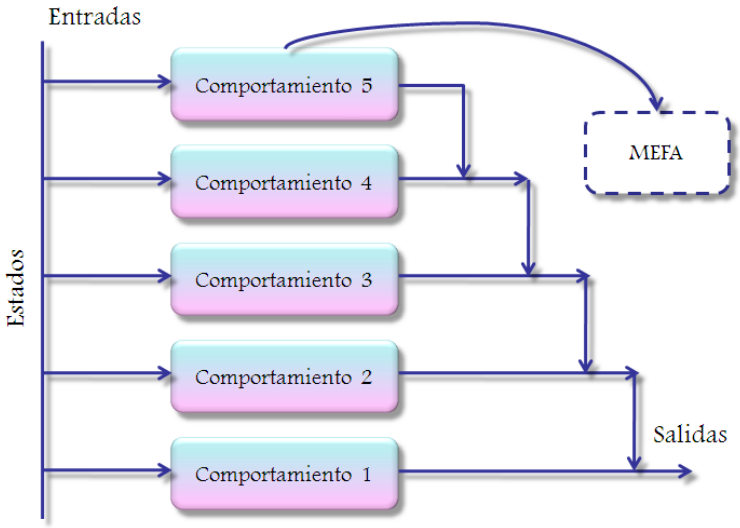


Figura 3.15: Arquitectura de Subsumción. Procesos de Inhibición y Supresión. Adaptada de (Brooks, 1986).

En cuanto a la arquitectura ARR, ésta se basa esencialmente en los principios del modelo de subsumción, aunque propone la sustitución de los módulos MEFA por sub-sistemas de procesamiento más complejos. Pese a las mejoras que supone la inclusión de estos sub-sistemas, AAR continua heredando algunas de las debilidades del modelo original de subsumción. Una de las limitaciones propias de ARR, que se debe a la gran explosión de estados que resultan de la integración incremental de los sub-sistemas, es la dificultad de que emerja un diseño bien estructurado que produzca una actividad coordinada. Esto a su vez impone la necesidad de crear algún tipo de auto-gestión, algo que no se concibe en la Arquitectura de Subsumción original ni tampoco en ARR.

Una segunda limitación es que la arquitectura con el tiempo empieza a presentar problemas en la identificación de las señales ambientales y en la producción de secuencias correctas de acciones o actividades, lo que produce un aumento en la complejidad del desarrollo de la tarea. Por otro lado, AAR no expone adecuadamente una teoría comprehensible sobre la cognición: debido a que el modelo de subsumción define una escalabilidad limitada del sistema, resulta difícil que la arquitectura exhiba facultades cognitivas de orden superior (i.e., el agente no puede explicar el funcionamiento del comportamiento auto-dirigido que emplea), lo que convierte a la arquitectura en un sistema que integra comportamientos complejos pero que sólo presenta cualidades de un sistema reactivo típico.

3.2.2.2. Arquitectura GWM: Competencia y Simulación

Shanahan (Shanahan, 2006) propone una arquitectura cognitiva llamada GWM (por su acrónimo en inglés de *Global Workspace Model*), que es biológicamente plausible e inspirada en el funcionamiento del cerebro humano a nivel neuronal. En GWM, las funciones cognitivas relacionadas con la anticipación y la planificación son llevadas a ca-

bo mediante un mecanismo que simula internamente las interacciones con el ambiente, mientras que la selección de acciones (tanto externas como internas), es mediada por un mecanismo afectivo.

En contraste con la manipulación de representaciones simbólicas y declarativas que realizan habitualmente las arquitecturas de enfoque cognitivista, GWM representa el conocimiento (de forma sub-simbólica) a través de mapas neuronales organizados topográficamente, donde la selección de acciones resulta de la activación de pautas que correlacionan entradas sensoriales del sistema con salidas efectoras específicas. Shanahan argumenta que tales representaciones sub-simbólicas resultan particularmente apropiadas para la cognición espacial (Shanahan, 2006), la cual resulta ser una capacidad cognitiva crucial, aunque significativamente compleja cuando se emplean aproximaciones tradicionales basadas solamente en la lógica.

Shanahan propone, en su arquitectura, un modelo de *Área de Trabajo Global* basado en la teoría de Baars (Baars, 2002), a partir del cual emergen una secuencia de estados que proceden de la interacción de múltiples procesos que operan de forma aislada y en paralelo, como se observa en la figura 3.16.a. Estos procesos especializados cooperan y compiten entre sí a fin de obtener el acceso al área de trabajo global. El (los) ganador(es) de la competición se hace(n) con el control consciente del agente y puede(n) difundir información hacia otros procesos especializados quienes continúan compitiendo por ser los ganadores en el siguiente ciclo de ejecución.

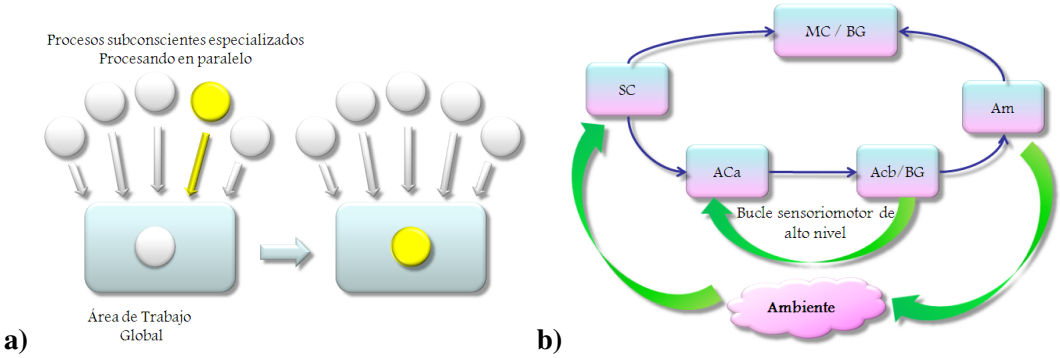


Figura 3.16: Arquitectura de Área de Trabajo Global. a) Coordinación de procesos concurrentes. b) Mecanismo de prospección que emplea la simulación sensoriomotriz. SC: Corteza Sensorial, MC: Corteza Motriz, BG: Ganglios Basales (selección de acciones), AC: Corteza de Asociación, Am: Amígdala (emociones). Adaptada de (Shanahan, 2006).

La arquitectura GWM se compone de dos bucles principales: un bucle sensoriomotriz de primer orden asociado con el mundo externo, y un bucle sensoriomotriz de orden superior conectado internamente con las memorias asociativas, como se observa en la Figura 3.16.b. El bucle de primer orden emula las funciones de la corteza sensorial y del ganglio basal (que controla la corteza motriz), provocando asimismo un arco reflejo de acción-percepción. El bucle de segundo orden comprende dos elementos corticales asociativos los cuales llevan a cabo simulaciones *offline* (es decir, durante un proceso fuera de línea) de las interacciones entre el sistema sensorial y la correspondiente respuesta motriz. La primera corteza asociativa simula una salida motriz, mientras que la segunda

simula los estímulos sensoriales esperados que dan lugar a la siguiente respuesta motriz, y así sucesivamente. Posteriormente, el bucle de mayor orden modula efectivamente la selección de acciones producida por el ganglio basal del bucle de primer orden a través de un componente influido por emociones que simula la función de la amígdala del cerebro.

La arquitectura es capaz de anticipar y planificar los comportamientos futuros a través de un proceso similar al de la "imaginación" (e.g., la simulación asociativa interna de los mecanismos sensoriomotrices del agente). Sin embargo, el modelo de área de trabajo global no corresponde con una área particular de la zona cortical cerebral sino que, por el contrario, presenta las características típicas de una red global de comunicaciones. En la Figura 3.16.b se ilustra grosso modo la manera en que la arquitectura logra coordinar los procesos de prospección y anticipación a través de la simulación sensoriomotriz. Por último, en cuanto a los módulos de memoria, la arquitectura está implementada como un sistema conexionista que emplea grupos de componentes denominados *G-RAMs*: *Memorias Generalizadas de Acceso Aleatorio* (Aleksander, 1990).

3.2.2.3. Arquitectura SASE: simulando niveles de consciencia

Weng (Weng, 2004) propone una arquitectura cognitiva emergente llamada SASE (por su acrónimo en inglés de *Self-Aware Self-Effecting*), la cual concibe la cognición como un proceso de "desarrollo evolutivo", esto es, que tanto el conocimiento como las facultades cognitivas no son aspectos que se especifican (o programan) a priori, sino que son el resultado de la interacción en tiempo real del agente con el ambiente, con otros agentes, y con humanos. Por tanto, la arquitectura no posee un diseño pre-definido que determine la manera de realizar ciertas tareas, sino que el sistema está dotado con capacidades para adaptarse a las necesidades ambientales y aprender la forma en que debe lograr dichas tareas.

Weng se refiere a su arquitectura como un sistema auto-consciente y auto-consecuente (Weng, 2004). Esto supone dos distinciones importantes en la arquitectura: (1) existe un conjunto de sensores y efectores que están acoplados con el ambiente mediante un mecanismo sensorial proprioceptivo, y (2) existe un conjunto de sensores y efectores que están asociados con el cerebro y con el sistema nervioso central (SNC) del agente. Según Wengs, sólo aquellos sistemas que definen mecanismos explícitos para percibir y afectar el SNC califican como arquitecturas SASE.

Algunas de las implicaciones asociadas a la construcción de agentes cognitivos que emplean la arquitectura SASE son: el sistema es configurado sin un conocimiento previo acerca de las tareas que tendrá que desempeñar el agente, el SNC del agente no es accesible directamente por los diseñadores (humanos) una vez está en ejecución y, como consecuencia, la única manera que un humano puede afectar el comportamiento del agente es a través de sensores y efectores externos. En la figura 3.17 se observa la manera en que interactúan los componentes internos de la arquitectura con el ambiente a través de los sensores y efectores.

La arquitectura SASE es muy fiel al paradigma de cognición emergente, en especial a la aproximación enactiva: su filogenia es inmutable y es sólo a través del desarrollo ontogenético que el sistema puede aprender a operar efectivamente en su ambiente. El

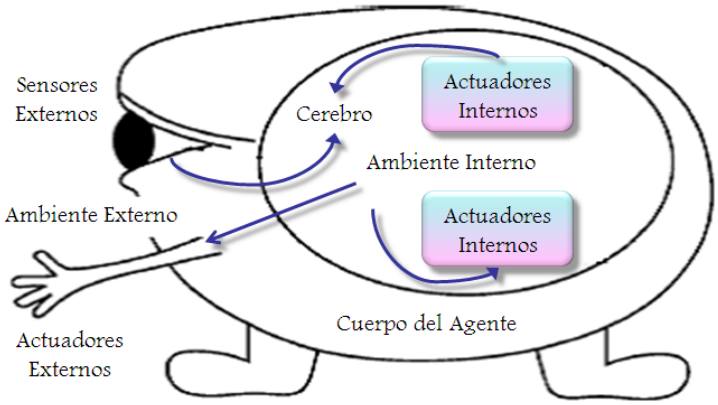


Figura 3.17: Arquitectura SASE. Adaptada de (Weng, 2004).

concepto de operación de SASE es similar al nivel dos de los principios organizacionales autopoiéticos introducidos por Maturana y Varela (Maturana y Varela, 1987) (como ejemplo, la auto-producción y el auto-desarrollo descritos en la sección 2.1.1).

Los procesos cognitivos en SASE se basan en procesos de decisión markovianos (PDM), específicamente, en procesos markovianos orientados a la observación autoconsciente y auto-consecuente (PDM-OOACAC). En un espectro de PDMs donde varían los grados de complejidad cognitiva y conductual, Weng situó esta particular arquitectura en un primer tipo de sistemas cuya principal característica es la capacidad de desarrollar evolutivamente sus propias estructuras cognitivas (Weng, 2004).

En cuanto al sistema sensoriomotriz de PDM-OOACAC, éste define tres tipos de cartografiado neuronal para los flujos de información internos: un cartografiado sensorial, un cartografiado cognitivo, y un cartografiado motriz. La razón para que exista más de una ruta cognitiva entre el cartografiado sensorial y el cartografiado motriz posee una motivación neuro-biológica: una ruta encapsula los comportamientos innatos (y las capacidades filogenéticas que posee el sistema), mientras que las otras encapsulan los comportamientos aprendidos (y las capacidades ontogénicas del sistema). Estas rutas son mediadas por un motor de conversión basado en el modelo de subsumción, el cual confiere una prioridad más alta a la ruta de desarrollo ontogénica.

La principal aplicación de esta arquitectura ha sido en plataformas robóticas. En estas implementaciones, el sistema sensoriomotriz ejecuta en cascada los tres tipos de cartografiado: el sistema de cartografiado sensorial imita la distribución neuronal jerárquica de las cortezas visuales del cerebro, el cartografiado cognitivo controla los comportamientos innatos y aprendidos, y el cartografiado motriz, a través del modelo de subsumción, arbitra los comportamientos producidos por el cartografiado cognitivo. Las implementaciones de SASE se basan en aproximaciones evolucionistas (en inglés, developmental approaches) enfocadas hacia tareas tales como el reconocimiento de objetos, la navegación guiada por el sistema de visión, el aprendizaje del habla, el aprendizaje comunicativo, y la adquisición de nuevas destrezas (empleando o no un profesor que guía el aprendizaje).

3.2.2.4. Arquitectura DARWIN: Dispositivos basados en cerebros robóticos neuromiméticos

Kirchmar (Krichmar y Edelman, 2006) ha desarrollado en la última década una serie de plataformas robóticas bajo el nombre de DARWIN, mediante las cuales es posible experimentar con agentes evolutivos. Estos sistemas se componen de "dispositivos basados en el cerebro" (DBC), los cuales incorporan un sistema nervioso artificial mediante el cual el agente puede desarrollar tanto una memoria espacial y episódica, como ciertas capacidades de reconocimiento de patrones. Como tal, los DBCs son una aproximación neuro-mimética del paradigma emergente que emplean principalmente modelos enactivos y conexionistas.

Darwin difiere de la mayoría de aproximaciones conexionistas en que el modelado de la arquitectura se ciñe mucho más en la estructura y organización que tiene el cerebro humano. Por ejemplo, los DBCs definen un sistema nervioso central basado en redes neuronales artificiales, el cual actúa como un todo en lugar de definir una implementación neuronal que controle independientemente las funciones de control, memoria, o reconocimiento de patrones, como suele definirse en la mayoría de arquitecturas basadas en modelos conexionistas. Los principales mecanismos neuronales de la aproximación DBC son: un sistema de refuerzo, un conjunto de procesos de conectividad re-entrante, un mecanismo de sincronización dinámica de la actividad neuronal, un proceso que controla la plasticidad sináptica, y un conjunto de unidades neuronales con propiedades de respuesta espacio-temporal.

La versión de Darwin VIII es capaz de discriminar objetivos visuales simples (formas geométricas con colores) mediante la asociación de una señal auditiva innata. Este sistema nervioso simulado contiene 28 áreas neuronales, aproximadamente 54.000 unidades neuronales, y alrededor de 1.7 millones de conexiones sinápticas. Desde la perspectiva de comportamiento, Darwin VIII está condicionado para seleccionar preferentemente un objetivo sobre otros, mediante la asociación de una señal auditiva o visual, lo que demuestra su capacidad para orientar el foco de atención hacia un objetivo específico.

Darwin IX puede navegar y categorizar texturas empleando para ello una especie de bigotes artificiales basados en la neuro-anatomía simulada de un sistema somatosensorial[9] de una rata, y cuyo sistema nervioso comprende 17 áreas neuronales, 1.101 unidades neuronales, y aproximadamente 8.400 conexiones sinápticas.

Por último, Darwin X es capaz de desarrollar una memoria espacial y episódica basada en un modelo del hipocampo y de las regiones circundantes del cerebro. Su sistema nervioso simulado contiene 50 áreas neuronales, 90.000 unidades neuronales, y 1.4 millones de conexiones sinápticas. Esta configuración incluye un sistema visual, un sistema de dirección de la cabeza, una formación hipocámpica, un cerebro basal, un sistema de refuerzo basado en una función del sistema dopaminérgico[10], y un sistema de selección

[9] El sistema somatosensorial o sensorial somático permite al ser vivo experimentar sensaciones en las distintas partes del cuerpo. Pueden ser sensaciones del tacto, temperatura, de la posición de las partes del cuerpo o del dolor. Los receptores del sistema somatosensorial se encuentran repartidos por todo el cuerpo y sirven para detectar los estímulos mecánicos, químicos y físicos (Kandel et al., 1991).

[10] Se trata de un sistema químico de gran importancia en la transmisión de impulsos en el cerebro, el cual regula ciertos tipos de comportamientos. Por otra parte, este sistema establece la producción de dopamina,

de acciones. Finalmente, el sistema de visión es usado para reconocer objetos y luego estimar su posición, de tal manera que el agente luego puede desempeñar actividades de navegación y exploración.

3.2.3. Arquitecturas de Aproximación Híbrida

3.2.3.1. Arquitectura HUMANOID: Interacción multi-componente

Burgart (Burghart et al., 2005; Burghart et al., 2007) presenta una arquitectura cognitiva híbrida para un robot humanoide. Esta arquitectura se basa en la interacción de los siguientes componentes: un sistema jerárquico de tres niveles para la descomposición de tareas, un sub-sistema de Memoria de Largo Plazo que emplea una base de conocimiento global (para lo cual utiliza una variedad de esquemas representacionales, incluidas Ontologías, Modelos Geométricos, Modelos de Markov Ocultos, y Modelos de Cinemática), un Gestor de Diálogos que media entre la percepción y la planificación de tareas, un Supervisor de Ejecución, un sub-sistema de Memoria de Corto plazo al cual tienen acceso todos los niveles de percepción y gestión de tareas, y un conjunto de comportamientos que operan en paralelo (como se observa en la figura 3.18).

El Sub-sistema de Percepción se compone de una jerarquía de tres niveles: percepción de bajo, medio, y alto nivel. El módulo de percepción de bajo nivel interpreta los datos sensoriales sin necesidad de acceder a la base de conocimiento central del sistema, lo que permite un control reactivo del agente. Este módulo se comunica con el módulo de percepción de nivel medio y con el módulo de ejecución de tareas. Por su parte, el módulo de percepción de nivel medio provee un conjunto de componentes para el reconocimiento de patrones, y se comunica tanto con la base de conocimiento del sistema (la memoria de largo plazo), como con los modelos activos (la memoria de corto plazo). Finalmente, el módulo de percepción de alto nivel provee un conjunto de capacidades de interpretación más sofisticadas, tales como el reconocimiento de situaciones, la interpretación de gestos y movimientos, y la predicción de las intenciones de otros agentes.

Por otro lado, el sub-sistema de ejecución de tareas se compone de una jerarquía de tres niveles que incluye: la Planificación, la Coordinación, y la Ejecución de Tareas. El sistema cognitivo planifica las tareas en el nivel simbólico superior empleando para ello conocimiento especializado sobre las tareas, el cual es adquirido mediante un proceso de aprendizaje (e.g., a través de demostraciones). El *planificador de tareas* interactúa con el módulo de percepción de alto nivel, la base de conocimiento del sistema (memoria de largo plazo), el nivel de coordinación de tareas, y el *supervisor de ejecución*, este último encargado de gestionar las tareas y los recursos del sistema durante la ejecución del plan. Como resultado de la planificación, una secuencia de acciones es generada y enviada al nivel de *coordinación de tareas*, quien jerarquiza luego las tareas que serán realizadas en el nivel más bajo de ejecución. En general, durante la ejecución de alguna tarea, el nivel de coordinación de tareas funciona independientemente del nivel de planificación de tareas.

---

sustancia encargada de mantener nuestros pensamientos y percepciones de acuerdo con la realidad del medio que nos rodea (Miñarro y Rodríguez, 1996).

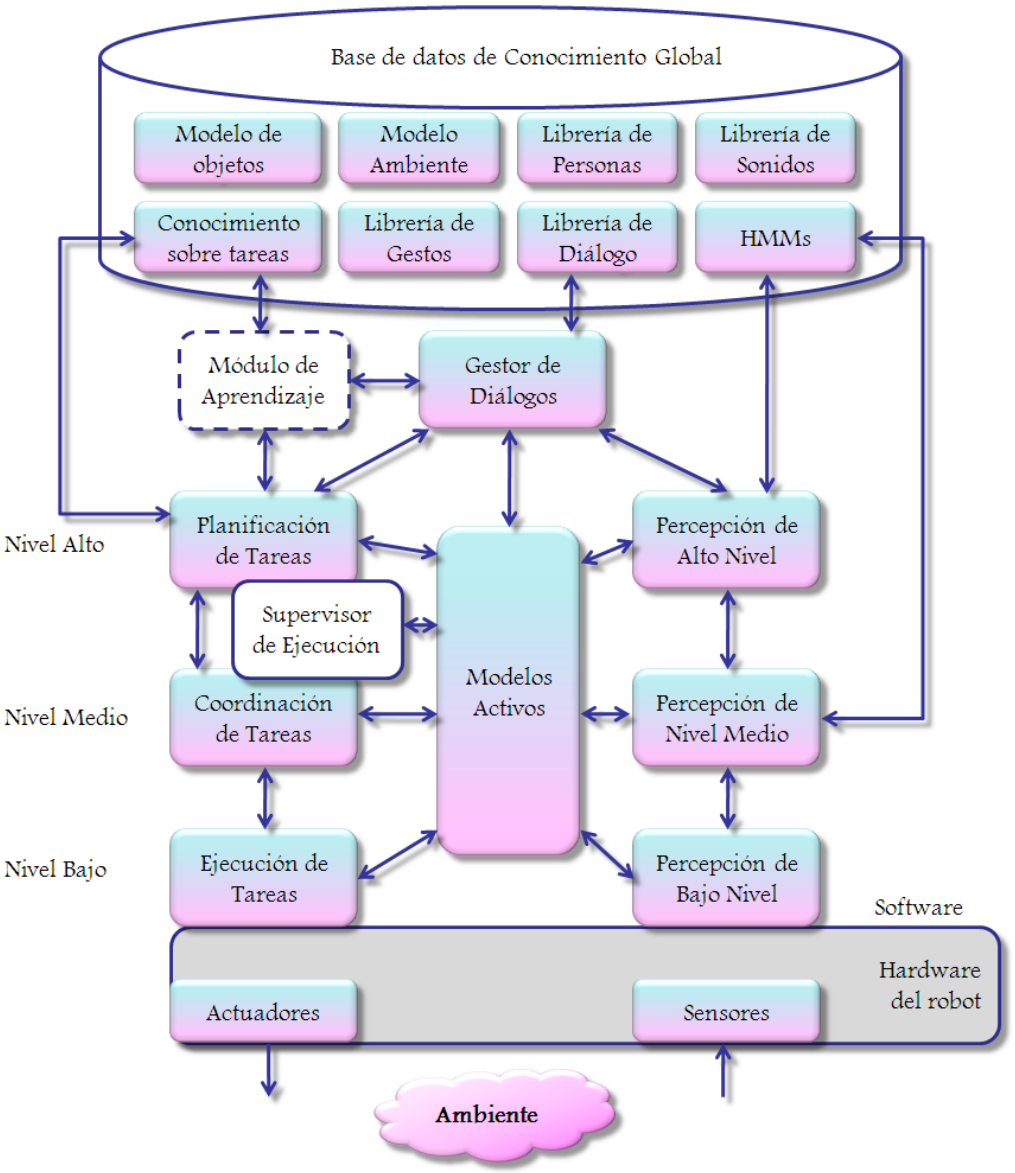


Figura 3.18: Arquitectura HUMANOID. Adaptada de (Burghart et al., 2005).

Por otra parte, un *Gestor de Diálogos*, que coordina la comunicación con el usuario e interpreta los eventos de comunicación, provee un puente entre el sub-sistema de percepción y el sub-sistema de tareas. Esta operación es propiamente cognitiva en el sentido que provee la funcionalidad de reconocer las intenciones y los comportamientos de los usuarios.

Finalmente, el *sub-sistema de aprendizaje* se encarga de integrar las tareas de aprendizaje actuales con las secuencias de acciones pre-programadas; aunque también define un proceso de aprendizaje en tiempo de ejecución basado en la imitación. Como tal, este último componente representa el proceso clave del mecanismo de aprendizaje.

3.2.3.2. Arquitectura CEREBUS

Horswill (Horswill, 2001; Horswill, 2007) argumenta que los sistemas de inteligencia artificial clásicos tales como SOAR, ACT-R y EPIC, son sistemas que almacenan todo el conocimiento de manera central en una base de datos simbólica, sobre la cual el razonamiento es efectuado principalmente a través de la búsqueda y actualización secuencial de los propios datos. Sin embargo, defiende que los SCA debieran ser sistemas distribuidos dotados con una gran cantidad de sensores, procesos de razonamiento, y procesos de control motriz, todos ellos débilmente acoplados unos con otros y operando en paralelo. Propone además, que cada uno de esos procesos debiera mantener su propia representación del mundo, puesto que no es realista el que cada uno requiera estar sincronizándose constantemente con una base de conocimiento central.

Horswill sostiene que los sistemas clásicos de razonamiento emplean una forma apropiada de dirigir la atención perceptual: asumen que toda la información relevante se encuentra almacenada previamente en la base de conocimiento o, que en su defecto, proveen un conjunto de acciones que disparan operadores perceptuales los cuales se ajustan a tareas concretas y modifican partes específicas de la base de conocimiento (justo como sucede, por ejemplo, con ACT-R). Ambas aproximaciones resultan inconvenientes: la primera falla en el problema típico relacionado con el uso de marcos (i.e., el desafío de representar todas las consecuencias posibles de las acciones del agente mediante un sistema de lógica, sin la necesidad de almacenar explícitamente un gran número de consecuencias no válidas y obvias), y la segunda requiere, de forma imperante, que el diseñador proponga una base de reglas mediante la cual puede asegurarse que las acciones apropiadas sean disparadas en las circunstancias correctas y en el momento preciso.

Horswill argumenta que el dividir todos los distintos modelos y representaciones en procesos distribuidos o sub-sistemas, los cuales no requieren de una sincronización con una base de conocimiento central, es una característica clave en el diseño de arquitecturas cognitivas. Por tanto, propone en Cerebus, la hibridación de modelos basados en comportamientos y algunas técnicas de IA simbólica (e.g., los sistemas de lógica de predicados que emplean mecanismos inferenciales con encadenamiento hacia adelante y hacia atrás). Combina además, un conjunto de sistemas sensoriomotrices basados en comportamiento, junto con una red semántica y una red de inferencia. La red semántica corresponde a la memoria declarativa de largo plazo, la cual almacena conocimiento reflexivo acerca de las propias capacidades del agente. Por otro lado, la red de inferencia permite razonar acerca del estado actual y los procesos de control. Juntos, implementan la característica clave de Cerebus: el uso de conocimiento reflexivo acerca de los sistemas sensoriomotrices del agente a fin de desempeñar un razonamiento efectivo sobre sus propias capacidades cognitivas.

En cuanto a la implementación del sistema basado en comportamientos, Cerebus emplea un lenguaje de programación funcional llamado GRL (por su acrónimo en inglés de *Generic Robot Language*). Este lenguaje hereda la mayoría de fortalezas del lenguaje LISP, tales como: el polimorfismo, las semánticas funcionales, los procedimientos de alto nivel, y las funciones de secuencia. Sin embargo, a diferencia de LISP, todas estas características son evaluadas parcialmente en tiempo de compilación a fin de reducir todas las instrucciones declarativas a una red de máquinas de estados finitos que opera en paralelo.

Adicionalmente, la arquitectura Cerebus puede aceptar instrucciones simples de un humano a través de una interfaz de diálogo. Por ejemplo, cuando se teclea el comando "muévete hasta el muro", su interprete (que es un conjunto de máquinas de estados finitos estructuradas en cascada) examina cada palabra individualmente, y luego las enlaza con los roles correspondientes. En el ejemplo, el interprete asociaría el comportamiento *muévete* con el rol de ⟨actividad⟩, y la palabra *muro* sería asociada con el rol de ⟨destino⟩. Cuando Cerebus detecta una interrupción, activaría como consecuencia el comportamiento *atender-con-urgencia*, el cual implementa las siguientes reglas:

- Si la condición de proximidad hasta el ⟨destino⟩ es falsa, entonces se continua con la ejecución de la ⟨actividad⟩
- Si el ⟨destino⟩ se encuentra dentro del campo de percepción y a una distancia mínima de proximidad, entonces se desactiva la ⟨actividad⟩
- Si la ⟨actividad⟩ se desactiva así misma, entonces desactivarse también a sí mismo

Cerebus no es fluido en sus diálogos en lo absoluto. Sin embargo, no es su intención demostrar que los sistemas basados en comportamiento podrían ser la técnica de implementación más adecuada para la generación de lenguaje natural. En lugar de esto, Cerebus demuestra que las redes paralelas de estados finitos pueden llegar a ser mucho más poderosas de lo esperado.

3.2.3.3. Arquitectura COG: Integración de múltiples Teorías de la Mente

Scassellati (Scassellati, 2002) presenta una arquitectura para un humanoide robótico llamada COG, la cual propone que la interacción social es un aspecto clave de la función cognitiva. Según Scassellati, un SCA que implemente alguna *teoría de la mente* debería ser capaz, por un lado, de aprender a partir del comportamiento observado en otro agente (artificial o humano) empleando para ello la interpretación de ciertas señales sociales y, por otro, de expresar su estado interno (e.g., emociones, deseos, objetivos, etc.) a través de interacciones sociales (no lingüísticas). Adicionalmente, tendría la capacidad para reconocer los objetivos y deseos de otros agentes, anticiparse a las actuaciones de estos, y modificar su propio comportamiento como consecuencia (Gold et al., 2009).

COG es una arquitectura que se basa en dos Teorías de la Mente: la teoría de Leslie (Leslie, 1994) y la teoría de Cohen (Baron-Cohen, 1995), las cuales descomponen el problema del modelado cognitivo en conjuntos de habilidades primarias y módulos de desarrollo (aunque cada teoría lo realiza de manera diferente). Scassellati propone un modelo híbrido llamado *"teoría de la mente para agentes personificados"*, la cual integra las dos teorías descritas anteriormente junto con un sistema perceptual y otro comportamental.

La Teoría de la Mente de Leslie enfatiza la independencia entre los diferentes módulos de dominio específico: un módulo mecánico, un módulo de actuación, y un módulo de actitud; cada uno de los cuales se encarga respectivamente de representar: el comportamiento de objetos inanimados, el comportamiento de objetos animados, y las creencias e intenciones de los objetos animados.

La Teoría de la Mente de Cohen propone tres módulos, uno de los cuales se ocupa de la interpretación de estímulos perceptuales (visual, auditivo, y táctil) asociados con el movimiento auto-propulsado, y otro que se encarga de la interpretación de estímulos visuales asociados con formas redondas. Estos dos módulos alimentan un tercer módulo de atención compartida que a su vez alimenta un módulo de la Teoría de la Mente que presenta el conocimiento intencional o los "Estados Mentales Epistémicos" de otros agentes.

El punto central de la Teoría de la Mente propuesta en COG, al menos en principio, es la creación de destrezas perceptuales y motrices primarias sobre las cuales se puedan construir capacidades mentales más complejas. Los comportamientos visuomotrices primarios (e.g., para el reconocimiento de caras y ojos) se basan en modelos motivacionales y estrategias de búsqueda visual (e.g., la distinción entre el movimiento inanimado y animado, la identificación de la dirección de la mirada, etc.) las cuales vienen integradas en la propia arquitectura.

3.2.3.4. Arquitectura KISMET: Aprendiendo a través de la interacción social

El rol que cumple el comportamiento emocional en la regulación de las interacciones sociales entre humanos y otros agentes ha sido examinado por Breazeal, empleando para ello un SCA antropomórfico articulado llamado Kismet (Breazeal, 2000). Breazeal argumenta que las emociones son un mecanismo importante para modular el comportamiento del sistema en respuesta tanto a los cambios en los estados internos como en los ambientales. Los estados emocionales preparan y motivan el sistema para responder de manera adaptativa, retroalimentan el aprendizaje de nuevo comportamiento, y actúan como un mecanismo para la homeóstasis comportamental. Así pues, el objetivo principal de Kismet es incentivar el aprendizaje a través de interacciones sociales.

Kismet tiene dos tipos de elementos motivacionales: los *Motivadores* y las *Emociones*. Los motivadores establecen las metas de alto nivel del SCA: e.g., interactuar con las personas (motivador social), emplear juguetes (motivador de diversión), y ocasionalmente descansar (motivador de fatiga). Por tanto, el comportamiento global del agente se enfoca en saciar sus motivadores. En comparación con las emociones, los motivadores operan cíclicamente en períodos de tiempo más prolongados: la intensidad del motivador incrementa con la ausencia de interacciones de satisfacción y disminuye con el acostumbramiento. Así pues, la meta interna del agente es mantener el nivel de cada motivador en alguna región homeostática entre la infra y la sobre-estimulación.

Las emociones (e.g., la ira, la frustración, el enfado, el miedo, la calma, la alegría, la aflicción, la sorpresa, el interés, el aburrimiento, etc.) evocan respuestas comportamentales específicas tales como lamentarse, distanciarse, huir, expresar placer, expresar lástima, o sobresalto, etc. En efecto, las emociones causan que el SCA entre en contacto con cosas que aumentan su "bienestar", evitando aquellas que no. Kismet propone una arquitectura compuesta por cinco módulos distintos: un Sistema Perceptual, un Sistema Motivacional, un Sistema Comportamental, un Sistema Atencional, y un Sistema Motriz. En la figura3.19 se observan los distintos sub-sistemas de la arquitectura KISMET.

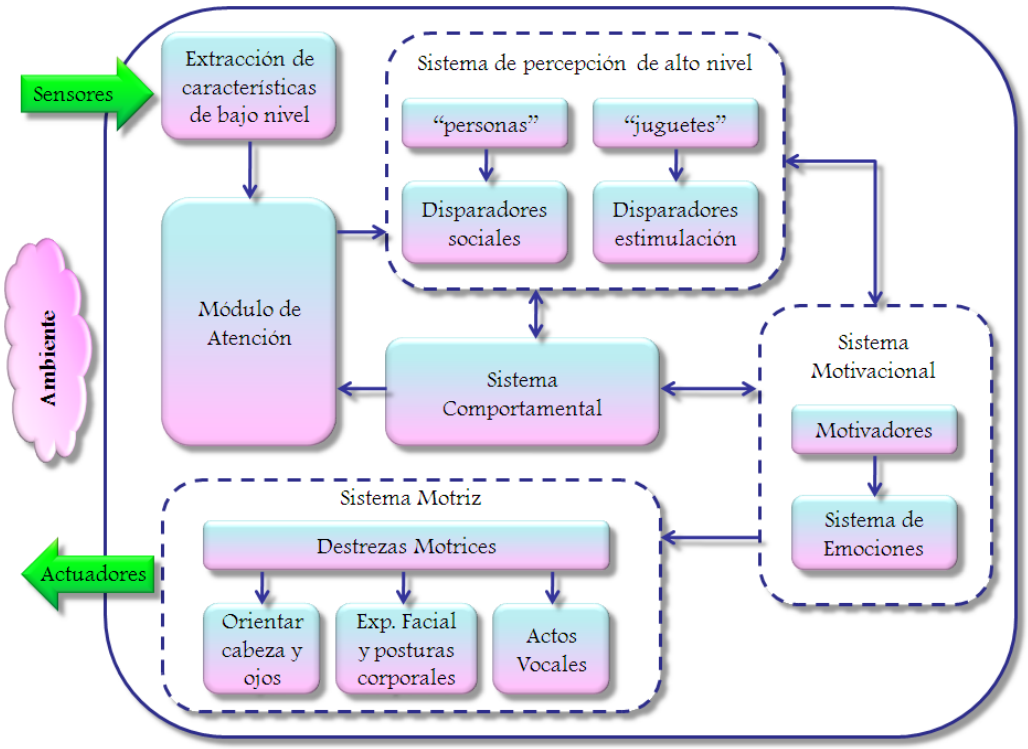


Figura 3.19: Arquitectura KISMET. Interacción entres sub-sistemas. Adaptada de (Breazeal, 2000).

El sistema perceptual de bajo nivel ejecuta las siguientes funciones: (1) percibe los estímulos visuales y auditivos; (2) extrae las características perceptuales (e.g., color, movimiento, frecuencia); (3) extrae descripciones afectivas a partir de las conversaciones; (4) orienta la atención visual; y (5) localiza las características más relevantes tales como caras, ojos, objetos, etc. Estas funciones son luego procesadas por el sistema perceptual de alto nivel y sirven como entradas para el sistema motivacional (emociones y motivadores) y el sistema comportamental. El sistema perceptual de alto nivel filtra la información que proviene del sistema perceptual de bajo nivel mediante un "disparador" de procesos que codifica el conjunto de creencias que mantiene el SCA acerca de su propio estado y su relación con el mundo.

En la arquitectura KISMET existen muchos tipos de disparadores, cada uno de los cuales es "configurado manualmente" por el diseñador del sistema. Cuando el nivel de activación de un disparador excede un umbral dado (basado en la intensidad de las entradas perceptuales, afectivas, motivacionales, y comportamentales), se envía su salida al sistema emocional para su procesamiento. Breazeal sostiene que "cada disparador puede ser visto como una evaluación cognitiva simple que combina características perceptuales de bajo nivel con medidas del estado interno a fin de crear categorías perceptuales que son significativas para el nivel comportamental" (Breazeal, 2003).

Adicionalmente, existe un proceso de valoración que etiqueta la salida del disparador con información afectiva pre-definida sobre su *nivel de estimulación* (cuánto estimula

al sistema), la *valencia* (qué prioridad tiene), y la *postura* (qué tan amigable es). Estos disparadores son luego filtrados por el *descubridor de emociones* que convierte cada tripleta del tipo EVP (Estimulo, Valencia, Postura) en emociones individuales. Luego, el sistema selecciona una emoción simple mediante un proceso de arbitraje que emplea una estrategia de *seleccionar-siempre-el-mejor*, la cual es enviada luego a los sistemas comportamental y motriz para provocar la expresión y postura apropiada.

Kismet es un sistema híbrido en el sentido que emplea esencialmente esquemas cognitivistas basados en reglas que determinan, por ejemplo, las condiciones anteriores, la operación de los disparadores de emociones, la valoración afectiva, etc., aunque el comportamiento global del sistema emerge de la interacción dinámica entre dichos subsistemas (Vernon et al., 2007).

3.2.3.5. Arquitectura CLARION: Un modelo de integración simbólico/sub-simbólico

CLARION (por su acrónimo en inglés de *Connectionist Learning Adaptive Rule Induction ON-line*) es una arquitectura híbrida que propone una distinción entre los procesos cognitivos que se llevan a cabo de forma explícita (simbólica) e implícita (sub-simbólica), y las interacciones entre estos dos (Sun et al., 2001; Sun y Zhang, 2004). El diseño de la arquitectura tiene dos objetivos: el desarrollo de agentes que pueden resolver ciertas tareas cognitivas, y la comprensión del aprendizaje humano y de los procesos de razonamiento observables en dominios similares. La arquitectura CLARION define cuatro módulos de memoria, cada uno compuesto de una representación dual del conocimiento (explícita-implícita): un sub-sistema centrado en acciones (ACS), un sub-sistema no centrado en acciones (NACS), un sub-sistema motivacional (MS), y un sub-sistema metacognitivo (MCS), como se observa en la figura 3.20. De ahora en adelante se empleará el término nivel superior para hacer referencia al nivel de representación explícita/simbólica de cada módulo, mientras que el término nivel inferior hará referencia al nivel de representación implícita/sub-simbólica.

En esencia, el módulo ACS se encarga de regular las acciones del agente, mientras que el módulo NCS se ocupa de mantener la base de conocimiento global del sistema (tanto explícito como implícito). Por otro lado, el módulo MS provee las motivaciones y estímulos que influyen sobre el funcionamiento de los sistemas de percepción, actuación, y cognición, mientras que el módulo MCS dirige y modifica las operaciones de los otros tres módulos. Cada uno de estos módulos adopta una representación localista-distribuida, donde la sección localista codifica el conocimiento explícito (i.e., en forma de reglas de producción) y la sección distribuida codifica conocimiento implícito (i.e., en forma de redes neuronales del tipo perceptrón multicapa – MLP).

El aprendizaje de nuevo conocimiento implícito es logrado mediante métodos de aprendizaje por refuerzo tales como el algoritmo Q-learning, o mediante métodos de aprendizaje supervisado como el algoritmo de retropropagación (back-propagation), los cuales pueden ser implementados empleando una red MLP (Sun et al., 2001). El conocimiento implícito adquirido por el nivel inferior es luego utilizado para crear el conocimiento explícito del nivel superior a través de un mecanismo de aprendizaje ascendente

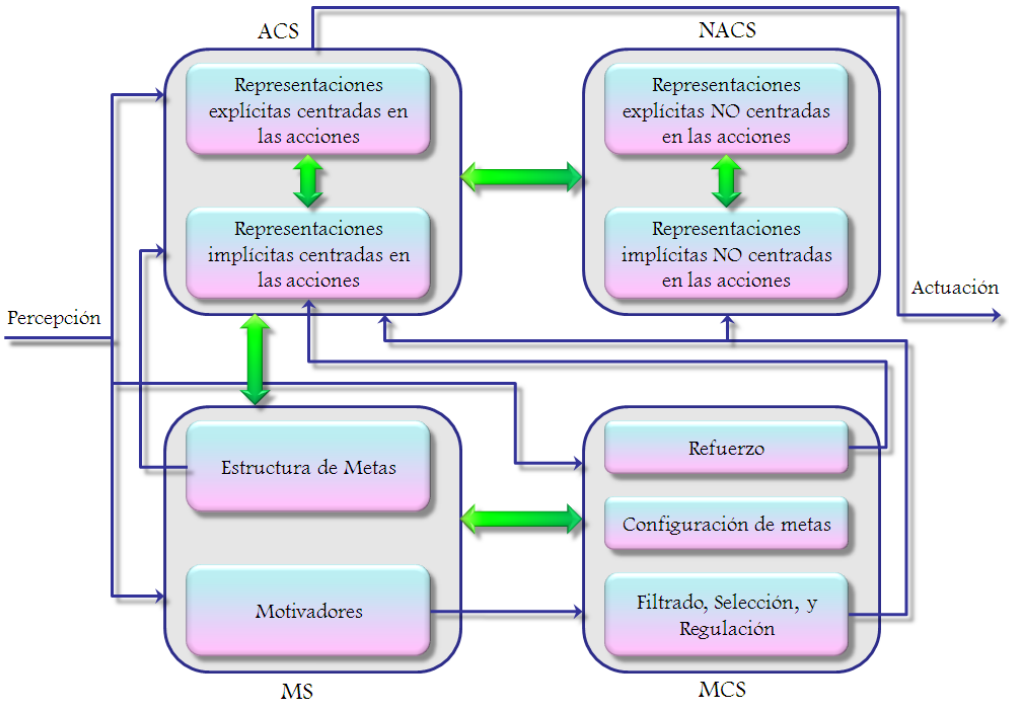


Figura 3.20: Arquitectura CLARION. Adaptada de (Sun et al., 2001).

(bottom-up). A su vez, este aprendizaje puede ser visto como una reconstrucción racional del conocimiento implícito en el nivel explícito de la arquitectura. Por otro lado, el aprendizaje descendente (top-down) puede lograrse mediante la pre-codificación de algunas reglas en el nivel superior, las cuales guían luego el proceso de aprendizaje por observación que efectúa el nivel inferior (Sun y Zhang, 2004).

Como tal, el proceso de toma de decisiones del sistema, que en las fases iniciales de aprendizaje se basa en el nivel superior (explícito/simbólico), gradualmente delega mayor responsabilidad en el nivel inferior (implícito/sub-simbólico). Esta arquitectura ha servido como plataforma de pruebas para simular una gran cantidad de micro-teorías psicológicas, así como también ha servido como un sistema complejo de toma de decisiones secuenciales que resuelve tareas de navegación en escenarios de campos de minas (Duch et al., 2002).

3.2.3.6. Arquitectura LIDA: El rol de la consciencia en los sistemas de memoria

La arquitectura LIDA (por su acrónimo en inglés de *Learning Intelligent Distribution Agent*) es un framework computacional para el diseño de agentes inteligentes, autónomos, y "conscientes", el cual implementa alguna de las ideas propuestas en el teoría del área de trabajo global (Franklin et al., 2005; Franklin, 2006). La arquitectura está construida sobre una plataforma más antigua llamada IDA, la cual fue inicialmente diseñada para automatizar todas las tareas y deberes que tenía que asignar un agente software a toda una plantilla de marineros.

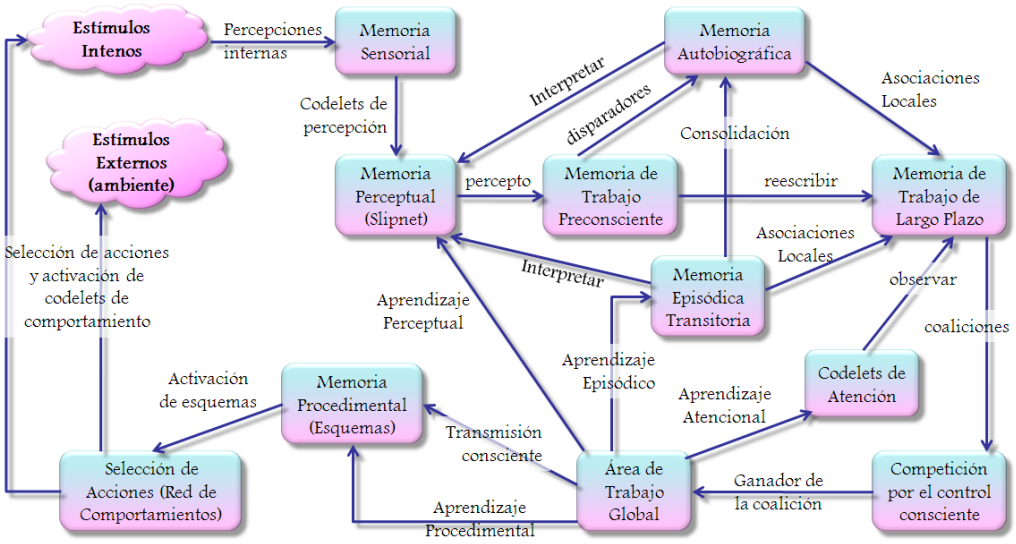


Figura 3.21: Arquitectura LIDA. Ciclo Cognitivo. Adaptado de (Franklin et al., 2005).

LIDA define una organización de la memoria que es en parte simbólica y en parte conexionista, donde todos los símbolos se fundamentan en las características del mundo físico, en el sentido que expone Brooks (Brooks y Stein, 1994). LIDA propone módulos distintos para: la percepción, la memoria de trabajo, las emociones, la memoria semántica, la memoria episódica, la selección de acciones, la expectación y la automatización (e.g., el aprendizaje de tareas procedimentales a partir de la experiencia), la satisfacción de restricciones, la deliberación, la negociación, la solución de problemas, la metacognición, y el comportamiento consciente simulado. La interacción entre algunos de estos módulos se ilustra en la figura 3.21.

La mayoría de las operaciones en LIDA son realizadas por los codelets, piezas de código o programas que desempeñan una tarea especializada e implementan los procesos subconscientes propuestos por la teoría del área de trabajo global. La arquitectura de LIDA incorpora tres nuevos módulos de aprendizaje dentro del antiguo modelo de IDA: el aprendizaje perceptual, episódico, y procedimental. El aprendizaje Perceptual concierne al aprendizaje de nuevos objetos, categorías, relaciones, etc., y puede tomar dos formas: el fortalecimiento o debilitamiento de los niveles de activación de los nodos, así como también la creación de nuevos nodos y enlaces en la memoria perceptual. El aprendizaje Episódico, por otro lado, implica el aprendizaje de eventos específicos (i.e., el qué, el dónde, y el cuándo). Este aprendizaje resulta de los eventos extraídos de los "contenidos conscientes" que son luego codificados en la memoria episódica (transitoria). Finalmente, el aprendizaje Procedimental concierne al aprendizaje de nuevas acciones y secuencias de acciones utilizadas para llevar a cabo nuevas tareas. Este proceso combina un *aprendizaje por selección* (i.e., la selección de unidades de conocimiento de un repertorio obsoleto) con un *aprendizaje por instrucciones* (i.e., la construcción de nuevas representaciones), todo ello modulado por un mecanismo de consciencia funcional que provee retroalimentación a las acciones del agente.

No existe duda de que esta arquitectura puede explicar muchas de las características del comportamiento mental, sin embargo, algunas de sus funciones cognitivas están aún en una etapa inmadura de desarrollo, como por ejemplo: la comprensión del lenguaje, la visión, y el razonamiento de sentido común basado en las percepciones (Duch et al., 2002).

3.2.3.7. Arquitectura Polyscheme

Polyscheme (Cassimatis, 2002; Cassimatis, 2007) es una arquitectura que integra múltiples métodos de representación, razonamiento, y esquemas de inferencia, en la resolución de problemas. Existen muchas razones (revisadas por Baars (Baars, 1988)) para creer que la mente se compone de múltiples "procesadores especializados" encargados de procesar la información percibida, representar el conocimiento, y hacer inferencias acerca de diferentes aspectos del mundo. Polyscheme se basa en esta hipótesis y propone los "especialistas" como los procesos especializados del sistema. Cada especialista modela un aspecto diferente del mundo empleando para ello: técnicas de representación e inferencia específicas, mecanismos de interacción con otros especialistas, y diversos métodos de aprendizaje. Así pues, un especialista puede emplear guiones (scripts), marcos, proposiciones lógicas, redes neuronales, y grafos, como técnicas de representación del conocimiento.

Los especialistas reflexivos de la arquitectura guían el mecanismo atencional de todo el sistema, mediante el suministro de esquemas de focalización que implementan diversos métodos de inferencia a través de la confrontación de guiones (script matching), la búsqueda inversa, el mantenimiento de la cadena de razonamiento, la simulación estocástica, y el razonamiento contrafactual. Mientras tanto, el razonamiento de grado superior es guiado por las políticas de alto nivel que enfocan la atención del agente.

En Polyscheme, la mayoría de los algoritmos emplea como operaciones básicas para la resolución de problemas: la inferencia hacia adelante, la descomposición en sub-metas, la representación de mundos alternativos, y la confrontación de identidad (Duch et al., 2002). Tales operaciones son gestionadas por los especialistas, quienes están dotados con diferentes representaciones (aunque todas ellas enfocadas sobre el mismo aspecto del mundo) y quienes integran también los procesos perceptuales y motrices de bajo nivel.

Por consiguiente, Polyscheme es una arquitectura cognitiva para sistemas robóticos que puede ser empleada tanto en tareas de razonamiento abstracto como en tareas de razonamiento de sentido común. La arquitectura ha sido empleada también en el modelado del proceso de razonamiento infantil, donde han sido analizados aspectos como la identificación de objetos, eventos, causalidad, y relaciones espaciales. El mecanismo de meta-aprendizaje que propone Polyscheme combina diferentes aproximaciones para la resolución de problemas y es considerado un aporte importante en la investigación de los procesos de razonamiento que simulan el sentido común en los humanos (Duch et al., 2002).

3.2.3.8. Arquitectura 4CAPS

La arquitectura 4CAPS (Just y Varma, 2007) define una implementación neuronal plausible mediante la cual se pueden diseñar agentes cognitivos que tratan con tareas complejas tales como la comprensión del lenguaje, la resolución de problemas, o el razonamiento espacial. La funcionalidad de la arquitectura ha sido validada a través de múltiples estudios comparativos que contrastan la operación interna de 4CAPS con los resultados que ofrecen las técnicas de neuro-imagen funcional para la medición de la actividad cerebral humana (Just y Varma, 2007). Así pues, la arquitectura ha sido empleada en la simulación cognitiva de: datos relacionados con el comportamiento humano (e.g., los tiempos de respuestas y las tasas de error), la resolución de problemas por analogía, la interacción humano-ordenador, la comprensión del discurso, y la manera en que personas tanto normales como con deficiencias mentales resuelven otras tareas complejas.

Uno de sus más importantes principios de operación, *"Pensar es el producto de la actividad concurrente de múltiples centros neuronales que colaboran en una red cortical a gran escala"*, conduce al diseño de una arquitectura basada en un número específico de centros (donde cada uno corresponde a un área del cerebro en particular) que emplean diferentes estilos de procesamiento (e.g., el área de Wernicke se especializa en la recuperación asociativa de información), y los cuales construyen y acceden selectivamente a las representaciones que se estructuran internamente de forma jerárquica y secuencial. A pesar de que cada centro posee una capacidad computacional limitada que constriñe su actividad, puede desempeñar y ser parte de múltiples funciones cognitivas (véase la figura 3.22).

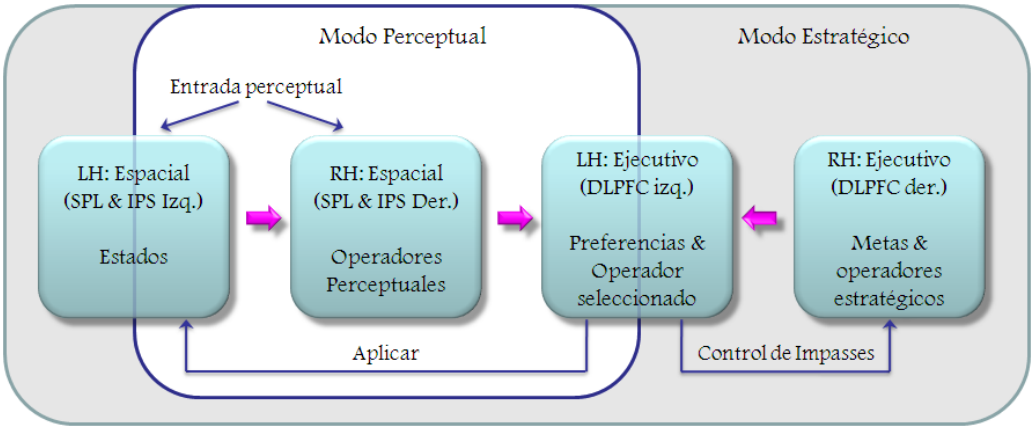


Figura 3.22: Arquitectura 4CAPS. Modelo para la resolución de un problema espacial. SPL: Lóbulo Parietal Superior. IPS: Lóbulo Parietal Inferior. DLPFC: Corteza Prefrontal Dorsolateral. LH: Hemisferio Izquierdo. RH: Hemisferio Derecho. Adaptada de (Just et al., 2007).

Las funciones cognitivas son asignadas a cada centro dependiendo de la disponibilidad de los recursos, y como consecuencia, la topología de toda la red a gran escala no es rígida. A pesar de que 4CAPS se basa en muchas ideas interesantes, no tiene como objetivo principal la simulación de comportamientos inteligentes, sino que su propósito

primordial es el modelado del desempeño de las funciones cognitivas del cerebro humano (Duch et al., 2002).

3.2.3.9. Arquitectura Novamente

Novamente (más conocida como *The Novamente AI Engine* (Looks et al., 2004)), es una arquitectura cognitiva basada en ideas teórico-sistémicas relacionadas con las complejas dinámicas mentales y las pautas emergentes asociadas. La arquitectura se inspira particularmente en el modelo *psynet* (Goertzel, 1997), y de forma más general en la "filosofía de las pautas mentales" (Goertzel, 2006), las cuales proveen una teoría moderadamente detallada acerca de las estructuras y las dinámicas emergentes de los sistemas inteligentes. En esta arquitectura, las funciones mentales tales como la percepción, la actuación, el razonamiento, y el aprendizaje procedimental son descritos en términos de interacciones entre múltiples agentes. Según esta arquitectura, las dinámicas de un sistema cognitivo se suponen deben ser gobernadas por dos fuerzas principales: la auto-organización y el comportamiento orientado hacia las metas.

La arquitectura Novamente propone que las estructuras cognitivas de orden superior, los procesos cognitivos, y los componentes de la memoria, emergen de la interacción entre redes de elementos más simples. Por tanto, las propiedades emergentes de dichas redes de elementos conducen hacia la organización de pautas o patrones mentales tanto jerárquicos como relacionales (heterárquicos)[11]. Estas pautas mentales son representadas mediante técnicas innovadoras que permiten la interacción efectiva entre los diversos procesos cognitivos. Así pues, el conocimiento es almacenado en forma de redes de átomos (empleando un nivel intermedio de representación), las cuales se asemejan estructuralmente a las redes semánticas clásicas excepto por la incorporación de aspectos dinámicos que se parecen más al funcionamiento de las redes neuronales. Esto permite la generación de una gran variedad de dinámicas cognitivas que requieren significativamente de menos memoria y tiempo de procesamiento que una aproximación de más bajo nivel, al estilo de las redes neuronales. Adicionalmente, el proceso de representación del conocimiento que define la arquitectura involucra dos niveles (y de allí su cualidad de híbrida): un nivel explícito, el cual está conformado por unidades discretas (átomos) que pueden ser de diversos tipos (e.g., nodos, enlaces, y contenedores); y un nivel implícito o emergente compuesto por estructuras auto-organizadas llamadas mapas.

Por otra parte, Novamente emplea dos algoritmos primarios a través de los cuales son construidos los procesos cognitivos: la Lógica de Términos Probabilísticos (PTL) y el Algoritmo de Optimización Bayesiana (BOA). PTL es un motor de inferencia altamente flexible, aplicable a dominios complejos y dinámicos, donde cada agente debe tratar una gran cantidad de incertidumbre en la información. Por otro lado, BOA es un algoritmo de optimización basado en poblaciones el cual tiene la capacidad de incorporar conocimiento anticipatorio. PTL se encarga de la creación local de pequeñas piezas de

[11] Para Minsky (Minsky, 1988), la mente humana es la resultante de una interacción de agentes simples que funcionan bajo las reglas de una sociedad. No obstante, tales interacciones funcionan bajo una lógica heterárquica. Una heterarquía puede concebirse como una síntesis entre orden jerárquico y caos anárquico, es decir, como una estructura flexible que posibilita la emergencia de nuevos "órdenes creativos" y "caos ordenados".

conocimiento a partir de otras piezas existentes; mientras que BOA está más orientado hacia la optimización global, donde la creación de nuevo conocimiento se logra mediante la integración de grandes cantidades de conocimiento existente. Estos dos algoritmos interactúan entre sí de diversas formas, representando la inter-dependencia necesaria entre la cognición local y la holística (Looks et al., 2004).

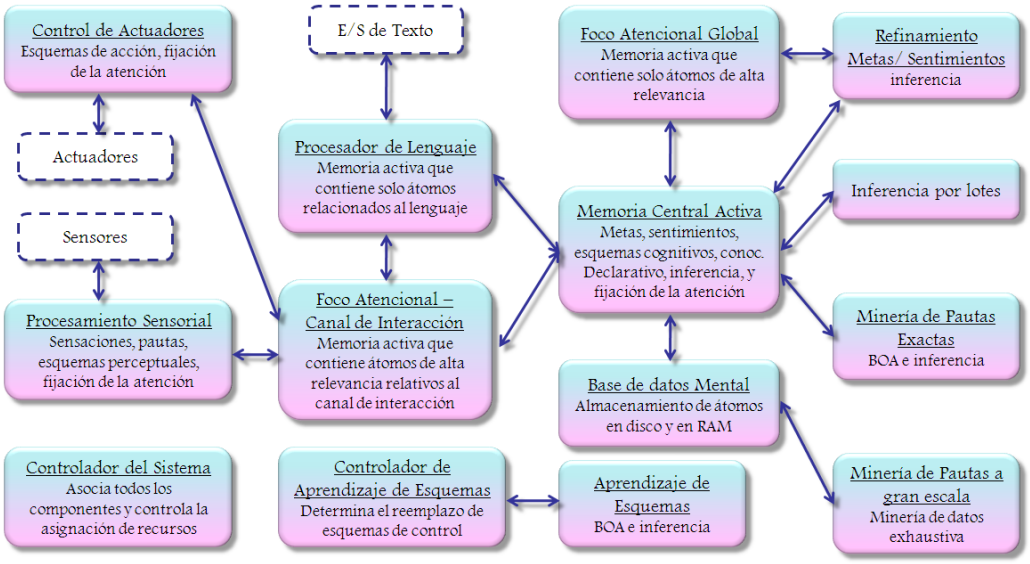


Figura 3.23: Arquitectura Novamente. Cada componente es un lóbulo que contiene múltiples tipos de átomos, cada uno coordinado por un agente diferente. Los lóbulos pueden distribuirse en múltiples máquinas y son controlados por esquemas, los cuales pueden ser reemplazados por otros mediante un mecanismo de aprendizaje de esquemas. El diagrama ilustra una configuración con un único canal de interacción, el cual contiene un conjunto específico de sensores, actuadores, y entradas lingüísticas; las implementaciones reales pueden contener múltiples canales, con diferentes propiedades. Adaptada de (Looks et al., 2004).

En vista que las representaciones básicas de conocimiento y los algoritmos cognitivos son reducidos a un núcleo mínimo en la arquitectura, las diversas especializaciones funcionales requeridas por la inteligencia general pragmática surgen como consecuencia, por un lado, de la determinación de un conjunto de nodos y enlaces que conforman las redes de átomos y, por otro, de la definición de una arquitectura de alto nivel compuesta por un conjunto de lóbulos especializados funcionalmente, donde todos ellos utilizan las mismas estructuras y algoritmos a fin de lograr propósitos particulares (véase la figura 3.23)

Originalmente, la arquitectura Novamente fue diseñada para operar con pequeñas cadenas de información, sin embargo, las versiones extendidas incorporan la capacidad de aprender nuevas unidades de conocimiento de longitud variable y representadas mediante estructuras arbóreas, las cuales son utilizadas para representar acciones, percepciones, y estados internos del agente. La arquitectura actualmente se encuentra en desarrollo, y sus principales aplicaciones se han enfocado en el área de la bio-informática, el procesamiento de lenguaje natural, y el descubrimiento de nuevo conocimiento. No obstante, las propiedades de escalabilidad de la arquitectura aún se desconocen (Duch et al., 2002).

3.2.3.10. Arquitectura DUAL

La arquitectura DUAL (Nestor y Kokinov, 2004) es otra aproximación inspirada en la teoría cognitiva de la "Sociedad de la Mente", propuesta por Minsky (Minsky, 1988). DUAL es una arquitectura híbrida, multi-agente, y de propósito general, que soporta la computación emergente y dinámica a través de una descripción unificada sobre las representaciones mentales, las estructuras de memoria, y los mecanismos de procesamiento, todos ellos llevados a cabo por pequeños micro-agentes interoperantes. Como resultado de la ausencia de un control central, el sistema está constantemente cambiando en función de su entorno.

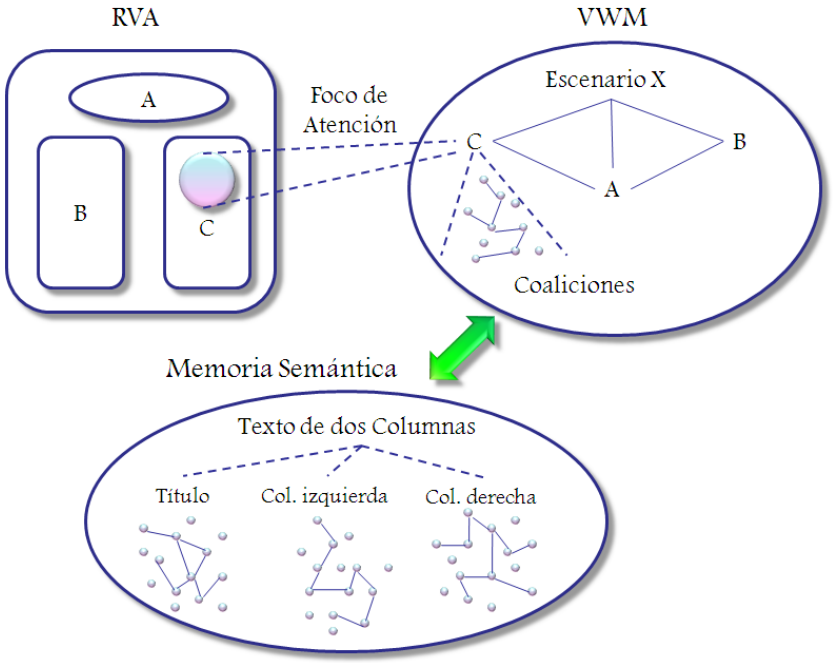


Figura 3.24: Arquitectura DUAL. La figura ilustra los tres componentes principales del modelo para el micro-dominio del análisis de textos: el Vector Visual Retinotópico (RVA), la Memoria de Trabajo Visual (VWM), y la Memoria Semántica de DUAL. El mecanismo de atención se enfoca en un área del vector visual donde se hallan los objetos que están activos en la VWM, mientras que las categorías de los objetos y del escenario correspondientes a los contenidos de la VWM, son recuperados de la memoria semántica. Adaptada de (Nestor, 2004).

Los agentes de DUAL interactúan formando grandes y complejas coaliciones, algunas de las cuales pueden materializarse. Estas coaliciones son representadas mediante una arquitectura multi-nivel: en el micro-nivel interactúa un conjunto de micro-agentes, en el meso-nivel tienen lugar las coaliciones dinámicas, y en el macro-nivel se coordinan las funciones cognitivas de alto nivel. A partir de estos niveles es posible describir las diferentes propiedades del modelo a través de un conjunto de interpretaciones psico-cognitivas. En el micro-nivel de la arquitectura, se emplean micro-marcos como método de representación simbólica de los hechos observados por el agente, mientras que el grado de relevancia o activación de dichos hechos (en un contexto particular) es representado,

en el meso-nivel, por una red de nodos interconectados cuya accesibilidad es especificada por las dinámicas de dispersión de activación entre los mismos nodos. Los enlaces entre los micro-agentes, determinados por las ranuras de los propios marcos y por los pesos de cada conexión, controlan la influencia que tiene un agente sobre la actividad de los demás.

DUAL describe un modelo de atención y percepción visual cuyo propósito es justificar la construcción y categorización de las representaciones de los objetos y los escenarios que se derivan a partir de la interpretación de los estímulos visuales percibidos en un micro-dominio específico. Los procesos de bajo nivel, que operan en paralelo, son combinados con un despliegue serial de la atención visual que posibilita la construcción de representaciones simbólicas y abstractas. Un memoria visual a corto plazo, encargada de mantener la información durante los cambios de atención, forma el núcleo del modelo de conexión entre las representaciones de bajo nivel producidas por los estímulos y la memoria semántica de DUAL. La figura 3.24 presenta un modelo simplificado de los componentes principales de la arquitectura.

La arquitectura DUAL ha sido empleada en un número considerable de proyectos: AMBR, un modelo sobre el proceso de razonamiento en los humanos que unifica los mecanismos de aprendizaje por analogía, aprendizaje episódico, deducción, y generalización; JUDGEMAP, un modelo sobre el discernimiento humano; PEAN, un modelo sobre la percepción humana; y un modelo para la comprensión del rol del contexto y de los estímulos desencadenantes (*priming effects*) en las dinámicas de los procesos cognitivos [12]. DUAL es una arquitectura interesante capaz de explicar muchos de los fenómenos cognitivos, sin embargo, aún no se ha demostrado cómo su funcionalidad puede ser escalada a problemas reales que requieren de un proceso de razonamiento más complejo (Duch et al., 2002).

En el Anexo A, se presenta un análisis comparativo que contrasta y valida un conjunto de características deseables, agrupadas por categorías, para cada una de las 30 arquitecturas cognitivas revisadas en el presente capítulo. Estas características se han extraído a partir de los principios esenciales que proponen las ciencias cognitivas, y son comúnmente empleadas como criterios de diseño en el modelado de arquitecturas cognitivas para agentes inteligentes.

[12] Para mayor información acerca de los proyectos que emplean la arquitectura DUAL, consultar el sitio web oficial: `http://www.nbu.bg/cogs/personal/kokinov/`.

Capítulo 4

Naturaleza de la Solución

"Aunque el procesamiento innato, la conducta instintiva, las motivaciones e impulsos orquestados interiormente, y el aprendizaje orientado de manera innata son elementos importantes e imprescindibles del repertorio cognitivo de un animal, no es probable que forma parte de ese territorio más esotérico de la actividad metal que asociamos con el pensamiento, el juicio y la toma de decisiones. Pero, ¿qué es el pensamiento, y cómo podemos reconocerlo en otras criaturas, si tiene lugar en el mas privado de los órganos, el cerebro? ¿Qué criterios de conducta pueden permitirnos distinguir entre el auténtico pensamiento que solemos creer que interviene en nuestras decisiones estéticas, morales y prácticas, y las intrincadas programaciones que pueden producir la ilusión de pensamiento en, al menos, algunos otros animales? ¿Podría ser, como sospechan los defensores de la inteligencia artificial, que todo pensamiento, incluido el nuestro, ***sea sólo la consecuencia de una hábil programación?****".*

— James Gould y Carol Grant, The Animal Mind, 1994.

4.1. Introducción

El psicólogo William Calvin (Calvin, 2001), define el comportamiento inteligente como el montaje escénico definitivo de la neurofisiología: el resultado final de muchos aspectos de la organización cerebral de un individuo, que le permite hacer algo que nunca ha hecho antes.

Puede que no sea posible explicar la inteligencia en todo su esplendor, pero ya se conocen algunos de los elementos de su explicación. Algunos son de conducta, otros son neurofisiológicos, y otros son procesos de tipo evolutivo que se desarrollan en cuestión de segundos. Incluso, se sabe algo de los principios de auto-organización que generan material emergente, como cuando distintas categorías y metáforas compiten por el territorio cerebral.

La gran cuestión para comprender la inteligencia (natural o artificial) no es *quién* tiene más, sino *qué* es, *cuándo* se necesita y *cómo* funciona. Entre los factores cognitivos

del *qué* figuran la intuición, la previsión, la rapidez, la creatividad y el número de cosas que se pueden manejar a la vez, entre otras.

Desde un punto de vista computacional, la simulación y representación de estos factores cognitivos debe estimar como principales las siguientes consideraciones:

- Los factores relativos a la cognición artificial usualmente se enmarcan dentro de un modelo cognitivo que representa el mecanismo interno de procesamiento de la información de una entidad autónoma. Dicha entidad autónoma que percibe el mundo y actúa sobre éste, se modela de manera más apropiada mediante el paradigma de los agentes software.
- La interacción de los factores cognitivos en un agente software debe poseer una estructura subyacente, a través de la cual se distinga su organización, sus propiedades, sus componentes y los mecanismos de comunicación entre los mismos. Esta estructura, es lo que llamaremos de ahora en adelante la *"Arquitectura Cognitiva"* del agente software.

Así pues, este capítulo presenta los principios teórico-sistémicos en los que se fundamenta la aproximación arquitectónica propuesta, atendiendo principalmente los problemas identificados en la sección 1.2, así como las debilidades más relevantes y comunes detectadas en el análisis comparativo que se expone en el anexo A. Adicionalmente, el presente capítulo intenta proveer, desde una perspectiva computacional, las respuestas a las cuestiones antes mencionadas (sobre el qué, el cuándo, y el cómo de la inteligencia) a través de un conjunto de criterios esenciales para el diseño de un modelo cognitivo innovador que exhiba las propiedades principales del comportamiento inteligente.

De esta manera, la solución computacional propuesta apunta al desarrollo de una arquitectura cognitiva para agentes inteligentes, la cual está inspirada en principios autoorganizativos y de comportamiento emergente, observados principalmente en los sistemas biológicos y en general, en los Sistemas Complejos Adaptativos. Este diseño e implementación de un modelo cognitivo resulta ser además una aproximación híbrida debido a que integra propiedades de los paradigmas Cognitivista y Emergente, no obstante, el principal aporte de la presente aproximación es el modelo adaptativo y evolutivo que se puede evidenciar en los diferentes niveles organizativos de la arquitectura cognitiva.

De ahora en adelante, se empleará el término de METÁFORA para hacer referencia a la aproximación arquitectónica propuesta. Los motivos de este nombre, podrán ser inferidos a lo largo de la lectura de este capítulo, no obstante, serán sintetizados al final del mismo en el apartado 4.6.

Este capítulo se encuentra organizado de la siguiente manera: en las secciones 4.2, 4.3, 4.4 y 4.5, se argumentan los principios biológicos, psicológicos, neurofisiologicos, y sistémicos que subyacen al diseño de METÁFORA. La sección 4.5 resulta ser particularmente interesante debido a que sirve de puente entre dichos principios y la solución computacional propuesta. La sección 4.6 sintetiza las metáforas en las cuales se basa la arquitectura. La sección 4.7 describe la gestación de la idea de METÁFORA como modelo cognitivo y cómo éste ha evolucionado en los últimos años de investigación del autor. Y finalmente, la sección 4.8 describe brevemente la metodología investigativa e ingenieril empleada en el desarrollo de METÁFORA.

A pesar de la importancia de este capítulo para el entendimiento del resto de la tesis, el lector, en el caso de que se encuentre familiarizado con los conceptos, puede optar por omitirlo y consultar directamente la sección 4.6 donde se sintetizan las metáforas sobre las cuales se inspira el diseño de la arquitectura.

4.2. Integración de micro-teorías cognitivas: Un modelo Híbrido de Cognición Artificial

> *"Modern science is categorised by its ever-increasing specialisation, necessitated by the enormous amount of data, and the complexity of techniques and of theoretical structures within every field. Thus, science is split into innumerable disciplines that continually generate new subdisciplines. As a consequence, the physicist, the biologist, the psychologist and the social scientist are, so to speak, encapsulated in their private universes, and it is difficult to get word from one cocoon to the other".*
> — Ludwig von Bertalanffy, General Systems Theory, 1928.

La primera perspectiva de integración en la que se fundamenta el diseño de la presente aproximación arquitectónica, es la *hibridación* de micro-teorías cognitivas. El propósito de esta perspectiva se basa principalmente en dos enfoques:

1. La unificación de teorías sobre el proceso de cognición, desde la perspectiva de la psicología cognitiva, la biología, la neurofisiología, la teoría general de sistemas, y la inteligencia artificial. Desde este enfoque, diversos módulos, componentes y mecanismos computacionales inspirados en las ciencias cognitivas son integrados en METÁFORA con el propósito de simular funciones cognitivas de orden superior.
2. La generación de un modelo híbrido que incorpora las características de los paradigmas Emergente y Cognitivista junto con mecanismos de integración y sincronización necesarios para fusionar los dos paradigmas. Las características tomadas en cuenta son las relacionadas con la *Representación de Conocimiento*, los *Esquemas de Inferencia*, las *Estrategias de Razonamiento*, los *Mecanismos de Toma de Decisiones*, y el *Aprendizaje* (descritas en la sección 3.1).

El modelo híbrido de METÁFORA tiene una finalidad principal: la compensación de debilidades y potenciación de las fortalezas entre los enfoques Emergente y Cognitivista. El enfoque Cognitivista, a través de representaciones simbólicas, permitirá al agente cognitivo guiar los procesos de aprendizaje y adaptación ejecutados por el enfoque Emergente en las primeras etapas de "interacción con nuevos entornos", actuando como un conocimiento innato o experto sobre el dominio del conocimiento. Mientras tanto, el enfoque Emergente guiará los procesos de descubrimiento de nuevas reglas sub-simbólicas (las cuales serán escaladas continuamente hasta los niveles simbólicos del enfoque Cognitivista a fin de construir explicaciones comprensibles de lo aprendido), cuando las representaciones simbólicas carezcan de validez para el agente.

En cuanto a la segunda perspectiva de integración, fueron considerados y adaptados a las necesidades particulares de METÁFORA, los principios sugeridos por Cassimatis

(Cassimatis, 2002): el principio de *Funciones Comunes* y el principio de *Implementación Múltiple*, los cuales proponen una teoría para el diseño de arquitecturas cognitivas que explica cómo la mente integra los procesos cognitivos empleando métodos computacionales que resultan difíciles de integrar.

El principio de Funciones Comunes argumenta que muchos algoritmos de los diferentes sub-campos del modelado cognitivo computacional, especialmente aquellos que son usados para modelar el razonamiento, pueden ser concebidos como formas diferentes de seleccionar las secuencias del mismo conjunto básico de *funciones comunes*. Por ejemplo, considérese el siguiente listado preliminar de funciones cognitivas que contendrá METÁFORA:

- *Representación del mundo*: dada una entrada sensorial, el agente acomoda y modifica las estructuras cognitivas internas de manera que representen el estado percibido.
- *Inferencia hacia adelante*: dado un conjunto de creencias y reglas procedimentales, se infieren otras que son consecuencia de las primeras.
- *Generación de Sub-metas*: dada la meta de establecer el valor de verdad de una proposición P, crear una meta que determine los valores de verdad de las proposiciones que implicarían la falsedad de P.
- *Simulación de mundos alternativos*: representa y crea inferencias acerca de los estados alternos, posibles, hipotéticos, o condicionados del mundo.
- *Correspondencia de identidad*: dado un conjunto de proposiciones acerca de un objeto, se encuentran otros objetos que pueden ser idénticos a éste.

El principio de *Funciones Comunes*, puede ser justificado demostrando cómo estas funciones son posibles de implementar mediante una gran variedad de algoritmos. Las siguientes caracterizaciones de los dos algoritmos usados con mayor frecuencia en el modelado cognitivo (representar el mundo y la inferencia hacia adelante) ilustran cómo los métodos de diferentes vertientes de la ciencia cognitiva formal pueden ser implementados empleando el mismo conjunto de funciones comunes:

- *Búsqueda*: "Cuando existe incertidumbre acerca de si A es verdadero, entonces: *representar el mundo* donde A es verdadero, emplear *inferencia hacia adelante*, *representar el mundo* donde A no es verdadero, y emplear *inferencia hacia adelante* nuevamente. Si la *inferencia hacia adelante* conduce a un nuevo estado de incertidumbre, repetir el proceso."
- *Simulación estocástica*: "Cuando A es más probable que ¬ A, entonces: *representar el mundo* donde A es verdadero y emplear *inferencia hacia adelante* con una mayor frecuencia que como se suele hacer para el mundo donde ¬ A es verdadero."
- *Razonamiento simbólico*: "Dado un conjunto de hechos observados por el agente, se *representa el mundo* donde el conjunto de hechos es verdadero, se buscan aquellas reglas simbólicas que emparejan con los hechos dados empleando *inferencia hacia adelante*, surgen nuevos hechos que son nuevamente representados, y nuevamente se aplica *inferencia hacia adelante* hasta que no resulten nuevos hechos".

Esto ilustra cómo los algoritmos de distintos enfoques de modelado cognitivo (búsqueda, simulación estocástica, y razonamiento simbólico) pueden ser concebidos como estrategias diferentes que seleccionan el mismo conjunto básico de operaciones (inferencia hacia adelante, representación del mundo, etc.). Explicar la integración de procesos cognitivos que son representados empleando diferentes modelos, puede ser una tarea compleja debido a que estos modelos están basados con frecuencia en algoritmos que son difíciles de conciliar entre sí. Si la mente ejecuta cada estrategia usando una secuencia de funciones comunes, la integración de las diferentes estrategias de razonamiento podría convertirse en una explicación de cómo la mente combina e intercala secuencias de funciones comunes.

Po otro lado, el principio de *Implementación Múltiple* argumenta que múltiples mecanismos computacionales y representacionales pueden implementar cada una de las Funciones Comunes. Por tanto, este principio ayuda a explicar cómo los procesos cognitivos y perceptuales de bajo nivel crean e influyen sobre procesos de alto nivel como el razonamiento y la resolución de problemas. Por ejemplo, en METÁFORA, la *inferencia hacia adelante* es implementada usando los siguientes mecanismos diferentes entre sí:

- *Activación de reglas de producción*: involucra la correspondencia entre un conjunto de reglas y un conjunto de hechos con el objetivo de inferir nuevos hechos.
- *Redes neuronales de propagación hacia adelante*: toma los hechos representados por la activación de las unidades de entrada, propaga estas activaciones hacia adelante, y como respuesta obtiene nuevos hechos representados por los valores de las unidades de salida.
- *Memoria*: el valor de un atributo en un instante de tiempo T2 puede ser inferido recordando su valor en un instante de tiempo anterior, T1, mientras que el intervalo de tiempo entre T1 y T2 se lo suficientemente breve.
- *Anticipación*: el valor de un atributo que se encuentra en algún estado futuro cercano puede ser inferido después de percibir el valor de dicho atributo en el presente.

Así pues, la aplicación de los principios de implementación múltiple y funciones comunes en METÁFORA, sugiere una forma de explicar cómo las estrategias de razonamiento se integran con los procesos perceptual, motriz y nemotécnico de nivel inferior. Si la mente implementa estrategias de razonamiento empleando secuencias de funciones comunes y si (de acuerdo con el principio de implementación múltiple) cada función común puede ser implementada por múltiples mecanismos de orden inferior, entonces esos mecanismos pueden influenciar en cada paso de los procesos cognitivos de nivel superior.

4.3. Divide y vencerás: Propiedades de Modularidad, Especialización, y Plasticidad

Cuando los teóricos de la evolución piensan en la inteligencia y sus orígenes, suelen señalar rutas evolutivas indirectas que han afectado a órganos multi-uso (Calvin, 2001).

En realidad, muchos órganos tienen múltiples aplicaciones, y su combinación de funciones cambia con el tiempo. Por ejemplo, ¿Cuándo evolucionó ese órgano intercambiador de gases de los peces, conocido como "vejiga natatoria" por su contribución a la flotación, hasta convertirse en un pulmón?.

Así pues, al preguntarse cómo se originó la maquinaria neuronal para la previsión, la toma de decisiones o el aprendizaje, hay que tener en cuenta que los mecanismos subyacentes podrían servir para múltiples funciones, cualquiera de las cuales pudo ser favorecida por la selección natural, beneficiando de paso a las otras. Podrían ser como lo que los arquitectos llaman *instalaciones básicas*, como los cuartos para las fotocopiadoras y los buzones de correos. La boca, por ejemplo, es una instalación básica multi-uso que puede usarse para beber, paladear, ingerir, vocalizar, y expresar emociones; en algunos animales, sirve también para respirar, refrescarse, y luchar.

> *"Al considerar las transiciones de órganos, es muy importante tener en cuenta la probabilidad de conversión de una función a otra".*
> — Charles Darwin, El origen de las especies, 1859.

Haciendo una analogía con las instalaciones básicas u órganos multi-uso, se puede pensar en una arquitectura cognitiva compuesta por un conjunto de módulos híbridos, de representación simbólica y sub-simbólica, cuya funcionalidad sea la de especializar las habilidades y comportamientos del agente. Estos módulos especializados, en principio, no tienen un objetivo específico definido a priori sino que su propósito es definido (y re-definido adaptativamente) durante las relaciones de interacción que tiene el agente cognitivo con su entorno. En el contexto de METÁFORA, estos módulos híbridos serán denominados *Comportamientos*.

Arteaga y Pimienta (Arteaga y Pimienta, 2004) hablan de una organización horizontal y vertical de la corteza cerebral. Mientras que el patrón de organización horizontal se refiere a la disposición en capas paralelas a la superficie cortical, el patrón vertical describe agrupamientos celulares, ortogonales a la superficie, que han recibido el nombre de columnas. Mountcastle (Mountcastle, 2003) estableció la unidad básica de la operación cortical en la micro-columna, una estructura vertical de entre ochenta y cien neuronas distribuidas a lo largo de todas las capas corticales, a excepción de la primera, y conteniendo todos los tipos de células de la corteza, constituyendo unidades de procesamiento complejo.

Pues bien, en METÁFORA la distribución arquitectónica es también híbrida: una distribución horizontal por capas de procesamiento, donde las capas superiores se encargan del procesamiento simbólico y la representación explícita del conocimiento (enfoque Cognitivista), y las capas inferiores se encargan del procesamiento sub-simbólico y la representación implícita del conocimiento (enfoque Emergente), las cuales procesan en paralelo, la información que proviene del entorno del agente así como la información de los estados internos del mismo; y, por otro, una distribución vertical conformada por módulos de procesamiento complejo (los *Comportamientos*, en analogía con las microcolumnas corticales), los cuales desempeñan tareas multi-propósito, pero que a través de la experiencia del agente con su entorno se especializan en tareas y metas más concretas. La propiedad de plasticidad se refiere entonces a la capacidad de adaptación que le permite a cada *Comportamiento* especializarse nuevamente en una tarea y/o conjunto de metas diferente, cuando las condiciones ambientales o el estado interno del agente lo requieren.

4.4. La clave está en el Comportamiento Emergente

"Emergencia es lo que ocurre cuando un sistema de elementos relativamente simples se organiza espontáneamente y sin leyes explícitas hasta dar lugar a un comportamiento inteligente".
— S. Johnson, Sistemas Emergentes, 2004.

Un enfoque clásico al observar el funcionamiento de la mente, es considerar la consciencia como un proceso centralizado que puede ser implementado directamente por un algoritmo, es decir, por una secuencia de pasos. Sin embargo, un enfoque más interesante sería ver la consciencia como una propiedad asociada a la interacción entre poblaciones de elementos. Aunque siempre se considere la consciencia como un flujo secuencial, ésta sólo puede ser reproducida a través de procesos distribuidos y en paralelo a través de los cuales el comportamiento inteligente emerge.

Johnson (Johnson, 2004), menciona que se observan conductas emergentes cuando los componentes individuales prestan atención a sus vecinos inmediatos sin esperar órdenes, es decir, sin un control central, y como consecuencia actúan localmente. La acción colectiva de estos componentes produce el comportamiento global.

En este orden de ideas, componentes de orden de complejidad inferior producen, mediante procesos auto-organizativos, componentes de orden de complejidad superior. Así pues, partículas sub-atómicas, átomos, moléculas, neuronas, funciones cognitivas, consciencia, inteligencia individual, inteligencia colectiva, etc. (omitiendo algunos niveles intermedios de organización), forman jerarquías de organización con una complejidad incremental entre cada nivel, produciendo un comportamiento inteligente a partir de las propiedades emergentes observadas.

Por consiguiente, el propósito de esta sección es enunciar los principios básicos sobre el fenómeno del comportamiento emergente, que sirvieron de base para la construcción de los diferentes niveles cognitivos de METÁFORA. A continuación se describen los

principios generales y en las siguientes secciones se describen al detalle los requisitos propuestos para la construcción del modelo cognitivo (Johnson, 2004):

1. *Más es mejor*: a mayor cantidad de elementos, mejor es la apreciación del comportamiento colectivo.
2. *La ignorancia es útil*: es mejor construir un sistema con elementos "estúpidos" y simples, aunque altamente interconectados, que luego permitan progresivamente la emergencia de nuevas conductas.
3. *Alentar los encuentros casuales*: los sistemas descentralizados dependen fuertemente de las interacciones casuales, sin orden específico.
4. *Buscar patrones en la información manipulada.*
5. *Prestar atención a los elementos vecinos*: la información local conduce a la sabiduría global.

4.4.1. Fenomenología de la Persistencia

Para comprender la vida y la mente, desde un punto de vista biológico y computacional, es necesario abandonar la tendencia natural a dividir el mundo en partes discretas. Los organismos son sistemas en flujo, sus elementos constitutivos cambian constantemente. La mente en realidad no es una agregación simple de "elementos" en un sentido convencional, sino que todos estos "elementos" pueden considerarse como remolinos dinámicos y cambiantes que interactúan produciendo un único flujo, por lo cual resulta necesario ser cuidadosos cuando se intenta definir los límites alrededor de estos.

Steve Grand (Grand, 2000), argumenta que el mundo natural está compuesto de una jerarquía de *fenómenos persistentes* en la cual la materia, la vida, la mente y la sociedad son simplemente niveles o aspectos diferentes de la misma cosa, y que esta jerarquía natural puede ser simulada por una equivalente en un universo paralelo virtual llamado ciberespacio. Según Grand, "en ordenador no puede ser inteligente, pero puede crear un universo paralelo en el cual las formas naturales de la inteligencia pueden ser replicadas". El fenómeno persistente puede evidenciarse como un patrón que se propaga haciendo copias de si mismo en el espacio y en el tiempo mientras que sus elementos se mantienen mutuamente *estables*. Los elementos pueden ser estables si se mantienen aislados (como suele suceder en la química por ejemplo), y dejan de serlo persistentes cuando se juntan. Estos elementos constitutivos (partículas, átomos, neuronas, etc.) reaccionan con otros produciendo algo nuevo que resulta ser más estable que los elementos por separados que dieron origen a éste. Cabe notar en este punto, que la habilidad de hacer copias de sí mismo es un truco ingenioso para asegurar la persistencia, algo que se verá con más detalle en la sección 4.4.3.

Así pues, es posible pensar en la inteligencia y la consciencia como un fenómeno persistente más, uno que se encuentra en una jerarquía de mayor complejidad y sofisticación. En un sentido reduccionista, se tendría una relación entre elementos tales como: fotones $\Rightarrow$ partículas $\Rightarrow$ átomos $\Rightarrow$ moléculas $\Rightarrow$ redes auto-catalíticas $\Rightarrow$ sistemas auto-replicadores $\Rightarrow$ sistemas adaptativos $\Rightarrow$ inteligencia y consciencia $\Rightarrow$ sociedad. Cada nivel de la jerarquía es un conjunto de elementos simples que interactúan en bucles

retroalimentados de causa y efecto, con la diferencia que estos bucles no actúan como cadenas simples sino como sistemas de redes, como un tejido vastamente interconectado. No existe nada realmente que sea la causa inicial de alguna circunstancia en particular; cada causa es el efecto de al menos otra causa, y usualmente de más de una. Esto último es un criterio que se debe tener en cuenta a la hora de diseñar un modelo cognitivo, a fin de evitar la visión simplista sobre la causa y su respectivo efecto.

Como se evidenciará a lo largo del desarrollo de esta tesis, el propósito de METÁFORA no es simular los elementos de los niveles más inferiores, sino que, desde un punto de vista altamente reduccionista, se pretende modelar distintos niveles de complejidad y organización (quizás desde el nivel neuronal y no desde el atómico, por su puesto), hasta lograr niveles superiores de complejidad donde el comportamiento inteligente emerge.

4.4.2. Auto-organización, Autopoiesis y Complejidad

> *"Estamos acostumbrados a la idea de que para diseñar un artefacto complejo, como un reloj, se necesita de un diseñador aún más complejo, como un relojero. Es de sentido común, como lo sigue siendo la física de Aristóteles (a pesar de ser errónea). Pero desde Darwin sabemos que las cosas complejas también pueden surgir (de hecho, auto-organizarse) a partir de cosas más simples".*
> — W. Calvin, Cómo piensan los cerebros, 2001.

Esta sección describe una aproximación sobre cómo los diferentes niveles organizativos en METÁFORA aumentan incrementalmente en complejidad, hasta producir un Sistema Complejo Adaptativo. La aproximación arquitectónica propuesta, basándose en los principios de auto-organización y autopoiesis, logra garantizar la persistencia en el espacio-tiempo de sus elementos y las relaciones complejas entre estos.

Esta sección describe una aproximación al diseño por niveles auto-organizativos de la arquitectura METÁFORA, cuya complejidad aumenta incrementalmente hasta exhibir comportamientos cognitivos de orden superior.

4.4.2.1. Sistemas Auto-organizados y Paralelismo

> *"In biological systems, self-organization is a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of the system. Moreover, the rules specifying interactions among the system's components are executed using only local information, without reference to the global pattern".*
> — Camazine, Self-Organization in Biological Systems, 2003.

La auto-organización es un proceso de atracción y repulsión en el cual la organización interna de un sistema, normalmente un sistema abierto, incrementa en complejidad sin ser guiado por un origen externo. Usualmente, los sistemas auto-organizados exhiben propiedades emergentes. El concepto de auto-organización se apoya generalmente en 4 principios (Goldspink y Kay, 2003):

1. *Retroalimentación positiva*: conocida también como causalidad acumulativa, se refiere a una situación donde algún efecto causa más de lo mismo, incrementando el efecto en amplitud y frecuencia.
2. *Retroalimentación negativa*: ocurre cuando la salida de un sistema actúa en oposición a los cambios de la entrada del sistema; como resultado, los cambios son atenuados. Si la retroalimentación del sistema es negativa, el sistema tenderá a ser estable.
3. *Balance entre explotación y Exploración*: la exploración realiza búsquedas en amplitud en el espacio de soluciones localizando así zonas prometedoras y favoreciendo la diversidad, mientras que la explotación realiza búsquedas en profundidad en dichas zonas obteniendo así las mejores soluciones y favoreciendo la convergencia.
4. *Múltiples interacciones*: relaciones bi-direccionales de causa y efecto entre los elementos constitutivos

Turing (Hodges, 1983), enunció tres conceptos brillantes acerca de la computación que pueden ser categorizadas como "máquinas organizadas", "máquinas no organizadas" y "máquinas auto-organizadas". Las dos últimas resultan ser más prometedoras que la primera, aunque la primera fue tan estupendamente exitosa que eclipsó las otras por casi medio siglo. Las máquinas no organizadas de Turing son lo que hoy en día se conoce como redes neuronales, mientras que sus máquinas auto-organizadas exploran uno de los procesos que ayuda a explicar el funcionamiento de los sistemas complejos, a nivel biológico, psicológico, económico, computacional, climático, entre muchos otros. La máquina organizada, la cual marcó las pautas para el estudio y simulación de los procesos mentales durante años, fue el ordenador digital. El ordenador digital es la invención más magistral del siglo veinte, incluso del segundo milenio (Grand, 2000). Pero aún así es sólo una máquina organizada.

En esencia, el problema es que el ordenador digital fue modelado basado en la apariencia exterior de los procesos mentales, en lugar de modelar las estructuras que dan su origen. Aunque se sabe sabemos que nuestros cerebros consisten de vastas cantidades de neuronas que operan en paralelo, actuamos como una sola mente. Esta mente pareciera que operara paso por paso, pensando en cómo llevar a cabo secuencias de acciones, una en cada instante. También tenemos la percepción de que nuestros pensamiento conscientes están en un nivel superior de la cadena de comandos: tomamos decisiones de forma consciente, pero delegamos la tarea de ejecutarlas a ciertas partes subconscientes de nuestro cerebro. Nuestra mente además nos da la impresión que emplea un enfoque descendente (top-down) que ejecuta una cadena de comandos, que es serial (i.e., sólo una mente por cerebro, operando un paso a la vez) y que es procedimental (i.e., trabaja en términos de procedimientos lógicos que pueden ser seguidos como una receta).

El ordenador digital es similar a una máquina serial porque sólo desempeña una operación a la vez. Es procedimental porque las unidades básicas de un programa son acciones que son ejecutadas (e.g., sumar dos números y almacenar el resultado). Éste también emplea un enfoque descendente (top-down), debido a que los programadores tienden a diseñar sus programas como jerarquías de control: un programa central desempeña comandos que emiten ordenes a las subrutinas, las cuales a su vez invocan rutinas subordinadas para llevar a cabo las funciones más atómicas. En otras palabras, lo que hace

parecer que una máquina posee comportamiento inteligente (e.g., la habilidad de jugar ajedrez) es solo, con frecuencia, la inteligencia almacenada de su diseñador, la cual es regurgitada secuencialmente.

En la naturaleza, todos los sistemas inteligentes son sistemas en paralelo (Grand, 2000). Incluso cuando creemos que somos una sola mente procesando sólo un pensamiento a la vez, en realidad somos una confederación débil de cientos de millones de máquinas muy pequeñas y estúpidas. Nuestros cerebros están hechos de un vasto número de neuronas, cada una operando en paralelo. No existe un controlador central ni serial, ejecutando procesos paso a paso. La inteligencia, contrariamente, es el resultado de millones de procesos no inteligentes que operan concurrentemente. En la práctica, la inteligencia es un constructo masivamente paralelo.

En términos computacionales, lo opuesto a emergente es algo pre-programado, ya que algo que es pre-programado dentro de un sistema puede ser descriptible enteramente en términos del lenguaje de sus partes. Un fenómeno emergente no es inminente en su substrato y no puede ser descrito en el mismo lenguaje de sus partes. Si cierto comportamiento es pre-programado en un sistema, no se le consideraría como inteligente. La emergencia requiere absolutamente del paralelismo porque de allí surgen las poblaciones de elementos que actúan en concierto. En realidad, actuar en "concierto" es una metáfora apropiada. ¿Cuánto aporta a la música el que muchas voces e instrumentos interpreten una pieza juntos a que si lo hiciesen uno tras otro de forma separada? (Calvin, 2001).

Resumiendo la idea de paralelismo que se quiere reflejar en METÁFORA, es que ésta debe ser necesariamente paralela en su naturaleza y que resulta aceptable simular este paralelismo empleando piezas discretas de tiempo en un ordenador serial. Después de todo, en el universo todo es paralelo: la historia no es una sino muchas; los procesos de causa y efecto no forman cadenas sino redes, etc.

4.4.2.2. Máquinas Autopoiéticas

Según Maturana y Varela (Maturana y Varela, 1980), desde una perspectiva mecanicista, una máquina o sistema autopoiético[1] es una "máquina auto-organizada" y homeostática que presenta una red de procesos u operaciones que la define como tal y la hace distinguible de los demás sistemas, y que puede crear o destruir elementos del propio sistema en respuesta a las perturbaciones del medio. Aunque la máquina cambie estructuralmente, su red de procesos permanece invariante durante toda su existencia, manteniendo la identidad de ésta. Los seres vivos son sistemas autopoiéticos que están vivos sólo mientras están en autopoiesis.

Maturana y Varela, proponen que todos los sistemas vivientes son autopoiéticos y cognitivos. Desde esta perspectiva, la cognición gestiona de forma auto-organizada, las interacciones entre un organismo y su entorno a través de bucles sensorio-motrices de bajo nivel, manteniendo así la autopoiesis del propio organismo. De algún modo, la cognición garantiza las condiciones necesarias para preservar la existencia del organismo (o sistema) como una red de procesos, a través de un acoplamiento con el entorno que

[1] El término autopoiesis fue definido anteriormente en la sección 2.1.1

permite su continuidad como entidad individual. En otras palabras, la cognición asegura la *"persistencia"* del sistema (natural o artificial) en el espacio y en el tiempo.

En (Varela, 1979), se explica cómo el nivel tradicionalmente considerado como cognitivo, en el cerebro de los animales multi-celulares, es en cierto modo la continuación del mismo proceso de autopoiesis identificado a nivel celular. Una visión interesante de Valera, es cómo el sistema nervioso y el sistema inmune en los vertebrado se comportan como sistemas autopoiéticos capaces de auto-regular sus procesos internos y de mantenerse en homeostasis con respecto a las perturbaciones del entorno (Varela y Coutinho, 1991).

En este orden de ideas, en METÁFORA, serán definidos múltiples sistemas autopoiéticos distribuidos en los distintos niveles de organización de la arquitectura. En el nivel inferior de procesamiento de la arquitectura se modelarán dos tipos de máquinas autopoiéticas: las Redes Neuronales Artificiales (RNA) y los Sistemas Inmunes Artificiales (SIA), en donde ambos sistemas operan como micro-sistemas cognitivos los cuales regulan sus propias funciones y auto-organizan sus estructuras cognitivas internas (i.e., neuronas y anticuerpos) a través de procesos de adaptación. En un nivel superior de organización, se proponen las Redes de Comportamientos (RC) como un sistema autopoiético a partir del cual emergen los procesos cognitivos de la consciencia.

En (Sarmiento, 2009) se expone una propuesta interesante que describe la autopoiesis como un fenómeno que se puede evidenciar en múltiples escalas del proceso cognitivo. Sarmiento afirma que los sistemas autopoiéticos se acoplan de tal forma que pueden generar tres tipos diferentes de sistemas auto-organizados: el sistema biológico, el sistema cognitivo, y el sistema social. A partir de esta propuesta se extraen cinco principios esenciales para el diseño de METÁFORA:

1. El ser humano participa en tres sistemas autopoiéticos, dos intrínsecos y uno extrínseco.
2. El primer sistema es de naturaleza biológica y su emergencia es la vida. Sus elementos constitutivos (las células) establecen relaciones entre ellos, cada vez más complejas. El mayor grado de complejidad surge en las células que componen el cerebro, pero la estructura que conforman, excepto en la neocorteza, es una variedad bidimensional.
3. Cuando la complejidad de un sistema autopoiético llega a un estado crítico emerge una nueva dimensión. La red neuronal humana como variedad bidimensional llega a su estado crítico en la neocorteza, emergiendo, gracias a su disposición columnar, una nueva dimensión (de variedad tridimensional) y con ella la *consciencia*, que todavía forma parte del primer sistema autopoiético humano.
4. El segundo sistema es de naturaleza simbólica y su emergencia es el pensamiento. La variedad tridimensional del cerebro humano alcanza su estado crítico y emerge una nueva variedad con una cuarta dimensión: el lenguaje.

5. El tercer sistema autopoiético, extrínseco, es de naturaleza comunicativa y su emergencia es la sociedad. El ser humano como elemento-substrato de este tercer sistema está sujeto a participar siempre en procesos cibernéticos de segundo orden y en su naturaleza no figura la posibilidad de acceder a una nueva variedad pentadimensional. Dicha posibilidad sólo la alcanzará la sociedad cuando llegue a su estado crítico.

En el diseño de METÁFORA, se consideran primordialmente los tres primeros principios descritos anteriormente, reemplanzando la concepción de ser humano por la de agente cognitivo, e intentando modelar el proceso cognitivo hasta el nivel emergente de la consciencia. Respecto al cuarto principio, la arquitectura exhibirá niveles de representación y razonamiento simbólico, deliberación, y planificación. En cuanto al quinto principio, éste será modelado parcialmente[2], mediante procesos auto-organizativos que serán dirigidos por un mecanismo de transmisión cultural de símbolos (i.e., un algoritmo memético), el cual estará soportado por un sistema multi-agente.

4.4.2.3. Sistemas Complejos Adaptativos

Un sistema complejo está compuesto por varias partes interconectadas, cuyos vínculos contienen información adicional que está oculta al observador. Como resultado de las interacciones entre elementos, surgen propiedades nuevas que no pueden explicarse a partir de las propiedades de los elementos aislados. Dichas propiedades se denominan propiedades emergentes (Waldrop, 1992).

La complejidad de un sistema puede ser de dos tipos: complejidad desorganizada y complejidad organizada (Weaver, 2007). En esencia, la complejidad des-organizada se encarga de un número muy grande de elementos, y la complejidad organizada se encarga del sistema en cuestión (muy posiblemente con un número limitado de partes) que exhibe propiedades emergentes.

En el contexto de los sistemas de complejidad organizada, el cerebro es modelado como un sistema celular complejo con dinámicas no-lineales. Mainzer (Mainzer, 2007), argumenta que la emergencia de los estados mentales (e.g., el reconocimiento de patrones, los sentimientos, los pensamientos, etc.) se explica por medio de la evolución de parámetros de orden macroscópico de ensamblajes cerebrales, que son causados por interacciones microscópicas no-lineales de células neuronales las cuales emplean estrategias de aprendizaje. Mainzer argumenta que la aproximación de los sistemas complejos no puede explicar qué es en realidad la mente, pero que es posible modelar las dinámicas de algunos de los estados mentales bajo ciertas condiciones.

Una disposición mental es comprendida como un estado global de un sistema complejo que es causado por interacciones locales no-lineales de sus partes, pero el cual no puede ser reducido a sus partes constitutivas. La aproximación de los sistemas complejos está muy lejos de poder explicar los sentimientos, la imaginación, y la creatividad, aunque se espera poder modelar algunos aspectos de sus dinámicas no-lineales (Arthur, 1997).

Los Sistemas Complejos Adaptativos (CAS, por su acrónimo en inglés de Complex Adaptive Systems) son un tipo específico de sistemas complejos. Algunas de las definiciones más aceptadas para este tipo de sistemas son:

- Según Holland (Holland, 1998), se definen como una red dinámica de muchos agentes (los cuales pueden ser representados por células, especies, individuos, empresas, naciones, etc.) que se desempeñan en paralelo, constantemente actuando y reaccionando frente a las actuaciones de otros agentes. El control de un CAS tiende a ser altamente disperso y descentralizado. Si existe algún tipo coherente de comportamiento en el sistema, éste tiene que emerger de la *competición y cooperación* entre los agentes mismos. El comportamiento general del sistema es el resultado de un gran número de decisiones tomadas a cada instante por muchos agentes individuales.
- Según Dooley (Dooley, 2002), un CAS se comporta y evoluciona de acuerdo con tres principios: el orden es emergente y opuesto a lo predeterminado, la historia del sistema es irreversible, y el futuro del sistema es con frecuencia impredecible. Los bloques de construcción básicos de un CAS son los agentes. Los agentes pueden explorar su entorno y desarrollar esquemas que representan reglas interpretativas y de acción. Estos esquemas son susceptibles al cambio y a la evolución.

Los modelos e ideas que conciernen a los CASs son esencialmente evolutivos, fundamentados en las perspectivas biológicas modernas sobre adaptación y evolución. La teoría de los CASs es un puente entre la teoría de sistemas y las ideas generalizadas sobre el darwinismo, la cual sugiere que los principios darwinianos sobre la evolución pueden explicar un amplio rango de fenómenos complejos, desde fenómenos cósmicos hasta fenómenos cognitivos y sociales (Waldrop, 1992). En la figura 4.1 se observan los elementos básicos de un CAS.

Lo que distingue a un CAS de un sistema multi-agente puro (SMA) es el enfoque que tiene este primero sobre las propiedades y características de alto nivel, tales como: la complejidad, la emergencia, y la auto-organización. Un SMA se define como un sistema compuesto de múltiples agentes que interactúan entre sí, mientras que en un CAS estos agentes son complejos y adaptativos. Los CASs están caracterizados por un alto grado de capacidad adaptativa, lo que les otorga gran flexibilidad en respuesta a las perturbaciones del entorno.

Algunas de las propiedades más importantes de los CAS (Waldrop, 1992) son: la adaptación, la homeostasis, la comunicación, la cooperación, la especialización, la organización temporal y espacial, y por su puesto, la reproducción. Estos sistemas se pueden encontrar en todos los niveles de organización del organismo: las células se especializan, se adaptan, y se reproducen tal como lo hacen los organismos superiores. La comunicación y cooperación toman lugar en todos los niveles, desde el nivel de agente hasta el nivel de sistema. Adicionalmente, las fuerzas que guían la cooperación entre agentes pueden ser analizadas por la teoría de juegos. Estas características de adaptación, evolución, cooperación y competición, serán descritas con más detalle en la sección 4.4.3, donde se habla acerca del proceso adaptativo a diferentes escalas.

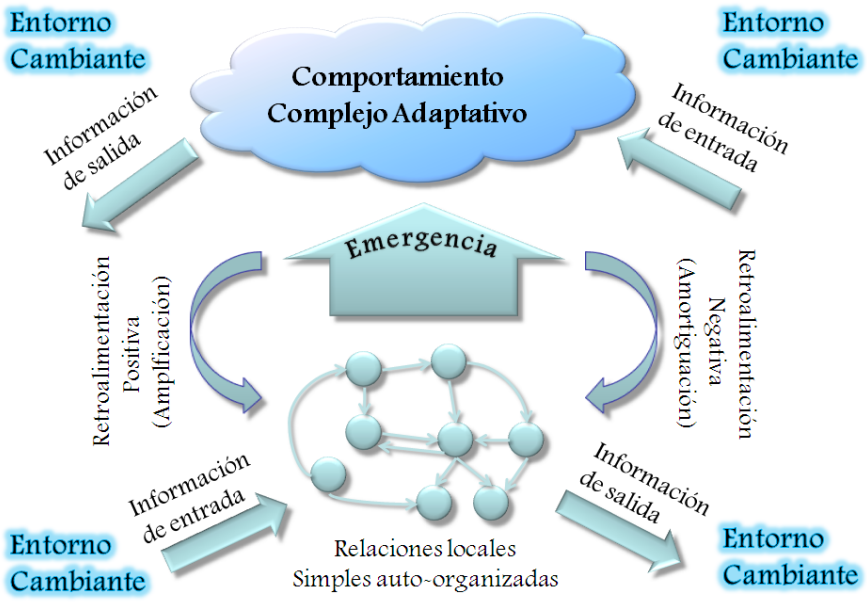


Figura 4.1: Sistema Complejo Adaptativo (Waldrop, 1992).

En analogía con los sistemas autopoiéticos descritos en la sección 4.4.2.2, los CAS delimitan sus propias fronteras, diferenciándose a sí mismos del entorno del cual reciben entradas de materia, energía e información. Si esto no sucediera, sería imposible distinguir el sistema del entorno. En (Goldspink y Kay, 2003), se propone una fusión entre la teoría autopoiética y la teoría de la complejidad, donde la teoría autopoiética provee las bases para el entendimiento de las características de los agentes de micro-nivel (e.g., células, neuronas, bloques de construcción, etc.), mientras que la teoría de los CAS provee las bases para la comprensión acerca del cómo estas características que surgen de la interacción en el micro-nivel, influyen en el rango y tipo de los agentes del macro-nivel (e.g., individuos y sociedades).

No existe una clara distinción entre los sistemas autopoiéticos y los CAS (Thompson y Varela, 2001), sin embargo, una de las principales diferencias es que los CASs definen claramente los procesos de evolución y adaptación del sistema ante el entorno cambiante, mientras que los sistemas autopoiéticos definen este tipo de procesos como un simple acoplamiento estructural con el entorno.

En METÁFORA, la interacción recíproca entre las máquinas autopoiéticas y los CAS resulta evidente: mientras las máquinas autopoiéticas (RNAs, SIAs, RCs, etc.) se encargan de los procesos auto-organizativos ocurridos en las estructuras del micro-nivel de cognición, por otro lado, los principios de la teoría de los Sistemas Complejos Adaptativos se aplican en la modulación de los procesos auto-organizativos de las estructuras complejas en el macro-nivel.

4.4.3. Lo que no se adapta...se extingue!: Comportamiento Adaptativo Multi-nivel

"–Pues en nuestro país –dijo Alicia, todavía un poco jadeante–, si corres muy deprisa durante mucho tiempo, como hemos estado haciendo nosotros, generalmente acabas llegando a algún sitio.
–¡Vaya país más atrasado! –dijo la Reina Roja–. Aquí, como ves, tienes que correr todo lo que puedas para mantenerte en el mismo sitio. Si quieres llegar a alguna otra parte, tienes que correr por lo menos el doble de rápido".
— Lewis Carroll, Alicia a través del espejo, 1871.

En analogía con lo que dice la Reina Roja, la *adaptación* es una manera especial de correr, imprescindible para la administración inteligente, la cual se convierte en un elemento necesario para garantizar la persistencia a largo plazo, en este caso, del agente cognitivo. En esta sección, se discutirán algunos de los principios de adaptación biológica, y cómo ésta puede dirigir los procesos cognitivos de METÁFORA desde los niveles de orden inferior hasta las estructuras complejas de orden superior, donde se supone emerge el comportamiento inteligente.

4.4.3.1. Retroalimentación, y más Retroalimentación...!

La retroalimentación, según Waldrop (Waldrop, 1992), es un proceso por el cual una señal viaja a través de una cadena de relaciones causales hasta relacionarse de nuevo consigo misma. La retroalimentación se puede dividir en dos categorías: positiva y negativa. La retroalimentación es positiva si el aumento de una variable, después de un retraso, provoca un nuevo aumento de esa misma variable. Las retroalimentaciones positivas se hallan cuando hay un refuerzo o amplificación en el sistema, que provoca un comportamiento exponencial. Por otra parte, una retroalimentación es negativa si el aumento de una variable provoca que posteriormente haya una disminución en esa misma variable. Las retroalimentaciones negativas llevan al equilibrio o estabilizan los sistemas (proceso denominado homeostasis), lo que produce un comportamiento asintótico o bien oscilante.

La forma más sencilla de explicar la retroalimentación es a través de un ejemplo. Considérese que existe un cambio brusco de temperatura en una habitación causado por una racha de aire frío. La caída de la temperatura puede desencadenar varios tipos de acciones. Por ejemplo, algunas personas en la habitación se podrían colocar un abrigo o desplazarse hacia otra parte más caliente de la casa. También puede ocurrir que el termostato ponga en marcha la calefacción. El efecto más importante de la actividad de la calefacción, es el calor que sale de los radiadores de la habitación, el cual provoca que la temperatura de la habitación se eleve.

Así pues, un *sistema de lazo cerrado* es una cadena circular de causalidad que sé "retroalimenta": la temperatura de la habitación disminuye y esto activa el termostato, el cual pone en marcha la caldera de la calefacción, la cual calienta los radiadores, que elevan la temperatura de la habitación, sin considerar las otras relaciones causales como el hecho de colocarse un abrigo. Un cambio en la temperatura de la habitación provoca otro cambio en la temperatura de la habitación. En el ejemplo del termostato, el efecto de los

dos cambios tiene sentido contrario: una disminución en la temperatura de la habitación provoca un posterior aumento de la temperatura de la misma. Si el termostato funciona correctamente, la temperatura en la habitación nunca debería alejarse mucho del ideal deseado.

Los sistemas de lazo cerrado se denominan sistemas retroalimentados. En un sistema retroalimentado, cada variable es –a la vez– causa y efecto. La temperatura de la habitación se afecta por la actividad de los radiadores, y afecta al termostato, el cual afecta a la caldera de la calefacción y al futuro estado de los radiadores. En un sistema retroalimentado un cambio en el estado real del sistema (e.g., un cambio en la temperatura de la habitación) lleva a una decisión (tomada por el termostato) que provoca una acción (poner en marcha la caldera de calefacción) que afecta al estado real del sistema (los radiadores elevan la temperatura de la habitación) y a su vez tienen influencia en futuras decisiones (la caldera de la calefacción se detiene una vez que se ha alcanzado la temperatura deseada en la habitación). Este ejemplo del termostato define un sistema con una retroalimentación negativa: un aumento en la temperatura de la habitación acaba provocando un descenso en la temperatura de la habitación, hasta alcanzar su valor inicial.

En un contexto más general, se puede pensar que las cosas en el mundo real no suelen ser simples sino por el contrario, los sistemas más interesantes suelen tener múltiples bucles de retroalimentación –algunos son positivos y otros negativos, algunos con respuestas rápidas y otros con respuestas un poco más lentas. Existe un bucle de retroalimentación de orden superior, llamado *"adaptación"*, el cual complica un poco más las cosas.

La adaptación es una característica de los sistemas biológicos, la cual ha sido denominada por Steve Grand (Grand, 2000), como la primera ley de la biología: esta puede ser resumida en la frase "la naturaleza es perezosa". La mayoría de los sistemas biológicos se ajustan automáticamente a sí mismos de manera que llegan a un estado de reposo cuando el mundo se encuentra en su estado promedio local. En otras palabras, estos sistemas rastrean un promedio cambiante de los estados del mundo y se adaptan lentamente a estos de tal manera que la mayor parte del tiempo están inactivos. Sólo responden a las desviaciones del promedio local. Así pues, la adaptación es uno de los trucos ingeniosos mediante el cual la naturaleza explota la retroalimentación.

En el resto de esta sección, se explicará como la adaptación podría ser útil en el diseño de los mecanismos de adaptación y aprendizaje a diferentes escalas en METÁFORA. Asimismo, se destacará la necesidad de comprender y manipular las complejas redes de bucles de retroalimentación que hacen posible que emerja el comportamiento inteligente y adaptativo en la arquitectura. Después de todo, la inteligencia es una herramienta que permite comprender el comportamiento de los bucles complejos de causa y efecto, y lo que asegurará la persistencia del agente cognitivo será su habilidad para desempeñar estos comportamientos y predecir el futuro a partir de su propio conocimiento sobre las relaciones causales.

La teoría de la complejidad (Kauffman, 1995) propone una metáfora interesante para explicar la retroalimentación positiva y negativa, llamada el *paisaje* del sistema. Esta metáfora describe un escenario bidimensional compuesto de valles, cordilleras, cráteres,

etc., donde el eje horizontal representa los posibles estados del sistema, y las montañas y valles en el eje vertical indican en qué estados es más probable que el sistema descienda y cuáles tenderá a evitar. La mayoría de sistemas reales tienen más de una variable –más de un aspecto que la retroalimentación puede afectar– así que representar este tipo de sistemas requeriría más de dos ejes. En esta metáfora, el sistema se comporta como una pelota que se desplaza sobre la superficie del paisaje, sorteando los precipicios e intentando mantenerse en los valles (estados de equilibrio u homeostasis), y tomando el suficiente impulso para escalar algunas colinas. En esta metáfora, los puntos máximos de la retroalimentación positiva y los puntos nulos de la retroalimentación negativa, representados por las depresiones y por las hendiduras de la superficie, son llamados los *atractores* del sistema, debido a que el sistema tiende a caer dentro de estos estados.

Una última demanda de la metáfora del *paisaje*, es que los bucles de retroalimentación pueden coexistir en diferentes regiones del paisaje, de la siguiente manera: la pelota (el sistema) puede rodar por una colina de retroalimentación positiva y circundar por un rato en el valle al descender la cuesta, después de que un bucle de retroalimentación negativa ha tomado el control. Los bucles de retroalimentación pueden interactuar también de maneras más sutiles. En este sentido, es muy posible imaginar diferentes jerarquías de retroalimentación, donde la función de retroalimentación de un bucle es *modulada* por la del otro. Por ejemplo, la ganancia de un bucle de retroalimentación negativa (la pendiente de la cuesta, y por consiguiente, la velocidad máxima a la cual desciende el sistema hasta llegar al equilibrio después de una perturbación) podría ser modulada por otro bucle de retroalimentación, como respuesta a alguna otra variable.

Ahora bien, en lo que respecta al diseño de METÁFORA, es posible modelar múltiples bucles interactuando en diferentes escalas de tiempo y en diferentes niveles o jerarquías. Pero es necesario considerar además, que en la metáfora del "paisaje", es posible encontrar dos o más pelotas (agentes) rodando simultáneamente, y por lo tanto es posible imaginar que una pelota puede perturbar el paisaje en el cual está rodando la otra. Surge entonces la necesidad de que estas dos pelotas se comuniquen de alguna manera o de lo contrario interferirán mutuamente en su camino. Ahora, esto tiene más sentido cuando se consideran los "paisajes evolutivos", donde las colinas significan algo adicional (e.g., niveles de aptitud y no solamente bucles de retroalimentación). Considerando que cada agente puede cambiar el paisaje de los demás, entonces cada uno tendrá que ser lo suficientemente astuto como para planificar los movimientos y tomar en cuenta no solamente la curvatura de las cordilleras sino también el cómo pueden otros agentes reaccionar y, por ende, qué cambios pueden ocasionar afectando su camino.

De estas ideas tan interesantes, surge un nuevo concepto llamado *"el comportamiento adaptativo"*, el cual podría describirse en los siguientes términos siguiendo la metáfora del paisaje. Todos los organismos (agentes) viven en el filo de la montaña: tienen recursos limitados de energía, y pueden dejar de ser persistentes en cualquier momento. El reto es mantenerse en el filo evitando resbalar tan lejos que luego no se pueda recuperar el camino. La habilidad de controlar el flujo de energía y cambiar el rumbo en respuesta a la información acerca de si se está deslizando cuesta abajo o no, es lo que se llama *comportamiento adaptativo* (Langley, 2005).

En realidad, el aspecto más importante del comportamiento adaptativo aplicado al diseño de METÁFORA, es que el agente cognitivo sea capaz de llegar al siguiente paso, y mirar hacia adelante, lo cual le permitirá *predecir* las necesidades de futuros cambios de dirección y crear planes en función de ello. Esta habilidad de *aprender* las relaciones de causa y efecto y usarlas para predecir el futuro, es lo que recibe el nombre de *inteligencia* (Grand, 2000), y esta inteligencia es lo que permite a los agentes inteligentes el poder generar planes, algo más que sólo "reacciones anticipadas". Así pues, METÁFORA estará en capacidad de predecir las relaciones causales mediante afirmaciones del tipo "que pasaría si…", las cuales le permitirán generar ideas innovadoras frente a situaciones desconocidas. Como mínimo, esta capacidad requiere que el agente pueda predecir secuencias no-lineales, esto es, el agente no sólo identificará dónde están las cuestas en determinado momento, sino dónde se hallará cuando otro agente lo empuje cuesta abajo.

Cabe notar en este punto, que cada nivel de resolución de problemas requiere una retroalimentación. El nivel más sencillo (el comportamiento adaptativo) requiere una retroalimentación negativa relativamente directa, para compensar los errores de dirección. Debido a que la firmeza del terreno del paisaje es desconocida y cambiante, incluso el nivel de amortiguamiento de los agentes al rodar cuesta abajo las colinas, puede necesitar ser controlado por la retroalimentación. De este mismo modo, la inteligencia requiere de varios niveles de retroalimentación. Algunos de ellos controlan el comportamiento, mientras otros controlan los *cambios en el comportamiento*, un proceso que es llamado *aprendizaje*. Los niveles más avanzados de inteligencia requieren retroalimentación incluso de los eventos que no han sucedido aún (Wiener et al., 1943).

El mundo exterior puede ser visualizado como un conjunto de bucles de retroalimentación, y lo que resulta interesante es que el mundo interior también puede ser visualizado de la misma manera. En este "espacio interior", los mecanismos de retroalimentación no sólo permiten que el agente pueda aprender, sino que también puedan representar el resultado actual de ese aprendizaje –estos son los bucles que determinan las respuestas del agente a los cambios de su entorno. En un sentido concreto, estos bucles de retroalimentación interna son la imagen en el espejo de lo que percibe el agente sobre su mundo exterior, debido a que la finalidad del agente es compensar las fuerzas externas que están amenazando su habilidad para persistir.

Si un bucle de retroalimentación positiva se presenta en el mundo real, como por ejemplo la trampa de la pobreza[3], la respuesta correcta del cerebro a esta situación puede ser representada por un bucle de retroalimentación negativa (una regla del tipo "recortar los gastos si el dinero está escaseando"). Debido a que los bucles de retroalimentación interna son un reflejo del mundo exterior, es posible pensar en estos como *modelos mentales* del mundo. De la misma manera, METÁFORA creará modelos mentales basados en bucles recursivos de retroalimentación interna, a través de los cuales podrá simular escenarios futuros y generar comportamientos anticipatorios.

Las teorías sobre el procesamiento de la información tienden a enfatizar la importancia de la retroalimentación en el aprendizaje puesto que el conocimiento de los resultados es necesario para corregir los errores y desarrollar nuevos planes. Por otro lado, las teorías

[3]La trampa de la pobreza se refiere a un defecto en el sistema social que no permite a los pobres mejorar su situación económica la cual cae en un ciclo vicioso.

comportamentales tales como las de Clark (Clark, 1935), Guthrie (Guthrie, 1930), Thorndike (Thorndike, 1932), y Skinner (Skinner, 1950), se enfocan en el rol del *refuerzo* en la motivación del individuo (en este caso del agente) con el propósito de que se comporte de cierta manera.

La retroalimentación por refuerzo conduce al agente a ir incrementalmente más rápido en la dirección que actualmente va, bien sea alejarse de su meta o ir hacia ésta. El refuerzo en los procesos de aprendizaje del agente puede ser de dos formas: positivo cuando se "recompensa" una acción correcta que se desea mantener como respuesta a situaciones similares futuras, y negativo cuando se "penaliza" una acción que se desea evitar en situaciones futuras. El refuerzo positivo está relacionado con la búsqueda del placer mientras el refuerzo negativo está relacionado con la evasión del dolor.

Una anotación final sobre los bucles de retroalimentación, es que son sólo eso, bucles que no poseen ni un principio ni un final. Por ejemplo, las glándulas suprarrenales no le indican al corazón que debe latir más rápido, simplemente responden a las señales secretando adrenalina; el corazón responde a la adrenalina latiendo más fuerte y rápido. Las glándulas suprarrenales no conocen en realidad qué es el corazón, y el corazón nunca ha escuchado acerca del sistema endocrino. La adrenalina por su parte no ha escuchado nunca acerca de ninguno de los dos.

Así pues, esta es la forma en que los organismos funcionan, y metafóricamente hablando, la forma en que la arquitectura cognitiva debería funcionar a distintas escalas: no existe un arquitecto, ni un controlador maestro que le diga a METÁFORA qué debe hacer. Sólo existe un vasto número de pequeñas entidades independientes que responden a las señales cuando les conviene, y emiten nuevas señales a destinos que desconocen.

4.4.3.2. Procesos Darwinianos y Máquinas Darwinianas

> *"Construimos modelos mentales que representan aspectos importantes de nuestro mundo físico y social, y manipulamos elementos de esos modelos cuando pensamos, planificamos y tratamos de explicar sucesos de ese mundo. La capacidad de construir y manipular modelos válidos de la realidad nos proporciona a los humanos nuestra característica* ventaja adaptativa. *Debemos considerarla como uno de los máximos logros del intelecto humano".*
> – Gordon H. Bower y Daniel G. Morrow.

El psicólogo William James decía que los procesos mentales funcionan de un modo darwiniano, y lo decía hacia 1890 (James, 1890), poco más de una década después de que Charles Darwin publicara el *El origen de las especies*. El concepto de ensayo y error lo desarrolló el psicólogo Alexander Bain en 1855 (Bain, 1855), pero James aplicaba además el pensamiento evolutivo.

Más de un siglo después, sólo se está empezando a añadir sustancia a esta idea con mecanismos cerebrales adecuados para el darwinismo. Durante varias décadas se ha hablado de la supervivencia selectiva de las sinapsis en las complejas redes neuronales en

el cerebro. Ahora se está empezando a vislumbrar también el "cableado" que podría hacer funcionar todo el proceso darwiniano, y probablemente en la escala temporal de la consciencia, de milisegundos a minutos.

Esta versión del darwinismo que da forma a lo improbable, implica generar grandes cantidades de copias de ciertas pautas de activación cerebral, con ligeras variaciones entre unas copias y otras, y dejar que estas variantes compitan por el dominio de un área de trabajo. En esta competición influye el modo, mejor o peor, en que estas pautas de activación espaciotemporal[4] se amoldan a los "bultos y baches de la carretera": las pautas memorizadas almacenadas en las uniones sinápticas. Toda esta visión mecanicista de los procesos darwinianos, es lo que Calvin denomina *"Máquina Darwiniana"* (Calvin, 1996).

El darwinismo no sólo podría modelar un cerebro mejor en dos millones de años sin que lo guiara de la mano un maestro alfarero; además, otro proceso darwiniano que actúa en el cerebro podría encontrar una solución más inteligente a un problema en la escala temporal del pensamiento y la acción: de milisegundos a minutos. La respuesta inmunitaria del cuerpo también parece ser un proceso darwiniano, por el que adquieren forma anticuerpos cada vez mejores y mejor adaptados a la molécula invasora, en una serie de generaciones que duran varias semanas.

Según el psicólogo evolutivo William Calvin (Calvin, 2001), los procesos darwinianos tienden a empezar por la base biológica: la reproducción. Siempre están haciendo copias. Las teorías sobre la toma de decisiones suelen argumentar que uno hace varios planes de movimiento –abrir la mano, hacer el signo de la victoria o formar una pinza de precisión con los dedos– y que estos planes alternativos *compiten* reproductivamente unos con otros hasta que uno de ellos "vence". Según estas teorías, se necesita una masa crítica de clones que de la orden para que se inicie la acción.

Pero el darwinismo implica mucho más que la mera reproducción y *competencia*. Cuando se intenta abstraer los rasgos esenciales de un proceso darwiniano a partir de lo que se sabe sobre la evolución de las especies y la respuesta inmunitaria, parece que una Máquina Darwiniana tendría que poseer seis propiedades fundamentales, las cuales tienen que estar presentes para que el proceso siga adelante:

- *Tiene que haber una pauta.* El caso clásico es la serie de bases de ADN que constituyen un gen. Tal como lo decía Richard Dawkins en *El gen egoísta* (Dawkins, 1992), la pauta también puede ser cultural, como una melodía. Dawkins acuñó el término *meme* para estas pautas. También podría tratarse de las pautas de actividad mental relacionadas con crear un pensamiento.

[4]Muchas veces se habla de las actividades mentales como si estuvieran subdivididas en fases: sentir, pensar y actuar: pero el problema es que muy pocas cosas ocurren en un punto concreto del tiempo y el espacio. Todas las actividades interesantes del cerebro implican pautas espaciotemporales de actividad celular, no muy diferentes de una melodía musical, donde el espacio es el teclado o la escala musical (Calvin, 1996). En el resto de la presente tesis, los términos *"pauta espaciotemporal"*, *"estructura cognitiva"*, *"modelo mental"*, y *"meme"*, serán empleados como sinónimos de un mismo concepto: estructuras mentales construidas por el agente a diferentes niveles representacionales (que van desde las estructuras más básicas o atómicas, hasta las estructuras complejas formadas por el auto-ensamblaje de las estructuras más simples) las cuales sirven como bloques para la construcción de procesos cognitivos de orden superior. La única consideración al respecto es que cuando se emplee el término *meme* se hará referencia a una estructura cognitiva susceptible de evolución, esto es, que a partir de su codificación cromosómica puede ser replicada, alterada, y adaptada.

- *De algún modo, se hacen copias de esta pauta.* Las células se dividen. La gente tararea o silba una melodía que ha oído. De hecho, la unidad de pauta (el meme) se define por lo que se puede copiar con semifidelidad: por ejemplo, la secuencia de ADN del gen se copia con semifidelidad durante la meiosis, mientras que los cromosomas o los organismos completos no se copian con ninguna fidelidad.
- *Las pautas cambian de vez en cuando.* Las mutaciones puntuales provocadas por los rayos cósmicos son, tal vez, las alteraciones más conocidas, pero son mucho más frecuentes los errores de copia y la recombinación (barajar las cartas) que tiene lugar en la meiosis.
- *Las copias compiten* por ocupar un espacio ambiental limitado.
- En el *éxito relativo* de las variantes influye un *ambiente polifacético.* A veces se suele decir que el ambiente "selecciona", o que hay reproducción selectiva o supervivencia selectiva. A esta diferencia de resultados, Charles Darwin la llamaba *selección natural.*
- La siguiente generación se basará en *las variaciones que sobrevivan hasta la edad de reproducirse* y encuentren pareja. La elevada mortalidad de los individuos jóvenes hace que el ambiente tenga mucha más importancia para ellos que para los adultos. Esto significa que las variantes supervivientes hacen sus apuestas por la supervivencia sobre una base que se ha desplazado, no sobre el centro original de las variaciones (esto es lo que Darwin llamaba el principio de la herencia). En la siguiente generación, se produce una nueva variabilidad alrededor de los últimos que tuvieron éxito. Muchas de las variantes nuevas serán peores que el promedio de los padres, pero puede haber algunas que estén aún mejor "adaptadas" al conjunto de condiciones del ambiente.

En este punto, es importante hacer hincapié en dos procesos que resultan ser necesarios no sólo en las Máquinas Darwinianas que definirá METÁFORA sino en los distintos niveles de complejidad y auto-organización de la propia arquitectura: *la competencia y la cooperación.* Estos dos procesos, como se verá durante el desarrollo de la tesis, están íntimamente relacionados con las estructuras auto-organizativas desde los niveles inferiores hasta los niveles superiores de METÁFORA (e.g., neuronas artificiales ⇒ anticuerpos ⇒ reglas ⇒ pautas espaciotemporales ⇒ bloques de conocimiento ⇒ memes ⇒ *Comportamientos* ⇒ planes ⇒ etc.).

La *competencia* promueve la "lucha por la supervivencia" entre los elementos constitutivos del sub-sistema, seleccionando los más "adaptados" y garantizando el éxito reproductor de los mismos, mientras la *cooperación* establece relaciones de altruismo entre los elementos constitutivos, lo cual promueve el trabajo en conjunto por un bien común.

Existen tres factores que pueden acelerar el proceso darwiniano: el sexo, el cambio climático, y las islas. El sexo (que consiste en barajar genes usando dos barajas) no es imprescindible para el proceso darwiniano, y tampoco lo es el cambio climático... pero le añaden sabor y velocidad, tanto si se desarrollan en milisegundos como en milenios. El

tercer factor que acelera el proceso darwiniano es la fragmentación y el aislamiento que la sigue –el proceso darwiniano funciona más deprisa en las islas que en los continentes. Para algunos procesos darwinianos que requieren velocidad (cosa que desde luego ocurre en la escala temporal del pensamiento y la acción), el proceso de fragmentación podría ser imprescindible.

> *"No se puede esperar que un hombre redondo encaje a la primera en un agujero cuadrado; hay que darle tiempo para cambiar de forma".*
> – Mark Twain.

Los cambios climáticos, en especial las repercusiones de los malos tiempos, son los que han promovido a lo largo de la historia de la tierra, los aspectos creativos de la evolución. La rapidez con que cambian las cosas es importante para cualquier modelo de inteligencia por incremento acumulativo, tanto si implica un cerebro más grande como si sólo se trata de un cerebro más organizado. En cualquier clima, puede evolucionar un cerebro especializado que resulte más eficiente que el generalizado sobrecargado; sin embargo, los cerebros generalizados poseen la versatilidad de conducta que les permite pasar de los agujeros redondos a los cuadrados: el aprendiz de muchos oficios tiene ventaja sobre el maestro de uno solo.

Los cambios climáticos pueden ser simulados por las perturbaciones ambientales que percibirá el agente. Por su puesto, si el agente sólo interactúa en un mismo entorno donde los elementos no sufren cambios considerables, el agente se especializará sólo en las tareas que requieran su atención para aquel entorno monótono, pero disminuirá su capacidad para adaptarse a entornos altamente cambiantes. Es necesario pues, encontrar un punto de equilibrio donde existan suficientes perturbaciones ambientales que promuevan la continua adaptación de los procesos cognitivos del agente pero que por otro lado, se eviten los cambios climáticos bruscos y veloces a los cuales el agente no logre adaptarse.

Retomando la metáfora de la Máquina Darwiniana aplicada al diseño de METÁFORA, se podría decir que las pautas cognitivas pueden necesitar múltiples copias, y que las copias de las pautas pueden tener que "competir" con copias alternativas en "islas" durante una serie de "cambios climáticos" mentales, para que evolucione rápidamente una conjetura inteligente.

Las copias de las pautas, pueden generar cierta resonancia y sincronización entre las mismas. Para entender un poco mejor esta resonancia y sincronización entre las pautas, es necesario referirse a una metáfora que se ajusta a este fenómeno neuronal: Si una persona canta en un coro, tiene que sincronizarse con los demás, y para ello escucha: por lo general, se oye a sí mismo entrando demasiado tarde o demasiado pronto. Pero, por supuesto, esa persona también influye en los demás. Aunque todos sean un poco duros de oído, al poco tiempo todos estarán sincronizados, gracias a esta "retroalimentación".

La situación en el coro es muy similar a la de una neurona piramidal superficial en el cortex (Schubotz y von Cramon, 2001), que recibe por todas partes impulsos excitantes

de sus vecinas. La sincronización se alcanza incluso con niveles muy pequeños de retroalimentación: dos péndulos idénticos, situados uno junto al otro, tienden a sincronizarse sólo por las vibraciones que producen en el aire y en el soporte. Aunque los osciladores armónicos, como los péndulos, tardan algún tiempo en sincronizarse, los sistemas no lineales, como la producción de impulsos en las neuronas, se puede sincronizar con mucha rapidez, aunque la fuerza de conexión mutua sea relativamente débil.

A continuación, se ofrece una analogía entre cómo podrían funcionar los procesos darwinianos en la corteza cerebral según (Calvin, 2001) y cómo podrían ser simulados dichos procesos darwinianos en METÁFORA. Supóngase que el agente percibe algo que pasa a toda velocidad justo a su lado, y desaparece bajo una silla. Al agente le ha parecido algo redondo, y tal vez amarillo o anaranjado, pero se movía muy deprisa y ahora ya no lo ve, así que no puede echarle una segunda mirada. Para saber qué era y conjeturar una respuesta obvia, lo primero que haría el agente (similar a como sucede en el cerebro), es encontrar algunos candidatos y después compararlos, a ver cuál parece más razonable.

Afortunadamente, las competiciones entre clonaciones pueden realizar esta tarea. Similar a como funciona un "código cerebral" en el cerebro[5], en el agente existe una pauta espaciotemporal (meme) tentativa para el objeto, formada por todos los detectores de rastreo que activó: color, forma, movimiento, y puede que el sonido al rebotar en el suelo. Esta pauta empieza a hacer clones de sí misma (auto-replicarse), por decirlo de algún modo.

Si el agente ya hubiera percibido muchas veces un objeto parecido con anterioridad, podría haber una resonancia perfecta... pero supóngase que no lo ha visto. Aún así, el código cerebral (meme) tentativo incluye componentes especializados en Redondo, Amarillo, y Rápido... Las pelotas de tenis poseen esos mismos atributos, y ⟨`pelota de tenis`⟩ obtiene una buena resonancia, de modo que en la zona adyacente se activa la melodía Pelota de Tenis (un aspecto muy positivo de los atractores es que pueden coger un patrón muy parecido y transformarlo en la pauta característica). Las clonaciones con poca resonancia hacen que se descarten algunos componentes, y es posible que la variante ⟨`mandarina`⟩ provoque una resonancia que sea captada en otra zona especializada de la arquitectura, a pesar de que el color no es del todo correcto.

Respecto a la competición entre clonaciones, en ese momento se estarían clonando los códigos cerebrales ⟨`desconocido`⟩, ⟨`pelota de tenis`⟩ y ⟨`mandarina`⟩. Es posible que también aparezca ⟨`manzana`⟩: si hace poco el agente observó a alguien comiéndose una manzana, podrían haber quedado huellas temporales de la pauta Manzana en la memoria de trabajo. Pero enseguida, la pauta Manzana es derrotada por la pauta Mandarina, que se está clonando en otra parte. Al otro lado del territorio de "Desconocido", Pelota de Tenis está progresando muy bien, y llega un momento en que derrota y sustituye a Desconocido, invadiendo incluso el territorio de Mandarina. Aproximadamente a estas alturas, el agente podría comunicar "Creo que era una pelota de tenis", porque en el coro de Pelota de Tenis hay ya clones suficientes para llegar a esta deducción.

W. Calvin (Calvin, 2001), propone que este tipo de clonaciones de pautas podrían sugerir también una manera en que los pensamientos subconscientes pueden vagar y de

[5]Los códigos cerebrales, al igual que los códigos de barras de los supermercados, son patrones abstractos que representan el objeto real (Calvin, 2001).

vez en cuando hacer surgir en el torrente consciente algún hecho relevante del pasado. Lo mejor de todo es que, como las propias variantes pueden clonarse hasta lograr un éxito momentáneo, el mosaico de pausas es creativo y puede dar forma a algo de calidad a partir de orígenes humildes. Incluso parece probable que surjan formas de relación de grado superior, como *las metáforas*, porque las pautas son arbitrarias y capaces de formar nuevas combinaciones.

La pauta espaciotemporal vencedora de la competición de copia, la que tiene el coro más grande compitiendo por la atención de las rutas de salida, parece un buen candidato a lo que los neurofisiologos llaman consciencia (Dennett, 1991; Calvin, 1996; Baars, 1997). El foco de atención variable podría ser otro clon que salta al primer plano. El subconsciente podría estar formado por los otros patrones dominates. No hay una zona concreta en la arquitectura que sea "el centro de la consciencia" durante mucho tiempo; enseguida hay otra que toma el control. En el apartado 4.6, se describirá con más detalle una de las metáforas que se empleará para la simulación de los procesos conscientes y subconscientes en la aproximación arquitectónica.



Figura 4.2: Resonancia de pautas espaciotemporales. El agente percibe un objeto (generalmente con características borrosas o con ruido) e inicia la competición entre las pautas espaciotemporales A y B por obtener el control del área de trabajo. Estas pautas corresponden a recuerdos previos del agente. Cada pauta realiza copias de sí misma, replicándose con mayor eficiencia –mayor resonancia del coro– la pauta B cuyo patrón se asemeja más al objeto percibido. Mientras tanto, débilmente se clona la pauta C que puede corresponder a un recuerdo lejano o borroso.

En la figura 4.2, se esquematiza la idea general de la competición entre pautas en el proceso darwiniano anteriormente descrito para el ejemplo de la Pelota de Tenis. Extra-

polando un poco este proceso, se podría pensar en una capacidad de simulación de planes futuros a través de procesos recursivos de clonación y competición entre pautas que condujeran a nuevas situaciones hipotéticas donde nuevamente se aplicarían los procesos darwinianos para decidir qué hacer a continuación.

Una importante aplicación de las Máquinas Darwinianas en METÁFORA es la de los planes deliberativos: la planificación en una escala temporal de segundos, horas, días, etc., permitiría ensayar combinaciones, juzgar los inconvenientes de cada una a través de simulaciones internas, buscar resonancias para las pautas que representan los escenarios más probables, etc. Bien se podría considerar que esto es una característica clave del comportamiento inteligente.

4.4.3.3. Simulando la Epigénesis, la Ontogénesis y la Filogénesis

Si se desea situar el problema de modelar un sistema cognitivo en un contexto biológico, es necesario desplazarse al dominio de la IA bio-inspirada. Tomando como referencia los procesos naturales de adaptación, se propone simular en METÁFORA tres enfoques biológicos que permiten el proceso de adaptación de las funciones cognitivas a diferentes escalas organizativas: la *Epigénesis*, la *Ontogénesis* y la *Filogénesis* (las cuales fueron descritas en los apartado 2.1.2).

En los sistemas cognitivos emergentes de tipo enactivo[6], con frecuencia se enfatiza la diferencia entre filogénesis, ontogénesis y epigénesis. La filogénesis se refiere a la configuración inicial del sistema cognitivo y su evolución de generación a generación. Por otro lado, la ontogénesis se refiere al desarrollo de las estructuras y procesos cognitivos del sistema, y la epigénesis al conocimiento que adquiere mediante aprendizaje durante la interacción del agente con el mundo.

En la postura emergente, existe una negociación entre la configuración filogenética, el desarrollo ontogenético, y el aprendizaje epigenético. La filogénesis determina el conjunto de habilidades y comportamientos innatos o pre-configurados. Por otra parte, el desarrollo ontogenético y el aprendizaje epigenético motivan la creación de capacidades cognitivas a partir de la configuración filogenética. Debido a que no tendría mucho sentido simular en METÁFORA las escalas de tiempo evolutivas que permitan la emergencia filogenética de un sistema cognitivo –no sería posible computacionalmente, al menos por ahora, esperar que el modelo cognitivo evolucionase a partir de cero– surge la necesidad de identificar un estado filogenético inicial de las estructuras cognitivas a partir del cual el sistema cognitivo continúe la evolución.

A continuación se enuncian dos aspectos complementarios los cuales serán tomados en cuenta para la construcción de la arquitectura cognitiva: el primero llamado *Co-determinación*, es la auto-organización del sistema como una entidad independiente lo cual se refleja a través de su filogenia, y el segundo, llamado *Co-desarrollo*, es el acoplamiento del sistema con el ambiente a través de sus mecanismos de percepción y actuación, reflejado a través de los procesos ontogenéticos y epigenéticos del sistema.

[6]Los sistemas enactivos son descritos en la sección 3.1.2.

La co-determinación resulta de la naturaleza autónoma de los sistemas cognitivos. Refleja el hecho de que un sistema autónomo se defina a sí mismo a través de procesos de auto-organización y subordine todos los demás procesos para la preservación de la autonomía (Varela, 1979). Sin embargo, la co-determinación también refleja el hecho de que todos los sistemas auto-organizados tienen un ambiente en el cual están inmersos, se diferencian de los demás, y donde cada uno se concibe como un sistema que se apoya de cualquier forma en este proceso de autonomía-preservación. De esta manera, el sistema y el ambiente son co-determinados: el agente cognitivo es determinado por su ambiente al sustentar su autonomía ante las perturbaciones ambientales, y al mismo tiempo el proceso cognitivo determina qué es real o significativo para el agente, exactamente por la misma razón. En este sentido, la co-determinación significa que el agente construye su modelo de la realidad como un resultado de su operación en el mundo.

Maturana y Varela han presentado de manera gráfica la manera de comunicar la naturaleza de los sistemas co-determinados, autónomos, y auto-organizados, perturbando y siendo perturbados por su ambiente (Maturana y Varela, 1987), como se observa en la figura 4.3. La flecha circular indica la autonomía, la auto-organización, y la auto-producción del sistema; la línea ondulada representa el ambiente; y las semi-flechas bi-direccionales perturbaciones mutuas –acoplamiento estructural– entre los dos. El diagrama de la izquierda denota un sistema autopoiético y el de la derecha un sistema autónomo capaz de evolucionar mediante la auto-perturbación –auto-modificación– de su propio sistema nervioso, de tal forma que puede acomodarse a un espacio efectivo de acciones del sistema, mucho más amplio.

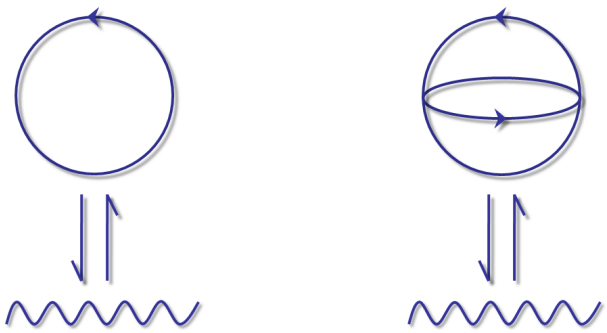


Figura 4.3: Sistema Autopoiético. Ideogramas de Maturana y Varela. El diagrama de la izquierda denota un sistema autopoiético. El diagrama de la derecha denota un sistema cerrado autónomo con un sistema nervioso central.

La Co-determinación requiere entonces, que el sistema sea capaz de ser autónomo como una entidad. Esto es, que tenga un proceso de auto-organización (como se describió anteriormente en el apartado 4.4.2) que demuestre la capacidad de percibir y actuar coherentemente, es decir, que el agente posea las condiciones esenciales para su supervivencia y desarrollo. Esto es exactamente a lo que se quiere referir con el término *configuración filogenética* de la arquitectura cognitiva: cada agente que implementa METÁFORA, considerado como un sistema autónomo, posee un conjunto de capacidades *innatas* que forman la base de cualquier desarrollo adaptativo subsecuente, las cuales son transformadas en cada generación a través de procesos evolutivos a nivel poblacional.

Por otro lado, el Co-desarrollo es justamente el proceso cognitivo de establecer y aumentar el espacio posible de acoplamientos mutuamente consistentes en los cuales la arquitectura puede emplearse. En este sentido, el espacio de posibilidades perceptuales del agente no se determina sobre un ambiente objetivo absoluto, sino sobre el espacio de posibles acciones que el agente puede realizar mientras se mantiene la consistencia del acoplamiento con el ambiente.

Estas perturbaciones ambientales no controlan el sistema debido a que no son componentes del sistema (y por definición, no forman parte del proceso de auto-organización), pero sí intervienen en el desarrollo ontogenético y epigenético del sistema. A través de dicho desarrollo, el agente construiría su propia epistemología, por ejemplo, su propia historia y su conocimiento (dependiente del contexto) acerca del mundo, conocimiento que tiene significado precisamente porque captura la consistencia e invariancia que emerge de la auto-organización dinámica producida en respuesta al acoplamiento ambiental.

Este proceso de desarrollo es logrado a través de la auto-modificación, debido a la presencia de un sistema nervioso central: éste no sólo provoca que el ambiente perturbe al sistema (y vice versa) sino que además el sistema se perturba a sí mismo y el sistema nervioso central se adapta como resultado. Consecuentemente, el sistema puede acomodarse a un espacio más amplio de acciones efectivas. Esto se puede observar en el segundo ideograma de Maturana y Varela en la figura 4.3, donde se agrega una segunda flecha circular al ideograma de autopoiesis para representar el proceso de auto-perturbación y auto-modificación. Así pues, el desarrollo ontogenético y epigenético que debe coordinar METÁFORA para cada agente, debe estar soportado por procesos de auto-perturbación y auto-modificación, lo que se podría denominar el sistema nervioso central del agente.

Los aspectos de co-determinación y co-desarrollo pueden influenciarse de manera mutua y recursiva. El sistema y el ambiente son co-determinados (a través del acoplamiento mutuo y la auto-organización emergente, como se explicó anteriormente), sin embargo algunos sistemas cognitivos pueden además adaptarse a través de un proceso continuo de co-desarrollo que resulta en nuevos acoplamientos co-determinados (Maturana y Varela, 1987). El co-desarrollo requiere además cierta plasticidad adicional en el proceso auto-organizativo. Si esto sucede, se obtienen dos cosas: las co-determinaciones condicionadas filogenéticamente entre el sistema cognitivo y su ambiente; y el potencial para el co-desarrollo ontogenético y epigenético del sistema mismo a lo largo de su "persistencia".

Es importante enfatizar que el proceso de desarrollo de cada agente ocurre de una manera muy especial. La actuación, la percepción, y la cognición están fuertemente acopladas en el desarrollo: no sólo se permite que la actuación organice la percepción y la cognición, sino que la percepción y la cognición son esenciales también para la organización de la actuación. Las actuaciones de los sistemas no parecen ser pre-diseñadas ni tampoco son determinadas por la experiencia, sino que éstas resultan de la operación del sistema nervioso central y las interacciones dinámicas con el ambiente.

Según Thelen (Thelen, 1995), la percepción, la cognición, y las motivaciones componen la interfaz entre los procesos cerebrales y las acciones. Consecuentemente, la cognición avanzada es el resultado de un proceso de desarrollo a través del cual el sistema adquiere progresivamente más destrezas y logra la habilidad de comprender eventos,

contextos, y acciones, inicialmente que tratan con situaciones inmediatas e incrementalmente adquieren una capacidad prospectiva o predictiva, de la que se habló en el apartado 4.4.3.2.

En este sentido, el concepto de *co-determinación* que fundamenta la idea de Maturana y Varela sobre un acoplamiento estructural de primer orden en los sistemas autopoiéticos, resulta similar a la causalidad circular de acción y percepción propuesta por Kelso (Kelso, 1995) donde es el sistema quien gestiona su interacción mutua con el mundo, y refleja además, los principios organizacionales heredados de los sistemas auto-gestionados de Bickhard (Bickhard, 2000). Por otra parte, el concepto de *co-desarrollo* tiene sus orígenes en los sistemas autopoiéticos de nivel dos y tres propuestos por Maturana y Varela (Maturana y Varela, 1987).

Ahora, es importante destacar el significado del término "interacción". En el contexto de las arquitecturas cognitivas, cuando se decanta por una postura de cognición emergente (como es el caso de METÁFORA), la interacción es una actividad compartida en la cual las acciones de cada agente influyen en las acciones de otros agentes concebidos en la misma interacción, resultando así un patrón de comportamientos compartidos que se construye mutuamente (Ogden et al., 2002). Esto es consistente con el paradigma de cognición emergente, especialmente con la interacción inspirada en conceptos de autopoiesis y acoplamiento estructural (Maturana y Varela, 1980), tal como se observa en la figura 4.4.

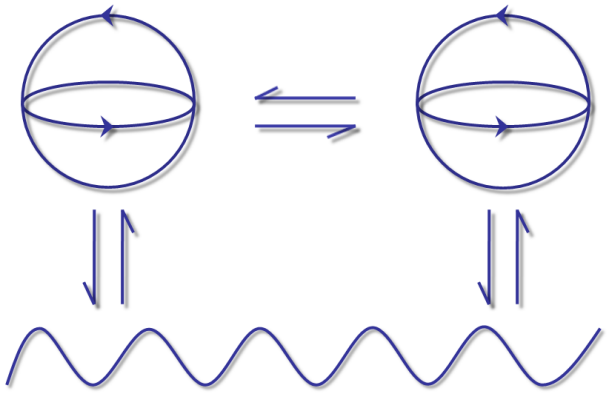


Figura 4.4: Co-desarrollo en Sistemas Autopoiéticos. Ideogramas de Maturana y Varela que denotan el co-desarrollo concebido por la interacción entre sistemas cognitivos.

Tales patrones de comportamientos complementarios construidos de forma mutua, son también destacados en la noción de actuación acoplada propuesta en (Clark, 1994). Además, el significado explícito no es necesario para comunicar algo en la interacción, simplemente es importante que los agentes estén involucrados mutuamente en una secuencia de acciones. El significado emerge a través de la experiencia consensual y compartida que es mediada a través de la interacción. Respecto al proceso de interacción entre los agentes cognitivos, METÁFORA define un mecanismo de enfoque filogenético a través del cual se intercambian y co-evolucionan pautas culturales o memes.

Una de las definiciones más aceptadas para la co-evolución natural, es la de Janzen (Jansen, 1980): "la co-evolución es el proceso por el cual dos o más organismos ejercen

presión de selección mutua y sincrónica (en tiempo geológico) que resulta en adaptaciones específicas recíprocas". Así, si no hay adaptación mutua, no puede hablarse de co-evolución. Por ejemplo, las ranas y las moscas; si las ranas adquieren una mutación, de una lengua más rugosa que permita atrapar más moscas, las moscas tendrán que evolucionar para lograr ser más escurridizas y evitar ser atrapadas por las lenguas mutadas de la rana. Lo mismo puede suceder si la mosca llega a sufrir cambios y mutaciones que lleven a la rana a adaptarse.

En METÁFORA, la evolución del conjunto de memes a través de un Algoritmo Memético (AM) trabajaría en dos niveles y en dos escalas temporales: en una escala temporal a corto plazo, un conjunto de agentes operaría en el espacio de búsqueda asociado al problema, mientras que en una escala temporal a largo plazo, los agentes adaptarían las heurísticas a ellos asociadas.

El AM de cada agente se vería afectado por procesos co-evolutivos, donde la adaptación del meme a puede influenciar en la adaptación de los memes $b, c, d, ...etc.$ con los cuales se encuentra relacionado. De esta manera, podría argumentarse que existe, una micro co-evolución a nivel ontogenético (en una escala temporal a corto plazo), donde cada agente adapta sus propios memes de manera recíproca a través de la interacción con el entorno (adaptación local); y una macro co-evolución de memes a nivel filogenético (en una escala temporal de largo plazo) que actúa cuando los memes evolucionan paralelamente (desde el punto de vista poblacional) en repositorios discretizados por tipo de meme, o en otras palabras, en el repositorio a evolucionarán los memes $a_1, a_2, a_3...a_n$, donde n es el número de agentes que se encuentran interactuando en el entorno, y así sucesivamente con cada uno de los tipos de memes presentes en el acervo $(a, b, c, d...)$.

Por lo tanto, la metáfora de los procesos co-evolutivos observados en la naturaleza, se reflejará en una co-evolución de memes en la cual se cumplen las siguientes condiciones:

- *Especificidad*: la evolución de cada rasgo (meme) en un *Comportamiento* es debida a presiones selectivas de otros rasgos (memes) de los demás *Comportamientos* del sistema.

- *Reciprocidad*: los rasgos en los *Comportamientos* participantes del sistema evolucionan conjuntamente.

- *Simultaneidad*: los rasgos en los *Comportamientos* participantes del sistema evolucionan al mismo tiempo.

En síntesis, los aspectos de co-determinación y co-desarrollo en METÁFORA, simulados a través de los procesos de filogénesis, ontogénesis y epigénesis, se reflejan de la siguiente manera: el enfoque filogenético definirá la base cognitiva innata a través de la cual se producirá el desarrollo epigenético y ontogenético subsecuente de la arquitectura; a partir de esto, el enfoque epigenético coordinará los mecanismos de aprendizaje de la arquitectura, los cuales son afectados por la interacción del agente con el entorno; por otra parte, el enfoque ontogenético tendrá la responsabilidad de gestionar la especialización de *Comportamientos*, la orquestación de los procesos conscientes, y la estructuración de los planes deliberativos, a través de procesos darwinianos; finalmente, el enfoque filogenético, retomando la base cognitiva modificada y adaptada por los procesos ontogenéticos y epigenéticos, se ocupará de coordinar, a través de procesos co-evolutivos (en un entorno multi-agente), la recombinación genética de tres tipos de memes: (1) las unidades básicas de construcción de los *Comportamientos* (reglas simbólicas y sub-simbólicas), (2) las redes de comportamientos, y (3) los planes deliberativos. El proceso continúa indefinidamente en un bucle de lazo cerrado.

A diferencia del modelo biológico, en el enfoque filogenético de METÁFORA no serán evolucionadas las características genotípicas relacionadas con el fenotipo corporal del agente, sino que serán evolucionadas las estructuras cognitivas, es decir, serán evolucionados los diferentes tipos memes y no los agentes como individuos (i.e., no se simulará el proceso de reproducción que dará origen a nuevos agentes), y por tanto, ésta será una evolución sólo a nivel cognitivo. Obsérvese que ésta podría ser una manera incipiente de transmisión cultural donde surge un nivel de "persistencia" de mayor jerarquía: *el comportamiento social entre agentes como fenómeno emergente*.

Para concluir, en cada uno de los enfoques (epigenético, ontogenético y filogenético) se implementarán procesos darwinianos a diferentes escalas, empleando las Máquinas Darwinianas descritas en el apartado 4.4.3.2 con el fin de *adaptar* cada nivel de organización de la arquitectura al entorno dinámico del agente.

4.5. Bloques de construcción: Lego vs. Puzzle

En esta sección se describirán, desde una perspectiva mecanicista, un conjunto de componentes o bloques de construcción cibernéticos de primer orden, los cuales pueden ser ensamblados entre sí para producir un modelo cognitivo articulable. La importancia de estos bloques, no se fundamenta en su capacidad de discretizar de forma física los componentes identificables de la arquitectura, sino que, desde un punto de vista completamente diferente, identifican las relaciones de causa y efecto y las redes de bucles de retroalimentación entre dichos componentes de la arquitectura (como se trató anteriormente en el apartado 4.4).

La metáfora que acá se pretende aplicar es, que el diseño estructural y relacional de los componentes de una arquitectura cognitiva no se corresponde con el modelo de un puzzle donde todos los elementos, mecanismos, módulos, componentes y demás, encajan de una manera precisa y absurdamente acoplada, sino que por el contrario, corres-

ponde más con la definición de un conjunto de bloques, en analogía a los usados en Lego[7], los cuales se puedan ensamblar libremente entre sí, creando diferentes estructuras y articulando nuevos patrones que puedan dar origen a diferentes configuraciones de arquitecturas cognitivas.

La idea de los bloques de construcción, se inspirará en principios observados en los sistemas complejos, tales como los sistemas biológicos, los sistemas electrónicos, los sistemas mecánicos, y los sistemas de vida artificial (Ashby, 1956; Levy, 1992; Grand, 2000). A continuación, se describen los bloques de construcción que se emplearán en el diseño de METÁFORA:

El Modulador: un transistor electrónico puede usar un flujo de causa y efecto –una señal que rebota de electrón a electrón a lo largo de un cable– con el fin de modular otro flujo. Éste se convierte en retroalimentación cuando el flujo que está efectuando la modulación es también el flujo que está siendo modulado. Si parte de la señal de salida es retroalimentada de nuevo al origen, entonces el flujo se controla a sí mismo. A nivel biológico, esto ocurre también con las enzimas, las cuales modulan la producción de sustancias químicas en el cuerpo. La modulación es directa cuando se comunican señales electroquímicas a lo largo de las fibras nerviosas, y es difusa cuando se transmiten señales químicas a través del flujo sanguíneo, mediante el fluido entre las células o entre las neuronas en el cerebro.

El cerebro humano está lleno de neurotransmirores y neuromoduladores que con frecuencia envían y reciben señales químicas que recorren largas distancias. Por ejemplo, el cerebro necesita percatarse de qué tan hambriento está el individuo en un determinado momento, y esta información no es accesible directamente del entorno, pero puede ser determinada por el monitoreo de los niveles de productos intermedios provenientes del sistema digestivo y del sistema respiratorio. METÁFORA empleará los *moduladores* para articular el nivel de concentración de las señales que se envían desde los diferentes módulos, permitiendo así acelerar o ralentizar los procesos entre dichos módulos. Los moduladores, serán también de gran utilidad en la amplificación o reducción del efecto de las señales de refuerzo (retroalimentación) que provienen tanto del ambiente, como del módulo Motivacional de la arquitectura.

El Transductor: por definición, en electrónica un transductor es un dispositivo que transforma el efecto de una causa física, como la presión, la temperatura, la dilatación, la humedad, etc., en otro tipo de señal, normalmente eléctrica. En biología, hace referencia a una entidad, por lo general una proteína o un conjunto de proteínas, que lleva a cabo la trasformación de una acción hormonal en una actividad enzimática. Una neurona en el cerebro que emite un neurotransmisor está llevando a cabo una transducción entre una señal eléctrica y una química. Cuando un sonido es convertido en señales nerviosas por el oído, realiza una conversión de información difusa a información directa; cuando una señal nerviosa en la glándula pituitaria causa la secreción de una hormona dentro del flujo sanguíneo, la transformación es de directa a difusa.

En METÁFORA, tanto las señales directas como las difusas, se encargan de transmitir información que produce la activación de algún componente, módulo, y/o mecanismo de la arquitectura. Por tanto, la diferencia entre estos dos tipos de señales radica en que

[7]Lego es una marca registrada de juguetes, principalmente bloques de plástico interconectables.

la señal directa transmite información de activación de forma inmediata, mientras que la señal difusa permite la acumulación paulatina de información (en analogía con la difusión química en el cerebro) hasta que se alcanza un umbral determinado de activación, después del cual es disparada una acción, una conducta, una emoción, etc. Así pues, el transductor se encargará de convertir estos dos tipos de señales en ambas direcciones.

Adicionalmente, existen otros dos tipos de información que requieren un proceso de conversión: la simbólica y la sub-simbólica. Como se ha discutido antes, METÁFORA hibridiza estos dos tipos de información (o conocimiento) en diversos módulos de la arquitectura. A fin de permitir la comunicación y sincronización entre los módulos de representación simbólica (Cognitivista) y los módulos de representación sub-simbólica (Emergente), se definen dos tipos de transductores: el *transductor-ad* que traduce las señales análogas (simbólicas) en señales digitales o binarias (sub-simbólicas), y el *transductor-da* que efectúa la acción contraria, traduce las señales digitales en señales análogas.

El Diferenciador: en ocasiones, lo que importa no es el nivel absoluto de una señal sino cuál es su tasa de cambio, o incluso, cómo "cambia" su tasa de cambio. Una de las principales razones por las que un diferenciador puede ser útil para un sistema biológico, es que éste permite determinar si la intensidad de una señal incrementa o disminuye. Por ejemplo, es posible que un par de reacciones químicas del cerebro produzcan alguna sustancia cuya concentración es la medida de qué tan rápido se eleva la concentración de otra sustancia. Este químico debe entonces activar un quimoreceptor en el cerebro que inicia una respuesta a partir de éste.

Este bloque de construcción será útil en el diseño de METÁFORA, principalmente en el control del foco de atención del agente. Por ejemplo, supóngase que el agente fija su atención en un objeto de su entorno, sin embargo, si este objeto no sufre cambio alguno con el paso del tiempo, el agente deberá perder gradualmente su interés sobre éste, a fin de cambiar el foco de atención en otros objetos que sí requieran su atención. En el momento en que aparezca repentinamente un objeto en el campo de percepción del agente, éste deberá recibir con mayor intensidad la señal de dicho objeto mientras las señales de los objetos ya presentes en el campo de percepción irán perdiendo poco a poco su intensidad, y el agente estará en capacidad de identificar este cambio de intensidad de la señal de los objetos percibidos y, decidir así, hacia dónde debe enfocar su atención nuevamente. Algo similar podría ocurrir con los elementos que el agente almacenará en la memoria de corto plazo, donde los elementos almacenados pierden gradualmente el "derecho" de permanecer en la memoria y son eliminados para dar espacio a nuevos elementos que se encuentren en un estado actual de procesamiento.

El Integrador: complementa la acción del anterior. En lugar de detectar los cambios en una señal, el integrador mide la cantidad total de señales que llegan en un período de tiempo. En ocasiones la integración es usada para convertir una señal de frecuencia modulada en una señal de amplitud modulada. Los impulsos nerviosos son enviados desde y hacia el cerebro como flujos de pequeñas "descargas eléctricas", cuanto más cerca se produzcan estas descargas eléctricas, más imperativa será la señal.

En el diseño de METÁFORA se emplearán los integradores en diversos componentes de la arquitectura. Por ejemplo, las redes neuronales recibirán señales (del entorno o del

estado interno del agente), y cuando estas señales acumulen la suficiente intensidad sobrepasando el umbral de activación de una neurona (determinado por un integrador), ésta se disparará y enviará un "impulso eléctrico" a las neuronas con quien esté conectada. Algo muy similar ocurre con las dinámicas de activación en las Redes de Comportamientos, o con las Redes Inmunes Artificiales, como se verá en capítulos posteriores. También se aplicarán los integradores, en cada componente híbrido especializado (*Comportamiento*) de la arquitectura, de manera tal que las distintas acciones recomendadas por cada componente del *Comportamiento*, se integrarán en una única salida que determinará una acción a ejecutar por el agente.

El Oscilador: todas las redes auto-organizadas que conforman la arquitectura contendrán bucles de retroalimentación, donde la salida de uno de los componentes cambiará finalmente su propia entrada, la cual a su vez cambiará su salida, y de allí nuevamente la entrada... Bajo ciertas circunstancias este proceso de causa y efecto nunca se detendrá, y el resultado es lo que se denominará un oscilador. En los circuitos electrónicos, los osciladores vienen en una gran variedad de tipos, pero indiscutiblemente involucran una retroalimentación positiva y un retardo (delay). Los osciladores en la naturaleza, definen los ritmos biológicos, como es el caso del sueño. En ausencia de la noche o del día, el cuerpo tiende a perder el ritmo circadiano[8], que luego se sincroniza con la rotación de la tierra.

Los osciladores serán de gran utilidad en el diseño de METÁFORA cuando se definan los ritmos temporales en los que deben activarse ciertos procesos internos en el agente. Por ejemplo, de acuerdo con el proceso darwiniano que se esté ejecutando en cierta escala de tiempo, se activarán ciertos procesos de clonación y recombinación (en anticuerpos, memes, etc.), los cuales volverán a ser ejecutados después de que el agente haya cumplido nuevamente un ciclo de interacción con el entorno. Por otro lado, los osciladores serán también útiles a la hora de simular motivadores y emociones básicas en el agente. Por ejemplo, si se desea simular el estado ⟨`sediento`⟩ en el agente, éste no será un estado que se mantendrá por siempre, sino que, por el contrario, aumentará en la medida que el agente realice desplazamientos y disminuirá en la medida que el agente satisfaga su deseo de beber agua, sin embargo, tampoco resulta conveniente que el agente beba agua indefinidamente, sino que el agente llegará luego a un estado de ⟨`saciado`⟩, el cual determinará que la cantidad de agua bebida ha sido suficiente, y de nuevo se inicia el ciclo: el agente sigue desplazándose y con el tiempo disminuirá su cantidad de agua almacenada, activándose nuevamente el estado de ⟨`sediento`⟩ ....

El Trinquete: es un dispositivo que dirige la adaptación del sistema hacia adelante, previniendo que los cambios se deshagan una vez hayan sido incorporados. En términos evolutivos, el trinquete de la selección natural opera en una población que varía aleatoriamente, manteniendo todos los cambios "buenos" (debido a que las criaturas más adaptadas tienen mayor probabilidad de pasar sus genes a la siguiente generación) y descartando aquellos que son "malos". En electrónica, la implementación de un trinquete es llamado "diodo": un dispositivo que permite que la corriente fluya a través de éste en una sola dirección y no en otra.

[8]Perteneciente o relativo a un período de aproximadamente 24 horas. Se aplica especialmente a ciertos fenómenos biológicos que ocurren rítmicamente alrededor de la misma hora, como la sucesión de vigilia y sueño.

La evolución probablemente tenga cierta tensión, debido a que mientras que las especies se adaptan cada vez más a su nicho, el número de mejoras futuras disminuye. Sin embargo, debido a que los entornos están en constante cambio y debido a que las carreras armamentistas evolutivas continúan aumentando la competencia, esta tensión es con frecuencia aliviada y de esta manera la evolución nunca se detiene por completo. La primera ley de la biología –la adaptación– frecuentemente yace en trinquetes que no producen tensión: parte del sistema se mueve hacia adelante tan rápido como puede, y luego se estanca en ese lugar mientras el resto se adapta lentamente hacia él (Grand, 2000).

En METÁFORA, se emplearán diversos trinquetes evolutivos, desde los niveles adaptativos de orden inferior como en las RNAs adaptativas y el mecanismo de adaptación de los SIAs, hasta los mecanismos de evolución en las RCs y Planes, y la co-evolución de memes. Como se verá más adelante, el trinquete evolutivo permitirá que la adaptación avance hasta cierto punto y luego se estanque en un punto estable del sistema (e.g., en el caso de los SIAs, produciendo una memoria inmune estable a través de la selección elitista de anticuerpos mejor adaptados), donde nuevamente el proceso evolutivo explotará en busca de diversidad y adaptación. A continuación se mencionan cuatro bloques de construcción que en cooperación con el trinquete, dirigen el proceso evolutivo darwiniano a diferentes escalas de la arquitectura: el replicador, el recombinador, el mutador, y el selector.

El Replicador: se encargará de crear "copias" o clones de una pauta específica. Dicha pauta puede ser una característica estructural de METÁFORA codificada como una cadena de genes, o puede ser también una pauta cultural, es decir, un meme. Esta información de los clones es copiada con cierta semifidelidad. En términos electrónicos, se podría hacer una analogía con un repetidor, el cual toma una señal débil o de bajo nivel y la retransmite a una potencia o a un nivel más alto, pudiendo así cubrir distancias más largas sin degradación o con una degradación tolerable de la señal. El replicador, "retransmitirá" las pautas espaciotemporales a fin de hacerlas más persistentes en el tiempo, y como consecuencia, las hará más "potentes" al producir una mayor cantidad de clones de una pauta específica, la cual competirá con otras pautas por lograr cierta resonancia al momento de conjeturar, realizar procesos de razonamiento, planificar, etc.

El Mutador: tiene como finalidad incluir una señal de ruido en la señal original. En términos biológicos, el operador de mutación inserta alteraciones en el código genético produciendo así errores de copia (como los producidos por los rayos cósmicos). La mutación de las copias de pautas espaciotemporales permitirá insertar diversidad en la exploración de nuevas soluciones, lo cual evitará que el sistema caiga indefinidamente en estados monótonos.

El Recombinador: se encargará de barajar las cartas, por decirlo de alguna forma. Tomará la información de dos pautas (que cumplirán el rol de pautas padres) a partir de las cuales se crearán dos nuevas pautas (pautas hijas) que contendrán información combinada de las dos pautas padres. Este componente también agrega diversidad al proceso darwiniano que se lleva a cabo en los diferentes niveles de organización de METÁFORA.

El Selector: se encarga de realizar un filtrado de la información que se transmite entre los módulos de la arquitectura. Debido a que los componentes de METÁFORA actúan como redes cuyo procesamiento se efectúa en paralelo, se pueden tener múltiples

elementos *compitiendo* por enviar su señal tanto al exterior como a otros componentes internos de la arquitectura. De esta manera, el selector se encargará de escoger aquellos elementos más aptos para que envíen su señal. En términos evolutivos, este componente se ocupará de dirigir el proceso de selección natural, que en el caso de METÁFORA, sería la selección "artificial" que se hace en los distintos procesos darwinianos llevados a cabo por las Máquinas Darwinianas descritas en 4.4.3.2.

Los bloques de construcción que se han descrito hasta el momento, pueden combinarse entre sí para crear bloques de un orden superior de complejidad (e.g., algunos bloques de construcción de orden superior son las Máquinas Autopoiéticas y las Máquinas Darwinianas). Cada uno de estos bloques pueden aplicarse en diferentes niveles del diseño de METÁFORA, por tal motivo, pueden considerarse como bloques reutilizables que resultan ser un buen punto de partida para el diseño de sistemas cognitivos como redes complejas de retroalimentación.

Con la definición de estos bloques, no se pretende dar una visión puramente materialista o mecanicista del proceso de cognición, sino una visión sistémica de la arquitectura, ya que estos bloques representan mejor las interacciones y relaciones entre las "partes", que simplemente a las entidades físicas, con lo cual se busca re-afirmar la idea gestáltica de que el todo es más que la simple suma de sus partes.

En METÁFORA, las neuronas, los anticuerpos, y a una mayor escala, los *Comportamientos*, son el epítome de los moduladores, y pueden estar constituidos por integradores, diferenciadores y osciladores. Por otro lado, los genes son una excelente metáfora para describir como éstas estructuras pueden ser combinadas dentro de circuitos más complejos, mientras al mismo tiempo permiten que los trinquetes evolutivos cambien el diseño de las estructuras cognitivas.

A pesar de que estos bloques de aproximación cibernética permiten la construcción de entidades de inspiración biológica, tales como neuronas, anticuerpos, genes, etc., no se pretende emplearlos del mismo modo en que aparecen en un contexto biológico. Así pues, lo que realmente se pretende replicar no es el diseño físico del cerebro sino su diseño "informativo".

Finalmente, estos bloques de construcción básicos pueden ser implementados en un ordenador, como componentes de primer orden –algoritmos que se comportan de forma similar a neuronas, anticuerpos, genes, reacciones químicas, etc. Lo que viene ahora, es intentar conectarlos en redes retroalimentadas a fin de conservar la *persistencia* de los elementos del sistema, y de esta manera lograr que dichos elementos inanimados se conviertan en un todo con capacidad de cognición. En la figura 4.5 se esquematizan los bloques de construcción descritos anteriormente, los cuales serán empleados más adelante en el diseño funcional de METÁFORA. En la sección 5.10 se enuncian los requisitos relativos a los bloques de construcción de la arquitectura.



Figura 4.5: Bloques de Construcción de primer orden

4.6. Una Arquitectura Dirigida por Metáforas

En esta sección se sintetizan las diferentes metáforas (mencionadas a lo largo de este capítulo) de inspiración biológica, psico-cognitiva, neurofisiológica, macanicista, y sistémica, sobre las cuales se cimentará el diseño arquitectónico de METÁFORA.

En este punto, es muy probable que el lector ya haya intuido el por qué del nombre de la arquitectura: METÁFORA, y muy seguramente estará en lo correcto. El nombre de METÁFORA, ha surgido de la propia naturaleza de la arquitectura, inspirada, como se dijo anteriormente, en diversos modelos metafóricos que, aplicados al diseño de la arquitectura, proporcionan una nueva perspectiva de lo que podría ser un modelo cognitivo híbrido con capacidades emergentes como la adaptación, el aprendizaje, la inteligencia, la consciencia, y en una dimensión más sofisticada, la creatividad. Sin embargo, el nombre de METÁFORA va aún más allá: además de simbolizar el conjunto de metáforas en que se inspira la arquitectura, representa una visión del trabajo futuro donde la aproximación arquitectónica propuesta estará en capacidad no sólo de aprender, categorizar, y relacionar conceptos, sino de trasladar dichos conceptos desde un contexto a otro sin que

pierdan su validez, es decir, que el sistema cognitivo estará en capacidad de crear sus propias "metáforas" acerca del mundo percibido.

A continuación, se describen brevemente las metáforas en las cuales se inspira la arquitectura:

- *Las Instalaciones Básicas*: enunciadas en el apartado 4.3, describen el conjunto de componentes híbridos *multi-propósito* en METÁFORA: los *Comportamientos*. Estos componentes carecen de finalidad inicial, y es a través de la interacción con el entorno, el aprendizaje (epigenesis), el desarrollo (ontogenesis), y los procesos de retroalimentación, que definen su propósito. Así pues, estos componentes se comportarían como las instalaciones básicas que diseñan los arquitectos para luego definirles una funcionalidad específica, o como las zonas corticales multi-uso, las cuales pueden ser redefinidas durante la existencia del organismo, brindando plasticidad al cerebro.
- *El Coro Aleluya*: esta metáfora mencionada en el apartado 4.4.3.2, hace referencia a la *resonancia* y *sincronización* neuronal que existe entre las pautas espaciotemporales y su analogía con un coro de personas que cantan en disonancia y se sincronizan finalmente. Para que una pauta espaciotemporal "persista" debe crear cierta resonancia con sus pautas vecinas (tal como sucede con ciertos grupos de neuronas piramidales en el neocortex), sincronizándose en el "canto" con las demás, gracias a la retroalimentación recibida. Las pautas espaciotemporales en METÁFORA, tendrán que crear coaliciones con otras pautas con el fin de lograr la suficiente resonancia y así obtener el control temporal en la gestación de un acto inteligente.
- *Las Máquinas Autopoiéticas*: descritas en el apartado 4.4.2.2, son la metáfora biológica por excelencia para representar los procesos de auto-regulación, auto-reparación, auto-organización, homeostasis, y retroalimentación, presentes en la generación de estructuras autónomas. Las Máquinas Autopoiéticas serán el motor de los procesos emergentes en METÁFORA, creando unidades básicas de conocimiento sub-simbólico en los niveles inferiores de la arquitectura, las cuales serán luego transformadas, por los niveles superiores, en unidades de conocimiento explicativo de tipo simbólico.
- *El Paisaje de la Retroalimentación*: esta metáfora enunciada en el apartado 4.4.3.1, esquematiza los bucles de retroalimentación que afectan un sistema complejo adaptativo, en este caso de METÁFORA. Esta metáfora sitúa al agente en un paisaje compuesto por valles y colinas, donde debe predecir un conjunto de factores externos (incluyendo la actuación de otros agentes) con el objetivo de mantenerse en movimiento constantemente, evitando caer en precipicios muy profundos, y acumulando la suficiente energía para escalar cuesta arriba las montañas. El comportamiento adaptativo del agente surge de esta habilidad para mantenerse en el filo de la montaña sin caer, y las características cognitivas de orden superior tales como el aprendizaje, la predicción de la conducta de otros, y la inteligencia, derivan como consecuencia del anidamiento de bucles de retroalimentación más complejos.
- *Las Máquinas Darwinianas*: mencionadas en el apartado 4.4.3.2, materializan los procesos darwinianos que se aplicarán a diferentes niveles jerárquicos y escalas temporales de METÁFORA. Esta metáfora, recopila los procesos básicos de clonación, recombinación, y selección, observados en el proceso evolutivo natural, y los

aplica en los diversos niveles organizativos de la arquitectura, donde son evolucionadas las diversas estructuras cognitivas y culturales. Dichas estructuras cognitivas, en todos los niveles de organización de la arquitectura, permanecerán en continua competencia y cooperación (como parte del proceso darwiniano), a fin de mantener su propia persistencia, y a gran escala, la del agente.

- *Simulación de la Epigénesis, Ontogénesis y Filogénesis*: esta metáfora, de inspiración biológica, es trasladada al contexto del proceso cognitivo. Así pues, el propósito de estos tres niveles de simulación biológica, es llevar a METÁFORA desde un nivel de desarrollo interno de sus estructuras mentales más básicas, hasta la evolución de las estructuras más complejas desde una perspectiva poblacional (i.e., empleando un sistema multi-agente). Esta metáfora determina las pautas sobre cómo y cuándo se deben aplicar las Máquinas Darwinianas en los diferentes niveles jerárquicos de la arquitectura, a fin de obtener un sistema que esté en capacidad de auto-organizar sus propias estructuras y procesos cognitivos internos.
- *Los Factores Aceleradores*: en el apartado 4.4.3.3, se mencionaron tres factores aceleradores del proceso evolutivo: el cambio climático, el sexo y las islas. Estos tres factores, cada uno representando una metáfora por separado, son el motor evolutivo que impulsa los procesos de epigénesis, ontogénesis, y filogénesis. El sexo por su lado, representa la forma en que las pautas espaciotemporales son clonadas y recombinadas en los diferentes niveles de METÁFORA con el objetivo de obtener pautas nuevas y mejor adaptadas. Por su parte, el cambio climático consiste en someter dichas pautas a cambios ambientales, aplicarlas en diferentes entornos, trasladarlas a distintos niveles de representación y entendimiento, etc., con el fin de hacerlas más generalizables y, por ende, más adaptables a las nuevas situaciones. Por último, la metáfora de las islas, es lo que se podría llamar un aislamiento mental, donde las pautas son separadas por períodos cortos de tiempo para poder así acelerar el proceso de adaptación en paralelo. Así pues, las islas son particularmente útiles en la especialización del conocimiento: así como existe una especialización evidente de los órganos funcionales entre individuos que habitan diferentes islas aún siendo de la misma especie, es posible conseguir que las pautas o estructuras cognitivas se especialicen y adapten con mayor rapidez.
- *Los Bloques de Construcción*: mencionados en el apartado 4.5, son una metáfora que intenta describir la arquitectura en términos de componentes reutilizables, multi-propósito, y enfocados a representar las relaciones de causa y efecto entre los elementos de METÁFORA sin describir su composición estructural. Esta metáfora de los bloques cibernéticos implica también el hecho de poder modelar la arquitectura como un conjunto de componentes y mecanismos con un bajo acoplamiento entre los mismos: los componentes de la arquitectura no se ensamblan entre sí como piezas de un puzle, sino que, por el contrario, se ensamblan como estructuras articulables y reutilizables como si fuesen piezas de Lego.

Para terminar el conjunto de metáforas aplicables a la arquitectura, se considerará una metáfora más: "la metáfora del *Teatro de la Consciencia*". Esta es una metáfora, de las muchas que existen en el campo de la psicología cognitiva, que explica la orquestación de los procesos conscientes en el cerebro humano. A continuación se describen brevemente algunas de las metáforas más aceptadas en el campo de la psicología cognitiva y

se hará hincapié en la del Teatro de la Consciencia, la cual será empleada en el diseño de METÁFORA.

4.6.1. Metáforas sobre la Consciencia

Debido a la complejidad inherente al estudio de la *"consciencia"*, en el campo de la psicología cognitiva, ésta suele argumentarse desde el punto de vista metafórico. Algunas de estas metáforas se enuncian a continuación.

La *metáfora de la consciencia como pegamento* (Treisman y Schmidt, 1982), propone que la consciencia aglutina y da una forma coherente a los miles de datos que llegan al cerebro en cada milisegundo, convirtiendo un torbellino de impulsos neuronales en una percepción comprensible del mundo y de la propia mente.

Por su parte, *la metáfora de los recursos* (Dennett, 1991), una metáfora de carácter económico, propone un modelo que compara la atención como un recurso que debe utilizarse en diversas actividades, de forma que su uso en alguna de ellas limita la cantidad disponible para utilizarlo en otras.

Baddeley (Baddeley, 1991) describe los contenidos de la memoria de corto plazo (MCP) como lo que "tenemos en mente", o sea, aquello de lo que somos conscientes. En su *metáfora del ejecutivo central*, Baddeley enuncia tres sistemas que componen la MCP. Uno de ellos llamado maestro, que actúa como un ejecutivo central, asigna los recursos de procesamiento a los otros dos que dependen de él: la agenda visoespacial (almacenamiento visual) y el lazo articulatorio (información lingüística).

Otro ejemplo de metáforas sobre la consciencia, es la *metáfora del foco de linterna* (Posner, 1980) la cual propone que la consciencia operaría como una persona en un mundo a oscuras donde sólo tiene una linterna. Los objetos iluminados por la linterna son los únicos que esa persona puede percibir (o ser consciente de), mientras que el resto del mundo sería invisible (subconsciente) para él. En esta metáfora, el foco de la linterna representa el mecanismo de la atención que se encarga de seleccionar qué ideas entran a la consciencia.

De la metáfora del foco de la linterna, se deriva la *metáfora del teatro de la consciencia* de Baars (Baars, 1997), que entiende la consciencia como un escenario donde los actores representan los procesos conscientes y el personal tras bambalinas, los procesos subconscientes. A continuación, se enuncian algunos de los principios básicos de esta metáfora, los cuales servirán de apoyo al diseño del mecanismo de *orquestación de procesos conscientes y subconscientes* en METÁFORA, como se verá en capítulos posteriores.

Como suele ocurrir en las teorías científicas, Baars (Baars, 1997) parte de una serie de presupuestos básicos en los que fundamenta su modelo. Estos son:

- La consciencia es vasta: las personas son conscientes y parecen actuar con control. Conversan, toman decisiones, aprenden textos, meditan, tienen en consideración si su conducta es apropiada o deseable socialmente, muestran emociones, empatía, mienten, experimentan placer mental, etc., sin embargo, al dormir dejan de tener

conducta y control. En resumen, la conciencia es un fenómeno evidente que no se puede soslayar.

- La consciencia es una variable: Baars, propone un método al que llama fenomenología de contraste, que permite estudiar las diferencias entre los distintos estados de la conciencia. A través de ésta, demuestra cómo la conciencia sobre el desarrollo de una tarea disminuye con la experiencia hasta convertirse en un proceso subconsciente (automatizado).
- Un continuo entre lo consciente y lo subconsciente: Baars propone que la experiencia coherente tiene lugar en una especie de teatro mental, donde se orquesta o se marca el ritmo de los múltiples módulos encargados de las distintas funciones cerebrales, como la percepción, la memoria, la atención, etc.

En la metáfora de Baars se pueden distinguir los siguientes componentes:

- El escenario (la Memoria de Trabajo): en el modelo de Baars, los contenidos que están sobre el escenario son conscientes o casi. Esto implica que, para ser consciente, un contenido determinado ha de estar en el escenario, pero no todos los contenidos en el escenario pueden ser conscientes a la vez. Este escenario corresponde a la Memoria de Trabajo de capacidad limitada.
- Los Actores (los contenidos de la consciencia): los actores (información sensorial externa, información sensorial "interna", imágenes visuales, habla interna, sentimientos...), *Compiten* y *Cooperan* por ser el protagonista de la escena en un momento dado, es decir, por el ser el contenido dominante en la conciencia.
- El foco (la atención): el foco luminoso que barre el escenario y resalta a uno u otro actor según quiera darle protagonismo, es un componente fundamental de esta metáfora, ya que el foco "elige" quién entra realmente en la conciencia. Sin embargo, el foco tiene una penumbra a su alrededor, perfilando levemente a otros actores, aunque sin iluminarlos de lleno, los cuales pueden convertirse en actores dominantes por efecto de repercusión. La atención, por tanto, es la encargada de controlar el foco que ilumina el escenario, en otras palabras, de seleccionar y dirigir la información hacia la conciencia.
- Los operadores contextuales tras bambalinas: son los responsables de montar el escenario en el que se mueven los actores, de dirigir el foco de la atención y de realizar tareas subconscientes igualmente importantes. Entre estos operadores es especialmente importante el que Baars denomina "director", que se identifica con ciertas áreas de los lóbulos frontales responsables del control ejecutivo. Según Baars, el director determina quien ha de ser iluminado mediante una estructura llamada el "árbol de metas", con frecuencia subconsciente, que incluye intenciones y expectativas inmediatas y a largo plazo, al que se ajusta todo lo que ocurre en el escenario.
- El público del teatro (La sociedad de la mente): el público de este teatro hace referencia a todos los módulos que componen el cerebro: los sistemas de identificación y procesamiento. La audiencia, se trata de la sociedad de la mente formada por módulos, locales, expertos en dominios específicos, que son movilizados gracias a la conciencia para cooperar en tareas nuevas, por metas inmediatas y a largo plazo.

> Los sistemas de memoria autobiográficos, declarativos, semánticos... los sistemas de aprendizaje, las habilidades motoras y los procesos automáticos, se disparan cuando sus condiciones se cumplen en la memoria de trabajo.

Adicionalmente, Baars propone un modelo que se adapta a su metáfora del teatro, llamado el Modelo del Área de Trabajo Global (ATG), el cual postula que la cognición humana se compone de una multitud de pequeños procesos de propósito específico, casi siempre actuando de manera subconsciente. Estos procesos especializados compiten y cooperan por el acceso al área de trabajo global, es decir, por el "estado de conciencia", tal y como se describió en los principios de la metáfora del teatro. Los ganadores de la competición adquieren el derecho al control del acceso global y pueden difundir información a los demás procesos especializados que compiten (procesos subconscientes), con el fin de reclutar otros procesos que participen en la solución de la situación actual pero que serán activados en el siguiente ciclo cognitivo.

En METÁFORA, se empleará la metáfora del Teatro de la Consciencia, aplicando todos sus elementos descriptivos: el escenario (Memoria de Trabajo), el foco de luz (la Atención), los operadores tras bambalinas (el director y los operadores de contexto), la audiencia (procesos subconscientes), y los actores (que para el caso de METÁFORA son representados por el conjunto de *Comportamientos* del agente).

4.7. Evolución de la Arquitectura Propuesta

> *"Hay que evitar el uso de conceptos matemáticos para deslumbrar, en lugar de para iluminar; hay que evitar las "demostraciones por falta de imaginación", como cuando llegamos a la conclusión, por arrogancia o por impaciencia, de que no existen alternativas a las respuestas que hemos encontrado. Y en particular, hay que tener mucho cuidado de situar nuestras teorías al nivel adecuado de explicación mecanicista".*
>
> — William Calvin, Cómo piensan los cerebros, 2001.

Al igual que lo mencionado por Calvin en su cita, el desarrollo del presente modelo cognitivo ha sido el resultado de una búsqueda incesante de alternativas de solución a una cuestión que surgió en mi hace no menos de diez años: ¿cómo poder modelar los procesos cognitivos humanos a fin de implementarlos en un modelo computacional? Así pues, en esta sección se presenta la línea "evolutiva" que ha tenido el desarrollo de la presente tesis doctoral. En especial, se puede observar cómo fue concebida inicialmente la idea y como ha venido adaptándose, sufriendo progresivas "mutaciones" y "recombinaciones" hasta el resultado actual... el cual ha abierto muchas posibilidades de continuar "evolucionando" el modelo en líneas de trabajo futuras.

- (2000–2002): se propone un modelo cognitivo para un Sistema Tutor Inteligente (STI) que sirve como apoyo a la enseñanza de las matemáticas básicas. El propósito inicial de este modelo cognitivo consistió en "adaptar" y "personalizar" un conjunto de estrategias pedagógicas de acuerdo con las necesidades particulares y el estilo cognitivo de cada estudiante, lo cual era logrado a través de un Sistema Clasificador

(SC) de inspiración biológica y de un modelo tridimensional de los estilos cognitivos. Este STI incluía además un Módulo Experto coordinado por un componente de razonamiento simbólico: un Sistema Experto con un mecanismo de inferencia hacia adelante (Romero et al., 2002; Romero, 2003).

- (2001–2002): paralelamente al diseño del STI, se realiza un proyecto de investigación en el área de "vida artificial". Este proyecto buscaba la simulación de comportamientos adaptativos en un entorno animat[9] donde fueron ajustados los parámetros de aprendizaje del Sistema Clasificador de Aprendizaje (Romero, 2002; Romero y Ramirez, 2002).
- (2004–2005): en colaboración con la Universidad Nacional de Colombia, se propone un modelo cognitivo multicapa para la navegación autónoma en robots móviles. Cada capa de procesamiento es modelada mediante una técnica computacional bio-inspirada: Redes Neuronales, Sistemas Inmunes Artificiales, y Sistemas Clasificadores. Es este trabajo se propone un algoritmo basado en principios inmunológicos, el cual resulta de una combinación entre la teoría de las redes inmunes y las técnicas de aprendizaje por refuerzo. La aproximación fue probada en diversos entornos robóticos simulados, donde se requerían capacidades de aprendizaje para la solución de problemas de navegación autónoma (Romero y Niño, 2006).
- (2005–2007): en este punto, habiendo iniciado el programa de doctorado en la UPM, surge la idea de una arquitectura cognitiva para agentes inteligentes. La propuesta inicial parte de tres principios: el primero, sobre la integración de técnicas bio-inspiradas (las RNAs, los SCs, los SIAs, los AGs, la PEG, entre otros); el segundo, sobre la concepción de una arquitectura de subsumción multicapa; y la tercera, la idea de un mecanismo auto-organizativo que especificará de forma dinámica, las relaciones de subsumción entre las capas (inhibición, supresión y agregación). El modelo es probado en un entorno animat que simula comportamientos del tipo "presa–depredador". La arquitectura, aunque posee un procesamiento en paralelo de las capas horizontales, no posee una especialización vertical de las mismas (Romero y De Antonio, 2007a; Romero y De Antonio, 2007b; Romero y De Antonio, 2008a).
- (2007–2008): se plantea una extensión al modelo cognitivo descrito en el punto anterior, la cual se fundamenta en tres pilares: primero, se propone una arquitectura híbrida que incluye representaciones tanto simbólicas (enfoque cognitivista) como sub-simbólicas (enfoque emergente); segundo, la arquitectura tiene una distribución tanto horizontal (por capas de procesamiento: las capas inferiores son de enfoque emergente y las superiores de enfoque cognitivista), como vertical (una división por *Comportamientos* especializados); y tercero, se incluye un mecanismo de orquestación de los *Comportamientos* empleando una aproximación evolutiva (Programación de Expresiones Genéticas –PEG) el cual modula, de forma adaptativa, la interacción entre los *Comportamientos*. Esta arquitectura es nuevamente probada en un entorno animat y los resultados observados evidencian la necesidad de definir niveles concretos de adaptación y auto-organización en una dimensión distinta a la meramente estructural (Romero y De Antonio, 2008b; Romero y De Antonio, 2009a).

[9] Animat es una contracción en inglés de la palabra *animal* y la palabra *robot* y se refiere a la simulación de animales artificiales en entornos virtuales.

- (2009–2010): en esta última fase del desarrollo de la arquitectura, se plantea un modelo cognitivo influido por tres enfoques de inspiración biológica: la epigénesis (la cual controla los procesos de aprendizaje), la ontogénesis (que dirige el desarrollo y adaptación de las estructuras internas de la arquitectura), y la filogénesis (que define el proceso evolutivo y la transmisión cultural de memes). Se modifica el mecanismo de orquestación de *Comportamientos*, incorporando la metáfora del Teatro de la Consciencia que es implementada a través de las Redes de Comportamientos (RC). Se propone una extensión al modelo de RC, integrando un mecanismo de programación evolutiva (PEG) que se encarga de evolucionar y adaptar la estructura y funcionalidad de las RCs. Se incluye un mecanismo de anticipación y simulación de planes, el cual es coordinado por un Sistema Clasificador Anticipatorio. Esta arquitectura es probada en entornos robóticos (simulados y reales) durante una estancia investigativa que realicé en el Imperial College London (Romero y De Antonio, 2009b; Romero, 2010).

4.8. Metodología de Diseño

En esta sección se describe brevemente la metodología empleada durante el desarrollo de la tesis doctoral, para lo cual se siguieron dos metodologías: una de tipo científico-investigativa y la otra de tipo ingenieril.

En cuanto a la metodología científico-investigativa, se empleó un método deductivo-experimental mediante el cual fue posible, a partir de un conjunto de premisas e hipótesis (descritas en el capítulo 5), llevar a cabo dos actividades: la formulación de una disertación teórica acerca de los criterios necesarios para el diseño e implementación de una arquitectura cognitiva para agentes que exhiben comportamiento inteligente y adaptativo, y la validación de la arquitectura cognitiva, sus procesos y mecanismos subyacentes, en contextos experimentales adecuadamente definidos. En cuanto a la metodología ingenieril, se empleó el modelo de Proceso Unificado (Jacobson et al., 2000) el cual se integra al proceso investigativo en las etapas de desarrollo de la solución computacional.

El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases se divide en un conjunto de iteraciones. Estas iteraciones ofrecen como resultado un incremento del producto que añade o mejora las funcionalidades del sistema en desarrollo.

Cada una de estas iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: modelado del negocio (o problema), requisitos, análisis y diseño, implementación, pruebas, despliegue, gestión del cambio y configuraciones, gestión del proyecto, y entorno. Aunque todas las iteraciones suponen la intervención de casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto. En la figura 4.6 se observa la distribución en fases, iteraciones y disciplinas.

Por último, la metodología global que se siguió durante el desarrollo de la tesis doctoral es estructurada en cuatro macro-fases de la siguiente forma:

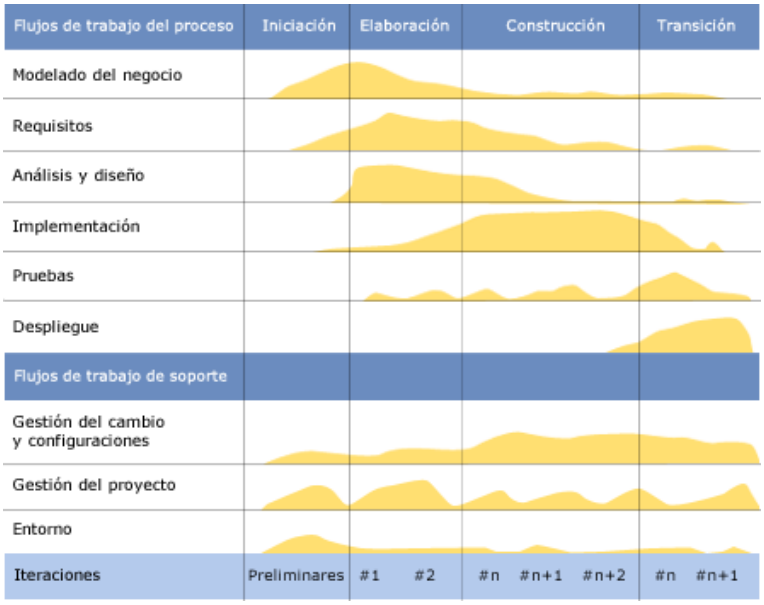


Figura 4.6: Proceso Unificado: esfuerzo de cada disciplina según la fase del proyecto (Jacobson et al., 2000).

1. Definición del Estado del Arte sobre Arquitecturas Cognitivas para agentes inteligentes, y elaboración de un Análisis Comparativo entre las diferentes aproximaciones de los enfoques Cognitivista, Emergente, e Híbrido.
2. Definición del conjunto de hipótesis y propuesta de una Arquitectura Cognitiva como una solución computacional de inspiración biológica.
3. Desarrollo de la solución computacional: en esta fase se siguen las siguientes disciplinas del proceso unificado de desarrollo de software: Análisis de Requisitos, Análisis y Diseño OO, Implementación, y Pruebas de los diferentes módulos de la Arquitectura Cognitiva. Adicionalmente, se lleva a cabo la disciplina de Gestión del proyecto.
4. Validación Experimental del funcionamiento de la arquitectura y sus componentes.

Capítulo 5

Hipótesis y requisitos del Sistema

"Los científicos que formulan hipótesis deberían construir sobre ellas grandes edificios que las alberguen".
– Jean Piaget.

En esta sección se enuncian los requisitos técnicos y funcionales de la arquitectura, así como las hipótesis en las que se basó la investigación. Estos requisitos e hipótesis se encuentran agrupados según las diferentes perspectivas que se tuvieron en cuenta en el capítulo 4 sobre la naturaleza de la solución.

5.1. Hipótesis y requisitos relativos al Modelo Cognitivo Integrado

A continuación se presentan los requisitos relativos a la sección 4.2:

- El intento de simular, en agentes software, los procesos mentales superiores relacionados con la inteligencia natural y la cognición, no responde a esfuerzos aislados e individuales por estudiar las distintas facultades cognitivas asociadas, sino que contrariamente, se presume que la definición de un modelo integrado permitiría simular el proceso cognitivo como un todo y no como la simple agregación de facultades individuales. Lo que es más, muchas de estas facultades cognitivas no pueden ser posibles sin la existencia simultánea de las otras.
- De lo anterior se deriva la siguiente hipótesis: los requisitos mínimos para simular adecuadamente un proceso cognitivo artificial incluirían la integración de al menos las siguientes facultades: Percepción, Atención, Aprendizaje, Representación del Conocimiento, Anticipación, Toma de Decisiones, Memoria, Planificación, Actuación, Adaptación, Mecanismos de Explicación de los esquemas de razonamiento y Motivadores (emociones primarias).

- La construcción de un modelo cognitivo debería incluir al menos dos sistemas subyacentes los cuales se complementan mutuamente: uno de enfoque cognitivista encargado de realizar representaciones simbólicas que permiten el acceso directo o explícito a unidades de conocimiento mediante etiquetas semánticas, y un enfoque emergente que construya representaciones sub-simbólicas del conocimiento y cuyo acceso sea implícito.
- La hibridación de los sistemas cognitivista y emergente de la arquitectura debería definir al menos dos mecanismos de integración: uno de enfoque ascendente (bottom-up) que construya representaciones simbólicas del conocimiento a partir de la extracción sub-simbólica de las mismas, y otro de enfoque descendente (top-down) que emplee las representaciones explícitas de conocimiento del nivel simbólico (cognitivista) para orientar luego el proceso de consolidación y aprendizaje de nuevas representaciones implícitas en el nivel sub-simbólico (emergente).
- La mente implementa estrategias de orden superior para el razonamiento y la resolución de problemas mediante la ejecución de secuencias de *funciones comunes*. Estas funciones pueden ser simuladas por la arquitectura cognitiva a través de mecanismos computacionales.
- La arquitectura cognitiva se compone de módulos especializados (los *Comportamientos*) que implementan las funciones comunes, empleando para ello diversos mecanismos computacionales y secuencias algorítmicas.
- Existe evidencia (Baars, 1997; Stroop, 1935), de que la mente, por cada área especializada del cerebro, emplea simultáneamente múltiples *funciones comunes* a fin de llevar a cabo una tarea específica. Por tanto, se asume que la arquitectura debe incluir un mecanismo de integración que permita por un lado, la combinación de las acciones recomendadas por todos los componentes que conforman un *Comportamiento* y, por otro, la unificación total de las acciones disparadas por los múltiples *Comportamientos* que puedan estar activos en un instante determinado.
- La arquitectura incluye componentes de enfoque tanto cognitivista como emergente debido a: (1) que el enfoque emergente dirige los procesos de aprendizaje y representación del conocimiento sub-simbólico, a partir del cual es posible generar representaciones simbólicas que pueden ser comunicadas y explicadas a un humano; y (2) que los componentes de enfoque cognitivista y emergente se compensan mutuamente: mientras el nivel cognitivista guía la interacción inicial del agente con un entorno nuevo a través del conocimiento "innato" almacenado, el nivel emergente se encarga de "aprender" y "descubrir" nuevas reglas, o adaptar las ya existentes, cuando las reglas innatas pierden validez.
- En este punto, cabe aclarar que el objetivo de emplear diferentes micro-teorías cognitivas en la construcción de una arquitectura, se encuentra enmarcado en el propósito de *simular* la funcionalidad de los procesos cognitivos a partir de los cuales emerge el comportamiento inteligente, y no en *emular* fielmente la actividad neuronal que da origen a los mismos. La diferencia que se quiere resaltar en estas líneas, es que mientras el proceso de emulación de los procesos mentales implicaría la imitación precisa del funcionamiento del cerebro humano desde un nivel muy bajo, la simulación de dichos procesos cognitivos sólo implicaría la representación del comportamiento de la mente sin la necesidad de reproducir fielmente todas sus

estructuras subyacentes (átomos, moléculas, células nerviosas, etc.) o su compleja neuro-anatomía.

- El desarrollo de la propuesta arquitectónica posee tres fases: la definición de una *arquitectura conceptual*, la especificación de una *concreción funcional*, y la caracterización de una *concreción contextual*. La *arquitectura conceptual* define un diseño preliminar basado en bloques de construcción cibernéticos, el cual sirve de puente entre los fundamentos teóricos de inspiración biológica y la visión mecanicista-computacional del sistema. En la *concreción funcional* se identifican los componentes, módulos, y mecanismos que deben ser especificados, diseñados y construidos a fin de lograr una implementación posible de la arquitectura propuesta, independientemente del contexto de aplicación de la misma. Por último, en la *concreción contextual* se determinan los requisitos que impone el dominio de aplicación seleccionado, se evalúan los componentes de la arquitectura según dichas restricciones, se identifican sus relaciones, y se diseñan y construyen sus procesos específicos.

5.2. Hipótesis y requisitos relativos a las propiedades de Plasticidad, Modularidad y Especialización

A continuación se presentan los requisitos relativos a la sección 4.3:

- Se proponen las instalaciones básicas, como un conjunto de habilidades o *Comportamientos*, que se encuentran organizados como módulos imbricados de carácter multi-propósito. Así pues, se define un conjunto de módulos o *Comportamientos* que especializan la conducta del agente, pero que a su vez, proporcionan cierta redundancia al proceso cognitivo puesto que es posible diseñar múltiples *Comportamientos* para el mismo propósito, esto con el fin de hacer más robusto el sistema: si un *Comportamiento* especializado no hace lo que se supone debe hacer, habrán otros que si lo harán de la manera correcta y, con el tiempo, buscarán sincronizarse entre sí a través de la retroalimentación.
- La articulación de los distintos módulos especializados o instalaciones básicas, concede a la arquitectura plasticidad y escalabilidad, permitiendo por un lado, que la estructura arquitectónica no sea rígida y que sus componentes sean fácilmente modificables y/o reemplazables, y por el otro, que nuevos módulos sean incorporados de forma independiente e incremental.
- La plasticidad de los módulos especializados estará condicionada a su capacidad de definir autonomamente su orientación a metas específicas, a la capacidad de aprender a generar modelos del mundo exterior y a la capacidad de redefinir su estructura interna.
- Se propone una arquitectura de distribución híbrida: una distribución horizontal por capas de procesamiento tanto simbólico como sub-simbólico, y una distribución vertical por componentes de procesamiento complejo y especializado (los *Comportamientos*). De esta manera, la arquitectura gestionaría distintos tipos de procesamiento y articularía, en componentes multi-propósito (en analogía con las

micro-columnas corticales del cerebro humano), las habilidades y destrezas del agente.

5.3. Hipótesis y requisitos relativos a la propiedad de Persistencia del Comportamiento Emergente

A continuación se presentan los requisitos relativos a la sección 4.4.1:

- En el comportamiento emergente, cada clase nueva de fenómeno persistente abre la puerta a la creación de nuevas clases de complejidad superior. Sin importar si se está tratando con los fenómenos persistentes mas simples o los más complejos, deben aplicarse los mismos conceptos y mecanismos básicos. Por tanto, se argumenta la hipótesis de que los procesos cognitivos más complejos, como la inteligencia y la consciencia, pueden ser simulados como patrones persistentes que emergen de la actividad de una red de bucles anidados regulados por señales de retroalimentación.
- Para crear inteligencia y consciencia artificial es necesario comprender la naturaleza de su jerarquía en niveles, implementar simulaciones de los operadores básicos de cada nivel, y construir sobre estas bases los niveles superiores del fenómeno persistente.

5.4. Hipótesis y requisitos relativos a las características de auto-organización y paralelismo del Comportamiento Emergente

A continuación se presentan los requisitos relativos a la sección 4.4.2:

- Una de las ideas principales que se quiere reflejar en esta tesis, es que el ordenador digital, como una máquina altamente organizada, puede convertirse en una máquina que alberga procesos no organizados y auto-organizados. Se propone emplear el ordenador serial, procedimental, y de enfoque descendente, como una herramienta para crear nuevas "máquinas" que sean paralelas, distribuidas, relacionales, y de enfoque ascendente (emergente), capaces de simular comportamiento inteligente.
- Los procesos cognitivos modelados por la arquitectura deben ser diseñados como procesos que operan en paralelo (que pueden ser simulados o no en ordenadores seriales, es decir, pseudo-paralelismo), los cuales procesarán concurrentemente la información externa percibida por el agente y accederán al conocimiento almacenado de forma independiente.
- La idea de emplear sistemas auto-organizados y distribuidos en el diseño de la arquitectura METÁFORA se fundamente en la premisa de que a partir de una población de agentes independientes y autónomos que interactúan de forma local únicamente, se puede producir un comportamiento "inteligente" global. Acá el término

"agente" no se refiere únicamente al sistema cognitivo como tal, sino a los múltiples componentes básicos que interactúan de forma "autónoma" en los distintos niveles de organización de la arquitectura.

5.5. Hipótesis y requisitos relativos al diseño de Máquinas Autopoiéticas

A continuación se presentan los requisitos relativos a la sección 4.4.2.2:

- La arquitectura estará compuesta por múltiples máquinas autopoiéticas de propósito general. Se propone el uso de *Redes Neuronales Artificiales*, de *Sistemas Inmunes Artificiales*, y de *Redes de Comportamientos*, como máquinas autopoiéticas encargadas de auto-regular sus procesos de producción, transformación y destrucción; auto-organizar sus componentes y relaciones entre los mismos, y mantener la homeostasis interna del sistema.
- Estas máquinas autopoiéticas, actuando en calidad de sistemas micro-cognitivos, se ocuparán de la generación emergente de estructuras cognitivas débilmente acopladas que representarán unidades auto-contenidas de conocimiento.
- En una primera aproximación al diseño de la arquitectura cognitiva, se propone un modelo jerárquico que representa el acoplamiento funcional entre los sistemas autopoiéticos de primer y segundo orden: Redes Neuronales Artificiales y Sistemas Inmunes Artificiales (sistema biológico); y Consciencia y Pensamiento simbólico (sistema cognitivo).

5.6. Hipótesis y requisitos relativos al diseño de la arquitectura como un Sistema Complejo Adaptativo

A continuación se presentan los requisitos relativos a la sección 4.4.2.3:

- La arquitectura cognitiva deberá exhibir las características propias de un CAS, tales como: niveles jerárquicos de complejidad organizada; auto-organización, auto-regulación y auto-producción de los elementos constitutivos; homeostasis de los procesos internos del sistema; y comportamiento emergente.
- La arquitectura modelará los CASs como agentes especializados que actúan en paralelo y de forma distribuida, generando un control descentralizado del sistema. Los agentes (y cada uno de sus elementos constitutivos) cooperarán y competirán entre sí como parte de las dinámicas auto-organizativas del sistema.
- Cada CAS definirá un conjunto de procesos adaptativos que garantizarán la supervivencia del agente ante los cambios de su entorno. Así mismo, el conjunto poblacional de CASs (i.e., el sistema multi-agente) se verá afectado por procesos evolutivos que conducirán a la conciliación a largo plazo de los procesos cognitivos y culturales de la sociedad de agentes.

5.7. Hipótesis y requisitos relativos a los bucles de Retroalimentación de la Arquitectura

A continuación se presentan los requisitos relativos a la sección 4.4.3.1:

- El resultado neto del comportamiento adaptativo de un agente cognitivo, es una retroalimentación negativa a través de la cual mantiene su homeostasis ante la gran variedad de matices ambientales. Sin embargo, la implementación de una ruta de retroalimentación global requiere de la interacción de muchos bucles de retroalimentación intermedios, algunos negativos y otros positivos. El diseño de METÁFORA deberá reflejar entonces dichos bucles intermedios de retroalimentación, los cuales ascienden verticalmente a través de los niveles organizativos de la arquitectura hasta lograr modelar los procesos cognitivos de orden superior.
- Para el diseño de la aproximación arquitectónica, resulta necesario pensar en la arquitectura METÁFORA como una red de bucles anidados de retroalimentación en lugar de una simple colección de componentes y módulos, de esta manera nuevos patrones de comportamiento inteligente emergerán.
- La retroalimentación viene dada en dos variedades (positiva y negativa), pero es a través de la variación de los puntos nulos, de las escalas temporales de los bucles de retroalimentación, y de la modulación de un bucle de retroalimentación a través de uno de orden superior, que la arquitectura cognitiva puede ser dotada con las cualidades de complejidad, robustez y persistencia.
- A diferencia de un control descendente (top-down) donde un controlador maestro está encargado del sistema por completo y donde la necesidad de conocer el número de componentes y relaciones del sistema incrementa exponencialmente, la inteligencia natural emplea un control ascendente (bottom-up) donde las partes no conocen cuál es su rol dentro del todo, pero donde el todo emerge de la cacofonía de cientos de miles de bucles simples de causa y efecto. Así pues, la simulación de la inteligencia y de los procesos cognitivos en METÁFORA debe estar conducida, en mayor medida, por un enfoque de control ascendente, aunque complementada, en menor medida, por un control descendente.
- La arquitectura incluirá diversos mecanismos de aprendizaje basados en bucles jerárquicos de retroalimentación positiva y negativa. Estos bucles de retroalimentación se ocuparán de controlar el comportamiento del agente y de controlar los "cambios" en el comportamiento. El aprendizaje a diferentes escalas (o niveles de organización de la arquitectura) conducirá a un comportamiento adaptativo y robusto global del agente.
- Se propone emplear múltiples mecanismos de aprendizaje automático (machine learning) desde tres enfoques diferentes: supervisado, no supervisado y por refuerzo[1]. La interacción entre estos mecanismos garantiza una disminución en la pendiente de la curva de aprendizaje global del sistema y compensa las debilidades

[1]El aprendizaje supervisado produce una función que establece una correspondencia entre las entradas y las salidas deseadas del sistema. Por otro lado en el aprendizaje no supervisado, todo el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formado tan sólo por entradas al sistema y no se tiene información sobre las categorías de esos ejemplos. Finalmente, en el aprendizaje por refuerzo el sistema

entre cada enfoque, mejorando como resultado el desempeño y la robustez de la arquitectura.

- Los agentes que implementen METÁFORA podrán aprender por sí mismos, sin asistencia, en entornos medianamente complejos y realistas. El agente reconocerá situaciones que ha experimentado anteriormente y aprenderá cómo responder ante éstas. Aprenderá también a generalizar del pasado, situaciones que le ayuden a reaccionar sensiblemente frente a nuevas circunstancias. Así mismo, podrá recordar y olvidar, con el objeto de retener las cosas más significativas. METÁFORA administrará el control básico de la atención, permitiendo que lo eventos importantes obtengan mayor foco de atención por parte del agente, filtrando de esta manera lo que debe ser aprendido y lo que debe ser descartado.
- Se propone el diseño de un mecanismo de predicción basado en la simulación de bucles jerárquicos de causa y efecto, los cuales obtienen una retroalimentación basada en el aprendizaje previo de relaciones causales con el entorno y con el estado interno del sistema, a fin de que el agente pueda construir planes predictivos sobre los cambios que sufrirá el entorno en estados futuros. Este mismo principio podría ser la base para simular procesos cognitivos de más alto nivel como por ejemplo la creatividad, sin embargo, este requisito resulta ambicioso para el alcance de esta tesis y se propone mejor como una línea de investigación futura.

5.8. Hipótesis y requisitos relativos a las Máquinas Darwinianas

A continuación se presentan los requisitos relativos a la sección 4.4.3.2:

- El mismo proceso darwiniano que da forma a una nueva especie a lo largo de milenios –o a un nuevo anticuerpo durante las pocas semanas de la respuesta inmunitaria– puede también dar forma a ideas en la escala temporal del pensamiento divergente y la acción. Como se ha mencionado antes, la simulación de los procesos darwinianos en METÁFORA podría actuar a diferentes escalas del proceso cognitivo: desde el comportamiento reactivo donde la toma de decisiones requiere respuestas en el orden de milisegundos, hasta comportamientos más deliberativos en la escala de minutos, o incluso, la planificación a largo plazo.
- El propósito de las Máquinas Darwinianas que simulan los procesos darwinianos a diferentes escalas del proceso cognitivo, consiste en aprender y crear variaciones de diferentes pautas espaciotemporales (estructuras cognitivas). La competición entre las variaciones determina qué pauta tiene mayor resonancia en relación a otras pautas, y ésta resonancia puede estar a su vez bajo la influencia de una serie de entradas sensoriales e impulsos emotivos.
- Para lograr que METÁFORA pueda conjeturar inteligentemente a partir de un conjunto de pautas espaciotemporales caóticas, creadas al azar por una máquina dar-

aprende observando el mundo que le rodea y su información de entrada es la retroalimentación que obtiene del mundo exterior como respuesta a sus acciones.

winiana, se necesita de: (1) una pauta distintiva; (2) un proceso de copia; (3) variación; (4) posibles competiciones por espacios de trabajo; (5) ambientes multifacéticos (tanto actuales como memorizados) que influyan en la competición; y (6) una siguiente generación en la que es más probable que haya variantes establecidas a partir de los clones con mayores territorios (los territorios grandes tienen perímetros mayores, que es donde las variantes pueden escapar a la tendencia de corregir y empezar a clonar su nueva pauta).

- Se necesita velocidad para que los procesos darwinianos en los diferentes niveles jerárquicos de organización de METÁFORA funcionen con la rapidez suficiente para conjeturar con inteligencia. Para lograr esto, se simularán tres aspectos aceleradores del proceso darwiniano: el sexo (intercambio de material genético entre las pautas cognitivas), el cambio climático (variaciones y perturbaciones en el entorno del agente que permiten la exploración de nuevas alternativas adaptativas), y las islas (donde se evolucionarán aisladamente y en paralelo las estructuras cognitivas de diferentes ordenes jerárquicos).
- Es posible llevar un poco más lejos la importancia de los procesos darwinianos en la construcción de una conjetura inteligente, permitiendo que METÁFORA genere estructuras más complejas como los planes y los escenarios alternativos de ejecución, a partir de la evolución de las pautas espaciotemporales más básicas que pueden variar continuamente hasta encontrar nuevas resonancias. Estas estructuras más complejas (planes y escenarios alternativos), podrían simular internamente acciones que se ejecutarían en el mundo real, ya que las estructuras cognitivas básicas que intervienen en un plan pueden ser validadas cotejándolas con las resonancias de los recuerdos de larga duración y de las entradas sensoriales del estado actual.
- Dos aspectos recurrentes se verán continuamente reflejados en el diseño de METÁFORA: la *Cooperación* y la *Competencia*. Los componentes de la arquitectura –en los diferentes niveles de organización: neuronas, anticuerpos, *Comportamientos*, memes, planes, etc.– competirán por la "dominancia" (e.g., hacer más copias de sí mismos, obtener el foco de atención en los procesos conscientes, etc.) y esta competencia se evidencia principalmente en el proceso de selección estocástica que ejecuta cada Máquina Darwiniana. Por otro lado, estos mismos componentes cooperan entre sí para formar coaliciones (e.g., la resonancia neuronal, la cooperación entre los elementos de las redes de comportamientos y las redes inmunes, etc.), y para intercambiar información a través del proceso de recombinación (donde los memes cooperan entre sí para generar memes mejor adaptados) coordinado por cada Máquina Darwiniana.

5.9. Hipótesis y requisitos relativos a la simulación de la Filogénesis, Ontogénesis y Epigénesis

A continuación se presentan los requisitos relativos a la sección 4.4.3.3:

- Los procesos de desarrollo y evolución de las estructuras cognitivas de la arquitectura se basarán en tres pilares fundamentales de inspiración biológica: (1) el enfo-

que epigenético, el cual busca la adquisición de nuevas habilidades cognitivas mediante procesos de aprendizaje y anticipación que requieren poco o ningún conocimiento previo, permitiendo así la auto-organización de estructuras sub-simbólicas que darán origen, a través de propiedades emergentes, a nuevo conocimiento simbólico más complejo; (2) el enfoque ontogenético que se encarga de adaptar y evolucionar las dinámicas internas que determinan la especialización y caracterización del conjunto de *Comportamientos* de la arquitectura; y (3) el enfoque filogenético, que define a nivel de la población de agentes cognitivos, los mecanismos encargados de evolucionar las configuraciones cognitivas que permitirán a los agentes adaptarse de forma adecuada a las condiciones del entorno con el cual interactúan.

- El enfoque epigenético concierne a aquellos procesos y estructuras auto-organizadas de METÁFORA (tales como las máquinas autopoiéticas de primer orden representadas por las redes neuronales artificiales -RNA- y por los sistemas inmunes artificiales -SIA), las cuales son afectadas directamente por el entorno y por los estados internos del agente. Adicionalmente, el aspecto epigenético definirá diferentes tipos de aprendizaje (e.g., supervisado, por refuerzo, y por inducción), de tal manera que cada algoritmo de aprendizaje compensa las debilidades de los demás y consolida las fortalezas mutuas.

- El enfoque ontogenético concierne al desarrollo cognitivo de cada agente cuya arquitectura subyacente sea METÁFORA. Este proceso explota los mecanismos de diferenciación funcional, como por ejemplo la especialización del conjunto de *Comportamientos* o de estructuras cognitivas de orden superior (e.g., los planes y las estrategias) de la arquitectura. Similar a la auto-reparación celular, cada *Comportamiento* puede potencialmente reemplazar algún otro (cuando éste último no funcione correctamente), o simplemente crear la redundancia necesaria para obtener el control en la competencia por el foco de atención. El enfoque ontogenético, se encargará además de crear estructuras de planes complejos (bien sea como una estructura jerárquica o secuencial de *Comportamientos*) a través de un proceso darwiniano que dirige la búsqueda hacia el plan óptimo.

- El enfoque filogenético concierne a la evolución de las estructuras cognitivas que el agente ha adquirido a través de su ontogénesis y epigénesis a lo largo de su interacción con el entorno. Las estructuras cognitivas (memes) que participarán em el proceso de evolución son: (1) las unidades básicas de conocimiento que conforman los *Comportamientos* del agente (reglas), (2) las redes de comportamientos, y (3) las estructuras jerárquicas y secuenciales de los planes deliberativos. Las asociaciones entre memes del mismo tipo producen memes de mayor jerarquía y complejidad, así, por ejemplo, la asociación de memes de reglas constituyen *Comportamientos*, que a su vez forman memes de redes de comportamientos, y la asociación de estos últimos estructuran memes complejos como los planes deliberativos.

- En cada uno de los enfoques epigenético, ontogenético y filogenético, se emplearán procesos darwinianos que operan en diferentes escalas temporales (segundos, minutos, horas, días, etc.), tal como se describió en los requisitos para las Máquinas Darwinianas.

- Los memes, como unidades de transmisión cultural, reflejan las pautas espacio-temporales de METÁFORA, las cuales son evolucionadas continuamente por los procesos darwinianos, en analogía a los procesos de copiado que ocurren en los circuitos neuronales de las capas de la corteza cerebral.
- Se propone un Algoritmo Memético de METÁFORA que operará en dos niveles y escalas temporales: un proceso micro co-evolutivo a nivel epigenético/ontogenético, que actúa en el corto plazo y que busca la adaptación local de los memes a través del aprendizaje y la interacción con el entorno; y un proceso macro co-evolutivo a nivel filogenético, que actúa en el largo plazo operando sobre poblaciones de agentes (transmisión cultural de memes).

5.10. Hipótesis y requisitos relativos al diseño de los Bloques de Construcción

A continuación se presentan los requisitos relativos a la sección 4.5:

- Se propone un enfoque de diseño de la arquitectura a través de componentes reutilizables los cuales permiten modelar las relaciones e interacciones entre las redes auto-organizativas de METÁFORA, en lugar de modelar simplemente las entidades físicas y estructurales que conforman la misma.
- Los bloques de construcción intentan capturar la analogía con otros componentes inspirados en sistemas complejos tales como los sistemas biológicos y los sistemas electrónicos. Estos bloques cibernéticos de primer orden, permitirán construir bloques compuestos con niveles de complejidad superiores, tales como las Máquinas Autopoiéticas, las Máquinas Darwinianas, las Redes de Comportamientos, entre otros.
- METÁFORA adquiere las propiedades de escalabilidad y plasticidad gracias a que el diseñador puede articular los diferentes bloques de construcción a fin de construir diferentes configuraciones de la arquitectura o extender su funcionalidad mediante la incorporación de nuevos módulos. De esta manera, METÁFORA cumple con el principio fundamental de la Ingeniería de Software, el cual persigue un bajo acoplamiento y una alta cohesión entre los componentes de la arquitectura de software.

5.11. Hipótesis y requisitos relativos al empleo de Metáforas para el modelado de la Arquitectura

A continuación se presentan los requisitos relativos a la sección 4.6:

- Se proponen un conjunto de metáforas inspiradas en la biología, la psicología cognitiva, la neurofisiología, los sistemas mecánicos y electrónicos, y la teoría general de sistemas, las cuales servirán de substrato al diseño de METÁFORA como una

arquitectura de enfoque emergente, auto-organizado, adaptativo, escalable, y de propósito general.

- Concretamente, se propone una adaptación de la metáfora del Teatro de la Consciencia a fin de orquestar los diferentes procesos conscientes y subconscientes determinados por METÁFORA. Esta metáfora será realizada mediante una implementación "extendida" del modelo de las Redes de Comportamientos, las cuales se auto-organizarán a través de los procesos adaptativos dirigidos por el enfoque Ontogenético de la arquitectura.

5.12. Aportaciones Científicas y Tecnológicas

A continuación, se sintetizan las principales aportaciones científicas y tecnológicas del presente trabajo de tesis.

- Aportaciones Científicas:
  - Se propone una Arquitectura Cognitiva Híbrida para Agentes Inteligentes inspirada en modelos Biológicos, Psicológicos, Neurofisiológicos, Sistémicos y de Inteligencia Artificial, la cual modula mecanismos auto-organizativos, adaptativos y retroalimentados con el propósito de simular la emergencia de comportamientos inteligentes en el agente.
- Aportaciones Tecnológicas:
  - Se propone una aproximación arquitectónica basada en bloques de construcción cibernéticos de primer orden, a partir de los cuales se pueden generar distintas concreciones funcionales y contextuales de la arquitectura.
  - Se propone una concreción funcional de la arquitectura, empleando mecanismos de la Inteligencia Artificial clásica (simbólica), así como de la Inteligencia Artificial bio-inspirada (emergente).
  - Se proponen extensiones y adaptaciones de los algoritmos bio-inspirados empleados en la concreción funcional de la arquitectura, con el objetivo de proveer un mejor desempeño, robustez, y capacidad de adaptación del sistema.

Parte II

Definición de la Arquitectura

Capítulo 6

Diseño Conceptual de la Arquitectura Cognitiva

"Not only is there but one way of doing *things rightly, but there is only one way of* seeing *them, and that is, seeing the whole of them".*
— John Ruskin, The Two Paths, 1994.

6.1. Introducción

Como se ha mencionado en el capítulo anterior sobre la Naturaleza de la Solución, ésta debe apuntar al diseño de una arquitectura cognitiva que refleje propiedades emergentes tales como la auto-organización, la complejidad, y la adaptación.

En el presente capítulo, se propone un diseño funcional y conceptual de la arquitectura donde resulta posible identificar los componentes de alto nivel y las relaciones causales entre los mismos. Sin detallar excesivamente su anatomía, se presentan a nivel descriptivo los módulos, componentes y mecanismos de METÁFORA, y se explica la manera en que ésta articula la interacción entre los mismos a fin de obtener comportamientos emergentes.

6.2. Arquitectura Modular de Alto Nivel

Una primera aproximación al diseño de la arquitectura cognitiva presenta los diferentes módulos, niveles, mecanismos y procesos, que interactúan entre sí para construir habilidades cognitivas de orden superior.

La estructura arquitectónica de METÁFORA se puede apreciar desde tres perspectivas diferentes aunque intrínsecamente relacionadas: la *Perspectiva Modular*, la *Perspectiva por Niveles de Procesamiento*, y la *Perspectiva por Comportamientos Especializados*. La Perspectiva Modular intenta articular el funcionamiento global del modelo cognitivo en

módulos discretos, de propósito específico, los cuales poseen cierta analogía con algunas zonas de procesamiento identificables en el cerebro. Por su parte, la Perspectiva por Niveles de Procesamiento presenta un enfoque híbrido donde son dispuestos, horizontalmente, los niveles simbólicos y sub-simbólicos de procesamiento (enfoque Cognitivista y Emergente). Por último, la Perspectiva por *Comportamientos* especializados, presenta una organización vertical de componentes especializados, cuya finalidad se asemeja a las micro-columnas corticales (tal como se describió en los requisitos del apartado5.2), las cuales se distribuyen transversalmente a lo largo de las capas o niveles de procesamiento.

En cuanto a la Perspectiva Modular, METÁFORA se compone de los siguientes módulos (como se observa en la figura 6.1): el módulo de Percepción, el módulo de Orquestación de la Consciencia, el módulo Comportamental, el módulo Declarativo, el módulo Intencional, el módulo Motivacional, el módulo Efector y el módulo Meta-cognitivo.

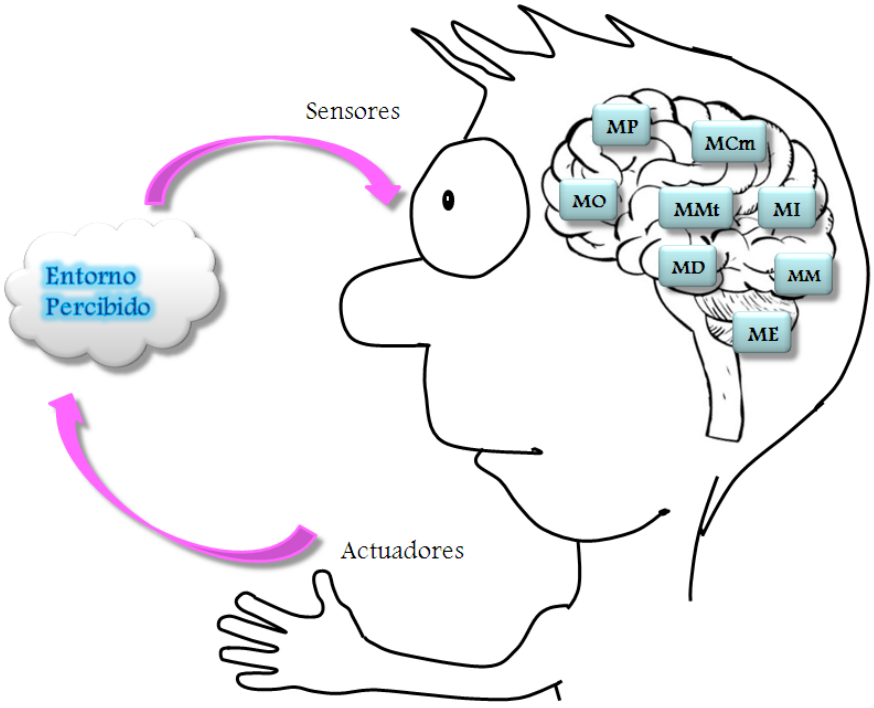


Figura 6.1: Representación caricaturizada de la Arquitectura Modular. MP: Módulo de Percepción, Mo: Módulo de Orquestación de la Consciencia, MCm: Módulo Comportamental, MMt: Módulo Motivacional, MD: Módulo Declarativo, MI: Módulo Intencional, MM: Módulo Meta-cognitivo y ME: Módulo Efector.

En la figura 6.2 se presenta la distribución estructural de los módulos de la arquitectura. Una descripción preliminar de estos módulos se expone a continuación, mientras que el diseño detallado de los mismos se especifica en los capítulos siguientes.

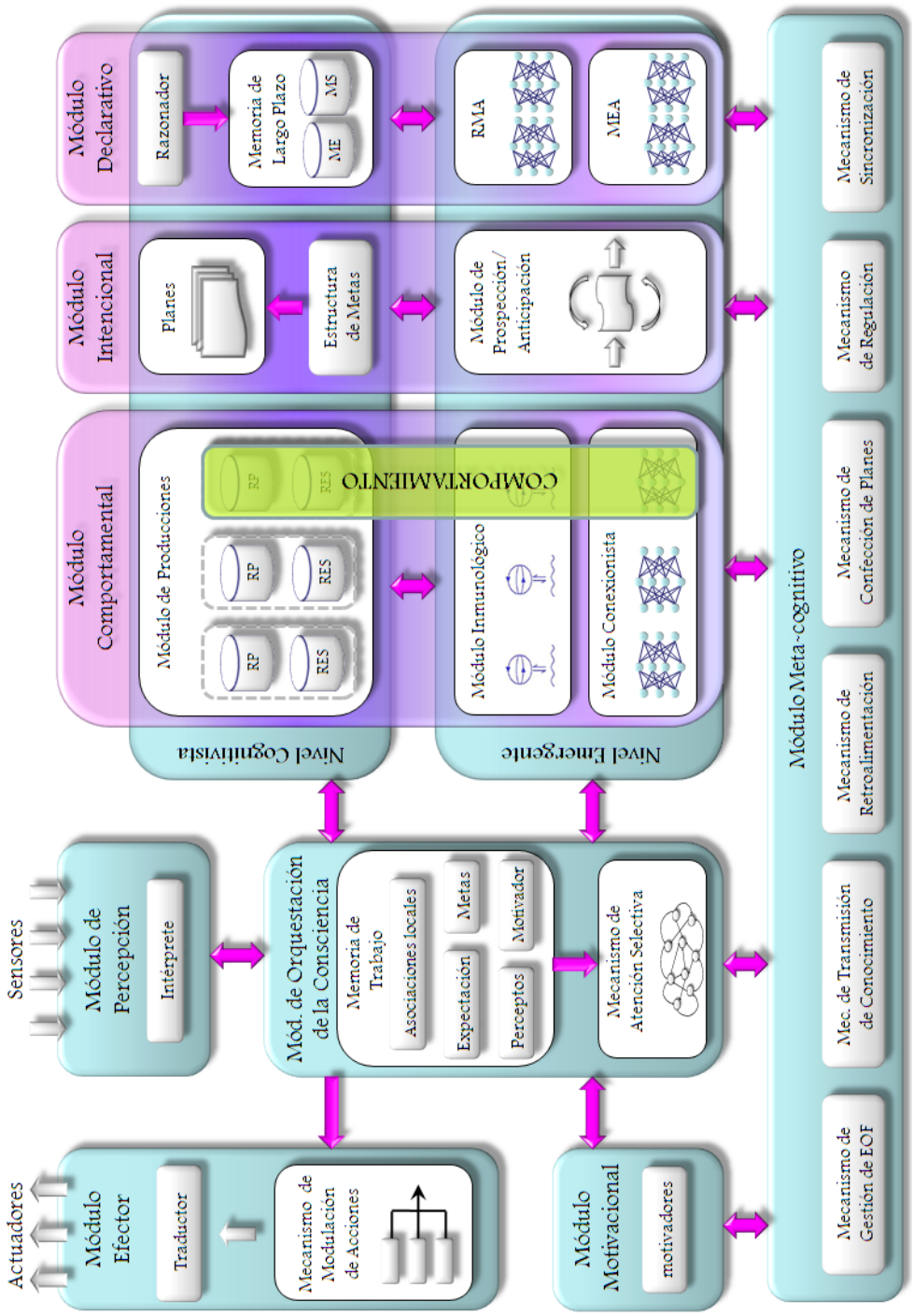


Figura 6.2: Imbricación de Módulos, Niveles de Procesamiento y Comportamientos Especializados en la Arquitectura.

- *El módulo de Percepción:* captura la información del mundo externo percibida a través de los *sensores* del agente, y posteriormente la interpreta en forma de *perceptos*[1], para que pueda ser procesada luego por los demás módulos.

[1]Percepto, por lo que conserva de su estructura de participio latino, se refiere a lo percibido o perceptum, es decir, el objeto tal como lo percibe el sujeto.

- *El módulo Comportamental:* define un conjunto de *Comportamientos* híbridos, los cuales cumplen funciones análogas a las micro-columnas corticales del cerebro. Cada *Comportamiento* emplea simultáneamente tres tipos de razonamiento procedimental: uno de tipo simbólico (módulo de Producciones) y dos de tipo subsimbólico (el módulo Conexionista y el módulo Inmunológico).

  - El módulo de Producciones está conformado a su vez por dos tipos de componentes: las Reglas Pre-determinadas (RP) que son reglas de producción del tipo ⟨`SI...ENTONCES...`⟩ las cuales representan el conocimiento innato del agente, y las Reglas de Extracción Sub-simbólica (RES) que representan unidades de conocimiento simbólico que extrae el sistema, en tiempo de ejecución, a partir del aprendizaje implícito realizado por el nivel emergente.
  - El módulo Conexionista se compone de un conjunto de Redes Neuronales Artificiales (RNA), encargadas de procesar la información sub-simbólica percibida por el agente.
  - El módulo Inmunológico, al igual que el módulo Conexionista, procesa la información sub-simbólica percibida por el agente y emplea un mecanismo de aprendizaje implícito. Este módulo define una colección de Sistemas Inmunes Artificiales (SIA)[2].

- *El módulo de Orquestación de la Consciencia:* se encarga de enfocar adaptativamente la atención del agente y de modular los procesos *conscientes* y *subconscientes*. Este módulo almacenará en la *Memoria de Trabajo*, los elementos percibidos del mundo exterior (perceptos) así como los elementos que describen el estado interno del agente (*metas, intensiones, emociones*, etc.). Este módulo emplea un "*Área de Trabajo Global*" en analogía con el escenario del Teatro de la Consciencia descrito en el apartado 4.6.1, donde los distintos *Comportamientos* (del módulo Comportamental) compiten entre sí por obtener el foco de atención y cooperan con otros (a través de las redes de comportamientos) con el fin de lograr una o varias metas del agente.

- *El módulo Intencional:* este módulo, en analogía con el lóbulo temporal del cerebro (Squire et al., 2004), se ocupa de administrar las *metas* del agente y de crear un *modelo anticipatorio* de los estados futuros del entorno. Adicionalmente, posee un *Planificador* que se encarga de confeccionar *planes* optimizados que se ajustan a las metas del agente.

- *El módulo Declarativo:* se ocupa del mantenimiento de las memorias declarativas de largo plazo en analogía con la función del hipocampo en el cerebro (Squire et al., 2004). Este módulo emplea cuatro tipos de memorias: en el nivel emergente, define las *memorias asociativas* y las *memorias episódicas abstractas*; y en el nivel cognitivista, define las *memorias episódica y semántica*.

- *El módulo Meta-cognitivo:* define el conjunto de meta-estrategias de regulación y control de los componentes de la arquitectura, así como los mecanismos de integración entre los módulos:

[2] En el capítulo 7, se determinan claramente las diferencias esenciales entre los módulos Conexionista e Inmunológico

- Mecanismo de Gestión de EOF: este mecanismo coordina los tres procesos de desarrollo y evolución (i.e., la epigénesis, la ontogénesis, y la filogénesis) encargados de la creación de las diferentes estructuras cognitivas del agente.
- Mecanismo de Transmisión de Conocimiento: una vez que son evolucionadas (a nivel global) las estructuras cognitivas del tipo meme, este mecanismo cumple con la función de transmitirlas a todos los agentes del sistema (simulando un aprendizaje colectivo).
- Mecanismo de Retroalimentación: se encarga de coordinar los distintos bucles y señales de retroalimentación para cada módulo de la arquitectura.
- Mecanismo de Confección de Planes: su finalidad es dirigir el proceso de generación de planes a partir del ensamblaje evolutivo de estructuras cognitivas más básicas.
- Mecanismo de Regulación: este mecanismo ajusta los parámetros que controlan el funcionamiento de los procesos cognitivos superiores del sistema (e.g., aprendizaje, planificación, modulación de acciones, etc.) a fin de hacerlos más eficientes.
- Mecanismo de Sincronización: coordina los procesos de sincronización e integración de las estructuras cognitivas creadas por cada uno de los módulos de los dos niveles horizontales de la arquitectura (el emergente y el cognitivista). Así pues, mediante este mecanismo, la arquitectura puede traducir las representaciones simbólicas creadas por el nivel cognitivista, en representaciones sub-simbólicas en el nivel emergente, y viceversa.

- *El módulo Motivacional:* define el conjunto de *motivadores* (emociones primarias) que orientan la conducta del agente, intentando simular la funcionalidad de la amígdala en el cerebro (Amunts et al., 2005).
- *El módulo Efector:* este módulo cumple con varias funciones cognitivas, en analogía a la función del cerebelo (Kandel et al., 2000). Por un lado, integra las diferentes acciones recomendadas por los componentes del *Comportamiento* que se encuentre activo en un determinado momento. Por otro lado, este módulo posee un Mecanismo de Modulación de Acciones que se encarga de determinar qué acción será ejecutada cuando múltiples acciones propuestas por los diferentes *Comportamientos* activos presentan conflicto. Finalmente, el módulo Efector traduce las acciones de los *Comportamientos* en señales motoras que pueden interpretar los actuadores del agente, o en acciones internas con un formato específico.

Así pues, el diseño modular de la arquitectura no pretende en ningún momento ser un fiel modelo del complejo funcionamiento del cerebro humano, sino que, por el contrario, intenta describir desde un enfoque reduccionista, los módulos y mecanismos necesarios para crear niveles incrementales de comportamiento inteligente.

Por otro lado, desde la Perspectiva por Niveles de Procesamiento, METÁFORA incorpora una nueva dimensión en la arquitectura donde los módulos orientados al procesamiento de unidades de conocimiento disponen de una representación tanto cognitivista como emergente, lo cual se traduce en representaciones simbólicas y sub-simbólicas del conocimiento respectivamente. Así pues, los módulos Comportamental, Declarativo, e Intencional, son de naturaleza híbrida ya que disponen de sub-módulos pertenecientes a

cada tipo de paradigma cognitivo, y se sincronizan entre sí a través de mecanismos de integración, modulación, y aprendizaje.

Por último, la Perspectiva por *Comportamientos* especializados, como se mencionó anteriormente, define un conjunto de *Comportamientos* híbridos dispuestos transversalmente y de forma imbricada a lo largo de los dos niveles de procesamiento (cognitivista y emergente). Estos *Comportamientos*, en principio, son de carácter multi-propósito (tal como se hablaba en la sección 5.2), sin embargo, a través de la interacción que tiene el agente con su entorno, se especializan en el desarrollo de habilidades y en la consecución de metas específicas. A pesar de su especialización, un *Comportamiento* pueden asumir la responsabilidad de otro *Comportamiento* cuando éste no se encuentra disponible (bien sea porque se encuentra desconectado o porque no existe en el repositorio de *Comportamientos* del agente), permitiendo que el sistema sea robusto y tolerante a fallos.

El motivo principal por el cual se ha decidido modelar los *Comportamientos* como unidades híbridas de procesamiento se debe fundamentalmente al *principio de implementación múltiple* descrito en la sección 4.2. Cada *Comportamiento*, empleará 4 algoritmos de procesamiento y aprendizaje (RP, RES, PMR y SIA) los cuales compiten y cooperan entre sí a fin de compensar sus debilidades y mejorar sus fortalezas. Por ejemplo, las redes RNAs y los SIAs resultan ser con frecuencia más apropiados, en comparación con las reglas de producción, en la categorización de objetos y el reconocimiento de patrones, mientras que las reglas se desempeñan mejor en inferencias que involucran cambios causales en las situaciones observadas por el agente. De esta manera, dependiendo de la situación, tendrá mayor relevancia cierto componente del *Comportamiento* sobre los demás componentes.

La incorporación de los dos tipos de algoritmos inspirados biológicamente (RNA y SIA) en el nivel emergente de cada *Comportamiento* de la arquitectura, responde a la necesidad de modelar diferentes componentes de procesamiento distribuido complementarios entre sí. La justificación que argumenta el uso de estas dos técnicas en lugar de otras técnicas de aprendizaje automático conocidas (e.g., los Sistemas Clasificadores, el Algoritmo *Q-learning*, etc.), se debe principalmente a dos razones: por un lado, existe evidencia biológica (Varela, 1992a) de que los sistemas cognitivos y auto-organizados por esencia son el sistema nervioso (compuesto por una amplia red de neuronas) y el sistema inmune (compuesto por redes de anticuerpos); y, por otro, debido a los estudios comparativos presentados en (Romero y Niño, 2006) queda patente la eficiencia de estos dos algoritmos en contraste con otros algoritmos de aprendizaje y clasificación.

Adicionalmente, se presenta el cuadro comparativo 6.1 que resume las diferencias y fortalezas de estos dos componentes bio-inspirados. Cabe resaltar, que el cuadro comparativo incluye además, las características principales de los Algoritmos Genéticos (AG), debido a que los SIA emplean tales algoritmos para guiar el proceso de adaptación del sistema. No obstante, gracias al conjunto de dinámicas y meta-dinámicas que incorporan los SIA, estos reflejan diferencias ventajosas sobre los AG[3] sobre los AG simples.

[3]En el cuadro comparativo son presentadas sólo las diferencias cualitativas. Las diferencias cuantitativas entre los SIA y otras heurísticas bio-inspiradas son reportadas por (N. De Castro y Von Zuben, 2001; Dasgupta, 2002)

Característica	AG (Optimización)	RNA (Clasificación)	SIA (Clasificación)
Componentes	Cadenas de cromosomas	Neuronas artificiales	Cadenas de atributos
Ubicación de Componentes	Dinámica	Pre-definida	Dinámica
Estructura	Componentes discretos	Componentes en red	Componentes discretos
Almacenamiento de conocimiento	Cadenas de cromosomas	Pesos en las conexiones	Concentración de componentes
Dinámicas	Evolución	Aprendizaje	Evolución y Aprendizaje
Meta-dinámicas	Reclutamiento y eliminación de componentes	Construcción y poda de conexiones	Reclutamiento, clonación, maduración y eliminación de componentes
Interacción de componentes	Cruce genético	Conexiones sinápticas	Reconocimiento y cruce genético
Interacción con el entorno	Función de aptitud	Estímulos externos	Función de Reconocimiento

Tabla 6.1: Comparación cualitativa entre tres técnicas bio-inspiradas: GA, RNA y SIA (Dasgupta, 2002).

Una característica que merece la pena destacar respecto a los *Comportamientos*, es que estos brindan gran escalabilidad a la arquitectura, ya que ésta puede ser extendida mediante la incorporación dinámica de cuantos *Comportamientos* se requieran para la solución de un problema en particular. Un aspecto que no se cubre en el alcance de esta tesis pero que resulta interesante para un desarrollo futuro, es el de proveer a la arquitectura de algún mecanismo para auto-definir *Comportamientos* a partir de la categorización de los filtros sensoriales, las metas a alcanzar, y las respectivas actuaciones permitidas.

Una visión global de la arquitectura donde es posible distinguir los distintos niveles, módulos, sub-módulos y mecanismos de METÁFORA, se puede apreciar en la figura 6.2. Es importante destacar que la arquitectura no posee un módulo de control central sino que dispone de un conjunto de módulos distribuidos y asíncronos que le permiten ser más robusta y flexible, ya que el funcionamiento de cada módulo no depende de forma directa del funcionamiento de los demás. De esta manera, se disminuye el acoplamiento entre módulos y se mejora la cualidad de tolerancia a fallos. Por ejemplo, el módulo Comportamental funciona de manera independiente a los demás módulos ejecutando concurrentemente los distintos sub-módulos internos, los cuales a su vez, seleccionan acciones candidatas en cada paso del ciclo cognitivo. No obstante, el módulo Comportamental no depende de una comunicación directa con los demás módulos de la arquitectura para llevar a cabo su procesamiento interno.

Como se puede observar en la figura 6.2, la estructura arquitectónica de METÁFORA se distribuye a través de niveles, módulos y componentes que se encuentran imbricados entre sí proporcionando una visión híbrida de la arquitectura. De esta manera, es posible identificar los distintos módulos especializados de la arquitectura, de los cuales, los módulos Comportamental, Declarativo e Intencional se disponen sobre los dos niveles de procesamiento (cognitivista y emergente). Adicionalmente, superpuesto al módulo Comportamental, se encuentra un conjunto de componentes híbridos transversales (los

Comportamientos) que integran distintas unidades micro-cognitivas (RNAs, SIAs, RPs y RESs).

Como comentario final, cabe mencionar que el proceso de cognición global de la arquitectrura METÁFORA se ocupa del mantenimiento de las estructuras y los mecanismos relativos a:

- El estado pasado: el cual almacena las representaciones asociadas con las experiencias previas del agente. Esta responsabilidad atañe a la Memoria de Trabajo y a los módulos Declarativo y Comportamental;
- El estado actual: el cual es actualizado continuamente para reflejar las representaciones acerca de lo que el agente percibe y las decisiones que toma en cada situación presente. El mantenimiento del estado actual es responsabilidad de los módulos de Percepción, Orquestación de la Consciencia, Comportamental, Motivacional, y Efector;
- El estado futuro: el cual mantiene las estructuras que representan situaciones futuras y los respectivos cursos de acción que pueden predecirse sobre dichas situaciones. El mantenimiento del estado futuro compete a los módulos Intencional, Orquestación de la Consciencia, y Comportamental.

"Declara el pasado, diagnostica el presente, predice el futuro".
– Hipócrates de Cos.

6.3. Substrato Computacional de Inspiración Biológica

A diferencia de la sección anterior que presenta una vision estática de la estructura modular y los elementos constitutivos de METÁFORA, esta sección describe las relaciones auto-organizativas entre dichos elementos y cómo a partir de éstas emergen los niveles incrementales de complejidad y las estructuras cognitivas de orden superior de la arquitectura.

6.3.1. Niveles Jerárquicos Auto-organizativos de la Arquitectura

En la actualidad se considera que detrás de los procesos de auto-organización hay un gran potencial creativo, puesto que, como se ha visto, un bucle recurrente puede llevar al surgimiento espontáneo de nuevas estructuras cognitivas y niveles de organización de la arquitectura. A continuación, se hace alusión a algunas ideas planteadas por otros autores, acerca del cómo podrían surgir en los sistemas autopoiéticos biológicos, los procesos cognitivos de orden superior (como la consciencia) a partir de bucles retroalimentados que crean nuevas dimensiones explicativas del fenómeno emergente. Seguidamente, se propone un esquema acerca de cómo esta emergencia por niveles auto-organizativos puede ser simulada a través de un modelo computacional con el propósito de evidenciar comportamientos inteligentes en METÁFORA.

La consideración de la "red" como patrón organizativo de los sistemas, es el punto de partida de las aportaciones de la cibernética y de la teoría de la información a los estudios de los sistemas vivos. Es posible entender cualquier sistema como una red no lineal donde la información circula a través de sus mallas formando ciclos. Estos bucles tienen efectos de autorregulación del propio sistema, tal como se describió en profundidad en el apartado 4.4.3.1. Los bucles de retroalimentación se pueden entender como disposiciones circulares de elementos que están conectados causalmente, de manera que una fluctuación inicial circula por todas las mallas en una cadena continua de multi-causas y multi-efectos que a la vez son causas de efectos posteriores, llegando a retroalimentar al nódulo inicial, haciéndolo causa de sí mismo (Kauffman, 1995).

En los manuales de Neurociencia a menudo se habla de circuitos nerviosos como una conjunción de agrupaciones o redes neuronales, con características funcionales diferentes, que realizan funciones mucho más complejas que las que podrían realizar un tipo único de neurona (Delgado et al., 1998). El cerebro se presenta como una estructura reticular o entramado de células cuya misión es la transmisión de estímulos mediante la comunicación entre ellas (Portellano, 2005). No obstante, tanto circuitos como redes pueden ser representados en planos, es decir, son variedades bidimensionales. La *consciencia humana* tiene como soporte las redes neuronales anteriormente aludidas, pero para emerger tiene que trascender esa bidimensionalidad de la red y adquirir otra nueva dimensión, ha de modificar la estructura reticular para alcanzar esa tercera dimensión que permita reconocer sus actos, sus reflexiones y a sí mismo como individuo. El proceso continua, creando niveles más complejos de cognición (e.g., la cognición social).

Según Sarmiento (Sarmiento, 2009), *elemento humano, sistema humano y meta-sistema humano*, son tres sistemas diferenciados que interactúan y emergen en un "elemento-sistema" consciente de su existencia y de la existencia de elementos–sistemas de su misma naturaleza. Como *"elemento humano"*, es un sistema biológico que se desarrolla según su constitución física y los procesos inherentes al propio sistema; como *"sistema humano"*, es un sistema cognitivo que se desarrolla sobre estructuras simbólicas formales o pseudoformales y que necesita del soporte sistémico de nivel inferior, de sus dimensiones; y como *"meta-sistema humano"* es un sistema social del que forma parte como elemento.

De manera análoga, en METÁFORA, es posible determinar cómo, a partir de múltiples niveles auto-organizativos, emergen en una nueva dimensión los procesos conscientes del agente. Así mismo, es posible distinguir en METÁFORA los tres sistemas propuestos por Sarmiento, pero ahora enmarcados en las dinámicas internas de cada nivel auto-organizativo de la arquitectura: *elemento cognitivo*, *sistema cognitivo* y *meta-sistema cognitivo. A continuación se describen brevemente dichos niveles auto-organizativos:*

- Un nivel auto-organizativo de primer orden, que corresponde al "sistema biológico" de METÁFORA, es constituido por dos tipos de máquinas autopoiéticas: las Redes Neuronales Artificiales (RNA) y los Sistemas Inmunes Artificiales (SIA). Este nivel, como *elemento cognitivo*, es un sistema dirigido por procesos epigenéticos, que construye los bloques básicos de conocimiento en el nivel emergente de la arquitectura a partir de los cuales se crean estructuras cognitivas más complejas.

- Un nivel auto-organizativo de segundo orden que emerge de la interacción entre el nivel anterior y un modelo conexionista basado en Redes de Comportamientos (RC), da paso a una nueva dimensión: la orquestación de los procesos *conscientes* y *subconscientes* de la arquitectura. Esta nueva dimensión donde emerge la consciencia[4], permite al agente modelar su realidad (tanto espacial como temporal), así como los procesos implicados en la toma de decisiones y en la valoración de sus propias actuaciones.
- Un nivel auto-organizativo de tercer orden emerge de la interacción de múltiples RCs: un Sistema de Prospección-Anticipación que predice estados futuros de interacción con el ambiente y crea planes contingentes. Este Sistema de Prospección-Anticipación simula internamente la interacción con el entorno en un doble bucle anidado: un primer bucle que emplea un Sistema Clasificador Anticipatorio (SA), el cual simula señales perceptuales de estados futuros del ambiente y determina las acciones respectivas mediante la activación de los *Comportamientos* del nivel inferior; y un segundo bucle que modula el acoplamiento estructural con el ambiente a través de la generación de planes anticipatorios. Este nivel junto con el anterior juegan el rol de *sistema cognitivo*, el cual se encarga del desarrollo de las estructuras cognitivas de alto nivel a través de los procesos ontogenéticos.
- Un nivel auto-organizativo de cuarto orden, que juega el rol de *meta-sistema cognitivo*, se identifica en el sistema social, es decir, en el sistema multi-agente. Cada agente, como individuo autónomo, interactúa continuamente con su ambiente y, como elemento social, participa en procesos adaptativos a nivel poblacional donde son evolucionadas las estructuras cognitivas de mayor complejidad: los memes. Estos procesos evolutivos del meta-sistema cognitivo son dirigidos por el enfoque filogenético.

A continuación, se detalla cada uno de estos niveles de organización de la arquitectura, en relación con los requisitos expuestos en los apartados 5.4, 5.5 y 5.6.

6.3.1.1. Máquinas Autopoiéticas de Primer Orden

Cómo se describió en la sección 4.4.2.2, en los organismos vivientes son considerados como sistemas autopoiéticos, el sistema nervioso y el sistema inmune, ambos capaces de auto-regular sus procesos internos y de mantenerse en homeostasis con respecto a las perturbaciones del entorno. De la misma manera, en METÁFORA se emplean estos dos sistemas autopoiéticos biológicos como inspiración al diseño de las máquinas autopoiéticas artificiales de primer orden: las Redes Neuronales Artificiales (RNA) y los Sistemas Inmunes Artificiales (SIA).

Tanto las RNA como los SIA se encargan de producir y mantener (auto-regular, auto-reparar, auto-organizar, crear, destruir, etc.) sus componentes internos –neuronas y anticuerpos respectivamente– los cuales representan las unidades básicas de conocimiento sub-simbólico en el nivel Emergente. Estas unidades básicas funcionan a un micro-nivel cognitivo, convirtiéndose en los bloques de construcción básicos de estructuras mentales

[4]Desde una perspectiva etológica y psicológica, descartando deliberadamente otros enfoques explicativos sobre el fenómeno de la consciencia como el filosófico, el evolutivo, el patológico, entre otros.

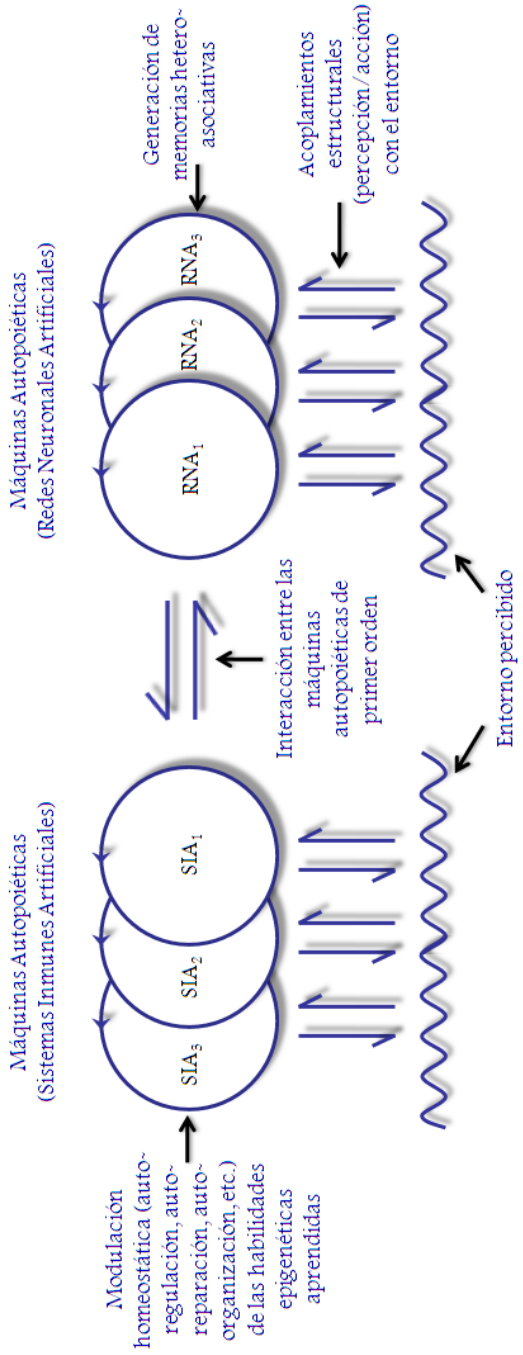


Figura 6.3: Máquinas Autopoiéticas de Primer Orden. Por cada *Comportamiento* existen dos MAs: una Red Neuronal Artificial ($RNA_1, RNA_2, RNA_3 \ldots$) y un Sistema Inmune Artificial ($SIA_1, SIA_2, SIA_3 \ldots$), las cuales interactúan entre sí y con el entorno para construir memorias hetero-asociativas[5] que representan las habilidades aprendidas por el agente.

más complejas tales como los *Comportamientos*, las redes de comportamientos, y los planes deliberativos.

En METÁFORA, las Máquinas Autopoiéticas (MAs) de primer orden se ocupan de generar memorias hetero-asociativas que relacionan los acoplamientos estructurales con el entorno a través de duplas de ⟨`percepción, acción`⟩. Las unidades de estas memorias hetero-asociativas son moduladas por los procesos homeostáticos de cada MA de primer orden, los cuales adaptan las dinámicas internas de regulación a los cambios observados en el entorno.

Empleando los ideogramas propuestos por Maturana y Varela (Maturana y Varela, 1987), en la figura 6.3 se representa la interacción entre las MAs de primer orden en METÁFORA. Cabe notar en esta figura, que se hace referencia a un conjunto determinado de *Comportamientos* del agente $C = \{(c_1, c_2, c_3 \ldots\}$, donde cada *Comportamiento* se compone de dos MAs que interactúan entre sí: $c_1 = \{RNA_1, SIA_1\}$; $c_2 = \{RNA_2, SIA_2\} \ldots$

El proceso de modulación realizado por las MAs de primer orden, corresponde al aprendizaje epigenético del agente, como se verá en la sección 6.3.2.1.

6.3.1.2. Máquinas Autopoiéticas de Segundo Orden

En un nivel superior de organización, se encuentran las Redes de Comportamientos (RC) que actúan como máquinas autopoiéticas compuestas. Las RC auto-regulan sus propios componentes, los *Comportamientos*, los cuales a su vez son sistemas auto-organizativos conformados por MAs de primer orden (RNA y SIA). En la figura 6.4 se puede apreciar la interacción de la MA de segundo orden con las MAs de primer orden que la conforman.

Como se observa en la figura 6.4, la MA de segundo orden realiza la modulación homeostática del conjunto de *Comportamientos* del agente, a través de la propia Red de Comportamientos auto-organizada. Esta modulación de la RC se efectúa desde dos enfoques: el primero, basado en las dinámicas internas de la red, determina la activación de los *Comportamientos* a través del envío de señales *exitativas* (que permiten la cooperación) y de señales *inhibitorias* (que promueven la competencia); y el segundo, basado en un proceso darwiniano dirigido por el desarrollo ontogenético del agente, el cual se encarga de modificar adaptativamente la funcionalidad y la estructura de la propia red de comportamientos, como se explicará en la sección 6.3.2.2.

Las MAs de segundo orden se acoplan estructuralmente al entorno a través de los estados percibidos por el agente y de las acciones que recomienda cada *Comportamiento* activado en respuesta a las perturbaciones ambientales.

Internamente, las MAs de primer orden (los componentes RNA y SIA de cada *Comportamiento*), se acoplan estructuralmente con el entorno a través de la propia historia de interacción entre los *Comportamientos* y dicho entorno. Esta historia es encapsulada en los *Comportamientos* mediante memorias hetero-asociativas que describen las relaciones de percepción/acción entre el agente y el entorno.

Adicionalmente, existe un acoplamiento de auto-modificación entre la MA de segundo orden y las MAs de primer orden, y viceversa. Las perturbaciones que genera la MA

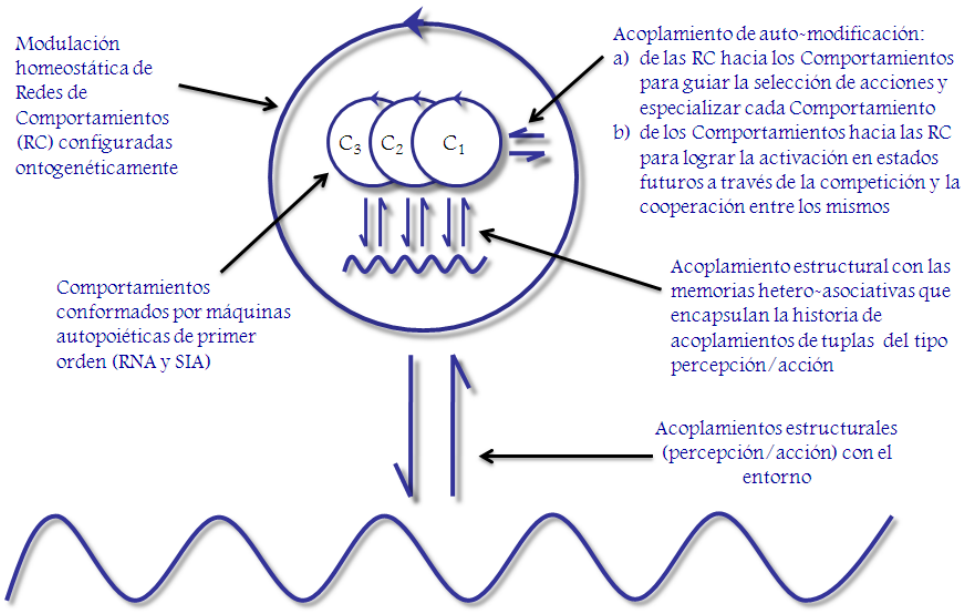


Figura 6.4: Máquinas Autopoiéticas de Segundo Orden. Las Redes de Comportamientos, como máquinas autopoiéticas compuestas, modulan la interacción de las máquinas autopoiéticas de primer orden.

de segundo orden sobre las MAs de primer orden corresponden a los cambios del estado percibido, producidos tanto por factores externos (cambios en el entorno) como por factores internos del agente (metas, motivadores, emociones, intenciones, etc.).

De esta manera, el acoplamiento de auto-modificación de la MA de segundo orden sobre las MAs de primer orden, permite que el mecanismo de selección de acciones de cada *Comportamiento* sea guiado, y además brinda las pautas necesarias para que cada uno se "especialice" –considerando que cada *Comportamiento* no define inicialmente una finalidad concreta sino que actúa como las instalaciones básicas descritas en la sección 4.3, y que es a través de la interacción con el entorno y de la modulación de dicha interacción, que logra especializarse.

Por otro lado, el acoplamiento de auto-modificación en el sentido contrario, de las MAs de primer orden hacia la MA de segundo orden, permite la competición y la cooperación entre los *Comportamientos* de la misma red mediante la modificación de las asociaciones entre dichos *Comportamientos*, con el objetivo de aumentar la probabilidad de ser activados en estados futuros.

En este punto, cabe resaltar que la interacción entre los *Comportamientos* y la auto-organización de los mismos a través de la máquina autopoiética de segundo orden (la Red de Comportamientos), induce la emergencia de un nuevo nivel de organización: "la consciencia artificial", desde el punto de vista que define la Metáfora del Teatro de la Consciencia (sección 4.6.1).

En este punto, cabe resaltar que los procesos auto-organizativos que dirige la máquina autopoiética de segundo orden (la Red de Comportamientos), inducen la emergencia de

un nuevo nivel de organización: "la consciencia artificial" del agente. El mecanismo de orquestación de los procesos conscientes y subconscientes será descrito al detalle en el capítulo 9.

6.3.1.3. Máquinas Autopoiéticas de Tercer Orden

Las máquinas autopoiéticas de tercer orden corresponden al Sistema de Prospección-Anticipación de METÁFORA. Esta MA, cuya representación se puede observar en la figura 6.5, modula la interacción entre las MAs de segundo orden (las Redes de Comportamientos).

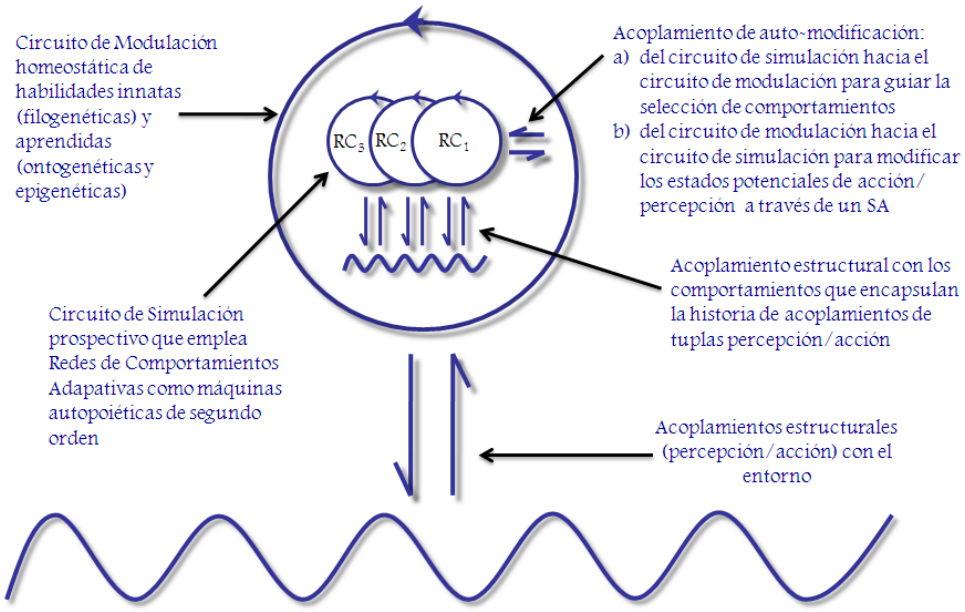


Figura 6.5: Máquinas Autopoiéticas de Tercer Orden. Un Sistema de Prospección-Anticipación, como máquina autopoiética de tercer orden, modula la interacción de las máquinas autopoiéticas de segundo orden, (i.e., las Redes de Comportamientos).

Esta MA de tercer orden permite crear un circuito modulatorio de las habilidades tanto innatas (definidas a través de procesos filogenéticos) como aprendidas (mediante procesos ontogenéticos y epigenéticos), con el propósito de "simular" estados futuros de interacción entre el agente y su entorno, y de crear planes anticipatorios acordes con estos estados.

Internamente, el Sistema de Prospección-Anticipación posee un circuito de simulación guiado por las Redes de Comportamientos. Estas redes poseen un acoplamiento estructural con el entorno a través de la historia de acoplamientos que es encapsulada en las MAs de primer orden (RNA y SIA). Estos acoplamientos estructurales, como se mencionó anteriormente, corresponden a memorias hetero-asociativas que representan la interacción con el entorno a través de duplas de $\langle$`percepción, acción`$\rangle$. El circuito de simulación actúa de la siguiente manera:

1. La máquina autopoiética de tercer orden posee internamente un Sistema Clasificador Anticipatorio (SA) (Stolzmann, 1998) que aprende latentemente a predecir estados futuros del entorno. A partir de la acción determinada por el agente para el estado actual percibido, el SA predice las consecuencias de ejecutar dicha acción (interna o externamente). Luego, el estado futuro pronosticado por el SA es almacenado en la Memoria de Trabajo como el "estado actual" percibido por el sistema.

2. Luego, se elige la MA de segundo orden (Red de Comportamientos) más apropiada para ejecutar sus dinámicas a fin de activar un *Comportamiento* como respuesta al "estado actual".

3. Una vez elegido el *Comportamiento*, éste activa sus MAs de primer orden (RNA y SIA) y los dos componentes simbólicos (RP y RES), e integra sus salidas respectivas. De esta manera se obtendría una recomendación de una acción concreta para ser ejecutada.

4. Esta nueva acción determinada por el *Comportamiento*, es enviada nuevamente al SA del paso 1, con el objetivo de predecir el siguiente estado futuro.

5. El bucle continua hasta que se alcanza una meta o hasta que no es posible predecir nuevos estados futuros. En este punto, el circuito de simulación obtiene una secuencia de *Comportamientos* y acciones, que pueden ejecutarse a manera de plan deliberativo.

Adicionalmente, existe un acoplamiento de auto-modificación desde el circuito de simulación hacia el circuito de modulación que determina cuál Red de Comportamientos será activada: debido a que cada RC posee un propósito diferente, éstas tienen que competir entre sí por obtener el control de la ejecución. El acoplamiento de auto-modificación en dirección contraria, desde el circuito de modulación hacia el circuito de simulación, se encarga de modificar los posibles estados futuros de acción/percepción del agente empleando un SA, como se describió anteriormente.

Como se puede observar hasta este punto, la estructura jerárquica de máquinas autopoiéticas de primer, segundo y tercer orden, permite que emerja una nueva dimensión cognitiva en el estado crítico del sistema: la prospección y anticipación a eventos futuros y la generación de planes deliberativos como consecuencia. Esta misma estructura jerárquica de MAs podría ser trasladada a otros contextos más allá de la planificación de secuencias de *Comportamientos* y acciones, como por ejemplo, a la simulación del pensamiento "creativo" y de los procesos imaginativos que surgen durante el sueño de fase REM[6], como sugiere (Holley, 2004), sin embargo, estos aspectos no se encuentran dentro del alcance de la presente tesis aunque sí serán considerados como una línea futura de investigación.

[6] Acrónimo en inglés para Rapid Eye Movement –Movimiento Rápido de Ojos–, que determina una fase profunda del sueño donde ocurren procesos creativos a través de la asociación de elementos que resultan en nuevas combinaciones útiles.

6.3.1.4. Sistema Complejo Adaptativo

Un último nivel de organización que emerge del anidamiento jerárquico de las máquinas autopoiéticas subyacentes, es el *Agente* como sistema autónomo. Este sistema autónomo, posee las propiedades características de un Sistema Complejo Adaptativo (CAS)(descritas en los requisitos de la sección 5.6) ya que: define niveles jerárquicos de complejidad organizada, demuestra procesos auto-organizativos de auto-regulación y auto-producción de los elementos constitutivos, mantiene la homeostasis de los procesos internos del sistema, exhibe un comportamiento emergente global, y presenta mecanismos que permiten la adaptación y evolución de sus componentes. En la figura 6.6 se aprecia la interacción entre dos agentes cuya autonomía emerge de los procesos auto-organizativos y homeostáticos internos.

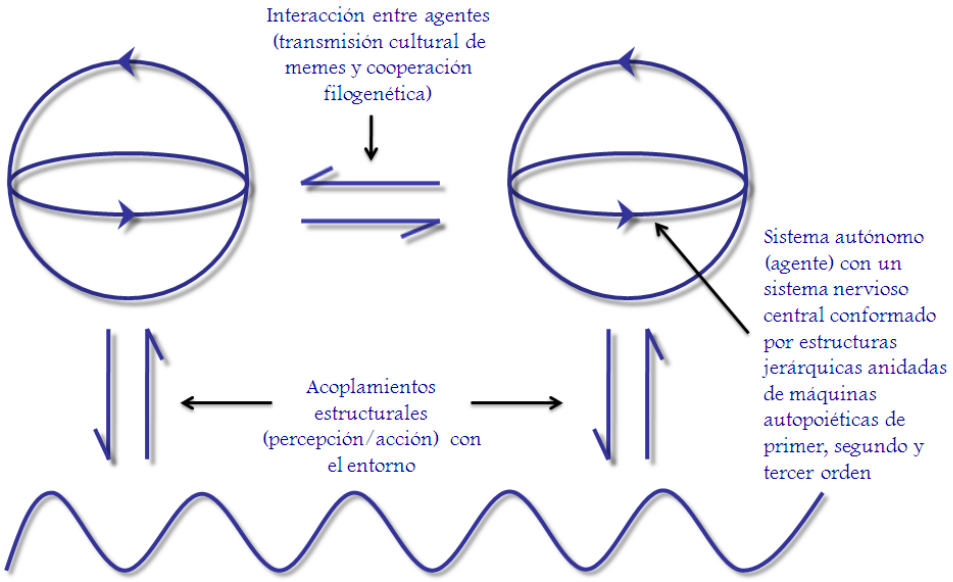


Figura 6.6: Representación de los agentes de la arquitectura como sistemas autónomos que poseen un sistema nervioso sintético.

Como se mencionó en la sección 4.4.2.3, la definición de los diferentes niveles jerárquicos auto-organizativos en METÁFORA se fundamenta en la tesis propuesta por Goldspink y Kay (Goldspink y Kay, 2003) acerca de crear un puente entre dos perspectivas: la teoría autopoiética y la teoría de la complejidad. La teoría autopoiética provee una base para la comprensión de las características de la cognición a micro-nivel en los agentes, mientras que la teoría de la complejidad provee una base para la comprensión de cómo estas características influyen en el rango y tipo de los comportamientos de macro-nivel que emergen de tal interacción.

De esta manera, se ha empleado la teoría autopoiética para describir, por un lado, la interacción recíproca entre el comportamiento a micro-nivel cognitivo (las máquinas autopoiéticas de primer, segundo y tercer orden que conforman el sistema nervioso sintético

– SNS – del agente[7]); y, por otro, el patrón resultante de comportamientos en el macro-nivel (el agente autónomo y el sistema multi-agente).

La teoría de la complejidad provee un modelo de cómo el fenómeno macro cognitivo emerge de la interacción compleja (y no lineal) entre los agentes heterogéneos que constituyen un sistema social. Así, la teoría de la complejidad permite explicar las dinámicas resultantes a través de la descripción de los procesos generativos que se derivan cuando estos agentes entran en interacciones recurrentes que los convierte en entidades estructuralmente acopladas. Desde esta perspectiva, el sistema multi-agente que define METÁFORA puede ser visto como una clase específica de sistema complejo, donde a través de la autopoiesis se clarifican las características distintivas de los agentes que lo conforman, y de allí, las clases de comportamientos que pueden emerger.

Cada agente, como un sistema complejo adaptativo, interactúa con otros agentes a través de la transmisión cultural de estructuras cognitivas complejas (memes) y participa en los procesos evolutivos de dichas estructuras. Así pues, cada agente *coopera* con los demás a través del legado cognitivo que aporta al proceso co-evolutivo; legado que se construye mediante la historia de interacciones del agente con el entorno, las cuales son encapsuladas en los diferentes niveles jerárquicos de la arquitectura.

Como se puede observar, los procesos duales de cooperación y competencia resultan ser aspectos esenciales y concomitantes en los diferentes niveles jerárquicos de METÁFORA. Las influencias de estos procesos pueden ser inextricables aunque complementarias, en la medida en que la competencia en un nivel jerárquico de la arquitectura puede ser considerada como cooperación en el nivel superior siguiente.

Por ejemplo, la *competencia* entre anticuerpos en el SIA (MA de primer orden) excluye los anticuerpos menos adaptados, lo cual genera mejores respuestas inmunes, es decir, mejores estructuras hetero-asociativas. Así, esta *competencia* optimiza los recursos del agente (memoria, tiempo de procesamiento, etc.) y además especializa la funcionalidad de los *Comportamientos* en cada RC (MA de segundo orden), lo que en este nivel de organización puede verse como una *cooperación*. Estos *Comportamientos* a su vez *compiten* por obtener el foco de control de la actuación del agente. De esta manera, a través de la *competencia*, los *Comportamientos* ganan el derecho a participar en la formación de planes que dirige el Sistema de Prospección-Anticipación en el siguiente nivel organizativo (MA de tercer orden), y dicha participación se convierte en *cooperación* una vez se ha estructurado el plan como una secuencia ordenada de *Comportamientos* que busca alcanzar una meta específica...

Otro aspecto interesante que cabe resaltar de la figura 6.6, es que se esquematiza un Sistema Nervioso Sintético (SNS) que coordina la autonomía de cada agente, conformado por una jerarquía de máquinas autopoiéticas de primer, segundo y tercer orden, las cuales aumentan en complejidad. Existe evidencia, tanto en la neurofisiología (Eccles, 1966) como en la teoría de sistemas (Milsum y Bertalanffy, 1978), de que el SNC de los animales ha crecido jerárquicamente a través de un desarrollo filogenético. Por ejemplo, el sistema neuromuscular del hombre sin duda tiene el arco reflejo espinal básico, pero sobre éste están sucesivamente superpuestas vías de "arcos más largos" a través del tal-

[7]Se emplea el término "sintético" para establecer la distinción con el sistema nervioso natural presente en ciertos organismos pluricelulares.

lo encefálico, del cerebelo y de la corteza cerebral. Los indicios actuales señalan que el reflejo espinal básico se conserva y es verdaderamente esencial para los reflejos rápidos de protección, pero que los arcos jerárquicos se usan para reflejos sucesivamente más refinados, estables y que requieran más destreza, así como para acciones voluntarias.

De manera similar, METÁFORA define para cada agente un SNC jerárquico conformado por el anidamiento de máquinas autopoiéticas. Las MAs de primer orden modelan los arcos reflejos básicos, cuya funcionalidad es dotar al agente de comportamiento "reactivo", es decir, de respuestas rápidas (en el orden de milisegundos) que se disparan de manera casi involuntaria, ya que son respuestas instintivas (innatas) o aprendidas, pero que en este último caso se han convertido en respuestas automáticas debido a la confrontación de la eficiencia de dicha respuesta reactiva ante situaciones ambientales similares.

Por su parte, las MAs de segundo y tercer orden se corresponden con arcos más largos, los cuales dotan al agente con un comportamiento más deliberativo y predictivo, donde se construyen planes y estrategias que serán efectivas en escalas de tiempo de segundos, minutos, o incluso, en el largo plazo. Sin embargo, como ya se ha mencionado anteriormente, estos arcos o niveles jerárquicos necesitan de los más básicos, en especial cuando se requiere "simular" o "ensayar" la secuencia de *Comportamientos* y acciones que ejecutaría un plan en el "mundo real".

6.3.2. Niveles Adaptativos de la Arquitectura

En esta sección, se describe cómo serán dirigidos los principios biológicos de desarrollo y evolución en los diferentes módulos y mecanismos de METÁFORA, por medio de la simulación de los procesos de epigénesis, ontogénesis y filogénesis descritos en el apartado 4.4.3.3. Así mismo, se darán las pautas iniciales para la realización de los requisitos de la sección 5.9.

6.3.2.1. Nivel Epigenético

Existen dos enfoques principales para representar la epigénesis computacionalmente. La primera justifica la utilización de estructuras de desarrollo que se expanden gradualmente basándose en la simulación de los procesos de reproducción celular (Bongard, 2002; Quick et al., 2003; Kasabov y Benuskova, 2004; Eggenberger, 2003), y la segunda considera la utilización de estructuras básicas con dinámicas internas las cuales se auto-organizan durante ejecuciones en tiempo real (Floreano y Urzelai, 2000; Bergfelt y Linaker, 2002; Ziemke y Thieme, 2002).

En el diseño de METÁFORA se emplearan estos dos enfoques de manera diferenciada. En cuanto al primer enfoque, se propone como estructuras de desarrollo las redes inmunes, donde los procesos de reproducción celular se corresponden con las meta-dinámicas del SIA encargadas de producir, clonar (replicar), recombinar y eliminar anticuerpos del repertorio inmune, expandiendo gradualmente de esta manera las redes inmunes a través

de la creación de nuevos lazos de interacción entre los anticuerpos. Por otro lado, el segundo enfoque se aplicará a través de las dinámicas auto-organizativas que define cada máquina autopoiética de primer orden (RNA y SIA).

Como se puede observar, los SIA participan en los dos enfoques epigenéticos mientras que las RNA sólo actúan en el segundo enfoque, es decir, las RNA no pueden expandir su estructura durante el desarrollo del agente sino que sólo pueden ajustar las conexiones sinápticas entre las neuronas como consecuencia de la modulación de una señal de retroalimentación que evalúa la correspondencia entre lo percibido y lo ejecutado por la red. Esta decisión de diseño obedece al hecho de que existe evidencia biológica (Rakic, 2002) de que la mayoría de las neuronas cerebrales sólo son generadas antes del nacimiento o durante el período neonatal temprano y las cuales persisten durante muchas décadas a lo largo de la vida del ser sin requerirse la producción de nuevas neuronas. Además, los estudios biológicos demuestran que la maduración cerebral generalmente involucra la reducción del número total de células y de conexiones sinápticas al transcurrir el tiempo (Chechik et al., 1999; Chalup, 2001), lo cual funciona como una estrategia para estimular el olvido de información que el cerebro no utiliza con frecuencia, disminuyendo así el gasto energético y optimizando la velocidad de comunicación entre neuronas (cuanto más cortos sean los caminos que comunican dos neuronas, mayor será la velocidad de procesamiento). No obstante, sería interesante proponer un modelo neuronal evolutivo donde las RNA se reproduzcan y expandan en tamaño, auto-definiendo sus propias topologías y conexiones sinápticas en tiempo de ejecución. Esto último, merece discutirse con más detalle en el capítulo 14 sobre el trabajo futuro.

De esta manera, la perspectiva epigenética reflejará por un lado, las meta-dinámicas que controlan los procesos auto-organizativos de las estructuras cognitivas de bajo nivel y, por otro, reflejará los diferentes procesos de *aprendizaje selectivo* que le permitirán al agente adquirir y descartar (olvidar) unidades de conocimiento que encapsulan la interacción con el entorno. Los diversos mecanismos de aprendizaje epigenético que emplea METÁFORA se categorizan de la siguiente manera:

- *Aprendizaje por Módulos*: esta categoría define mecanismos independientes de aprendizaje en diferentes módulos de la arquitectura con el objetivo de adaptar su funcionalidad a las necesidades ambientales. Concretamente, los módulos que emplearán mecanismos de aprendizaje son:
  - *El módulo Conexionista y las Memorias Asociativas de largo plazo*: estos módulos emplean dos tipos de aprendizaje: supervisado y por refuerzo. En el aprendizaje supervisado se presentan los patrones de entrada a la red neuronal y las respectivas salidas deseadas, y se emplea una fórmula matemática de minimización del error que ajusta los pesos sinápticos para dar la salida más cercana posible a la salida deseada. Por otro lado, usualmente las RNA de retropropagación emplean un aprendizaje por refuerzo (Sutton y Barto, 1998) por el cual se obtiene una señal ambiental de retroalimentación como respuesta a las acciones que propone la red neuronal. Concretamente, se empleará el algoritmo de *Q-learning* propuesto por Watkins (Watkins, 1989) y extendido por Kuzmin (Kuzmin, 2002). De este modo, el módulo Conexionista y las Memorias Asociativas pueden emplear (a criterio del diseñador) un

aprendizaje supervisado (en los casos en los que se disponga de conocimiento experto en el nivel cognitivista, que guíe el aprendizaje) o un aprendizaje por refuerzo dirigido por un algoritmo de *Q-learning* (en el caso de que se desee que la retroalimentación ambiental sea quien guíe el aprendizaje), con el fin de adquirir nuevas estructuras de conocimiento procedimental (en el módulo Conexionista) y declarativo (en las Memorias Asociativas).

- *El módulo Inmunológico*: este módulo empleará un proceso de aprendizaje por descubrimiento el cual permitirá la adquisición de nuevas estructuras hetero-asociativas de conocimiento, las cuales se formarán mediante la compleja interacción entre un algoritmo genético, las meta-dinámicas de clonación y maduración, y un aprendizaje por refuerzo dirigido por un algoritmo de distribución de recompensas.
- *El módulo Anticipatorio*: este módulo adquiere nuevo conocimiento empleando el mecanismo de aprendizaje psicológico de "control comportamental anticipatorio", el cual es un método de aprendizaje por refuerzo empleado en entornos tanto markovianos como no markovianos (Métivier y Lattaud, 2003). Este método adapta la idea del algoritmo de *Q-learning*, incorporando funciones de predicción a los refuerzos futuros a fin de obtener una política óptima de aprendizaje.
- *El módulo de Reglas de Producción y las Memorias Declarativas Simbólicas*: estos dos módulos definen un mecanismo de aprendizaje por inducción por el cual adquieren nuevas unidades de conocimiento en forma de reglas simbólicas. El aprendizaje por inducción realiza tres tareas básicas: (a) evaluar y perfeccionar las reglas disponibles, (b) generar nuevas reglas, y (c) formar asociaciones y jerarquías de reglas con el fin de crear estructuras de conocimiento más amplias. Estas tareas se logran mediante dos mecanismos inductivos: el refinamiento de las reglas existentes y la generación de nuevas reglas mediante procesos lógicos de inferencia.

- *Aprendizaje Vertical*: esta categoría hace referencia al aprendizaje que involucra la consolidación de conocimiento entre los diferentes niveles de procesamiento de la arquitectura, mediante procesos de asimilación y extracción. Los dos tipos de aprendizaje vertical son:

  - *Aprendizaje Descendente (Asimilación)*: este proceso define la consolidación de conocimiento desde las representaciones simbólicas (en el nivel cognitivista), hacia las representaciones sub-simbólicas (en el nivel emergente). En otras palabras, este es un aprendizaje supervisado donde el nivel cognitivista sirve como un "profesor" que modela conocimiento experto y que guía la *asimilación* de conocimiento en el nivel emergente. Este aprendizaje define un parámetro que indica la frecuencia del aprendizaje supervisado que es articulado por el nivel cognitivista con relación al aprendizaje por refuerzo en el nivel emergente (e.g., m pasos de aprendizaje supervisado cada n pasos de aprendizaje por refuerzo).
  - *Aprendizaje Ascendente (Extracción)*: Este tipo de aprendizaje *extraerá* conocimiento implícito del nivel emergente y formará nuevas estructuras de conocimiento explícito en el nivel cognitivista, las cuales serán posteriormente

> validadas a través de la experimentación. La idea básica del proceso de validación es el siguiente: si una acción seleccionada por el nivel emergente ha sido "exitosa" (e.g. satisface cierto criterio), entonces se extrae una regla de representación simbólica. En las interacciones subsecuentes del agente con el entorno, la regla es refinada a través del análisis de las consecuencias de aplicar dicha regla: si la consecuencia es exitosa, la condición de la regla puede ser generalizada para hacerla más universal, de lo contrario, la condición de la regla se constriñe a un dominio más específico de la situación actual, es decir, se especializa. Ésta es una versión en línea (on-line) de los procesos de pruebas de hipótesis que han sido estudiados (en diferentes contextos) como por ejemplo en (Bruner et al., 1956; Nosofsky et al., 1994). El objetivo de la extracción, es crear representaciones simbólicas (concretamente, Reglas de Extracción Sub-simbólica - RES), a partir del conocimiento sub-simbólico que se ha adquirido mediante descubrimiento en el nivel emergente, de esta forma, el conocimiento adquirido por el agente puede ser comunicado y transmitido de manera comprensible.

Hasta el momento se han definido los tipos de aprendizaje que coordina el enfoque epigenético de METÁFORA, no obstante, es necesario hacer hincapié en que la naturaleza en sí del aprendizaje es adaptativa, por lo cual, éste puede apreciarse igualmente en los niveles ontogenético y filogenético de la arquitectura en forma de *adaptación evolutiva*. Por ejemplo, el nivel ontogenético se enfoca en el desarrollo adaptativo de las estructuras cognitivas de orden superior (*Comportamientos*, Redes de Comportamientos, Planes, etc.), y este desarrollo adaptativo implica un aprendizaje a largo plazo acerca de "cuáles han sido las estructuras cognitivas mejor adaptadas". De manera similar, el proceso de evolución de memes en el nivel filogenético implica un aprendizaje a largo plazo de "los memes mejor adaptados".

6.3.2.2. Nivel Ontogenético

La contextualización de la perspectiva ontogenética en la arquitectura se evidencia en el desarrollo adaptativo de sus estructuras flexibles. Debido a que METÁFORA posee un conjunto de estructuras cognitivas jerárquicas (tales como los *Comportamientos*, las Redes de Comportamientos, los Planes, etc.) cuya configuración estructural varía según el contexto ambiental donde se diseñe la concreción de la arquitectura, surge la necesidad de definir un mecanismo evolutivo que determine adaptativamente, la manera más adecuada, eficiente, y robusta de desarrollar dichas estructuras cognitivas durante el ciclo de interacción del agente con el entorno, o en otras palabras, un mecanismo que se encargue de coordinar los procesos de adaptación de las estructuras cognitivas durante el desarrollo del agente. Concretamente, el proceso ontogenético define los procesos de desarrollo funcional y estructural de dos tipos de pautas cognitivas de orden superior: las redes de comportamientos y los planes. A continuación se describe grosso modo el proceso de desarrollo de estas dos estructuras cognitivas.

Las diversas dinámicas de cooperación y competencia existente entre los *Comportamientos* de la arquitectura, da origen a una estructura conexionista llamada la Red de

Comportamientos (RC). La RC se fundamenta en el modelo de dispersión de energía de activación propuesto por (Maes, 1989). Esta red define cómo se relacionan los *Comportamientos* a través de enlaces excitativos que promueven la activación de otros *Comportamientos* en el siguiente ciclo cognitivo, y de enlaces inhibitorios que suprimen incrementalmente dicha activación. Intuitivamente, la idea principal es que los *Comportamientos* empleen estos enlaces para activarse e inhibirse mutuamente de tal manera que, después de cierto tiempo, la energía de activación se acumule en los *Comportamientos* que representen las "mejores" secuencias de acciones que se puedan ejecutar, dado un estado y unas metas actuales. Una vez el nivel de activación de un *Comportamiento* sobrepase cierto umbral, éste se convierte en un *Comportamiento* activo y puede ejecutar acciones reales.

Cabe resaltar en este punto, que la activación de los *Comportamientos* en la RC no puede ser representada por un modelo lineal o determinista, sino que contrariamente, corresponde a un modelo no-lineal donde la secuencia de activación de *Comportamientos* no puede ser inferida desde un estado inicial hasta el logro de una meta específica. Por tanto, la RC debe demostrar cierta capacidad de adaptabilidad a las condiciones ambientales, sin embargo, las conexiones de esta red necesitan ser pre-cableadas y el ajuste de los parámetros es estático (Dorer, 2005). Se propone de esta manera, una extensión al modelo de Maes que se centra en un desarrollo ontogenético basado en computación evolutiva, más precisamente en un algoritmo de Programación de Expresiones Genéticas (PEG) (Ferreira, 2001), el cual evoluciona, en tiempo de ejecución, las características funcionales y estructurales de cada RC. Los aspectos fundamentales en los que se enfoca la extensión propuesta son:

- La red típica de Maes no mantiene un registro de sus dinámicas de activación pasadas. De manera que, un error de activación en un *Comportamiento* que afecta la planificación puede cometerse reiteradamente en el mismo plan, impidiendo en la mayoría de los casos que el agente pueda alcanzar la meta actual. El desarrollo ontogenético permitirá que la red auto-repare y reasigne los enlaces entre los *Comportamientos*, favoreciendo la rápida convergencia y evitando de esta manera los bucles infinitos, los cuellos de botella y los errores en la planificación que produce la red.
- En el modelo de Maes es necesario especificar a priori el conjunto de parámetros globales que caracteriza la red, lo cual produce dos desventajas: la primera es que se requiere de un diseño predeterminado, y la segunda, que el sistema es más propenso a caer en soluciones locales. En este punto surge un aspecto destacable que aporta la presente tesis en la extensión del modelo de Maes: el desarrollo ontogenético asumirá la responsabilidad de ajustar adaptativamente los parámetros de configuración de cada red de comportamientos[8], de acuerdo con la orientación e intencionalidad del agente (e.g., RCs orientadas a metas, RCs orientadas a la situación actual, RCs deliberativas, RCs reactivas, etc.). Este ajuste adaptativo permitiría encontrar máximos globales en el espacio de búsqueda (e.g., una secuencia

[8]Se asume un conjunto de redes de comportamientos que forman el acervo genético donde opera el enfoque ontogenético a través de procesos darwinianos de selección, replica, cruce y mutación, los cuales son coordinados por una Máquina Darwiniana.

de *Comportamientos* que optimice el número de pasos y de recursos empleados en la ejecución de un plan).

- El modelo propuesto por Maes requiere de un diseño a priori donde se determinan los enlaces excitativos, inhibitorios y de conflicto entre los *Comportamientos* que la constituyen. El diseño de esta red puede aumentar en complejidad exponencial en la medida que se incluyan nuevos *Comportamientos* y nuevas relaciones entre los mismos (Dorer, 2005). Así pues, el modelo típico de una RC no incorpora un mecanismo de "aprendizaje" que determine durante la propia interacción, la manera de realizar cambios estructurales en la red sin que ésta pierda su organización. Considerando este punto, el enfoque ontogenético que se propone, permite que cada red se auto-organice a través de procesos evolutivos internos dirigidos por el algoritmo PEG, donde se establecen dichos enlaces de manera adaptativa y sin requerirse un diseño a priori. Como consecuencia, la estructura de la red se auto-define y se optimiza en el tiempo.

- El enfoque ontogenético considera que el sistema puede estar determinado por cierto nivel de "inercia" con respecto a los cambios del entorno y de las metas. De esta forma, en entornos altamente dinámicos, es necesario que el modelo adaptativo ajuste lentamente los parámetros globales de las RCs o de lo contrario éstas nunca alcanzarán las metas a largo plazo ni sobrellevar los estados actuales que requieren respuestas reactivas.

- El enfoque ontogenético permite que los nuevos *Comportamientos* que se incorporan en el reportorio del agente sean rápidamente conectados e integrados a las dinámicas de dispersión de activación de la red. Así mismo, las referencias a *Comportamientos* desconectados son rápidamente actualizadas.

- El proceso adaptativo guiado por el enfoque ontogenético es "robusto", debido a que las RCs siempre activan el *Comportamiento* más apropiado (o el menos inadecuado), incluso en presencia de enlaces de conflicto entre los *Comportamientos*, o en casos donde la topología de la red requiere optimización. En cualquiera de los casos anteriores, la red se adapta y re-configura los enlaces entre los *Comportamientos*.

- En el modelo típico de Maes, el proceso de selección de acciones en cada comportamiento es menos "racional" que el producido por los sofisticados planificadores deliberativos construidos por la IA clásica (Maes, 1989). Este aspecto es mejorado considerablemente en METÁFORA, a través de la utilización de componentes híbridos (los *Comportamientos*) los cuales encapsulan diferentes técnicas de razonamiento, representación del conocimiento y aprendizaje (siguiendo el principio de implementación múltiple), y que poseen un desarrollo epigenético independiente, tal como se describió en la sección anterior.

- En biología, uno de los aspectos característicos de la ontogenesis es que define los procesos que generan la diversidad celular (diferenciación). En METÁFORA, el enfoque ontogenético produce una diferenciación cualitativa en los *Comportamientos* asignándoles una o varias metas a alcanzar, un conjunto de premisas (precondiciones) que los activan, y un conjunto de enlaces excitativos e inhibitorios con otros *Comportamientos* a través de los cuales cooperan y compiten entre sí.

El otro proceso de naturaleza ontogenética en METÁFORA, es la construcción de estructuras cognitivas de orden superior: los Planes. De manera similar al proceso ontogenético que genera RCs a partir del conjunto de *Comportamientos* del agente, los Planes son generados por la composición de RCs que interactúan entre sí mediante relaciones que generan una jerarquía de ejecución. Estos planes, se representan de forma similar a las Redes de Tareas Jerárquicas (RTJ) donde las dependencias entre las acciones se construyen en forma de redes (Sacerdoti, 1977; Tate, 1990; Wilkins, 1988).

En la planificación con RTJ, se obtienen tareas complejas mediante la *descomposición* de éstas en sub-tareas más simples. El proceso de planificación se encarga de descomponer recursivamente dichas tareas en tareas más simples, hasta que son generadas secuencias primitivas de acciones. Estas acciones forman un plan que realiza las tareas de alto nivel.

En el enfoque ontogenético de METÁFORA, este proceso se ejecuta de forma inversa: las acciones primitivas generadas por cada RP, RES, RNA y SIA compiten y cooperan para formar un *Comportamiento* especializado. Los *Comportamientos*, a través de las dinámicas de dispersión de energía de activación entre los mismos, forman las RCs. Éstas a su vez, compiten y cooperan para formar estructuras de mayor nivel de complejidad, los Planes como Redes de Tareas. Así pues, las estructuras de alto nivel se forman por la *composición* emergente de estructuras subyacentes más simples y auto-organizadas. Así pues, lo que merece especial atención en este punto, es entender que el proceso ontogenético en METÁFORA puede generar estructuras cognitivas complejas a partir de otras más simples, a través de la emergencia de diversos niveles auto-organizativos y jerárquicos.

Finalmente, se desea destacar la influencia de los procesos darwinianos dirigidos por la aproximación ontogenética, en los procesos prospectivos y anticipatorios de METÁFORA. Cada estructura cognitiva compleja generada por los procesos de selección, replicación, recombinación y mutación, es considerada como una alternativa de actuación en el mundo donde se encuentra inmerso el agente. Cada una de estas estructuras cognitivas es ensayada a través de un mecanismo de simulación interna, donde compiten entre sí por "ganar el derecho" de ser ejecutadas en el ambiente. Como consecuencia, la ejecución iterativa de este procedimiento le concede a METÁFORA la capacidad de generar planes deliberativos y reactivos que se anticipan a los hechos que pueden acontecer en el futuro, lo cual permite al agente actuar de forma coherente e inteligente.

6.3.2.3. Nivel Filogenético

En biología, la filogénesis estudia cómo funcionan y evolucionan los genomas a través de los siglos, produciendo una amplia diversidad de especies animales. El proceso filogenético ha sido un área de modelado computacional dirigido, principalmente, hacia el logro de un entendimiento más profundo acerca de las relaciones evolutivas entre los animales, y la manera en que las condiciones ambientales guían dicho proceso evolutivo (Symeonidis y Tollis, 2004; Kosiol et al., 2006; Arvelakis et al., 2005). Usualmente, los estudios computacionales especifican los modelos parametrizados de cada escalón evolutivo, y luego intentan identificar los valores de los parámetros que mejor explican un

conjunto de datos biológicos observados, con el propósito de formular conclusiones relativas a los procesos evolutivos naturales (Kosiol et al., 2006; Bohl y Lancaster, 2006; Jensen y Pedersen, 2000).

En contraste con lo anterior, la aproximación arquitectónica no se centra en el modelado de las dinámicas naturales que son guiadas por la filogénesis ni en la explicación del fenómeno relacionado. Por el contrario, los procesos filogenéticos naturales sirven de inspiración en el diseño de un mecanismo que es capaz de evolucionar computacionalmente las estructuras cognitivas de METÁFORA. Específicamente, se emplea un homólogo simplificado de la filogénesis conocido, con el nombre de "computación evolutiva", como un mecanismo de optimización capaz de diseñar efectivamente estructuras computacionales complejas (Roggen et al., 2003; Surmann y Maniadakis, 2001; Rovithakis et al., 2004). A continuación, se justifica la conveniencia de las técnicas evolutivas como soporte a la configuración de sistemas cognitivos artificiales inspirados en el funcionamiento del cerebro.

La aproximación filogenética se enfoca en dos aspectos del diseño adaptativo de METÁFORA: la optimización de parámetros globales y la evolución de estructuras cognitivas. La justificación del primer aspecto filogenético viene dada por el hecho de que el ajuste "manual y predeterminado" de todos los parámetros globales de la arquitectura a través de procesos de prueba y error, resulta ser una tarea tanto complicada como restrictiva, ya que elimina la capacidad de la arquitectura para adaptarse dinámicamente a los cambios ambientales. Como soporte a esta iniciativa, los métodos de la computación evolutiva facilitan la estimación de estos valores de los parámetros, consiguiendo configuraciones cercanas a la optimalidad global que se puede obtener para este tipo de sistemas altamente no-lineales.

Por otro lado, la justificación del segundo aspecto viene dada por la afirmación de que una aproximación evolutiva incremental podría resultar más prometedora hacia el desarrollo exitoso de mecanismos y estructuras cognitivas artificiales. Teniendo en cuenta esta aproximación, existirán casos en los que las estructuras cognitivas parciales de METÁFORA necesitan ser re-diseñadas en la medida que son integradas gradualmente con nuevas sub-estructuras en el modelo. Las técnicas de computación evolutiva son particularmente apropiadas para este fin, debido a que pueden adaptarse fácilmente a las condiciones operacionales cambiantes (DeJong, 1992). Además, en cada paso de re-diseño, son capaces de evitar que el sistema caiga en los óptimos locales, mediante la estimación de una nueva solución eficiente la cual cumple con el conjunto de objetivos incrementados. Este enfoque de adaptación filogenética en METÁFORA se efectúa a través de un proceso co-evolutivo dirigido por métodos de programación evolutiva.

Los psicólogos evolucionistas han propuesto ya con anterioridad la co-evolución de áreas cerebrales parciales durante el proceso evolutivo humano (Klein et al., 2002; Whiting y Barton, 2003). La analogía computacional con esta hipótesis es que la aproximación co-evolutiva puede extenderse aún más hacia una co-evolución cooperativa. Afortunadamente, al adoptar esta aproximación, las estructuras cognitivas de la arquitectura pueden especificarse de manera más apropiada, debido a que el proceso de exploración es confinado a un dominio más pequeño el cual permite una búsqueda más acotada.

Considerando que, un cromosoma[9] encapsula las distintas estructuras cognitivas –desde reglas primitivas de actuación hasta planes deliberativos– en forma de secuencias de memes y memeplexes[10], resulta evidente que, la aproximación co-evolutiva puede ser formulada de forma apropiada mediante la separación, probablemente una por cada gen del cromosoma, de las "funciones de aptitud" que guiarán el diseño de cada estructura cognitiva (enfoque multimodal). Esto contrasta con el proceso evolutivo unimodal que exige una sola función de aptitud para todo el cromosoma, impidiendo que se considere el desempeño mismo de cada estructura cognitiva. En otras palabras, la co-evolución provee facilidades para el diseño avanzado de componentes distribuidos debido a que se puede enfatizar explícitamente en las características independientes de las sub-estructuras cognitivas (mediante funciones de aptitud individuales) e implementar al mismo tiempo sus acoplamientos en un sistema unificado: el cromosoma.

Específicamente, la aproximación co-evolutiva exhibe las siguientes propiedades (Potter y DeJong, 2000), las cuales son significativas para el diseño de mecanismos y estructuras cognitivas:

- Facilita el diseño de sistemas distribuidos aproximando el problema al nivel de estructuras parciales utilizadas como bloques de construcción,
- Permite destacar los distintos roles de las estructuras parciales en un modelo compuesto o jerárquico,
- Exige la cooperación entre los componentes del sistema. En METÁFORA, esta cooperación se da entre cada estructura cognitiva que compone una jerarquía de estructuras más complejas,
- Es independiente de los procesos computacionales empleados para desarrollar cada sub-estructura del mecanismo cognitivo,
- Facilita la evolución simultánea de diferentes estructuras computacionales (estructuras cognitivas), cada una exhibiendo las características deseadas para representar un sub-sistema en particular del sistema cognitivo.

Aún cuando la metodología co-evolutiva parece ser apropiada para abordar el problema investigado en el presente modelo arquitectónico, las aproximaciones convencionales no permiten co-evolucionar un gran número de componentes cooperativos y organizados jerárquicamente como es el caso de las estructuras cognitivas de METÁFORA. Este problema, es resuelto por un lado, mediante una clasificación de los componentes en grupos con objetivos comunes, como se propone en (Maniadakis, 2006), los cuales son co-evolucionados en la misma dirección y, por otro, mediante un algoritmo de computación evolutiva que permita cierta flexibilidad de representación de dichos grupos, en una estructura cromosómica, en este caso, la Programación de Expresiones Genéticas (PEG)

[9]En la teoría de la computación evolutiva, cada cromosoma codifica, en un alfabeto determinado, una solución específica que representa un punto en el espacio de soluciones.

[10]La mayor parte del estudio de los memes se enfoca en los grupos de memes llamados memes complejos, o "memeplexes". Similar al comportamiento de los genes complejos identificados por la biología, los memeplexes se encuentran presentes con frecuencia en el mismo individuo. Aplicando la teoría universal del darwinismo, los grupos de memeplexes se agrupan debido a que de esta manera, los memes pueden copiarse a sí mismos con mayor éxito cuando se encuentran cooperando en equipo (Dawkins, 1992).

(Ferreira, 2001). Estos grupos pueden ser luego combinados, representando así gradualmente las estructuras cognitivas más complejas de la arquitectura, y la combinación de estos grupos puede ser luego evolucionada por medio de un proceso de nivel superior. Siguiendo esta aproximación, el mecanismo co-evolutivo cooperativo es organizado de manera jerárquica, con el fin de facilitar la solución de problemas de diseño complejo.

En síntesis, METÁFORA define un mecanismo co-evolutivo que combina, de forma "cooperativa", cada tipo de estructura cognitiva jerárquica: Reglas, *Comportamientos*, Redes de Comportamientos y Planes. En este punto, es necesario enfatizar que cada una de estas estructuras cognitivas es representada mediante un meme o un memeplexe (en el caso de las estructuras compuestas), ya que de esta manera pueden ser tratadas análogamente a como si fuesen genes sobre los cuales operan la clonación, la recombinación y la mutación.

6.3.2.4. Integración de los Niveles Epigenético, Ontogenético y Filogenético

En la figura 6.7 se esquematiza el modelo ascendente (bottom-up) de las estructuras cognitivas en METÁFORA y cómo éstas forman jerárquicamente, estructuras más complejas y robustas a través de los procesos Epigenéticos, Ontogenéticos y Filogenéticos (EOF).

En el nivel inferior, la configuración epigenética se encarga de definir dos tipos de estructuras cognitivas: los *Comportamientos* Genéricos y los *Comportamientos* Contextuales. La diferencia básica entre estos dos tipos de *Comportamientos*, es que los *Comportamientos* genéricos funcionan como plantillas que definen un conjunto de precondiciones de activación, y un conjunto de efectos esperados y no esperados que se suponen serán alcanzados o no, después de la ejecución del *Comportamiento*; mientras que, los *Comportamientos* contextuales son instancias de un *Comportamiento* genérico específico que definen además restricciones propias de su implementación (e.g., enlaces con otros *Comportamientos* contextuales, y tipos concretos de elementos sensoriales o motivacionales adicionales a los que define la plantilla).

Por ejemplo, supóngase el *Comportamiento* genérico ⟨`recoger-objeto`⟩, el cual puede tener dos *Comportamientos* contextuales: ⟨`recoger-bolígrafo`⟩ y ⟨`recoger-televisor`⟩. Estos dos *Comportamientos* contextuales pueden emplear el mismo conjunto de reglas y heurísticas para recoger el objeto que define cada uno, sin embargo, son activados cada uno por diferentes tipos de estímulos y cada uno puede definir distintos tipos de acciones, restricciones y señales de refuerzo (retroalimentación) – el *Comportamiento* ⟨`recoger-televisor`⟩ empleará acciones que implican una mayor fuerza de ejecución que las empleadas por el *Comportamiento* ⟨`recoger-bolígrafo`⟩ que requerirá probablemente acciones más precisas.

En el nivel ontogenético, se definen estructuras cognitivas de mayor complejidad, las RCs y los Planes. Las RCs definen topologías de nodos (*Comportamientos*) interconectados, los cuales cooperan entre sí para cumplir con una o varias *Metas*. A partir de estas RCs se forman estructuras cognitivas compuestas, los Planes, que se estructuran en forma de Redes de Tareas Jerárquicas, como se explicó en el apartado 6.3.2.2. Finalmente, estas estructuras cognitivas (*Comportamientos*, RCs y Planes), son enviadas al nivel filo-

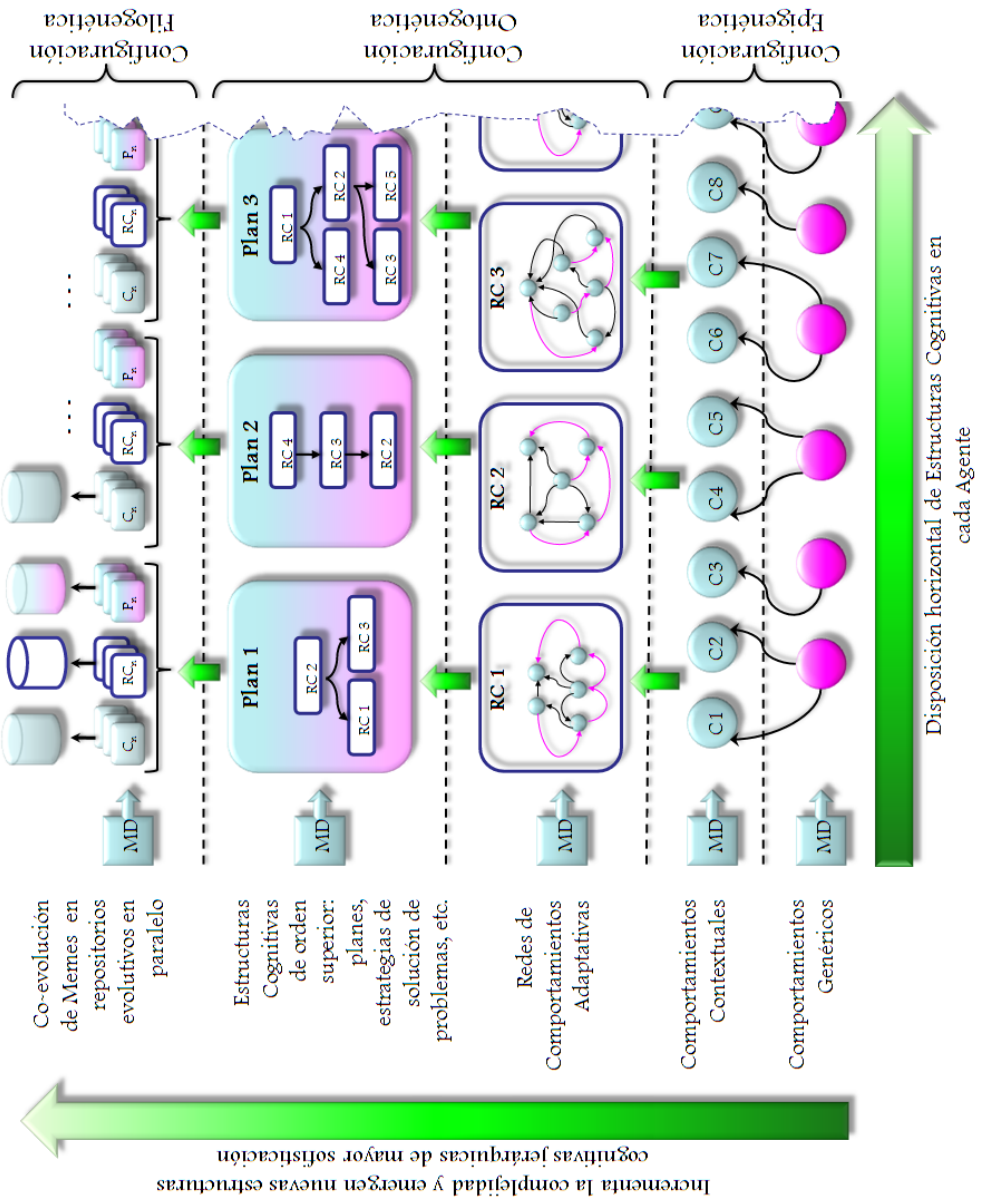


Figura 6.7: Modelo Ascendente (Bottom-Up) de la Arquitectura. Donde $\{C_1, C_2 \dots C_n\}$ corresponde al conjunto de *Comportamientos* del agente; $\{RC_1, RC_2 \dots RC_n\}$ corresponde al conjunto de Redes de Comportamientos que define cada agente; $\{P_1, P_2 \dots P_n\}$ corresponde a la librería de planes que construye cada agente; y MD son las Máquinas de Máquinas Darwinianas empleadas en cada nivel adaptativo de METÁFORA.

genético, donde son tratadas como Memes y Memeplexes, y es allí donde se efectúan los procesos de co-evolución cooperativa entre dichas estructuras.

Lo que cabe destacar en este punto, es que cada nivel jerárquico de la arquitectura aporta mayor complejidad al siguiente nivel superior, formando de esta manera, estructuras cognitivas de orden superior. Los procesos de epigénesis, ontogénesis y filogénesis,

dirigen el desarrollo y la evolución de estas estructuras cognitivas a través de las "Máquinas Darwinianas" (descritas en 4.4.3.2). Estas Máquinas Darwinianas, presentes en cada nivel jerárquico de desarrollo y evolución de la arquitectura, determinan pues, los procesos darwinianos que operan sobre las pautas y estructuras cognitivas a diferentes escalas temporales, permitiendo así diferentes tipos de actuación en el agente: respuestas reactivas determinadas por el nivel epigenético (en la escala de milisegundos), actuaciones deliberativas y razonamiento prospectivo modulado por el nivel ontogenético (en la escala de segundos), y transmisión de estructuras cognitivas refinadas (memes) dirigida por el nivel filogenético (en la escala de minutos o en el largo plazo).

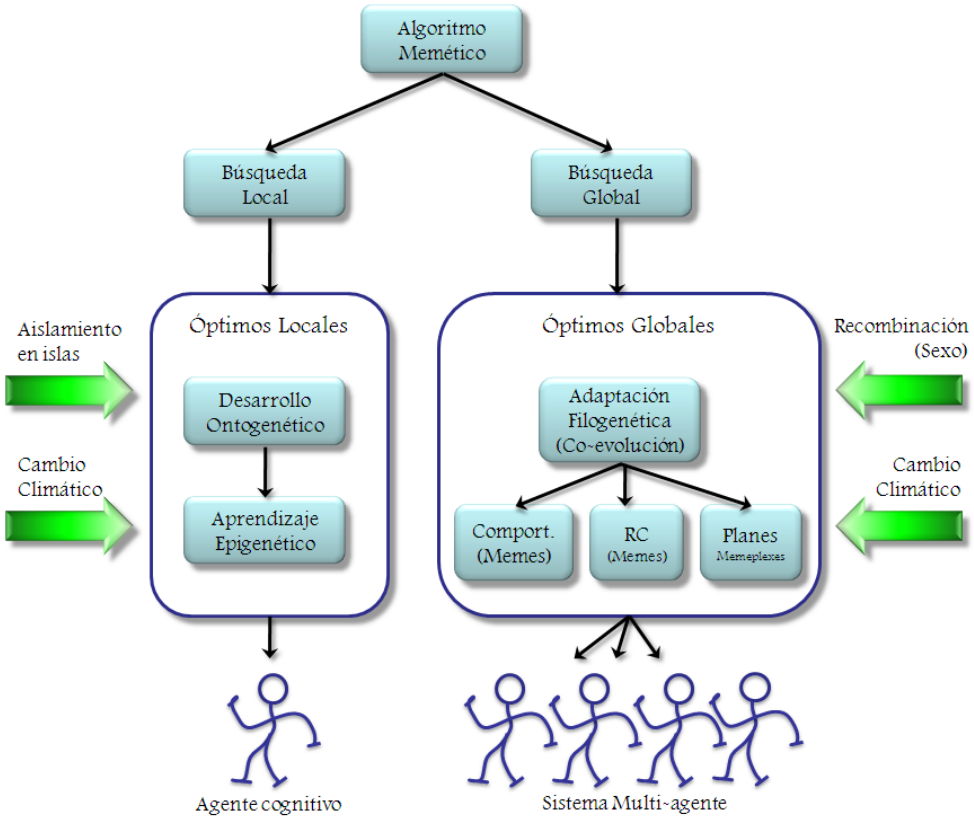


Figura 6.8: Integración de las aproximaciones Epigenética, Ontogenética y Filogenética a través de un Algoritmo Memético.

Una vez definida la organización ascendente de los niveles adaptativos de la arquitectura, es posible pasar a una visión más integradora de estas tres aproximaciones de inspiración biológica. Para esto, obsérvese primero la figura 6.8. En una escala de adaptación más alta, METÁFORA define un Algoritmo Memético (AM) encargado de modular la integración entre los procesos EOF. El AM, a través de la combinación sinérgica de meta-heurísticas basadas tanto en poblaciones como en búsquedas locales, se encarga de evolucionar el conjunto de memes o "individuos". Acá el término individuo, se hereda de la jerga empleada en los Algoritmos Evolutivos, para denominar cada solución tentativa que manipula el proceso evolutivo. En el caso de METÁFORA, cada uno de estos individuos corresponde a un tipo específico de estructura cognitiva o meme.

El proceso evolutivo dirigido por el AM comprende dos partes: la primera es una Búsqueda Local, la cual en gran medida es desempeñada por los procesos epigenético y ontogenético, donde cada estructura cognitiva es aprendida, refinada continuamente y puesta a prueba en tiempo real a través de la interacción del agente con su entorno; la segunda es la Meta-heurística Poblacional o Búsqueda Global, que en este caso será un mecanismo filogenético dirigido por programación evolutiva, más concretamente, mediante la Programación de Expresiones Genéticas (PEG) (Ferreira, 2001).

Así pues, la búsqueda local llevada a cabo por cada agente a través de sus procesos epigenéticos y ontogenéticos, es una estrategia "autónoma" que encuentra óptimos locales en el espacio de búsqueda. De esta manera, el AM puede verse como una colección de agentes que realizan una exploración autónoma del espacio de búsqueda, cooperando en ocasiones a través de la recombinación, y compitiendo por recursos computacionales a través de los mecanismos de selección y reemplazo. En cuanto a la búsqueda global, el proceso filogenético, guiado por el mecanismo co-evolutivo, produce óptimos globales al combinar las soluciones locales (memes y memeplexes) encontradas por cada agente durante su interacción con el entorno.

Adicionalmente, el AM incorpora tres aceleradores evolutivos descritos en la sección 4.4.3.2: el sexo, el cambio climático y el aislamiento en islas. El concepto de sexo en el proceso adaptativo, se vería reflejado por la recombinación genética entre dos instancias de un tipo de estructura cognitiva o meme específico; el cambio climático por su parte, se aplica cada vez que los memes son trasladados de un contexto a otro, permitiendo así crear nuevas generalizaciones acerca de cuándo y cómo aplicar dicho meme; y el aislamiento en islas se presenta en la medida en que cada agente encapsula y especializa la construcción de sus propias estructuras cognitivas acelerando la evolución de las mismas.

Así pues, METÁFORA define distintos tipos de aislamiento que aceleran la evolución de las estructuras cognitivas:

- Los *Comportamientos*: cada uno de los *Comportamientos* agrupa y especializa un conjunto de reglas y heurísticas controladas por los diferentes sub-componentes (RNA, SIA, RES y RP). De esta manera, las reglas evolucionan más rápidamente al estar acotado el espacio de búsqueda y no tener que "aprender" a identificar relaciones causales en un espacio de búsqueda más amplio. Por ejemplo, el *Comportamiento* ⟨`recoger-bolígrafo`⟩ sólo se enfoca en aprender y dirigir el descubrimiento de reglas y heurísticas que le permiten al agente recoger el objeto bolígrafo, y no por el contrario, de reglas y heurísticas para ⟨`abrir-la-puerta`⟩.
- Las Redes de Comportamientos y los Planes: estas dos estructuras cognitivas también evolucionan en repositorios separados, dependiendo del conjunto de metas asociadas. De esta manera, las RC y los planes se especializan y cumplen una función específica. En el proceso evolutivo de estas estructuras, sólo compiten entre sí instancias del mismo tipo, evitando por ejemplo, que un plan para ⟨`conducir-hasta-el-trabajo`⟩ compita con un plan para ⟨`preparar-la-cena`⟩.
- El Agente: el propio agente, como se mencionó anteriormente, cumple la función de aislamiento debido a que el agente puede desarrollar las capacidades cognitivas que sólo deriven de su interacción con el entorno. Es evidente que, otro agente

situado en otro entorno o en el mismo pero en otra ubicación, tendrá experiencias diferentes al primero y, como consecuencia, construirá estructuras cognitivas diferentes. Es por esto que resulta importante modelar una "transmisión" cultural o un intercambio de estructuras cognitivas entre agentes a través del proceso co-evolutivo que articula el enfoque filogenético.

6.3.3. Dinámicas del Sistema Nervioso Sintético

Inspirado en los trabajos previos de (Minsky, 1988; Maes, 1992; Breazeal, 2000), el sistema nervioso sintético (SNS) del METÁFORA es implementado como como una red altamente distribuida compuesta por elementos interactuantes. Cada elemento computacional (o nodo) recibe un conjunto de mensajes que provienen de aquellos otros elementos que están conectados a sus entradas, realizando algún tipo de procesamiento específico basado en dichos mensajes, y enviando luego los resultados a aquellos elementos conectados a sus salidas. Los elementos se conectan para formar redes, y las redes se conectan para formar los sub-sistemas del SNS. A continuación se describen los principios básicos que gobiernan el comportamiento de los elementos del SNS de METÁFORA.

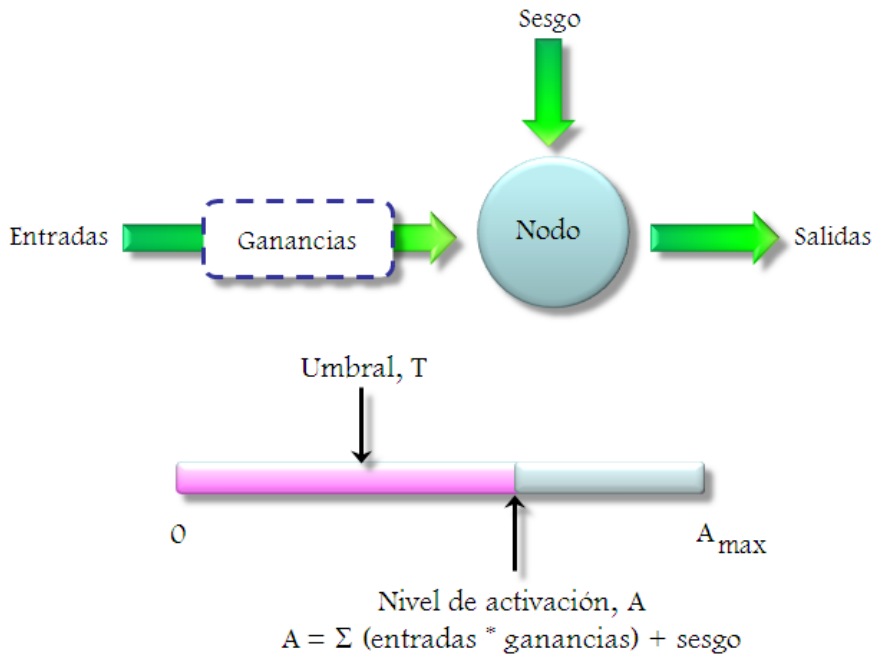


Figura 6.9: Esquema de un proceso computacional básico. El proceso se activa cuando el nivel de activación A excede el umbral T.

Unidades Básicas Computacionales: en la implementación de METÁFORA, cada *proceso computacional básico* es modelado como se presenta en la figura 6.9. El *nivel de activación* A del proceso es calculado por la siguiente ecuación:

$$A = (\sum_{n}^{j=1} w_j \times i_j) + b \tag{6.1}$$

donde: i_j son las entradas con valores enteros del nodo. w_j son los pesos de las entradas. b es el sesgo sobre el número de entradas n.

Los pesos de las entradas pueden ser negativos o positivos; un peso positivo corresponde a una conexión excitativa y un peso negativo corresponde a una conexión inhibitoria. Cada proceso es responsable de procesar su propio nivel de activación y de determinar cuándo puede hacerse activo. El nivel de activación de un proceso (nodo) es acumulativo, es decir, incrementan en valor en proporción a los aportes de energía que hacen las entradas de dicho proceso, utilizando para ello un bloque de construcción cibernético del tipo Integrador (como se mencionó en la la sección 4.5). El proceso se hace activo cuando su nivel de activación excede un *umbral de activación*. Cuando está activo, el proceso puede enviar energía de activación a otros nodos a fin de incrementar su nivel activación. Éste puede además realizar cierto procesamiento especial, enviar mensajes de salida a los procesos conectados, y/o expresarse mediante acciones motoras que son enviadas a los actuadores del agente. Cada *percepto*, *unidad de memoria de trabajo*, *neurona*, *regla simbólica* (procedimental, semántica, y episódica), *Comportamiento*, red de comportamientos, *motivador*, *emoción primaria*, etc., es modelada como un tipo diferente de proceso computacional básico que es ajustado al rol específico que cumple dentro de la arquitectura cognitiva. Así pues, a pesar de que difieren en funcionalidad, todos ellos siguen el mismo esquema básico de activación.

Redes de Unidades: las unidades básicas se conectan cooperativamente para formar redes de procesos interactuantes que permiten la emergencia de procesos cognitivos más complejos. Esto implica la conexión de las salidas de una unidad con las entradas de otras unidades. Cuando una unidad está activa, además de comunicar los mensajes a sus unidades conectadas, puede también comunicar su propia energía de activación. Esto es lo que se conoce con el nombre de *dispersión de la activación*, el cual es un mecanismo que puede influir sobre la activación o inhibición de otras unidades (Maes, 1992). Este mecanismo fue originalmente conceptualizado por (Lorenz, 1973) en su Modelo Hidráulico. Minsky (Minsky, 1988) emplea un esquema similar en sus ideas acerca de la formación de las memorias de largo plazo.

Sub-sistemas de redes: los grupos de redes interconectadas formas los sub-sistemas. Dentro de cada sub-sistema, los nodos activos desempeñan procesamientos especiales par llevar a cabo las tareas de dicho sub-sistema. Para realizar esto, los mensajes que son comunicados entre y dentro de tales redes deben compartir una "moneda común", de manera que la información contenida en los mensajes pueda ser procesada y combinada de una forma específica (McFarland y Bosser, 1993). Adicionalmente, en la medida

que el sub-sistema se hace más complejo, es posible que algunos agentes puedan entrar en conflicto con otros (e.g., cuando compiten por recursos compartidos). En este caso, los agentes deben tener algunos medios para competir contra los demás. Si cada agente procesa su relevancia en términos de una moneda compartida, los agentes en conflicto pueden competir basándose en dicho valor.

Moneda Común: ésta plantea un aspecto importante respecto a la comunicación dentro y entre los diferentes sub-sistemas. El comportamiento observable del agente es un producto de muchos procesos interactuantes. Por ejemplo, la etología, la psicología comparativa, y la neurociencia, han demostrado que el comportamiento observable está bajo la influencia de factores internos (i.e., motivadores, experiencia pasada, etc.) así como por factores externos (i.e., la percepción). Esto implica que los sub-sistemas pueden comunicarse e influenciarse mutuamente a pesar de sus diferentes funciones y modos de procesamiento. Esto ha permitido a etologistas tales como (McFarland y Bosser, 1993) y (Lorenz, 1973), el proponer que debería existir una moneda común que pueda ser compartida por los sub-sistemas perceptual, motivacional, declarativo, y comportamental.

En este esquema, el sub-sistema perceptual genera los valores basándose en los estímulos ambientales (perceptos), y los sub-sistemas motivacional y declarativo generan los valores basándose en sus factores internos (i.e., motivadores y asociaciones locales respectivamente). En METÁFORA, estos valores serán representados por unidades comunes (monedas) de la forma ⟨`propiedad, valor`⟩. Estos conjuntos de valores son comunicados luego al sub-sistema comportamental, donde los *Comportamientos* los emplean para procesar su relevancia y poder así competir, basados en este valor, por obtener el foco de control. Internamente cada sub-sistema puede operar independientemente sobre sus propias monedas. Este es el caso del módulo Comportamental (capítulo 7), el módulo Motivacional (sección 8.4), y el módulo de Orquestación de la Consciencia (capítulo 9) de METÁFORA. Sin embargo, la moneda que es comunicada entre los diferentes sistemas debe ser también compartida (y por tanto, este es el propósito de definir unidades comunes del tipo ⟨`propiedad, valor`⟩ que pueden ser compartidas entre los diferentes sub-sistemas de la arquitectura).

SNS basado en Valores: tomando como fundamento esta idea, el SNS del agente es implementado como un sistema basado en valores. Esto simplemente significa que cada proceso computa los valores numéricos (representados en una moneda común) correspondientes a sus entradas. Estos valores son comunicados como mensajes (o como energía de activación) a través de la red, bien sea dentro de un sub-sistema o entre subsistemas. Conceptualmente, la magnitud de un valor representa la intensidad de la contribución en términos de su influencia sobre otros procesos. Al emplear una aproximación basada en valores se obtiene el efecto elegante de permitir que el nivel de influencia pueda ser graduado en intensidad, en lugar de simplemente corresponder a un valor de "activado" o "desactivado" (Breazeal, 2000). Otros procesos computan su relevancia basándose para ello en la entrada de energía de activación o en los mensajes, y emplean su nivel de activación para competir contra otros y poder ejercer su influencia sobre el SNS.

6.4. Aproximación Cibernética al Diseño de la Arquitectura

Esta sección presenta un diseño preliminar de la arquitectura en términos de las relaciones de causa y efecto, los bucles de retroalimentación y los flujos de control entre sus componentes, y no en términos de la distribución física de los mismos.

Implícito en la estructura jerárquica está el concepto de que METÁFORA, a cualquier nivel dado, controla el funcionamiento de sus distintos sub-sistemas subyacentes. Este control está basado en la optimización de un índice de funcionamiento del sistema combinado. Por desgracia, tanto en los sistemas biológicos como en los artificiales, es mucho más difícil deducir cuantitativamente los índices pertinentes de funcionamiento en los distintos niveles jerárquicos, como los descritos en la sección 6.3.1, por ejemplo, que aceptar el concepto cualitativo (Milsum y Bertalanffy, 1978).

Cuando se habla de control jerárquico en METÁFORA, no se habla de la coordinación centralizada de las dinámicas y procesos entre los componentes internos en cada nivel de organización –ya que estos componentes se proponen como elementos distribuidos encargados de definir su propia organización, es decir, son auto-organizativos– sino que, contrariamente, se habla de la coordinación de las dinámicas de activación a un macro nivel, es decir, un nivel jerárquico dado controla la activación de los sub-sistemas que se encuentran en niveles inferiores. Por ejemplo, una RC define qué *Comportamientos* serán activados de acuerdo con el estado actual del agente, pero no determina las dinámicas internas del *Comportamiento* relativas a la selección de reglas, heurísticas, aprendizaje, y la ejecución de acciones.

En este punto, resulta necesario hacer una distinción entre (1) la optimización del *diseño* de las estructuras cognitivas a través del proceso evolutivo a largo plazo y (2) la optimización del *funcionamiento* de dichas estructuras que pueden cambiar significativa y frecuentemente durante el tiempo de "vida" o de interacción del agente con el entorno. Por supuesto, el diseño optimizado de una estructura cognitiva tiene lugar en el largo proceso evolutivo, pero su "rápida" capacidad de adaptación en una fracción de minuto durante la interacción entre el agente y su entorno la hace funcionalmente diferente.

Según la teoría de sistemas (Milsum y Bertalanffy, 1978), la optimización de un diseño biológico se denomina *estática* cuando se supone que ésta se mantiene constante en esencia durante el desarrollo del organismo. Sin embargo, a través de los cambios evolutivos ésta puede modificarse poco a poco durante la historia de las especies dadas. Este principio de optimización estática se refleja en METÁFORA, en los procesos filogenéticos que guían los cambios adaptativos históricos de las estructuras cognitivas a través de la evolución de los memes, tal como se explicó en el apartado 6.3.2.3. Por otro lado, la optimización *dinámica* durante el funcionamiento a través del control de retroalimentación, es articulada por los procesos epigenéticos y ontogenéticos de la arquitectura, como se describió en los apartados 6.3.2.1 y 6.3.2.2.

En la teoría de sistemas, el término optimización durante el funcionamiento se usa, debido a que puede ocurrir durante el tiempo real de vida de un sistema. Como tal, parece que implica la necesidad de un lazo de retroalimentación adaptativo sobre la marcha que sea más rico en información que el simplemente necesario para asegurar su homeostasis, y es por tanto, un control jerárquico. Más adelante se podrá apreciar cómo estos lazos de

retroalimentación adaptativa dirigen los procesos homeostáticos en cada nivel jerárquico auto-organizativo de METÁFORA.

En general, la retroalimentación negativa en cada nivel jerárquico de METÁFORA proporciona la base para el control homeóstatico y de optimización. Esto hace posible la comparación entre las condiciones reales y deseadas, que es característica del control convencional de *lazo cerrado*. Sin embargo, como se podrá apreciar más adelante, el proceso evolutivo es esencialmente de *lazo abierto* a un nivel, porque las estructuras cognitivas pueden adaptarse o no, en gran parte de acuerdo con la dotación de su "genotipo". Dicho con otras palabras, en su forma elemental, el estado queda establecido al principio (la configuración filogenética de cada agente) y la acción se desarrolla hacia el éxito o el fracaso, sin que durante el proceso se disponga de medidas de control. Sin embargo, y en contraste, se puede considerar con toda seguridad que el proceso evolutivo a mayor escala (en el siguiente nivel jerárquico de organización), comprende una acción de retroalimentación negativa.

A partir de estas consideraciones se puede definir operacionalmente una forma importante de control jerárquico: un *sistema de "reflejo" de lazo abierto*, junto con un *sistema de comprobación de lazo cerrado* a más largo plazo. El arco reflejo viene caracterizado por un receptor[11] el cual reconoce un estímulo en particular, seguido por la transmisión al módulo Efector que potencia una acción reactiva como respuesta. Si no se lleva a cabo una acción reactiva satisfactoria con esta transmisión de "lazo abierto", entonces entra en acción la retroalimentación, puesto que el estímulo seguirá existiendo para desencadenar otra acción de reflejo. Esto mismo se evidencia a otra escala superior, cuando el agente planifica una secuencia de movimientos o acciones como un control de lazo abierto, pero en el caso de que ésta no resulte ser lo suficientemente precisa o eficiente, se produce una acción correctora del plan, es decir, un control de lazo cerrado.

A nivel biológico, existe evidencia (Jerison, 1990) acerca de la posibilidad de relacionar la naturaleza pre-programadora del desarrollo de los reflejos neuro-musculares y de las acciones prácticas, con el desarrollo de los reflejos cognitivos, que pueden no desembocar en acciones inmediatas, y ser almacenados y elaborados, sólo neuronalmente, en los centros integrantes del sistema nervioso central. De este modo se puede considerar el conocimiento como un resultado de reflejos neuronales pre-programados, así como los procesos intelectuales del pensar, evaluar, y tomar decisiones. A continuación, se describe cómo los distintos flujos de control entre las jerarquías sucesivas de la arquitectura, desde los reflejos reactivos básicos hasta los reflejos cognitivos, son modelados en METÁFORA.

6.4.1. Diseño de bloques de construcción para el SIA

En la figura 6.10 se observa el diagrama de bloques para una de las Máquinas Autopoiéticas de primer orden: el Sistema Inmune Artificial (SIA). Intuitivamente, en la figura se puede apreciar una conexión lineal de componentes que forma un control de reflejos de lazo abierto: *Transductor-ad* $\Rightarrow$ *Modulador* $\Rightarrow$ *Transductor-da*.

[11] Este receptor puede ser un conocimiento innato generado por la configuración filogenética, o preprogramado en las reglas simbólicas del módulo de Producciones o en las memorias declarativas.

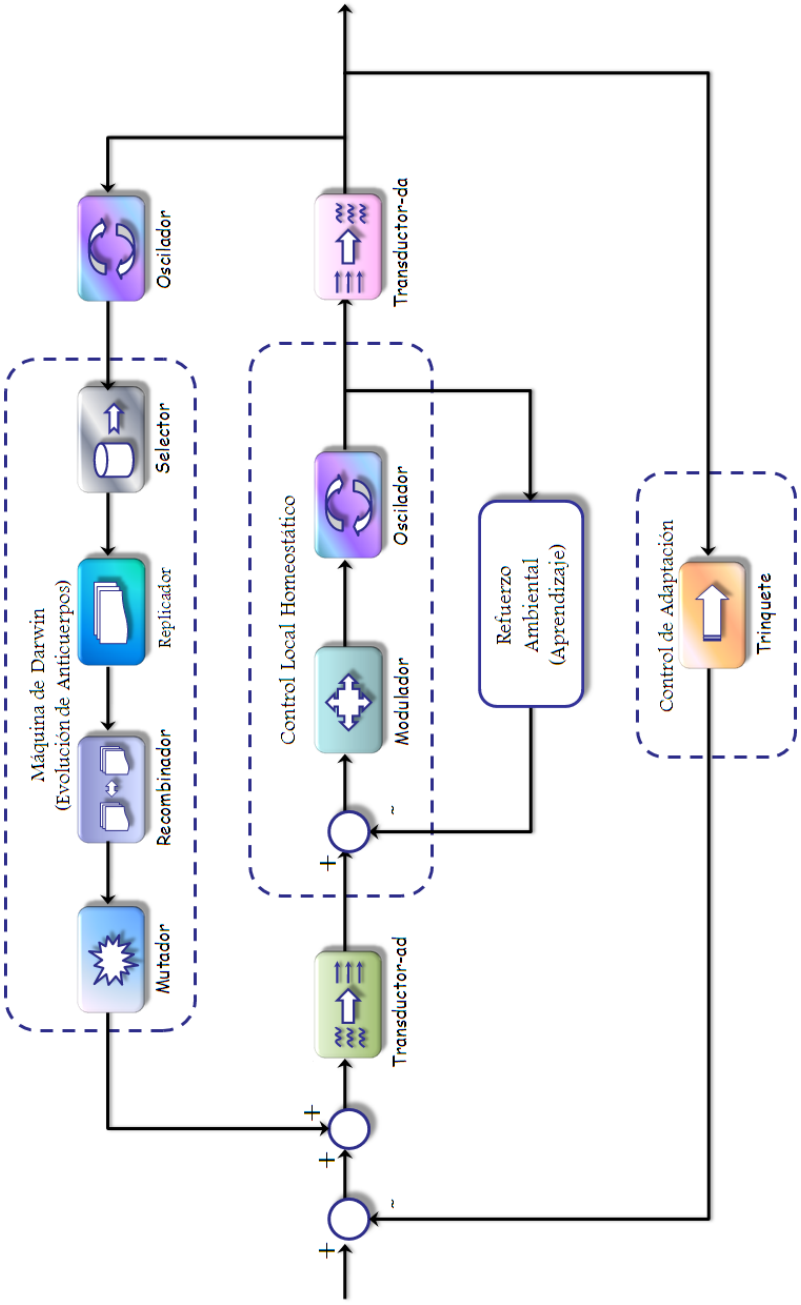


Figura 6.10: Diagrama de bloques para el Sistema Inmune Artificial.

El *Transductor-ad* convierte en información sub-simbólica, la información simbólica que proviene de los módulos declarativo y motivacional de la arquitectura. Este transductor convierte además, a un formato comprensible por el SIA (información sub-simbólica expresada en un alfabeto binario), la información que proviene de otros componentes del

nivel cognitivista del mismo *Comportamiento*[12]. El *Modulador* por su parte, se encarga de articular el proceso de selección de los anticuerpos que mejor identifican la entrada ambiental percibida (en términos inmunológicos, el antígeno) y de enviar una recomendación de actuación (i.e, una respuesta inmune) al módulo Efector. El *Transductor-da*, recibe la señal directa de la respuesta inmune del SIA y, por un lado, la convierte a un formato que sea comprensible por el módulo Efector y, por otro, la transforma a una acción simbólica que se comunica a otros módulos de la arquitectura, en especial al módulo Motivacional y a otros *Comportamientos*.

Es posible además identificar en la figura 6.10, tres lazos de retroalimentación. Un lazo interno de retroalimentación negativa que se encarga de dirigir las dinámicas de aprendizaje, y dos lazos externos, uno de retroalimentación positiva que evoluciona los anticuerpos e inyecta diversidad en el repertorio inmune, y otro de retroalimentación negativa que adapta los nuevos cambios en el acervo de anticuerpos.

El lazo interno de retroalimentación negativa se encarga de reforzar, a través de una señal ambiental, las acciones que envían los anticuerpos como respuesta a los estímulos de entrada. De esta manera, este lazo dirige el *aprendizaje* en el SIA, "premiando" aquellos anticuerpos que envían acciones correctas y "castigando" aquellos que producen acciones incorrectas (e.g., acciones no viables o que producen consecuencias negativas para el agente). Este lazo interno incorpora además, un *Oscilador* que determina los ciclos en que deben ser activadas las meta-dinámicas del SIA (i.e., procesos de producción, destrucción, auto-reparación, maduración de anticuerpos, proceso de recorte, entre otros).

Retomando un argumento que se mencionó en apartados anteriores, el proceso evolutivo del SIA está caracterizado esencialmente por una dinámica de retroalimentación positiva, pues dicho proceso se encarga de aumentar la exploración en el espacio de búsqueda y de amplificar los cambios evolutivos incorporados en la población de anticuerpos. En este lazo externo es posible distinguir ahora un nuevo bloque de construcción compuesto, la Máquina Darwiniana (MD), conformada por un *Selector*, un *Replicador*, un *Recombinador* y un *Mutador*.

Estos cuatro bloques que conforman la MD del SIA coordinan el proceso evolutivo del acervo inmune a través de las siguientes dinámicas: el *Selector*, como su nombre lo indica, selecciona probabilísticamente un sub-conjunto de anticuerpos "mejor adaptados" en base a un indicador de desempeño que ha adquirido cada uno durante los procesos de activación de la red inmune; luego, por medio del *Replicador*, se realizan copias (de acuerdo con cierto criterio de clonación) de los anticuerpos en dicho sub-conjunto; después, la información que contiene cada par de anticuerpos es intercambiada formando nuevos anticuerpos por medio del *Recombinador*; y finalmente, el *Mutador* actúa como un mecanismo que inyecta diversidad, generando pequeños cambios en la estructura de cada anticuerpo recombinado. De esta manera, la MD como se mencionó antes, tiene el objetivo de evolucionar y promover la diversidad del acervo de anticuerpos, combinando las propiedades de exploración y explotación de nuevas soluciones. Adicionalmente, este lazo externo incorpora un *Oscilador* que determina los ciclos de activación de las dinámicas evolutivas coordinadas por la MD.

[12]En la sección 6.2 ya se había comentado la propiedad híbrida e integradora de los *Comportamientos* al incluir diferentes mecanismos de aprendizaje y de procesamiento de la información.

Finalmente, se observa un lazo externo de retroalimentación negativa que se encarga de establecer el control adaptativo general del SIA a través de un *Trinquete* evolutivo que representa el nivel de adaptación y aprendizaje del sistema. Este lazo externo actúa en el largo plazo permitiendo que el sistema, primero, evolucione hasta cierto punto (creando una memoria inmune que se forma a partir de la selección de los anticuerpos mejor adaptados), y luego, se estabilice por medio de un proceso homeostático de retroalimentación negativa, permitiendo así que el proceso evolutivo sea activado nuevamente por el *Oscilador*.

6.4.2. Diseño de bloques de construcción para la RNA

En la figura 6.11, se observa el diagrama de bloques para la segunda Máquina Autopoiética: las Redes Neuronales Artificiales (RNA). Al igual que en el diagrama de bloques para el SIA, las RNA disponen de dos tipos de transductores que convierten la información directa en difusa y viceversa. Cabe notar que, en el flujo de control de lazo abierto (*Transductor-ad* $\Rightarrow$ *Diferenciador* $\Rightarrow$ *Modulador* $\Rightarrow$ *Integrador* $\Rightarrow$ *Transductor-da*) aparecen ahora dos bloques de construcción nuevos, el *Diferenciador* y el *Integrador*.

El *Diferenciador* determina la tasa de cambio en la intensidad de las señales que entran y se dispersan a través de la red, lo cual se corresponde con el aporte que hace cada conexión sináptica (determinado por un peso w) a cada nodo o neurona de la red. Estos pesos entre las conexiones se reajustan continuamente con el objetivo de reducir la señal de error que resulta, de comparar la salida de la RNA con las salidas deseadas ó, del reajuste que sugiere una señal ambiental de refuerzo (retroalimentación). Por su parte, el *Modulador* es el encargado de dirigir estas dinámicas de propagación y corrección de la señal de error a través de la red (algoritmo de retropropagación) y del reajuste subsecuente de los pesos. Cada neurona de la red funciona en realidad como un *Integrador*, donde las entradas que recibe cada una son acumuladas hasta superar un umbral[13], momento en el cual se dispara una señal a la siguiente neurona (en el caso de neuronas situadas en las capas ocultas de la red) o una señal al *Transductor-da* (en el caso de neuronas en la capa de salida).

Esta relación sinérgica entre el *Diferenciador*, el *Modulador* y el *Integrador*, junto con un lazo interno de retroalimentación negativa, es la que articula el proceso de aprendizaje de la red neuronal. Esta retroalimentación negativa viene dada por una señal ambiental del error, la cual se propaga hacia atrás de la red (backpropagation), y a partir de esta propagación del error son luego ajustados los pesos sinápticos entre las neuronas empleando un algoritmo de *Q-learning*. De esta manera, este lazo provee un circuito de regulación que corrige, en tiempo de ejecución, las acciones tomadas por el reflejo de lazo abierto.

En el diagrama de bloques para la RNA se observa además, un lazo externo de retroalimentación positiva coordinado por un *Oscilador*, el cual se encarga de activar cíclicamente una MD (la cual aparece simplificada en un sólo bloque). Similar al propósito de la MD en el diagrama de bloques del SIA, ésta dirige el proceso evolutivo de la RNA. Dicho proceso evolutivo consiste en la exploración de nuevas configuraciones topológicas de la

[13] Calculado por una función de activación o transferencia. Entre las más usuales están: la binaria, la lineal, la sigmoidal, y la tangente hiperbólica, entre otras.

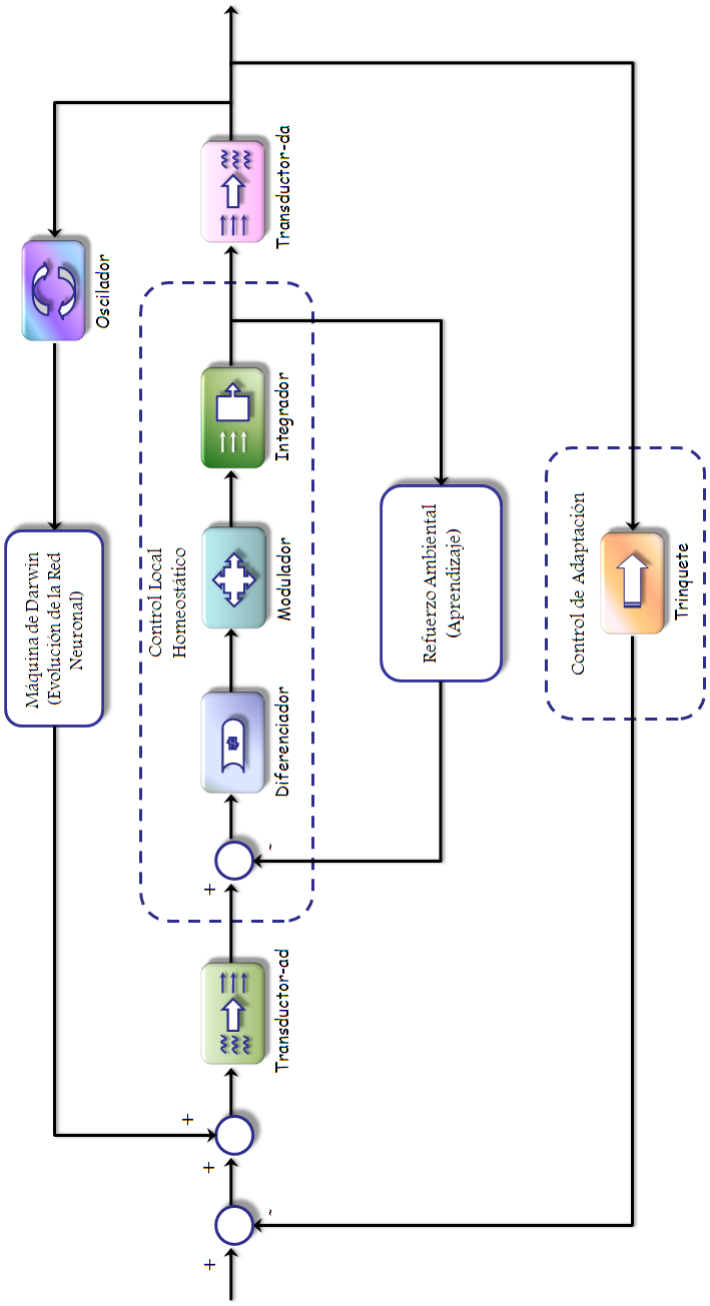


Figura 6.11: Diagrama de bloques para las Redes Neuronales Artificiales

red (variaciones en el número de neuronas, en el número de conexiones por neurona, y en los pesos entre tales conexiones) que conducen a adaptaciones funcionales de la red a largo plazo.

Finalmente, un lazo externo de retroalimentación negativa que involucra un trinquete evolutivo, dirige el proceso adaptativo de toda la red neuronal. Este lazo modula el proceso evolutivo que lidera la MD, llevando al sistema a un estado estable. No obstante,

recuérdese que por lo menos en el alcance de la presente tesis, las RNA sólo tendrán la capacidad de ajustar sus pesos sinápticos aunque no podrán cambiar o ampliar su estructura topológica (como se discutió en la sección 6.3.2.1). En un trabajo futuro, será considerada la capacidad evolutiva de las redes neuronales.

6.4.3. Diseño de bloques de construcción para las RC

En la figura 6.12 se observa el diagrama de bloques para la Red de Comportamientos (RC). Como se habló en la sección 6.2, los *Comportamientos* son unidades híbridas que integran múltiples componentes de procesamiento y aprendizaje: RP, RES, SIA y RNA. Por su parte, una RC es un modelo conexionista de *Comportamientos* que define un conjunto de dinámicas de activación/inhibición entre los mismos. En el diagrama se puede apreciar entonces, un reflejo básico de lazo abierto conformado por los bloques *Transductor-ad* $\Rightarrow$ *Diferenciador* $\Rightarrow$ *Modulador* $\Rightarrow$ *Integrador* $\Rightarrow$ Comportamientos $\Rightarrow$ *Integrador* $\Rightarrow$ *Transductor-da*, donde los *Comportamientos* y sus componentes aparecen como bloques simplificados. Este reflejo de lazo abierto representa la activación reactiva de un *Comportamiento* ante una situación que demande una respuesta innata o instintiva. Por ejemplo, si en una situación dada, el agente activa el *Comportamiento* $\langle$`beber-café`$\rangle$, y mientras lo hace ocurre que la taza de café *caliente* se desliza entre sus manos por un mal agarre, el agente activaría en ese caso el *Comportamiento* reactivo $\langle$`saltar-hacia-atrás`$\rangle$ activado por el reflejo de lazo abierto a fin de no quemarse con el café, sin embargo, esta decisión no ha requerido un proceso deliberativo sino que se ha activado de manera instintiva como un reflejo básico de lazo abierto.

La orquestación de los procesos conscientes en METÁFORA, es realizada principalmente a través de las dinámicas internas de las RCs y su interacción con el Área de Trabajo Global (ATG), esta última encargada de dirigir la atención del agente y administrar la Memoria de Trabajo (MT). Esta orquestación de la consciencia es articulada por 3 bloques de construcción: el *Diferenciador*, el *Modulador* y el *Integrador*.

El *Diferenciador* determina el cambio en el nivel de interés o atención que tiene el agente sobre los elementos percibidos. Si un objeto no presenta cambios interesantes para el agente durante cierto intervalo de tiempo, el agente empezará a reducir gradualmente su atención hacia ese elemento, incrementando quizás la atención en otros elementos que si sufren "cambios" o que aparecen repentinamente en su campo perceptual. Los elementos que no se encuentran dentro del foco de atención del agente empiezan a perder gradualmente su "derecho" de permanecer en la memoria de trabajo, perimiendo así el almacenamiento de nuevos elementos que se encuentren en un estado actual de procesamiento. De esta manera, el *Diferenciador* identifica el cambio de intensidad en los elementos percibidos, capturando el estado dinámico del entorno, lo cual influye a su vez en la activación de los diferentes *Comportamientos* y en el cambio del foco de atención del agente.

El *Modulador* se encarga de coordinar las dinámicas de dispersión de la energía de activación entre los *Comportamientos*. Las RCs definen un conjunto de parámetros globales que influyen en la activación de tales *Comportamientos*: un umbral de activación de cada *Comportamiento*, una cantidad de energía de activación por cada estado observado,

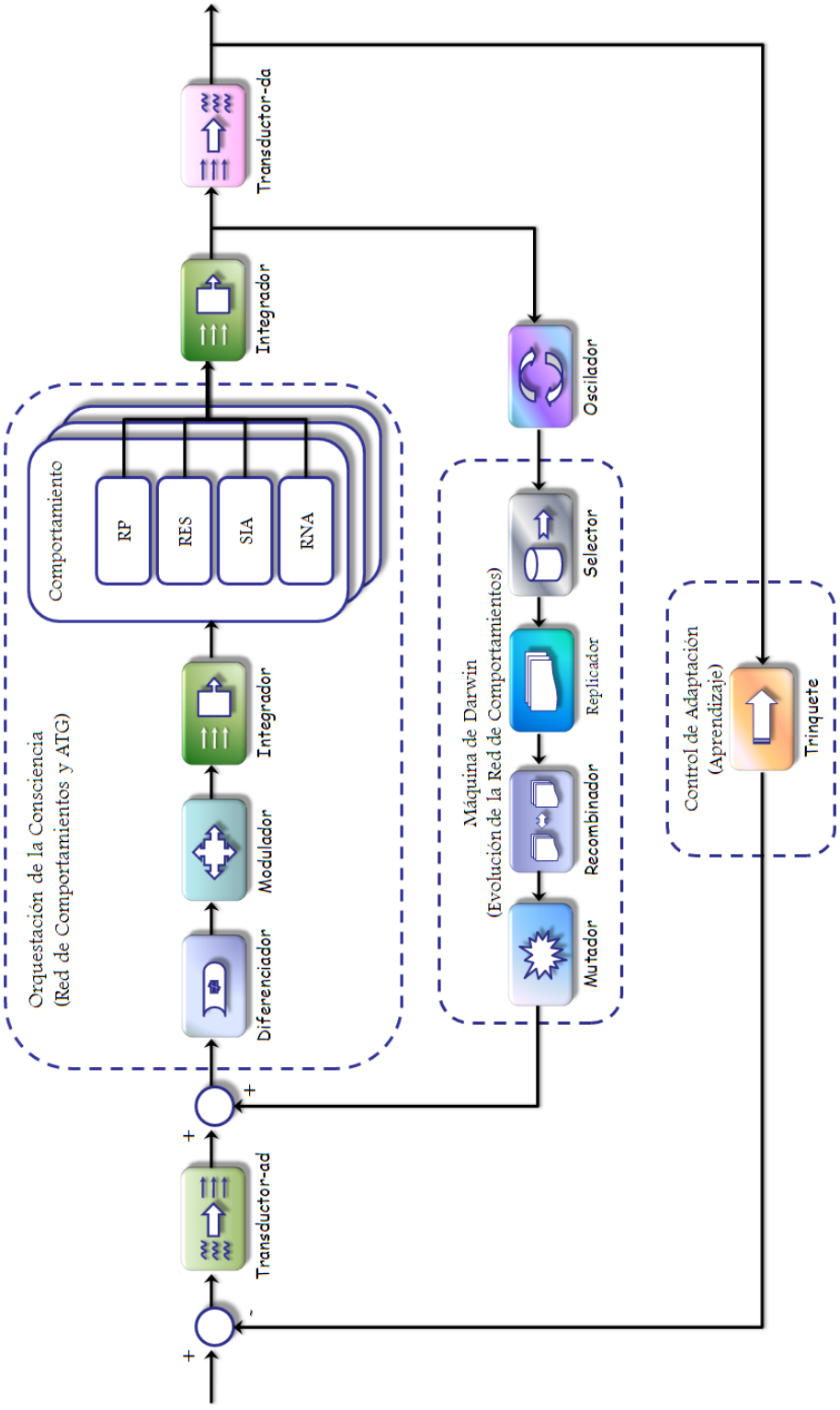


Figura 6.12: Diagrama de bloques para las Redes de Comportamientos

una cantidad de energía de activación que entra en la red por cada meta, y una cantidad de energía que sale de la red por cada meta lograda. Estos parámetros determinan la cantidad de activación que cada *Comportamiento* dispersa hacia adelante, hacia atrás o hacia afuera de la red. La modulación de estos parámetros puede derivar en intrincadas relaciones de activación e inhibición entre los *Comportamientos*, produciendo diferentes respuestas como consecuencia.

El *Integrador*, análogamente a la función del mismo en las RNA, se encarga de acumular las entradas de activación que llegan a cada *Comportamiento* (nodo) desde otros nodos de la red y desde factores externos a la red (estado percibido por el agente, metas, motivadores, etc.). Cuando supera un umbral de activación determinado, el *Comportamiento* se convierte en candidato para ser ejecutado. Adicionalmente, un segundo *Integrador* es propuesto con el objetivo de unificar en una sola, las salidas enviadas por cada uno de los componentes (RP, RES, SIA y RNA) del *Comportamiento* que ha sido activado. Este *Integrador* forma parte del *Mecanismo de Modulación de Acciones* del módulo Efector, y puede ser implementado empleando diferentes aproximaciones algorítmicas (en concordancia con el principio de implementación múltiple), las cuales se describen al detalle en el capítulo 7.

Nuevamente se observa en este diagrama de bloques, un lazo de retroalimentación positiva articulado por un *Oscilador* y una MD, donde esta última se encarga de explorar y explotar nuevas configuraciones de RCs a través de un proceso darwiniano. Este lazo es a su vez controlado por otro lazo exterior de retroalimentación negativa que guia el proceso de adaptación y aprendizaje de las RCs, generando un *Trinquete* evolutivo que, como se ha explicado anteriormente, lleva al sistema a un estado homeostático.

6.4.4. Diseño de bloques de construcción para el SPA

En la figura 6.13 se esquematiza el diagrama de bloques para el Sistema de Prospección-Anticipación (SPA) de METÁFORA. Nuevamente es posible identificar un flujo de lazo abierto conformado por los bloques *Transductor-ad* $\Rightarrow$ *Modulador* $\Rightarrow$ *Redes de* Comportamientos $\Rightarrow$ *Integrador* $\Rightarrow$ *Transductor-da*. Este flujo de lazo abierto representa el proceso de orquestación de la consciencia del agente, donde se puede destacar la actuación principal de un conjunto de RCs que se presentan de manera simplificada en el diagrama. En este flujo, cabe resaltar la función tanto del *Modulador*, encargado de articular el proceso de activación de las RCs con respecto al conjunto de metas activas, como la del *Integrador*, responsable de unificar las distintas recomendaciones de activación que propone cada RC en un sólo *Comportamiento* activo que se hará con el control consciente del agente. Este *Integrador* forma parte también del *Mecanismo de Modulación de Acciones* del módulo Efector, cuyo propósito es similar al *Integrador* de salidas de los *Comportamientos* que se comentó en el diagrama de bloques para la RC.

Un primer lazo de retroalimentación positiva dirige el proceso de evolución de RCs (como se explicó anteriormente) a través de una MD que ahora se representa de forma simplificada. Luego, dos lazos de retroalimentación negativa convergen en un sólo punto: un lazo de control adaptativo que, a través de un *Trinquete* evolutivo, modula el aprendizaje de nuevas y mejores RCs; y un lazo de retroalimentación negativa que representa

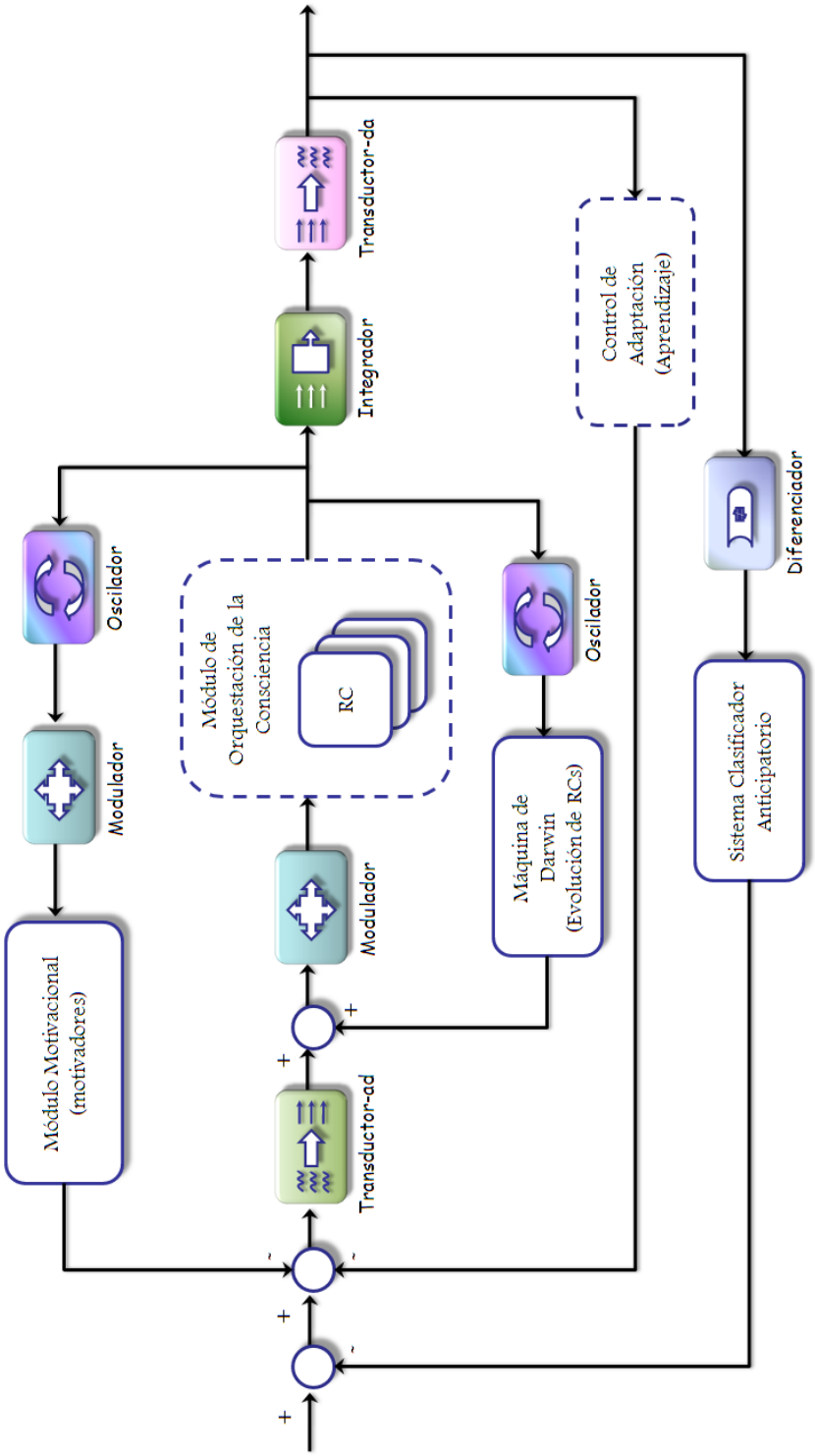


Figura 6.13: Diagrama de bloques para el Sistema de Prospección-Anticipación

la influencia "motivacional" sobre el proceso de toma de decisiones. Este último lazo, es controlado por un *Oscilador* y un *Modulador*.

La función del *Oscilador* en este lazo motivacional, es la de controlar la activación periódica de ciertos motivadores, necesidades e incluso algunas emociones básicas simuladas en el agente, mientras que la función del *Modulador* es la de controlar los niveles de concentración de los estímulos que activan tales motivadores y emociones. Por ejemplo, si el agente modela el comportamiento de un organismo artificial, por decir un ratón, éste se verá sujeto a la activación periódica de motivadores (en este caso, *instintos*) que se atribuyen a la conservación de su supervivencia, tales como el ⟨instinto-reproductor⟩, el ⟨instinto-de-alimentación⟩, el ⟨instinto-de-fatiga⟩, el ⟨instinto-de-curiosidad⟩, etc. Cada uno de estos motivadores tiene diferentes estados de satisfacción, e.g., el instinto de alimentación tendría estados como: ⟨con-hambre⟩, ⟨medianamente-satisfecho⟩, y ⟨sin-hambre⟩. Si el ratón se encontrase en el estado ⟨con-hambre⟩, éste influiría en la activación de los *Comportamientos* que condujesen al agente a saciar su instinto (motivador) de alimentación, sin embargo, este instinto no puede perdurar indefinidamente, pues de ser así el ratón sólo se preocuparía por comer incesantemente. Es aquí cuando el *Modulador* se encarga de medir los niveles de concentración y satisfacción de las necesidades del agente, y de activar nuevos estados para el motivador, en este caso el estado de ⟨medianamente-satisfecho⟩ o, si ha comido lo suficiente, el estado de ⟨sin-hambre⟩.

De esta manera, tanto los motivadores como las emociones básicas operan cíclicamente, incrementando con la ausencia de interacciones de satisfacción y disminuyendo con el acostumbramiento. El propósito entonces de este lazo de retroalimentación (reflejo cognitivo) es mantener el nivel motivacional del agente en alguna región homeostática entre la estimulación baja y la estimulación alta. Como comentario adicional, cabe notar que los planteamientos que se hacen sobre una jerarquía en desarrollo de los reflejos cognitivos basada sucesivamente en los reflejos más bajos ya logrados, son enteramente compatibles con las jerarquías *somático-sensoriales-emocionales* propuestas por Erikson (Erikson, 1973).

Ahora bien, el lazo abierto y los tres lazos cerrados internos descritos hasta el momento (MD, motivacional, y de adaptación), conforman el sistema de control básico para la toma de decisiones y la orquestación de los procesos conscientes en METÁFORA (incluyendo todos los flujos de control subyacentes que se han expuesto con anterioridad y que se presentan de forma simplificada en el diagrama). Un aspecto interesante que se propone en la aproximación arquitectónica, es la re-utilización de estos lazos anidados en la simulación interna de situaciones ambientales a partir de las cuales es posible generar, de manera emergente, planes jerárquicos más complejos.

El SPA reemplaza las entradas sensoriales por entradas simuladas obtenidas a partir del conocimiento anticipatorio que posee el agente acerca del ambiente. De esta manera, el SPA define un lazo articulatorio externo de retroalimentación negativa que se ocupa de guiar el proceso de prospección[14] como resultado de la interacción entre un *Sistema Clasificador Anticipatorio* (SA), un mecanismo de simulación interna, y un *Diferenciador*. La idea básica de este lazo externo, es la generación estimativa de entradas sensoriales al sistema, que se corresponden con posibles estados futuros del entorno. Así pues, el *Di-*

[14] Según RAE, el término prospección puede referirse a la exploración de posibilidades futuras basada en indicios presentes.

ferenciador identificaría la tasa de cambio de las condiciones ambientales en cada estado simulado, el SA haría una predicción sobre el siguiente estado ambiental empleando para ello heurísticas que se basan en un proceso de aprendizaje anticipatorio (Métivier y Lattaud, 2003), y el mecanismo de simulación interna mantendría toda esta interacción en un ciclo continuo (simulado) hasta lograr una meta específica del agente.

Por tanto, ahora es posible distinguir un sistema de lazo cerrado que se mantiene en constante flujo: primero, simulando entradas ambientales al sistema; luego, recibiendo entradas motivacionales (motivadores y emociones) e intencionales (metas); después, seleccionando la RC más conveniente que cumpla con las metas y el estado actual simulado; posteriormente, activando los *Comportamientos* apropiados que resultan de las dinámicas de la RC elegida; y finalmente, recomendando las acciones concretas destiladas por el proceso decisorio del *Comportamiento*, las cuales servirían de entrada para generar nuevas predicciones de estados futuros... y así sucesivamente hasta que se "logre" un conjunto de metas globales (y probablemente jerárquicas), o hasta que no exista suficiente información disponible para hacer una nueva predicción. En ese momento, el registro de toda la interacción cognitiva simulada, es estructurada dentro de un plan, el cual es almacenado en una librería de planes, donde competirá con otros planes por ser seleccionado para su posterior ejecución.

6.4.5. Diseño de bloques de construcción para el SMA

En la figura 6.14 se esquematiza el diagrama de bloques para el Sistema Multi-agente (SMA) de la arquitectura. En este diagrama se presentan, de forma simplificada y encapsulada en cada bloque denominado "Agente", los múltiples lazos de control subyacentes que conforman los procesos cognitivos en METÁFORA, los cuales han sido explicados en los apartados anteriores.

De este diagrama cabe destacar el lazo interno de retroalimentación positiva modulado por un *Oscilador* y conectado con una MD. Esta MD es la encargada de guiar el proceso evolutivo (a través del enfoque filogenético) de las estructuras cognitivas simples y compuestas (memes y memeplexes). Este enfoque evolutivo dirigido por la MD se corresponde con una aproximación co-evolutiva donde participa un conjunto de agentes que interactúan en el mismo o en distintos entornos (como se explicó en la sección 6.3.2.3). Este proceso co-evolutivo, que se ocupa de explorar nuevas soluciones (memes) dentro el espacio de búsqueda, posee un lazo externo de retroalimentación negativa que controla el proceso adaptativo global llevando al "acervo memético" –formado por la aportación de memes que hace cada uno de los agentes– a un estado homeostático o de reposo.

El lazo de control adaptativo, permite que el sistema alcance un nivel homeostático a través del aprendizaje de pautas meméticas parcialmente optimizadas, y mediante los mecanismos meta-cognitivos que auto-regulan las dinámicas internas de la arquitectura. Una vez que el control adaptativo induce al sistema a llegar a un estado homeostático, éste asegura, mediante un *Trinquete*, los cambios evolutivos eficientes que se hayan aplicado sobre el acervo memético hasta ese momento, y es sobre ésta base filogenética alcanzada que se inicia nuevamente el proceso de desarrollo y evolución de los memes. Cabe resaltar en estas líneas, que el proceso filogenético de la co-evolución de los memes se activa

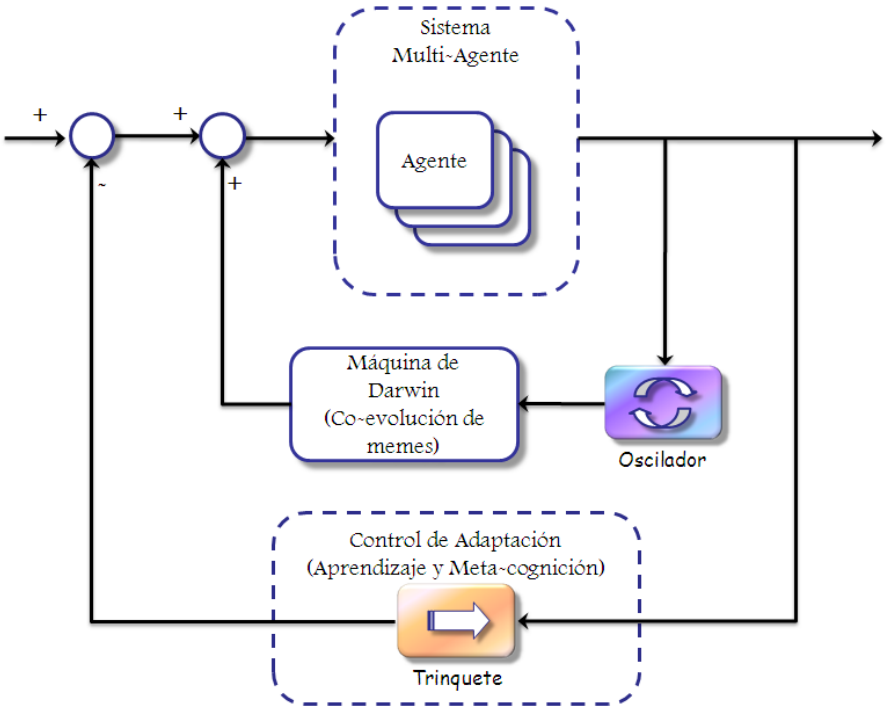


Figura 6.14: Diagrama de bloques para el Sistema Multi-Agente

cíclicamente, después de que se han ejecutado internamente (en cada agente) una serie de bucles retroalimentados cuyo objetivo es desarrollar tales estructuras cognitivas (memes), mediante los procesos ontogenéticos y epigenéticos mencionados con anterioridad.

Como consideración final, nótese bien que uno de los criterios importantes del diseño de METÁFORA es que, en el caso de una paralización en lo más alto de la jerarquía del sistema, los sub-sistemas en cada nivel jerárquico subyacente deben seguir siendo viables. Esto confirma la robustez global del sistema y la autonomía de cada nivel auto-organizativo de la arquitectura.

6.5. Ciclo Cognitivo de Alto Nivel

Esta sección describe el ciclo cognitivo de alto nivel de METÁFORA, donde es posible observar las dependencias entre los módulos, los mecanismos en paralelo y la secuencialidad del proceso cognitivo global. Debido a la complejidad intrínseca de los procesos cognitivos que coordina METÁFORA, su representación detallada en un único diagrama dificultaría su comprensión. Es por esta razón, que los procesos detallados y las dinámicas entre los distintos módulos de la arquitectura serán precisados en los capítulos posteriores.

Considérense, los principios básicos sobre el ciclo cognitivo humano propuestos por (Franklin et al., 2005):

- *Hipótesis*: el procesamiento cognitivo humano es efectuado a través de una continua iteración de ciclos cognitivos.
- *Duración*: cada ciclo cognitivo toma aproximadamente 300 milisegundos.
- *Efecto en cascada*: los distintos ciclos cognitivos pueden incluir sub-procesos que se ejecutan simultáneamente y en paralelo.
- *Secuencialidad*: el proceso de la consciencia mantiene un orden serial y la ilusión de continuidad.

La gran mayoría de las funciones que atañen a la cognición humana operan a través de interacciones continuas entre los *procesos conscientes*, los distintos *tipos de memoria* y el proceso de *toma de decisiones*. Estas interacciones son llamadas ciclos cognitivos (VanRullen y Koch, 2003). Mientras que estos ciclos pueden imbricarse, produciendo acciones en paralelo, deben a su vez preservar la secuencialidad de la consciencia. Tales hallazgos han sido reportados previamente por neurocientíficos (Halgren et al., 2002; Fuster et al., 2000; Lehmann et al., 1998; Freeman, 2003). Así pues, cada ciclo cognitivo en METÁFORA especifica los roles funcionales de la memoria, la consciencia, y la toma de decisiones en el proceso de cognición, tomando como base los principios propuestos en la metáfora del Teatro de la Consciencia de Baars (Baars, 1997).

La cognición humana se compone de ciclos cognitivos similares donde cada uno toma entre 100 y 500 milisegundos aproximadamente, pero que debido al solapamiento de estos, pueden ocurrir potencialmente a una tasa de entre dos y diez ciclos por segundo (Freeman, 2003). De esta manera, en METÁFORA, la ejecución de procesos cognitivos de mayor complejidad (deliberación, planificación, resolución de problemas, etc.) se conforma por la interoperabilidad entre ciclos cognitivos más básicos, flexibles e imbricados, que no superan los 500 ms cada uno, los cuales se encadenan en secuencias de entre dos y diez ciclos por segundo.

En el modelo cognitivo de METÁFORA se sugiere por consiguiente, que los eventos conscientes ocurran como una secuencia discreta y coherente de episodios que están separados por periodos muy cortos de *procesos no conscientes*. Estos últimos operan (compitiendo y cooperando entre sí, generando copias de sí mismos, etc.) en paralelo y de manera distribuida, entre cada intervalo de activación consciente del ciclo cognitivo.

Tomando en cuenta lo anterior, se enuncian a continuación los supuestos básicos sobre los cuales se modela cada ciclo cognitivo de alto nivel en la arquitectura:

- Cada agente autónomo opera en ciclos iterativos de la forma $\langle$*percepción–procesamiento–actuación*$\rangle$.
- El proceso cognitivo global de cada agente controlado por METÁFORA funciona a través de una cascada continua de ciclos cognitivos básicos.
- Los procesos cognitivos de alto nivel (deliberación, planificación, resolución de problemas, etc.) son producidos por las secuencias operativas de tales ciclos cognitivos básicos.
- Cada ciclo cognitivo básico no debe superar un tiempo de ejecución mayor a 500 milisegundos.

Grosso modo, las funciones que se definen para cada ciclo cognitivo en la arquitectura son:

- El agente percibe su entorno (externo o interno).
- El sistema construye una representación interna de la situación actual.
- El sistema decide en qué partes de lo percibido se debe enfocar la atención el agente.
- El sistema selecciona una acción apropiada para ser ejecutada.
- El agente ejecuta la acción (interna o externamente).

En la figura 6.15 se puede apreciar el flujo de eventos de cada ciclo cognitivo de METÁFORA. Estos eventos se describen a continuación.

1. Los estímulos sensoriales (externos o internos) son capturados a través de los sensores y enviados sin ningún procesamiento al módulo de Percepción, donde los procesos de reconocimiento y categorización tienen lugar. El módulo de Percepción transforma los estímulos, a través del Interprete, en bloques de información que se ordenan en parejas del tipo *[propiedad, valor]*, llamados perceptos. Este módulo posee además una memoria sensorial temporal que facilita el proceso de reconocimiento a través del almacenamiento de los perceptos previamente interpretados, cuya intensidad se desvanece incrementalmente en un intervalo de tiempo del orden de milisegundos, tal como propone Hofstadter (Hofstadter y Mitchell, 1994). Posteriormente, los perceptos son almacenados en la Memoria de Trabajo (MT) donde son asociados con perceptos existentes (Baddeley, 1993).
2. Empleando los perceptos enviados por el módulo Perceptual y aquellos residuales que aún se mantienen en la MT, se realizan *asociaciones locales* y en paralelo con:

   a) El módulo Declarativo, desde donde es recuperada la información de las memorias de largo plazo: de representación simbólica (memorias Semántica y Episódica) y sub-simbólica (memorias Asociativas y Episódicas Abstractas).
   b) El módulo Intencional, desde donde son recuperadas las metas actuales del sistema y el plan acorde con el logro de las metas actuales (en el caso de que exista algún plan disponible, de lo contrario se genera uno en sucesivos ciclos cognitivos, como se mencionará más adelante).
   c) El módulo Motivacional, desde donde son recuperados los motivadores y las emociones básicas que influyen sobre el comportamiento del agente en ese momento.

   La MT en los humanos posee una tasa de desvanecimiento que es medida en segundos, y su capacidad de almacenamiento es generalmente estimada entre 7 y 9 elementos o grupos de elementos (Hofstadter y Mitchell, 1994). Estas consideraciones serán tomadas en cuenta en el diseño de la MT de METÁFORA.
3. Una vez estructurada la información en la MT, el módulo de Orquestación de la Consciencia se encarga de activar las dinámicas que dirigen el foco de la atención del agente y la modulación de *Comportamientos*. En este paso del ciclo se ejecutan los siguientes procesos.

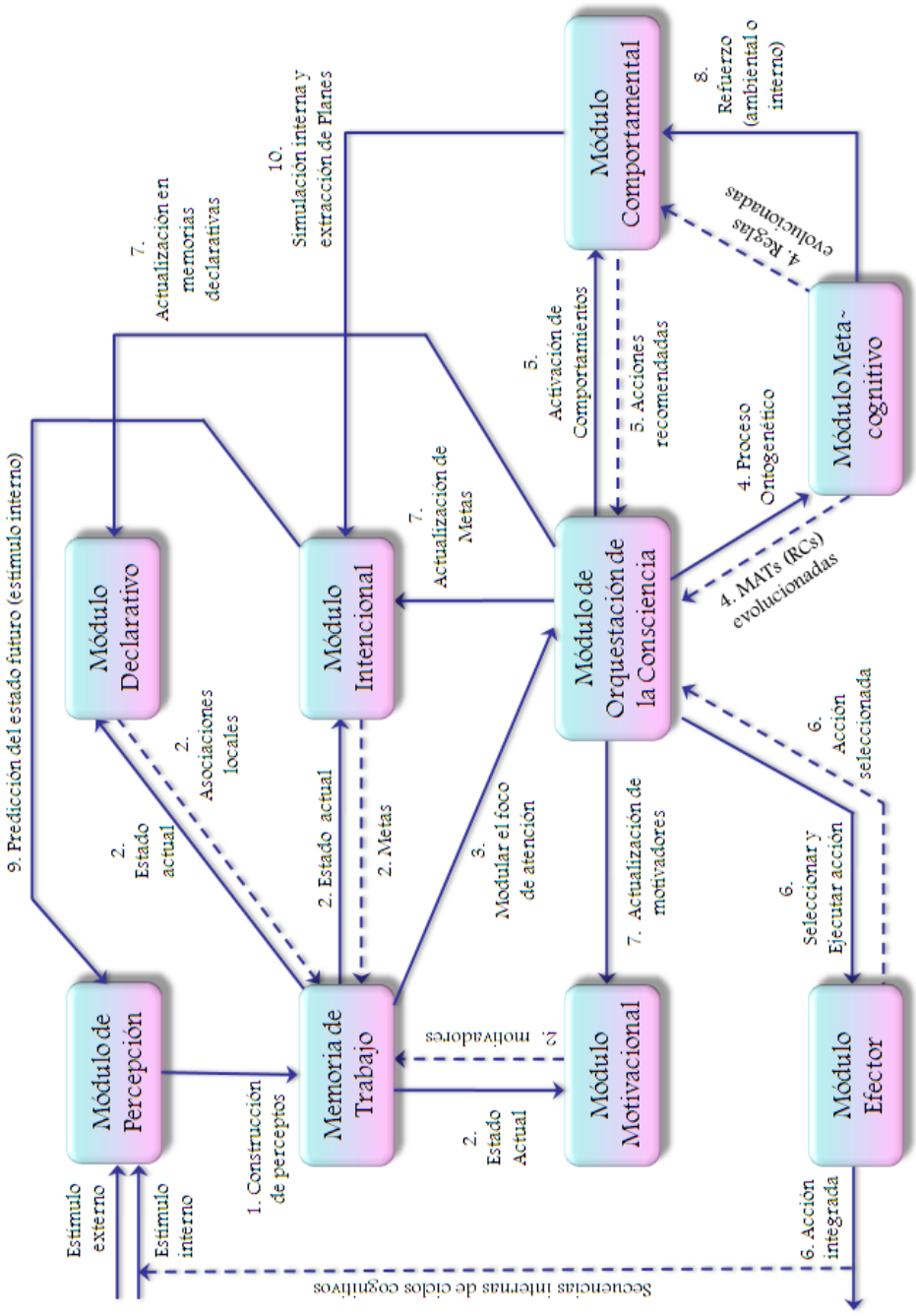


Figura 6.15: Ciclo Cognitivo de Alto Nivel.

a) En el caso de existir un plan activo, se escoge la RC propuesta por el plan. En caso contrario, las RCs compiten por ser elegidas. El criterio de selección de la RC depende del nivel de concordancia entre el propósito de la RC y el estado actual almacenado en la MT.

b) Dirigidas por la RC activa, se ejecutan las dinámicas de activación e inhibición entre los *Comportamientos* contextuales, donde estos cooperan y compiten por obtener el foco de atención. Con el fin de promover la conducta reactiva en el agente[15], los *Comportamientos* recibirán una mayor intensidad de activación por parte de los estímulos externos que por parte de los internos (metas, planes, motivadores, etc.).

c) Si la RC ha encontrado un *Comportamiento* aplicable, el proceso continua en el siguiente paso, de lo contrario se realiza una iteración más desde el paso a). En el caso de que ningún *Comportamiento* haya logrado sobrepasar el umbral de activación después de un número determinado de iteraciones, es seleccionado el *Comportamiento* con mayor nivel de activación.

4. Desde el módulo de Orquestación de la Consciencia se envía un mensaje de activación al módulo Comportamental, donde es ejecutado el *Comportamiento* que ha "ganado" el derecho de controlar el foco de atención después de la competición. Este módulo, ejecuta en paralelo, las diversas dinámicas de cada componente asociado (RP, RES, SIA, y RNA). Una vez se han procesado los distintos algoritmos de los componentes del *Comportamiento* activado, son enviadas nuevamente al módulo de Orquestación de la Consciencia las recomendaciones de acción propuestas por cada uno.

5. El módulo de Orquestación de la Consciencia envía las distintas recomendaciones de acción propuestas por el *Comportamiento* activado al módulo Efector. Éste, empleando un mecanismo de modulación de acciones, integra las distintas recomendaciones en una sola acción consolidada. En el caso de que hayan sido seleccionados múltiples *Comportamientos* para su ejecución en paralelo, el mecanismo de modulación determinará el orden y prioridad en que se ejecutarán las acciones. El mecanismo Traductor del módulo Efector se encargará de convertir las acciones en señales motoras apropiadas que serán ejecutadas en el entorno por los actuadores. El módulo Efector envía la acción consolidada de vuelta al módulo de Orquestación de la Consciencia para su posterior dispersión.

6. Una vez el módulo de Orquestación de la Consciencia recibe la acción unificada generada por el módulo Efector, retransmite ésta a los módulos Motivacional, Intencional, Declarativo y Comportamental de la arquitectura, con el objetivo de que estos realicen los procesos de actualización de sus elementos internos. Ésta, es una forma de cooperación entre los procesos conscientes y los subconscientes, donde parte de la activación del *Comportamiento* activado es dispersa entre lo procesos subconscientes (principalmente los *Comportamientos* no activados) que entrarán nuevamente a competir por obtener el foco de atención en el siguiente ciclo cognitivo. Esto resulta consistente con la teoría de Baars (Baars, 1997) en cuanto a

[15]Es deseable de algún modo, que el agente produzca respuestas casi inmediatas ante situaciones especiales que requieran un comportamiento reactivo. Por ejemplo, suponiendo que el agente tiene que desplazarse del punto A al punto B, y para lograr esta meta ha activado el *Comportamiento* ⟨`desplazarse-hasta-destino`⟩, pero en algún momento de su desplazamiento se encuentra con un muro, el cual debe evadir reactivamente, es en este momento donde el *Comportamiento* ⟨`evadir-obstáculo`⟩ debe prevalecer sobre el *Comportamiento* ⟨`desplazarse-hasta-destino`⟩, aunque después de evadir el muro, el agente retome el *Comportamiento* inicial que lo conducirá hasta B.

que los procesos conscientes (en este caso el *Comportamiento*, la RC y el plan activado), difunden hacia los procesos subconscientes [16] las decisiones y acciones tomadas por estos. En este punto se finaliza el ciclo cognitivo básico, después del cual el proceso es reiniciado tras la entrada de un nuevo estímulo (interno y/o externo).

Algunas consideraciones adicionales que se deben precisar sobre el ciclo cognitivo básico se enuncian a continuación.

Sobre el Refuerzo Ambiental

7. Al inicio de un nuevo ciclo cognitivo, el sistema recibe una señal de refuerzo ambiental (retroalimentación) que luego es enviada al módulo Meta-cognitivo, para que éste se encargue de recompensar o penalizar las acciones ejecutadas por el agente en el paso anterior. Esta señal de refuerzo retroalimenta la decisión de la RC de activar un determinado *Comportamiento*, y a su vez, guía el aprendizaje en cada uno de los componentes del *Comportamiento* activado.

   No obstante, existen escenarios concretos donde el refuerzo no establece una correspondencia biunívoca con la acción ejecutada en cada paso del ciclo cognitivo. Por un lado, las acciones ejecutadas por el agente no siempre conducen a consecuencias inmediatas, y por lo tanto es posible que sean ejecutados varios ciclos cognitivos sin que se perciba una señal de retroalimentación ambiental. Por otro lado, es posible que cuando la señal de refuerzo se reciba, ésta se mantenga con cierta intensidad que disminuye gradualmente con el transcurso de los ciclos cognitivos (situación que se da en aquellos casos donde las consecuencias de las acciones producen un efecto durante un cierto intervalo de tiempo).

Sobre la Conducta Reactiva

Como se mencionó en el flujo cognitivo básico, la RC puede actuar reactivamente ante estímulos externos a través de la modificación de los parámetros globales de la red, otorgando una mayor relevancia a la activación por el estado percibido que a la activación por metas, y permitiendo de esta manera, que los estados cambiantes del entorno produzcan entradas con mayor intensidad dentro de las dinámicas de activación de la RC.

Adicionalmente, y con el fin de minimizar el tiempo de respuesta de cada ciclo cognitivo ante situaciones que requieren que el agente exhiba una conducta reactiva, METÁFORA genera, durante su interacción con el entorno, un registro de la secuencia de *Comportamientos* activados y de sus respectivas acciones ejecutadas (sólo en el caso de que éstas tengan un refuerzo positivo). Estos registros son almacenados como plantillas de conductas reactivas, lo que bien se podría llamar, un *mini-plan reactivo*. La activación de uno de estos mini-planes reactivos depende de dos criterios:

- En el caso de que los estímulos externos o internos produzcan la activación de un *Comportamiento* que no conduce al logro directo o indirecto de al menos una de

[16] Es decir, todos aquellos procesos que se encuentran tras bambalinas según la metáfora del teatro de la consciencia

las metas iniciales del agente, pero que atiende las necesidades externas o internas inmediatas, éste puede ser considerado como un *Comportamiento* reactivo.

- Los perceptos construidos por el módulo Perceptual se corresponden con las precondiciones definidas por alguno de los mini-planes reactivos, y además no se encuentra ningún *Comportamiento* aplicable en la primera iteración de activación de la RC (paso 3).

En el caso de que no se cumplan estos dos criterios, el agente seguirá el ciclo cognitivo básico y ejecutará normalmente las dinámicas de activación de la RC respectiva. La ejecución de los mini-planes reactivos estará supervisada por el módulo de Orquestación de la Consciencia, de esta manera, si los mini-planes reactivos dejan de ser válidos dentro de un contexto, éste activará nuevamente el ciclo cognitivo básico como una medida de contingencia.

Sobre los Procesos Cognitivos de Alto Nivel

Como se mencionó al inicio de esta sección, así como en los requisitos expuestos en la sección 5.7, los procesos cognitivos de alto nivel (planificación, deliberación, resolución de problemas, etc.) requieren de la interacción de múltiples ciclos cognitivos retroalimentados.

En la figura 6.15 puede observarse una línea punteada entre el módulo Efector y el módulo de Percepción. Esta conexión indica que las acciones ejecutadas por el módulo Efector no necesariamente tienen que ser señales motoras que producen cambios en el ambiente, sino que pueden ser asimismo, acciones internas que sirven de entrada (como un estímulo interno o como un estado ambiental simulado) a un nuevo ciclo cognitivo básico que itera hasta cumplir alguna de las siguientes condiciones: 1) hasta que el sistema alcance un objetivo especifico (e.g., una meta concreta en el caso de la planificación y la resolución de problemas), o 2) hasta que se cumpla un número de ciclos determinados por el diseñador de la arquitectura, o 3) hasta que no exista más información disponible sobre la cual se pueda seguir iterando y se requiera información ambiental.

Concretamente, la extracción de planes deliberativos en METÁFORA propone un ciclo cognitivo alternativo donde son considerados los siguientes pasos después del ciclo básico:

8. Las acciones recomendadas por el módulo Comportamental y unificadas por el módulo Efector, son enviadas al módulo Intencional, donde a través de un Sistema Clasificador Anticipatorio (SA) se predice el siguiente estado futuro del ambiente (estado anticipado). El módulo Intencional envía este estado anticipado al módulo Perceptual, simulando y reemplazando el estímulo externo que normalmente recibiría. El módulo Perceptual identifica los perceptos e inicia nuevamente el ciclo cognitivo.
9. Los planes son confeccionados por el proceso ontogenético, en función de las metas actuales y los estados internos del agente. Cada plan deliberativo es generado entonces, como la secuencia de activaciones de RCs, *Comportamientos* y acciones concretas. Estos planes son evaluados mediante el mecanismo de simulación interna que trabaja en conjunción con el SA para producir secuencias de estados

ambientales simulados. Finalmente, el plan es almacenado en la librería de planes que posee el módulo Intencional y activado cuando el agente inicie nuevamente su interacción con el ambiente.

Sobre el Algoritmo Memético (AM)

Simultáneamente al ciclo cognitivo básico de METÁFORA, se ejecutan los procesos Epigenéticos, Ontogenéticos y Filogenéticos. Estos se describen en un flujo cognitivo alternativo, el cual es ejecutado paralelamente al ciclo cognitivo básico aunque en escalas de tiempo diferentes, como se explica a continuación:

10. Evolución de estructuras Cognitivas:

    a) Una base filogenética es creada inicialmente, la cual determina el conjunto de *Comportamientos* y de RCs[17] de cada agente. Por cada *Comportamiento*, son inicializados los componentes simbólicos y sub-simbólicos. Esta inicialización, puede ser aleatoria (en el caso de los componentes sub-simbólicos) o determinada por el diseñador (en el caso de los componentes simbólicos). Esta base filogenética es la configuración mínima sobre la cual se generarán los desarrollos adaptativos subsecuentes, y tiene como fin acelerar el proceso de aprendizaje inicial de las estructuras cognitivas al no tener que partir de un diseño desde cero.

    b) A nivel epigenético, son activados los mecanismos de aprendizaje de la arquitectura durante la interacción del agente con su ambiente. En cuanto a los componentes SIA y RNA, se efectúa un aprendizaje por refuerzo ambiental, mientras que para el componente RES se emplea un aprendizaje de extracción de conocimiento sub-simbólico.

    c) A nivel ontogenético, se evolucionan las RCs (definidas por la base filogenética) y los planes deliberativos (extraídos por el módulo de Prospección y Anticipación).

    d) Después de cierto número de interacciones del agente con el ambiente, es activado el proceso filogenético (búsqueda global del AM). Las estructuras cognitivas que ha desarrollado cada agente (reglas, *Comportamientos*, redes de comportamientos y planes), son encapsuladas en estructuras específicas que son susceptibles de ser manipuladas por el proceso evolutivo, estas estructuras son los memes. Aquellos memes que han demostrado un mejor "desempeño" durante la interacción del agente con su ambiente, tienen mayor probabilidad de ser clonados y recombinados durante el proceso evolutivo. Luego, los memes de todo el conjunto de agentes son evolucionados a través de un proceso co-evolutivo.

    e) Los memes resultantes del proceso filogenético, son transmitidos nuevamente a cada uno de los agentes, donde son incorporados en cada tipo específico de repositorio (Planes, RCs, *Comportamientos*, y reglas[18]), reemplazando

[17] Por cada RC se definen las respectivas relaciones de activación e inhibición entre *Comportamientos*.

[18] Reglas que se incluyen en el nivel simbólico (RP Y RES) y que son asimiladas en el nivel sub-simbólico (RNA y SIA) a través del *Aprendizaje Descendente*.

así aquellos memes que han tenido un desempeño poco eficiente. Estos nuevos memes, forman la base filogenética (conocimiento innato) a partir de la cual tendrán lugar los desarrollos epigenéticos y ontogenéticos subsiguientes (búsqueda local del AM). El proceso continua cíclicamente.

6.6. Resumen

En este capítulo se ha presentado una aproximación al diseño conceptual de la arquitectura, cuyo objetivo principal ha sido establecer un puente entre la naturaleza de la solución y el diseño detallado de la arquitectura. Mientras que en la naturaleza de la solución se mencionaron las bases biológicas, neurofisiológicas y psico-cognitivas que actúan como metáforas que sirven de inspiración a la aproximación arquitectónica, en este capítulo se presentan pautas más concretas acerca de cómo son modeladas tales metáforas en un diseño conceptual.

Al igual que la construcción de una casa requiere un diseño previo que exhiba diferentes perspectivas y niveles incrementales de especificación por parte de un arquitecto, en este capítulo se presentan tres perspectivas principales de la arquitectura: una del tipo modular estática, otra de inspiración biológica/cibernética, y una tercera donde se aprecia la interacción dinámica entre los módulos de la arquitectura en cada ciclo cognitivo.

En el diseño modular de la arquitectura, se observan los diferentes módulos, componentes y sub-sistemas que emplea METÁFORA. Esta es una perspectiva estática donde se aprecian las relaciones estructurales entre sus elementos y donde es posible destacar la imbricación de capas en la arquitectura. Se proponen tres perspectivas de estructuración de la arquitectura: una por módulos especializados, otra por niveles de procesamiento discretizados según el paradigma cognitivo (emergente o cognitivista), y una última que designa los componentes híbridos transversales de la arquitectura: los *Comportamientos*.

El principal aporte innovador que se hace con este modelado modular es el de otorgar a la arquitectura la cualidad de híbrida. En cuanto a lo estructural, el modelo combina una distribución horizontal o de procesamiento por niveles con una distribución vertical de componentes que procesan la información en paralelo (*Comportamientos*). En cuanto a lo representacional, los módulos de procesamiento cognitivo de la arquitectura son híbridos debido a que incorporan mecanismos de procesamiento tanto emergentes como cognitivistas.

En el diseño de inspiración biológica/cibernética se proponen cuatro perspectivas. En primera instancia, un modelado que propone los niveles de auto-organización jerárquicos de la arquitectura compuestos por máquinas autopoiéticas, las cuales crean relaciones anidadas entre sí hasta dar forma a un Sistema Adaptativo Complejo. En segunda instancia, se propone un modelado de los niveles de desarrollo y evolución de las estructuras cognitivas, basados en principios de la epigénesis, la ontogénesis y filogénesis, y se describe cómo estos interactúan y se integran en un proceso continuo de optimización. La tercera perspectiva propone un Sistema Nervioso Sintético compuesto jerárquicamente por unidades básicas computacionales, redes de unidades, y sub-sistemas de redes. Finalmente, se presenta una aproximación cibernética al diseño jerárquico de la arquitectura, la cual

emplea bloques de construcción para representar los flujos de control y retroalimentación de cada sub-sistema.

Un aspecto innovador que cabe destacar sobre este diseño de inspiración biológica, es la aproximación al modelado de la arquitectura cognitiva a través de diversos tipos de diagramas: los ideogramas autopoiéticos, los diagramas del enfoque ascendente, y los bloques de construcción cibernéticos. Estos diagramas en conjunto, intentan reflejar, de una manera novedosa, las complejas dinámicas de auto-organización, los bucles anidados de retroalimentación, y los procesos darwinianos presentes en los diferentes niveles jerárquicos de la arquitectura.

Por último, en el diseño del Ciclo Cognitivo, se presenta un flujo cognitivo básico donde se describen las relaciones dinámicas entre los módulos y componentes de la arquitectura en cada ciclo de *percepción-procesamiento-actuación*. Se presentan adicionalmente, algunos flujos alternativos que bien extienden el ciclo cognitivo básico, o bien se activan paralelamente a éste.

6.7. Discusión

Una vez se ha llegado a este punto, conviene enunciar algunas reflexiones acerca del diseño obtenido hasta el momento y sus repercusiones en el desarrollo subsecuente de la arquitectura.

- A pesar de que el modelo cognitivo propone una aproximación computacional de inspiración biológica, éste no intenta en ningún momento emular el complejo funcionamiento biológico del cerebro ni la intrincada organización de sus estructuras neuronales. Por el contrario, la presente aproximación se apoya en un enfoque híbrido que permite "acelerar", mediante técnicas simbólicas, el surgimiento de comportamientos complejos e inteligentes que emergen a partir de unidades sub-simbólicas. Sin embargo, un modelado más cercano a la neurofisiología cerebral supondría una emergencia "pura" de las facultades cognitivas, a través de las cuales, probablemente, podrían ser mejor comprendidos los procesos cognitivos biológicos. Esto es algo que definitivamente se escapa del alcance del presente trabajo de tesis doctoral, aunque si abre líneas futuras de investigación como las que persigue el proyecto *Blue Brain*[19].
- Como se ha mencionado antes, los *Comportamientos* de METÁFORA integran múltiples técnicas de procesamiento y de aprendizaje automático: Reglas de Producción (Reglas Predeterminadas y Reglas de Extracción Sub-simbólica), Redes Neuronales Artificiales (que pueden emplear un aprendizaje por refuerzo mediante el algoritmo – *Q-learning* –, o un aprendizaje supervisado mediante el algoritmo de retropropagación), y Sistemas Inmunes Artificiales (que emplean Algoritmos Genéticos y un sistema de distribución de recompensas). Previamente al diseño de la arquitectura fueron analizados otros algoritmos de aprendizaje, principalmente

[19]El propósito del proyecto Blue Brain es generar un modelo del cerebro de los mamíferos a través de técnicas de ingeniería inversa, con el objetivo de comprender mejor las funciones y disfunciones del cerebro a través de simulaciones detalladas. Para más información visitar: `http://bluebrain.epfl.ch/`

los llamados Sistemas Clasificadores (básicos, extendidos, anticipatorios y evolutivos) cuyos resultados de desempeño son discutidos en (Tamee et al., 2008).

No pretendiendo que con esto se haya abarcado el amplio espectro de mecanismos representacionales y como consecuencia quizás, no se hayan seleccionado los algoritmos de aprendizaje más apropiados[20], conviene realizar un análisis comparativo con otros algoritmos que pudiesen compensar, de manera más adecuada, las debilidades de los ya propuestos para cada *Comportamiento*. Algunos ejemplos de estos algoritmos podrían ser (Alpaydin, 2005; Nilson, 2006): los modelos probabilísticos (e.g., las Redes Bayesianas, los Métodos Paramétricos, la Regresión Multivariada, los Modelos Markovianos, etc.), el aprendizaje inductivo (e.g., los Árboles de Decisión), el aprendizaje por refuerzo (e.g., el método de Temporal Difference, el método SARSA, etc.), el aprendizaje deductivo (e.g., el Aprendizaje Basado en Explicaciones y Basado en Ejemplos), el aprendizaje por discriminación lineal (e.g., las Máquinas de Soporte Vectorial, la Discriminación por Regresión), entre otros.

Gracias a la cualidad de escalabilidad de METÁFORA, algunos de estos algoritmos podrían ser incluidos fácilmente en cada *Comportamiento*, y podrían definirse nuevos mecanismos de integración para determinar una salida unificada, o bien definirse ciertas heurísticas que determinaran la activación de un mecanismo mientras los demás son inhibidos.

- Los modelos de diseño del sistema a través de los bloques de construcción y de los ideogramas autopoiéticos, probablemente resulten ser demasiado abreviados para describir las complejas dinámicas de METÁFORA. Desde una perspectiva reduccionista, estos modelos simplifican y omiten muchos de los detalles de implementación interna de cada sub-sistema, lo cual indica que estos modelos encapsulan niveles de detalle más precisos. Por ejemplo, en el caso concreto del módulo motivacional y del módulo meta-cognitivo, estos aparecen en los diagramas de bloques cibernéticos como bloques simples que hacen parte de alguna señal de retroalimentación del sistema, pero internamente estos módulos llevan a cabo complejas dinámicas que bien valdría la pena modelarlas al detalle.

  La decisión de no representar tan minuciosamente estos modelos, a través de los bloques de construcción, recae en el simple hecho de que estos diagramas pertenecen a una fase descriptiva de la solución del problema. Existen otros modelos de representación que permiten llegar a un entendimiento más claro de lo que hace cada módulo de la arquitectura, pero resulta más apropiado utilizarlos en los diseños detallados que se presentarán en los siguientes capítulos.

  Un diseño siempre puede ser susceptible de optimización. Por tanto, la utilización de estos diseños preliminares son tan sólo un atisbo, un boceto, el cual requiere aún de procesos darwinianos mentales que lo conduzcan a un estado más refinado.

[20] La valoración de cuan apropiado puede ser un algoritmo de aprendizaje, en el diseño de los *Comportamientos* de METÁFORA, depende no sólo del conjunto de políticas que determinan su desempeño, sino del nivel de complementariedad y correlación que ejerza cada algoritmo sobre los demás.

"Es interesante resaltar que el proceso de especialización y complejidad crecientes observados en la naturaleza, tiene un paralelo en la ingeniería. *Así, la capacidad de un diseño inicial suele ampliarse, o "estirarse", a medida que pasa el tiempo, y esta mejora se obtiene casi siempre por refinamientos sucesivos que, de hecho, complican aún más el sistema".*

— John H. Milsum, La base jerárquica para los sistemas vivientes, 1994.

Capítulo 7

Los Actores: El Proceso de Toma de Decisiones Guiado por Comportamientos

"Todos los individuos con sistemas nerviosos complejos tienen que afrontar la cuestión que la vida plantea a cada momento: ¿Qué debo hacer a continuación?".
– Sue Savage-Rumbaugh.

En esta sección se describe el proceso de toma de decisiones llevado a cabo por el módulo Comportamental. Como se ha mencionado anteriormente, éste módulo está compuesto por un conjunto de *Comportamientos* cuyo tamaño es en principio determinado por el diseñador de la arquitectura. Un *Comportamiento* se compone de cuatro componentes de naturaleza híbrida (i.e., RP, RES, RNA, y SIA), donde cada uno de los cuales define sus propios mecanismos de: procesamiento de la información, representación del conocimiento, razonamiento, aprendizaje, selección de acciones, entre otros, todos ellos descritos en el presente capítulo. Adicionalmente, este capítulo expone los diversos mecanismos de integración de acciones que emplea el módulo Efector para unificar las distintas recomendaciones de actuación que propone cada componente del *Comportamiento* activo. El proceso de activación de *Comportamientos* será descrito en el capítulo 9.

7.1. Inspiración del Modelo Comportamental

"Behavior is primarily adaptation to the environment under sensory guidance. It takes the organism away from harmful events and toward favorable ones, or introduces changes in the immediate environment that make survival more likely".
— Donald Hebb, A textbook of Psychology, 1972.

El modelo de *Comportamientos* propuesto en la presente tesis, se inspira principalmente en el rol que cumplen los *actores* de la metáfora del Teatro de la Consciencia descrita en el

apartado 4.6.1. Grosso modo, los actores son unidades de procesamiento que representan los contenidos de la experiencia consciente. Cada actor coopera y compite con otros a fin de convertirse en el protagonista de la escena en un momento dado, es decir, con el propósito de obtener el foco de control del agente; y su actuación se encuentra bajo la influencia de los procesos sub-conscientes que operan tras bambalinas (los cuales serán descritos en el capítulo 8).

Por otro lado, desde un punto de vista biológico, los procesos de representación, aprendizaje, y ejecución que coordina cada *Comportamiento* se inspiran en la función conjunta que desempeñan diversas estructuras neuronales en el cerebro: las *Micro-columnas Corticales*, los *Ganglios Basales*, el *Cuerpo Estriado*, el *Tálamo*, y el *Pallidum*.

Una *Micro-columna Cortical* es la unidad básica funcional de la corteza cerebral – el siguiente nivel en organización después de la neurona – la cual agrupa de forma vertical, diversos tipos de células nerviosas que procesan distintos tipos de estímulos (Bonin, 1971). De manera análoga, un *Comportamiento* en METÁFORA agrupa, de forma vertical, distintos tipos de representaciones del conocimiento (i.e., en forma de *reglas* simbólicas y sub-simbólicas), y diferentes mecanismos para el procesamiento de estímulos y la generación de respuestas (acciones).

La justificación de diseñar cada *Comportamiento* como una unidad de procesamiento híbrida (compuesta por múltiples componentes que procesan la información paralelamente), se fundamenta en el propósito de hacer más robusto y eficiente el proceso de toma de decisiones mediante la integración, y la mutua compensación que surge como consecuencia, de los diversos mecanismos de razonamiento y representación del conocimiento empleados por cada componente del *Comportamiento* (siguiendo para ello los principios de *funciones comunes* e *implementación múltiple* descritos en la sección 4.2).

Por otro lado, los *Ganglios Basales* se corresponden con un área del cerebro de los mamíferos que se encarga principalmente de funciones como la adquisición de los hábitos, los procesos de aprendizaje, la resolución de conflictos, el control motriz, y la selección y ejecución de acciones. Un área cortical que corresponde con el almacenamiento de *reglas* en el cerebro es el *Cuerpo Estriado*, parte de los *Ganglios Basales* que se supone desempeña funciones de reconocimiento de patrones (como lo sugieren los estudios previos de (Amos, 2000; Frank et al., 2000; Houk y Wise, 1995; Wise et al., 1996)). Esta porción de los *Ganglios Basales* se proyecta hacia un número de pequeñas regiones conocidas colectivamente como el *Pallidum*. Las proyecciones del *Pallidum* efectúan reacciones inhibitorias sobre las células en el *Tálamo*, el cual filtra los estímulos sensoriales y dirige la selección de acciones en la corteza cerebral.

Graybiel y Kimura (Graybiel y Kimura, 1995) han sugerido que esta configuración crea un sistema del tipo "el ganador lo toma todo" (*winner-takes-all*), tal que, las proyecciones activas del *Cuerpo Estriado* inhiben de manera drástica aquellas neuronas del *Pallidum* que representan la acción seleccionada (las cuales a su vez, cesan la inhibición de células en el *Tálamo* después de que se ha producido la acción). Este es un mecanismo por el cual la regla ganadora se convierte en dominante. De acuerdo con Middleton y Strick (Middleton y Strick, 2000), por lo menos cinco regiones de la corteza frontal reciben proyecciones del *Tálamo* y son controladas por el ciclo del *Ganglio Basal*, jugando así un rol importante en el control del comportamiento.

Existe evidencia (Houk y Wise, 1995; Mao, 2005; Middleton y Strick, 2009) acerca de que los *Ganglios Basales* cumplen una función importante en el reconocimiento de patrones bajo la influencia de procesos de aprendizaje por refuerzo y de aprendizaje supervisado. Así mismo, se sugiere la idea de que los *Ganglios Basales* dirigen el aprendizaje competitivo de reglas y el aprendizaje procedimental de secuencias de acciones.

Por consiguiente, en analogía con los *Ganglios Basales*, cada *Comportamiento* define un conjunto de componentes que actúan como mecanismos de reconocimiento de patrones en relación a la función que desempeña el *Cuerpo Estriado*; a su vez, cada componente posee un mecanismo de resolución de conflictos que cumple con la función que desempeña el *Pallidum*; y finalmente, el módulo Efector, donde se modulan, integran y ejecutan las acciones recomendadas por cada componente del *Comportamiento*, simula la función del *Tálamo*.

El conjunto de representaciones simbólicas y sub-simbólicas (que modelan la memoria procedimental) y los procesos de adquisición de nuevo conocimiento que define cada componente del *Comportamiento*, se corresponden con la afirmación de que los *Ganglios Basales* favorecen el "aprendizaje procedimental" en el cerebro (Saint-Cyr et al., 1988; Ashby y Waldron, 2000; Hikosaka et al., 2001).

Una motivación adicional, de inspiración psicológica, que se ha adaptado en el modelado híbrido de los *Comportamientos* de METÁFORA, corresponde a la dicotomía entre el aprendizaje implícito y el aprendizaje explícito (ó aprendizaje consciente vs. aprendizaje subconsciente) (Berry y Broadbent, 1988; Reber, 1989; Stanley et al., 1989; Willingham et al., 1989; Lewicki et al., 1992). Se habla de "memoria implícita" y "aprendizaje implícito" para referirse a una situación donde la recuperación de la información y la adquisición de nuevo conocimiento no se hace con esfuerzo consciente. Por el contrario, los términos "memoria explícita" y "aprendizaje explícito" se refieren a la recuperación y adquisición de nuevo conocimiento que se lleva a cabo conscientemente. La memoria explícita se almacena a través de representaciones simbólicas, mientras que la memoria implícita es almacenada en representaciones sub-simbólicas.

Según Anderson (Anderson, 1993b), el aprendizaje explícito consiste en transformar el conocimiento explícito (simbólico) en conocimiento implícito (sub-simbólico). Para esto, se parte de la observación de que cuando se aprende una habilidad nueva se empieza por adquirir conocimiento explícito (simbólico, accesible a la conciencia, y reglado), es decir, instrucciones, pero que el uso repetido de este conocimiento da lugar a unos procedimientos específicos, que se activan automáticamente, sin la necesidad de un control ejecutivo. La existencia de este tipo de conocimiento se puede constatar en tareas dobles, de atención dividida: se pueden hacer dos cosas a la vez si las dos están automatizadas. Por otro lado, existe el aprendizaje de habilidades complejas el cual no parte de conocimiento explícito (simbólico): este es el dominio de lo que se conoce como aprendizaje implícito (Berry y Dienes, 1993). En este caso, el sistema cognitivo puede llegar a formular las pautas que gobiernan la información sub-simbólica percibida, y por lo tanto, el conocimiento explícito derivaría del implícito (Stanley et al., 1989).

En contra de lo que se podría pretender con una arquitectura exclusivamente conexionista (sub-simbólica), es que no se puede eliminar el conocimiento explícito a favor del implícito. El conocimiento explícito es importante porque permite la aceleración del proceso de aprendizaje, facilita la transferencia de las habilidades de un dominio conocido

a otro nuevo, permite analizar qué pasa cuando la ejecución de una decisión no es exitosa, y desempeña un papel clave en la transmisión del conocimiento, y por lo tanto, en la instrucción. De esta manera, resulta necesario definir un mecanismo de "convivencia" entre estas dicotomías (conocimiento simbólico vs. sub-simbólico, aprendizaje explícito vs. implícito), que encapsule, en unidades especializadas, la adquisición de nuevas habilidades cognitivas (en este caso, en los *Comportamientos*).

Así pues, tomando como inspiración los fundamentos psicológicos mencionados en los párrafos anteriores, se propone un conjunto de *Comportamientos* coordinados por el módulo Comportamental, los cuales modelan el aprendizaje de habilidades. Cada *Comportamiento* define un conjunto de entradas sensoriales que filtran el estado actual percibido, así como un conjunto de salidas que determinan señales efectoras específicas las cuales son enviadas al módulo Efector para su procesamiento. Internamente, cada *Comportamiento* define un conjunto de componentes de naturaleza tanto explícita (en forma de reglas de producción simbólicas), como implícita (en forma de patrones de activación y de reglas sub-simbólicas). Específicamente, cada *Comportamiento* define tres sub-módulos, uno de procesamiento explícito (en el nivel cognitivista) denominado *Módulo de Producciones*, y dos de procesamiento implícito de naturaleza autopoiética (en el nivel emergente) denominados *Módulo Conexionista* y *Módulo Inmunológico*.

Cada *Comportamiento*, define además dos mecanismos de aprendizaje vertical, un aprendizaje descendente (top-down) donde el conocimiento explícito adquirido en el nivel cognitivista se asimila en el nivel emergente en forma de representaciones implícitas, y un aprendizaje ascendente (bottom-up) donde las representaciones implícitas se extraen del nivel emergente para formar estructuras explícitas en el nivel cognitivista. Esto coincide con las ideas de fundamento psicológico sobre el aprendizaje implícito y explícito mencionadas anteriormente.

Como observación adicional, antes de describir los sub-módulos de cada *Comportamiento*, cabe resaltar que los componentes que se sitúan en el nivel emergente del *Comportamiento* son de naturaleza *teleoreactiva* (un término acuñado por (Benson y Nilsson, 1995)): estos producen respuestas reactivas respecto al estado actual del mundo percibido (simulando los arcos reflejos básicos, mencionados en la sección 6.3.1.4), y a su vez toman en consideración la meta actual que debe ser alcanzada por el agente (y por consiguiente, son teleológicos[1]).

7.2. Estructura de un Comportamiento

La estructura común que comparten los *Comportamientos* (tanto genéricos como contextuales) corresponde a una tupla de elementos como la siguiente: $\langle$`P, ED, END, C, E, S`$\rangle$. Cada uno de estos elementos se describe a continuación:

- `P`: es el conjunto de pre-condiciones que deben cumplirse para que el *Comportamiento* pueda ser un candidato a la competición por el control consciente del

[1] Según RAE, la teleología es una doctrina filosófica que estudia las causas finales de las cosas. Norbert Wiener (Wiener et al., 1943) llamó sistemas teleológicos a los sistemas cibernéticos cuyo funcionamiento puede describirse como orientado a un fin.

agente. Estas pre-condiciones pueden conformarse por diversas unidades representacionales (e.g., perceptos, unidades de memoria declarativa, metas, motivadores, estados anticipados, etc.) expresadas en un formato del tipo ⟨`propiedad, valor`⟩ y almacenadas todas ellas en la Memoria de Trabajo.

- `ED`: es la lista de efectos deseados del *Comportamiento*, o en otras palabras, el conjunto de condiciones (ambientales e internas) que se esperan obtener después de la ejecución del *Comportamiento*. Esta lista puede considerarse como el aspecto teleológico del *Comportamiento* asociado. Estos efectos son expresados como parejas del tipo ⟨`propiedad, valor`⟩.
- `END`: es la lista de efectos no deseados. Contrariamente a la anterior, la lista `END` especifica las condiciones (ambientales e internas) que se deben evitar durante la ejecución del *Comportamiento*. Especifica además el estado en que debe permanecer el sistema después de la ejecución del *Comportamiento*. Estos efectos son también expresados como parejas del tipo ⟨`propiedad, valor`⟩.
- `C`: corresponde al conjunto de componentes que conforma el *Comportamiento* (i.e., RP, RES, RNA, y SIA).
- `E`: define las entradas específicas que procesarán todos los componentes del *Comportamiento*, una vez se satisfagan todas sus pre-condiciones y haya sido elegido por el módulo de Orquestación de la consciencia como el *Comportamiento* que controlará la conducta consciente del agente. Estas entradas se corresponden con un sub-conjunto de unidades que se encuentran activas en la MT[2], las cuales poseen una doble representación (tanto simbólica como sub-simbólica) y se encuentran expresadas como parejas ⟨`propiedad, valor`⟩. De lo anterior se puede deducir el hecho de que los *Comportamientos* atienden selectivamente un sub-conjunto de elementos que se encuentran activos en la MT, proceso que es llevado a cabo mediante el "filtrado" de las entradas (tanto externas como internas) al sistema.
- `S`: define el conjunto de salidas del sistema. Estas salidas pueden ser tanto acciones externas que se comunican a los actuadores del agente, como acciones internas que sirven como nuevas entradas a ciclos cognitivos de procesos más complejos. Las salidas son expresadas de igual forma como parejas del tipo ⟨`propiedad, valor`⟩.

En el presente capítulo se especificará el funcionamiento de los diversos componentes de cada *Comportamiento*, es decir, el modus operandi del elemento `C`. Los elementos restantes (`P`, `ED`, y `END`) describen las relaciones de activación, excitación, e inhibición entre los diferentes *Comportamientos* del agente, tema que será discutido en el capitulo 9. Por último, cabe resaltar que la diferencia esencial entre un *Comportamiento* genérico y uno contextual, es que este último *"extiende"* la funcionalidad del primero mediante la inclusión de nuevas unidades representacionales en los elementos `P`, `ED`, `END`, `E`, y `S`.

[2]Las dinámicas que determinan la permanencia y vigencia temporal de las unidades en la MT, las cuales sirven de contexto a la ejecución de los procesos conscientes y subconscientes, serán descritas al detalle en la sección 8.7. De momento, asúmase que las unidades de la MT corresponden tanto a estados externos o ambientales (i.e., perceptos) como estados internos del agente (i.e., motivadores, metas, estados anticipados, unidades de memoria declarativa, etc.) expresadas como parejas ⟨`propiedad, valor`⟩.

7.3. Módulo Conexionista (PMR)

Hasta hace unas décadas, la psicología cognitiva clásica no daba importancia a la base orgánica de la vida mental. Cabe recordar que incluso sus propios postulados básicos le llevaron a considerar que la lógica existente en los procesos mentales se puede comprender y explicar con independencia de su realización en un sistema físico, según lo que afirma la tesis del funcionalismo (Browne y Sun, 2001). El conexionismo, sin embargo, toma al cerebro como modelo de procesamiento y su inspiración neurofisiológica aproxima la brecha entre la psicología y la biología.

Aunque existen representaciones conexionistas de redes neuronales que intentan ser modelos de los procesos reales de partes del cerebro, las que interesan a las ciencias cognitivas no intentan imitar el comportamiento del sistema nervioso sino el de la mente. Las redes neuronales artificiales (RNAs) intentan reproducir las características principales del comportamiento del cerebro, particularmente de su estructura fisiológica básica, la neurona, y de su agrupación en estructuras complejas o redes de neuronas (McLeod et al., 1998).

En el módulo Conexionista de METÁFORA, el conocimiento es almacenado de forma *implícita* a través de las RNAs, lo que impide que éste pueda ser accedido directamente, aunque no, el que pueda ser procesado computacionalmente. A continuación, se enuncian los criterios en los cuales se basó la decisión de emplear las RNAs como un mecanismo de procesamiento de conocimiento sub-simbólico.

- *La simplicidad en el procesamiento*: puesto que se requiere que el procesamiento sub-simbólico permita al agente generar respuestas rápidas, en tiempo real, ante los cambios en las variables continuas del entorno percibido.
- *Clasificación de la información*: las RNAs pueden clasificar, de forma autónoma, las entradas que le son presentadas, es decir, realizan una clasificación de patrones en un espacio de búsqueda complejo, sin que sea necesario una supervisión o una función objetivo explícita.
- *Capacidad de Generalización*: las RNAs permiten al agente extrapolar respuestas para condiciones de entrada desconocidas, a partir de algunos patrones ya conocidos y deseables en el agente (estos patrones son definidos como reglas simbólicas en el módulo de Producciones de la arquitectura y representan el conocimiento innato del agente).
- *Representación distribuida y no localista*: los sistemas localistas retienen la información en unidades discretas, mientras que en las RNAs la información se retiene en poblaciones de elementos (neuronas), donde cada uno de ellos colabora de forma parcial al almacenamiento de la información. Esto permite que cada *Comportamiento* sea más robusto, ya que las consecuencias de contener información errónea en la RNA (e.g., un ajuste inadecuado de los pesos de los nodos) son menos devastadoras que en los sistemas localistas (e.g., en los sistemas simbólicos, como los sistemas expertos). No se puede dar una interpretación cognitiva a cada una de las unidades o neuronas; la interpretación cognitiva (el decir, por ejemplo, cómo se relacionan los conceptos) es algo válido sólo para la red en su conjunto, y en todo caso, para los patrones de los pesos de las unidades.

- *Computación sin reglas*: mientras que en el nivel de procesamiento simbólico de cada *Comportamiento* se elaboran reglas explícitas que recogen los aspectos esenciales que el sistema debe captar, con las RNAs es posible conseguir que el sistema reconozca patrones sin que sea necesario presentarle dichas reglas explícitas de reconocimiento, sólo basta con presentarle los ejemplos adecuados y luego entrenar la red. Esta peculiaridad lleva a caracterizar a las RNAs como sistemas dinámicos auto-adaptativos (Anderson, 1993a).
- *Degradación elegante y tolerancia a fallos*: la recuperación de la memoria en las RNAs distribuidas es capaz de darse aún en los casos en que la información de entrada es confusa, incompleta, e incluso errónea (lo que se suele llamar tolerancia a fallos respecto a los datos). La razón por la que las RNAs son tolerantes a fallos, es debido a la existencia de un cierto grado de redundancia en el procesamiento y almacenamiento de la información, es decir, a su carácter distribuido y paralelo.

Concretamente, en el módulo Conexionista se emplean redes neuronales del tipo Perceptrón Multicapa con propagación hacia atrás o Retropropagación (de ahora en adelante, PMR). Algunos autores (Rumelhart y McClelland, 1986; Sun et al., 2001; Seger, 1994; Cleeremans et al., 1998), han proporcionado suficiente evidencia con respecto al argumento de que las unidades en una PMR son capaces de llevar a cabo procesamiento computacional complejo aunque, debido a su característica de distribuidas, son unidades sub-simbólicas que carecen de significado por sí solas; esto es, que generalmente no tienen asociadas etiquetas semánticas.

La PMR está formada por múltiples capas que le permiten resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón simple (Anderson, 1993a). Las capas de una red PMR pueden clasificarse en tres tipos: una capa de neuronas de entrada constituida por aquellas neuronas que introducen los patrones de entrada en la red, en la cual no se produce procesamiento; una o varias capas de neuronas ocultas formadas por aquellas neuronas cuyas entradas provienen de capas anteriores y cuyas salidas pasan a las neuronas de las capas posteriores; y una capa de neuronas de salida cuyos valores de salida se corresponden con las salidas de toda la red. Usualmente, se emplea un algoritmo de aprendizaje de propagación hacia atrás (o retropropagación) el cual se encarga de corregir el error desde las capas de salida hacia las capas intermedias, modificando gradualmente los pesos sinápticos. Para información detallada acerca del modelo de Redes Perceptrón Multicapa y la importancia de las capas ocultas, el lector puede consultar a (Haykin, 1999).

7.3.1. Representación

Las entradas de la red neuronal se corresponden con el subconjunto de las unidades de información que almacena temporalmente la memoria de trabajo. Esta unidades recibidas por la red (y organizadas como parejas ordenadas del tipo ⟨`propiedad, valor`⟩) representan la información tanto ambiental (perceptos) como interna (e.g., motivadores, metas, información proprioceptiva del sistema, etc.) percibida por el agente.

Por otro lado, las salidas de la red neuronal corresponden a recomendaciones de acciones, las cuales pueden ser de dos tipos:

1. Acciones externas, las cuales ejecutan una acción concreta en el ambiente en el cual se encuentra inmerso el agente. Estas acciones se representan igualmente en un formato del tipo ⟨`propiedad, valor`⟩, donde cada *propiedad* se comunica con un tipo específico de *actuador* (e.g., un brazo robótico), y cada *valor* indica un comando que se pueda ejecutar sobre dicho actuador (e.g., rotar horizontalmente 45º).
2. Acciones internas, las cuales sirven como nueva entrada en la cadena causal que emplean los procesos cognitivos de orden superior (planificación, deliberación, etc.).

Un *patrón de activación* de la red neuronal es la pauta de activación espaciotemporal que representa un objeto, una acción o una abstracción (una idea, por ejemplo), como los códigos de barras de los envases, que representan una cosa sin parecerse a ella. Cuando el agente recibe información sensorial de un plátano, por ejemplo, son activadas varias neuronas en la capa de entrada de la red: algunas de esas neuronas están especializadas en el color amarillo; otras en las cortas líneas rectas tangentes a la curva del plátano, etc. Evocar un recuerdo consiste, simplemente, en reconstruir esa pauta de actividad, según la hipótesis de la asamblea de células, postulada por el psicólogo Donald Hebb (Hebb, 1949).

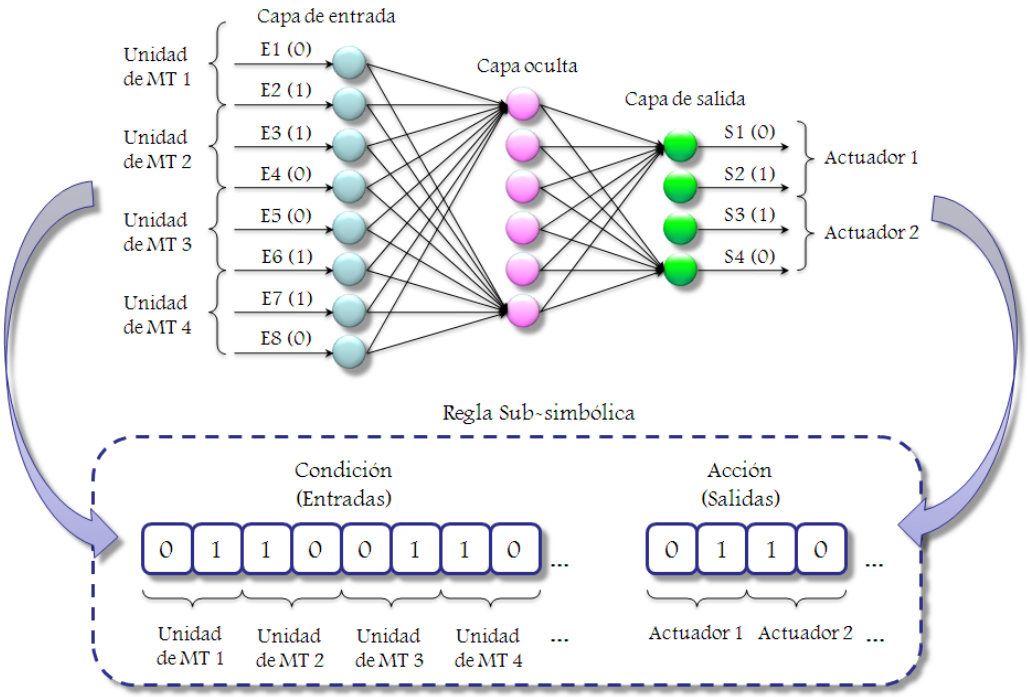


Figura 7.1: Representación de una red PMR. Las conexiones entre los nodos de la red aparecen simplificadas con el fin de no complicar visualmente el entramado sináptico de la red. En la parte inferior se observa una regla sub-simbólica extraída de la operación de la red PMR.

En la figura 7.1, se presenta un ejemplo de una red PMR. En la capa de entrada, la red recibe un conjunto de datos con valores de "0" ó "1" *(E1, E2,... En)*. Cada entrada y su

respectivo *valor* corresponden a las pautas ⟨`propiedad, valor`⟩ que componen cada elemento percibido y almacenado en la MT, donde la *propiedad* indica la característica percibida y el *valor* indica si está presente o no dicha característica, con un valor de 1 o 0 respectivamente. De la misma manera, las salidas *(S1, S2,... Sn)* se encuentran agrupadas por pautas del tipo ⟨`propiedad, valor`⟩, donde cada salida y su respectivo *valor* corresponden a un comando que se envía a un actuador específico del agente.

Considérese un agente que navega en un ambiente virtual. Tomando como ejemplo la red PMR de la figura 7.1, se tiene un primer *percepto* (unidad-MT 1) que identifica la presencia de un objeto determinado justo en frente del agente, el cual tendría dos pautas posibles: ⟨`objeto, presente`⟩ u ⟨`objeto, no-presente`⟩; un segundo *percepto* (unidad-MT 2) que identifica la presencia de un obstáculo: ⟨`obstáculo, presente`⟩ u ⟨`obstáculo, no-presente`⟩; un tercer percepto (unidad-MT 3) para identificar la presencia de otros agentes: ⟨`otros-agentes, presentes`⟩ o ⟨`otros-agentes, no-presentes`⟩; y una meta actual (unidad-MT 4): ⟨`buscar, objeto`⟩.

Por otro lado, el *Actuador 1* tendría las salidas: ⟨`moverse, hacia-adelante`⟩ o ⟨`moverse, hacia-atrás`⟩; y el *Actuador 2* tendría las salidas: ⟨`girar, derecha`⟩ o ⟨`girar, izquierda`⟩. Dado que la meta del agente es buscar un objeto, cuando éste lo percibe en frente, las salidas de la red enviarán comandos a los actuadores que permitan que el agente se aproxime al objeto, mientras que si lo que percibe el agente en frente es un obstáculo, las salidas corresponderán a un comportamiento evasivo. Este ejemplo, aunque demasiado simple, resulta útil para describir la agrupación en pautas ⟨`propiedad, valor`⟩ tanto en las entradas como en las salidas de la PMR, y cómo estás crean relaciones hetero-asociativas a través de la red PMR con el fin generar un comportamiento dirigido hacia el logro de las metas. Esta característica de estructuración en pautas ⟨`propiedad, valor`⟩ se verá reflejada además en los distintos componentes de procesamiento del *Comportamiento*, como se verá en secciones posteriores.

Adicionalmente, en la figura 7.1 se observa el formato de extracción de una regla sub-simbólica conformada por una parte condición y una parte acción, las cuales corresponden, respectivamente, a un patrón de activación específico de entradas y salidas de la red. Esto es lo que habitualmente se conoce por memoria asociativa: cuando se aplica un estímulo (dato de entrada), la red responde con una salida asociada a dicho estímulo. Más concretamente, en la red PMR se produce una *hetero-asociación*, que se refiere al caso en el que la red aprende parejas de datos *[(A1, B1), (A2, B2),... (An, Bn)]*, de tal forma que cuando se presente cierta información de entrada *Ai*, la red responde generando la correspondiente salida asociada *Bi*. Una red hetero-asociativa podría considerarse como aquella que computa cierta función, que en la mayoría de los casos no podría expresarse analíticamente, entre un conjunto de entradas y un conjunto de salidas, correspondiendo a cada posible entrada una determinada salida. Esta correspondencia entre los datos de entrada y de salida es extraída por el módulo Conexionista como una regla sub-simbólica hetero-asociativa, la cual puede ser luego traducida a una regla simbólica (e.g. una regla de producción RES) a través de los respectivos *Transductores*.

Esta regla sub-simbólica es extraída durante la operación de la red neuronal, en el momento en que las consecuencias de las acciones (externas o internas) asociadas a un

patrón específico de entradas, son evaluadas como "favorables". La determinación de si las acciones de la red neuronal son favorables o no, viene dada por la retroalimentación ambiental. El proceso de extracción de las reglas sub-simbólicas y su utilidad en la construcción de nuevas estructuras simbólicas en el nivel cognitivista, es descrito en la sección 7.5.2.

En cuanto al diseño de la arquitectura de cada red PMR, existen tres consideraciones importantes que son tomadas en cuenta:

1. ¿Qué tipo de asociaciones deben definirse en el diseño de la red?
2. ¿Cuántas capas debería tener la red y cuántas neuronas en cada capa?
3. ¿Qué funciones de activación son las más adecuadas para la red?

Estas cuestiones han sido tratadas por diversos investigadores de forma independiente. El resultado más interesante ha sido el formulado por (Cybenko, 1989) y (Hornik et al., 1989), quienes proponen que una arquitectura de red PMR de *tres capas* con funciones de activación *sigmoidales* en la capa oculta, y funciones de activación *lineales* en la capa de salida, funcionan adecuadamente como *aproximadores universales*. Es importante destacar que las redes hetero-asociativas, como la red PMR, al asociar informaciones de entrada con diferentes informaciones de salida, precisan al menos de tres capas, una para captar y retener la información de entrada, otra que sirva como memoria asociativa, y otra para mantener la salida con la información asociada. Si esto no fuese así, se perdería la información inicial al obtenerse un dato asociado de la capa de entrada, lo cual no debe ocurrir, ya que en el proceso de obtención de la salida se puede necesitar acceder varias veces a esta información que, por tanto, deberá permanecer en la capa de entrada, mientras la capa oculta sirve de capa de procesamiento y asociación.

En cuanto a estas cuestiones, lo que no se ha llegado a especificar concretamente es el número de neuronas que debe tener la capa oculta. Este último punto se ha tratado en diversos trabajos llegando siempre a resultados poco concluyentes o demasiado difíciles de aplicar en la práctica. En general, las soluciones apuntan a una estrategia iterativa de "prueba y error" donde se logra una convergencia aceptable de la red a través de continuas modificaciones en el número de neuronas en la capa oculta y de ajustes posteriores en las fases de entrenamiento. Como se mencionó anteriormente, las redes PMR funcionan como aproximadores universales, y con respecto a esto Haykin (Haykin, 1999) ha demostrado que cuantas más neuronas haya en la capa oculta, la aproximación será más exacta. No obstante, cuantas más neuronas existan en la capa oculta, se requerirá mayor procesamiento por parte de la red y mayor será el tiempo de entrenamiento requerido para alcanzar la convergencia en un máximo global.

Los resultados presentados por (Cybenko, 1989), pueden dar a entender que no hay necesidad de utilizar más de una capa oculta y/o mezclar diferentes funciones de activación diferentes de las mencionadas. Sin embargo, es posible que la precisión o la facilidad de aprendizaje aumente utilizando arquitecturas más sofisticadas. En particular, cuando en la función que se desea aprender, la complejidad de la relación entre entradas y salidas es alta, se puede mejorar la precisión en la aproximación con arquitecturas más complejas. Si embargo, dado que la implementación, el aprendizaje y el análisis de la

red resultante pueden complicarse, es habitual utilizar una sola capa oculta con neuronas sigmoidales o tangentes hiperbólicas (Haykin, 1999).

Considerando lo anteriormente expuesto, como criterio de diseño en METÁFORA se define por defecto, para cada red PMR, una sola capa oculta con un número de neuronas equivalente al doble de la capa de entrada, en el caso de que ésta sea la capa con mayor número de neuronas, de lo contrario, equivalente al doble de neuronas de la capa de salida. En las neuronas de la capa oculta, se empleará por defecto una función de activación sigmoidal (ecuación 7.1) y en las neuronas de la capa de salida, una función de activación lineal (ecuación 7.2). Con esto se pretende garantizar una convergencia aceptable de la red.

$$f = \frac{1}{1 + e^{-x}} \tag{7.1}$$

donde: *x* corresponde al vector de entrada.

$$f = \sum_{j=1}^{n} w_j \times x_j \tag{7.2}$$

donde: *w* corresponde al peso entre la neurona oculta y la neurona de salida *j*.

Las redes con activación por umbral son difíciles de entrenar debido a que la función de error se vuelve constante poco a poco, y por tanto el gradiente o no existe o es cero, haciendo imposible realizar una retropropagación o emplear métodos de entrenamiento basados en gradiente que sean más eficientes. Por tal motivo, en la capa oculta de cada red PMR se emplean (como se dijo anteriormente) funciones de activación sigmoidal, las cuales incorporan la característica de "no-linealidad" deseada para tratar con datos de entrada continuos y no lineales (propiedad típica de la red perceptrón multicapa) (McCullagh y Nelder, 1989). En las neuronas con funciones sigmoidales, un pequeño cambio en los pesos usualmente producirá un cambio en las salidas, lo cual permite determinar si el cambio en los pesos ha sido "bueno" o "malo". Por otro lado, si se emplean funciones lineales, un pequeño cambio en los pesos con frecuencia no producirá ningún cambio en las salidas.

En cuanto a la capa de salida, la función de activación lineal resulta ser una buena opción si los valores objetivo son continuos y sus límites no son conocidos (Jordan, 1995). Sin embargo, cada concreción contextual de la arquitectura es susceptible de ser parametrizable, a través del módulo Meta-cognitivo, de acuerdo con el tipo de situaciones donde se empleará el agente. De esta manera, es posible configurar funciones de activación tales como la función binaria o escalón, la función identidad, la función tangente hiperbólica, o una función gaussiana. Según la función escogida se tendrá un mayor o menor determinismo en el comportamiento del modelo y la selección de la función dependerá directamente del tipo de datos objetivo que se requieran para la concreción contextual en particular.

Tómese en cuenta la representación propuesta para la red PMR descrita en la sección anterior. Suponiendo el caso de que se desea diseñar un agente que interactúa con entornos moderadamente realistas, donde una cantidad de 128 entradas sensoriales resulta

un valor aceptable, y asumiendo que cada una de estas entradas puede estar simplemente activa o inactiva (1 ó 0); el número total de posibles situaciones que se pueden representar asciende a 2^{128}, un número descomunalmente alto si se pretende que la red aprenda cada uno de estas situaciones.

Por otro lado, supóngase que el agente puede ejecutar 16 tipos diferentes de acciones para cada situación percibida y que no posee conocimiento alguno de cuál ejecutar en cada momento. Una aproximación de prueba y error no sería lo suficientemente apropiada para encontrar la acción correcta a cada situación (no sería viable ni eficiente que el agente probase por cada entrada, cada una de las 16 acciones).

Debido a estas suposiciones, que en efecto podrían tener consecuencias aún más dramáticas (e.g., un mayor número de entradas sensoriales o de salidas posibles), surge la necesidad de definir cuatro mecanismos que solucionan el problema antes descrito: el aprendizaje, el olvido, el refuerzo, y la generalización.

- *Aprendizaje:* la red PMR aprende nueva información que se registra de forma distribuida en los pesos asociados a las conexiones entre neuronas. Por tanto, puede imaginarse una red PMR como cierto tipo de memoria que almacena datos de forma estable, datos que se graban en dicha memoria como consecuencia del aprendizaje de la red y que pueden ser leídos a la salida como respuesta a cierta información de entrada. De esta manera la red se comporta como una memoria asociativa: los estímulos (o datos de entrada) generan conexiones con las neuronas conceptuales (ubicadas en las capas ocultas o intermedias de la red), con el fin de almacenar el reconocimiento de situaciones específicas, y a su vez, las neuronas conceptuales generan conexiones con las neuronas de salida apropiadas, con el fin de establecer una asociación entre la información de entrada y la salida activada.
- *Olvido:* debido al inmenso número de situaciones diferentes en las que el agente puede verse envuelto, no resultaría conveniente almacenar cada tupla de ⟨`entrada, salida`⟩ que el agente experimenta. Es por esto que resulta necesario aplicar un mecanismo de "olvido", el cual consiste en reducir los pesos entre aquellas conexiones sinápticas que dejan de emplearse o que tienen muy poca activación. Este proceso de auto-ajuste de los pesos sinápticos permitirá, a la red PMR, retener en la memoria asociativa aquello que es importante y olvidar aquello que es irrelevante. En la presente tesis, se emplean dos mecanismos de aprendizaje/olvido para la red PMR (los cuales serán explicados al detalle en el siguiente apartado): un aprendizaje supervisado basado en el algoritmo de retropropagación (backpropagation) donde la red es estrenada a partir de un conjunto de patrones deseables, y un aprendizaje por refuerzo basado en el algoritmo *Q-learning* donde la red aprende/olvida como consecuencia de la modulación de una señal de retroalimentación que ajusta los pesos sinápticos.
- *Refuerzo:* como se ha mencionado en capítulos anteriores, las neuronas funcionan como *integradores* de señales: reciben señales de otras neuronas y cuando la acumulación de éstas sobrepasa un umbral de activación, la neurona se dispara[3]. En el

[3]Esta definición concuerda con dos ideas expuestas en capítulos anteriores: el funcionamiento del bloque cibernético *Integrador*, y el propósito de uno de los procesos computacionales básicos que conforman el SNS del agente.

caso de las neuronas de salida, se dispara una acción que el agente ejecutará interna o externamente. Sin embargo, estas acciones no siempre conducen a consecuencias favorables. Si una acción resulta ser ventajosa ante una situación dada, el agente debe asegurarse de que dicha acción sea ejecutada con mayor probabilidad en el futuro, lo que se logra mediante una conexión que realiza la red entre la condición y la acción que excita las neuronas de salida cuando dicha condición se cumple. Por otro lado, si en una situación dada la acción resulta ser inconveniente para el agente, éste debe asegurarse de que en el futuro esa misma situación descarte la acción recomendada previamente, tendiendo gradualmente a inhibir la activación de la neurona de salida.

El proceso por el cual se "recompensan" las decisiones favorables y se "castigan" las poco convenientes, se denomina *refuerzo*, el cual es un tipo de retroalimentación que entra al sistema y se encarga de actualizar los pesos sinápticos entre las neuronas de la red, "aprendiendo" aquellas decisiones que pueden ser repetibles en el futuro, y "olvidando" aquellas decisiones que no conducen a experiencias beneficiosas para el agente. Sin embargo, en entornos realistas, este refuerzo no siempre se recibe de forma inmediata, y no siempre es discreto, sino que por el contrario, puede que se reciba con cierto desfase e ir disminuyendo/aumentando gradualmente. Por tal motivo, la señal de refuerzo requiere de un *modulador*, encargado de amplificar o reducir dicha señal con el objetivo de que el refuerzo pueda recibirse de forma continua, dirigiendo gradualmente la adquisición u olvido de conocimiento en la memoria de la red. El mecanismo de refuerzo que gestiona METÁFORA se explica con más detalle en la sección 10.2.

- *Generalización*: se entiende por generalización, la habilidad de una red neuronal de almacenar, en sus pesos sinápticos, las características que son comunes a todos los patrones de activación aprendidos, de tal manera que cuando se presente a la red una nueva situación de entrada, ésta sea capaz de calcular la salida más aproximada que corresponda. Cuando un nuevo conjunto de entradas es almacenado, la fuerza sináptica entre las conexiones existentes, representadas por los pesos, son modificadas sutilmente. Si estas modificaciones refuerzan los pesos previamente almacenados, entonces ocurre la generalización. Los patrones generalizados son la suma de todas las modificaciones aisladas que han ocurrido. Es decir, la red neuronal es capaz de generalizar los patrones de activación que le son presentados en la medida que module los cambios ocurridos en sus pesos.

  En el caso de que la red PMR emplee un aprendizaje supervisado (algoritmo de retropropagación) se requiere tener en cuenta algunas consideraciones de diseño en cuanto a los patrones de entrenamiento[4] a fin de que la red pueda realizar una generalización correcta de los mismos[5]:

  - Si el problema es de clasificación, los ejemplos que formen parte del entrenamiento deben representar todas las categorías. La red puede aprender mejor

[4] Los patrones de entrenamiento son un conjunto de pares representativos entrada/salida (donde las entradas son los posibles estados observados del mundo y las salidas son las respuestas deseadas de la red), los cuales son aprendidos y generalizados durante una fase de entrenamiento de la red neuronal.

[5] Estas consideraciones no resultan necesarias para el caso del algoritmo *Q-learning* puesto que el aprendizaje, en este caso, está en función de los refuerzos ambientales que reciba la red en tiempo de ejecución, y no de un conjunto de patrones de entrenamiento definidos a priori.

si se dispone de una mayor cantidad de ejemplos para cada una de las categorías.

- Si el problema es de aproximación o predicción, se deben incluir ejemplos de valores continuos a través de todo el rango. Es decir, de la nube de ejemplos de entrenamiento, la muestra debe contener ejemplos de los bordes y del centro para un mejor entrenamiento.

7.3.2. Aprendizaje

La estructura y conectividad de cada red PMR puede ser definida a priori o bien puede ser ajustada en tiempo de ejecución (*on-line*) a través de un mecanismo de aprendizaje (epigénesis de la red PMR). El primer caso corresponde a los *arcos reflejos* innatos que son modelados por los procesos filogenéticos del agente o mediante un diseño a priori. El segundo caso, representa la capacidad de la red de auto-organizarse durante la continua interacción con el entorno. Esta auto-organización implica un reajuste de los pesos sinápticos a través del aprendizaje, el cual puede ser supervisado (dirigido por el nivel cognitivista), o por refuerzo (dirigido por una señal de retroalimentación). En este orden de ideas, se propone un algoritmo de retropropagación de errores (*backpropagtion*) que coordina el aprendizaje supervisado, y un algoritmo de aprendizaje denominado *Q-learning* que coordina el aprendizaje por refuerzo.

7.3.2.1. Aprendizaje Supervisado

En el caso de disponer de un conjunto de patrones de entrenamiento (e.g., conocimiento innato en el nivel cognitivista, patrones pre-establecidos por un diseñador, etc.), es posible aplicar directamente un algoritmo de retropropagación que emplea, como medida del error, la diferencia absoluta entre la salida deseada y la salida actual. Considérese un modelo de tres capas como el descrito en la sección anterior. El cálculo de la salida de la red sería:

$$z_k^\mu = \sum_j w'_{kj} h_j^\mu - \theta'_k = \sum_j w'_{kj} f(\sum_i w'_{ji} x_i^\mu - \theta'_j) - \theta'_k \tag{7.3}$$

donde: z_k^μ corresponde a la salida de la red. x_i^μ es la capa de entrada con un umbral de disparo θ'_k. h_j^μ es la capa oculta con un umbral θ_j. w' son los pesos sinápticos entre las neuronas de cada capa.

Supóngase un conjunto de patrones de entrada x^μ, ($\mu = 1 \dots p$, con p el número de patrones disponibles). Así pues, la función de error que se intenta minimizar corresponde al error cuadrático medio:

$$error(w_{ji}, \theta_j, w'_{kj}, \theta'_k) = \frac{1}{2} \sum_\mu \sum_k \{t_k^\mu - f(\sum_j w'_{kj} h_j^\mu - \theta'_k)\}^2 \tag{7.4}$$

donde: t_k^μ es la salida deseada (target).

Después de obtenida la primera salida de la red y de haber calculado el error, se calculan los gradientes locales encargados de ponderar el cambio de los valores en los pesos sinápticos. Los pesos entre las neuronas de la capa oculta y la capa de salida se calculan de la siguiente manera:

$$\Delta w_{kj} = \alpha x_{kj} \delta k \tag{7.5}$$

donde: $\delta_k = error_k o_k(1 - o_k)$. w_{kj} es el peso asociado con j-ésima entrada a la neurona de salida k (que proviene de la j-ésima neurona de la capa oculta). x_{kj} es la j-ésima entrada a la neurona de salida k. α es una tasa de aprendizaje.

De esto se obtiene que:

$$\delta_k = error_k o_k(1 - o_k) \tag{7.6}$$

donde: o_k es la salida resultante para la neurona de salida k.

El ajuste de los pesos entre las neuronas de la capa de entrada y la capa oculta se calcula de la siguiente manera:

$$\Delta w_{ji} = \alpha \delta_j x_{ji} \tag{7.7}$$

donde: w_{ji} es el peso asociado entre la i-ésima neurona de entrada y la neurona oculta j. x_{ji} es la i-ésima entrada a la neurona oculta j. α es la tasa de aprendizaje.

De esto se obtiene que:

$$\delta_j = o_j(1 - o_j) \sum \delta_k w_{kj} \tag{7.8}$$

donde: o_j es la salida de la neurona oculta j. k denota todo el flujo de activación de las. neuronas en la capa de salida. w_{kj} es el peso asociado entre la entrada j y la k-ésima neurona de salida. δ_k es igual al $error_{Ok}(1 - o_k)$.

El pseudocódigo empleado para el algoritmo de retropropagación se presenta en el cuadro 7.1. [6]

Cabe resaltar en este punto, que los patrones de entrenamiento que emplea la red PMR en el aprendizaje supervisado guiado por el algoritmo de retropropagación, provienen de dos fuentes:

1. Del conocimiento experto almacenado en forma de Reglas de Producción, las cuales son predefinidas por el diseñador en el nivel cognitivista del *Comportamiento*.
2. Como resultado del proceso filogenético donde nuevos "memes" son generados y transmitidos entre los agentes del sistema. En este caso, los memes son incorporados en el nivel simbólico de cada *Comportamiento* respectivo y representan el conocimiento innato que guiará los procesos ontogenéticos y epigenéticos durante el ciclo de vida del agente.

[6] A fin de identificar los parámetros de entrada y de salida que emplea el algoritmo, las variables se denotan con el símbolo $\downarrow$ o $\uparrow$ respectivamente. Esta notación se empleará de ahora en adelante en todos los algoritmos propuestos.

Método: retropropagar

```
1.  Variables
2.     pt            ↓ conjunto de patrones de entrenamiento.
3.     red           ↑↓ estructura de la red neuronal.
4.  Inicio Procedimiento
5.     red ← inicializar() //Establece aleatoriamente los pesos y umbrales
       iniciales de la red.
6.     Mientras que todos los patrones hayan sido clasificados
       correctamente o hasta que se alcance algún criterio de paro
       (definido por el módulo Meta-cognitivo) hacer:
7.        Para cada patrón μ del conjunto de patrones pt hacer:
8.           z_k^μ ← ejecutarRedHaciaAdelante(red) //se lleva a cabo una fa-
             se de ejecución para obtener la respuesta de la red empleando como
             entrada el patrón μ - ésimo, según la ecuación 7.3.
9.           errorSalida ← calcularErrorCapaSalida(red.salida)
             //se calcula la señal de error asociada a la capa de salida, empleando
             la ecuación 7.4.
10.          gradientes ← calcularGradientes(red) //se computan los
             gradientes locales entre la capa oculta y la capa de salida, y entre la
             capa de entrada y la capa oculta. Se emplean las ecuaciones 7.5 y 7.7.
11.          actualizar(red) //se actualizan los pesos y umbrales de la red.
12.       Fin Para
13.    Fin Mientras
14. Fin Procedimiento
```

Algoritmo 7.1: Algoritmo de Retropropagación para las redes PMR.

7.3.2.2. Aprendizaje Por Refuerzo

El algoritmo de *Q-learning*, propuesto por Watkins (Watkins, 1989; Watkins y Dayan, 1992), pertenece al grupo de los algoritmos de aprendizaje por refuerzo. El algoritmo *Q-learning* define la función $Q(x, a)$ como aquella que estima el máximo refuerzo acumulativo que puede ser recibido en el estado actual x, después de que la acción a ha sido desempeñada, es decir, emplea una política de aprendizaje basada en el retorno a largo plazo:

$$\forall x \forall a : Q(x, a) = \max_{a_i : i=1,2,3...} \left(\sum_{i=0}^{\infty} \gamma^i r_i\right) \tag{7.9}$$

donde: γ es un factor de descuento ($0 \leq \gamma \leq 1$) que favorece los refuerzos que se reciben inmediatamente en contraste con aquellos que se reciben en pasos posteriores. a_i es la acción que puede ser desempeñada en el paso i (con $a_0 = a$). r_i es el refuerzo recibido en el paso i (el cual puede ser positivo, negativo o nulo).

La secuencia de acciones se construye a partir de los valores Q, los cuales proveen una estimación del refuerzo total en un enfoque paso a paso. La actualización de $Q(x, a)$ se basa en:

$$\forall x \forall a : \Delta Q(x, a) = \alpha(r + \gamma \times e(y) - Q(x, a)) \tag{7.10}$$

donde: γ es un factor de descuento. α es una tasa de aprendizaje. y es el nuevo estado resultante de ejecutar la acción a en el estado x. $e(y) = \max_b Q(y, b)$. Nótese que x y y son conjuntos que se componen de las unidades activas en la MT (i.e., perceptos, metas, motivadores, etc.).

La tasa de aprendizaje (α) determina hasta qué punto la información nueva adquirida reemplazará la información antigua. Un factor de 0 hará que el agente no aprenda nada, mientras que un factor de 1 haría que agente considerara sólo la información más reciente. Por otro lado, el factor de descuento (γ) determina la importancia de futuros refuerzos. Un factor de 0 hará al agente más "oportunista" al considerar sólo los refuerzos actuales, mientras que un factor de 1 lo haría esforzarse por obtener un refuerzo más alto en el largo plazo.

Por consiguiente, la actualización se basa en una *diferencia temporal* que evalúa el estado actual y la acción escogida. En la fórmula 7.10, la función $Q(x, a)$ estima el refuerzo total a ser recibido en el estado actual *antes* de que la acción a sea ejecutada, y $r + \gamma e(y)$ estima el máximo refuerzo total que se podría obtener desde el estado actual hasta el estado resultante *después* de que la acción a sea ejecutada; de esta manera, la diferencia temporal resultante de evaluar una acción permite el aprendizaje de los valores Q, los cuales se aproximan al refuerzo total. Intuitivamente, es posible observar que el algoritmo *Q-learning* permite la emergencia de un comportamiento secuencial y causal, debido a las estimaciones que realiza sobre los refuerzos que obtendrá la red en estados futuros.

Existe evidencia (Lin y Mitchell, 1992; Kuzmin, 2002; Sun et al., 2001) de que el algoritmo de *Q-learning* puede dirigir el proceso de aprendizaje en redes neuronales del tipo PMR. Cuando se aplica el algoritmo de *Q-learning*, el entrenamiento de retropropagación de la red se enfoca en la minimización del error en cada paso, como se observa en la ecuación 7.11:

$$\forall x \forall i : error_i = \begin{cases} r + \gamma e(y) - Q(x, a_i), & \text{si } a_i = a; \\ 0, & \text{de lo contrario.} \end{cases} \tag{7.11}$$

donde: i es el índice para un nodo de salida de la red que representa la acción a_i. a es la acción que se ejecuta actualmente. x es el estado actual.

Basado en esta medida del error, el algoritmo de retropropagación es aplicado para ajustar los pesos internos de la red. Nótese que estos pesos son inicializados aleatoriamente antes de iniciar el entrenamiento de la red. En el caso de que existan múltiples tipos de acción[7], este proceso es llevado a cabo para cada una de estas acciones, separadamente. El pseudocódigo del algoritmo de *Q-learning* para las redes PMR se presenta en el cuadro 7.2.

Algunos de los aspectos más importantes por los cuales resulta útil emplear un aprendizaje por refuerzo, y en especial el algoritmo de *Q-learning* son:

[7]Por ejemplo, en el caso de modelar un animal virtual, éste podría definir una acción diferente para cada actuador: una para el movimiento del brazo, una para el movimiento de la cabeza, etc.

Método: ejecutarQlearning

```
1.  Variables
2.     red            ↑↓ representa la estructura de la red neuronal.
3.  Inicio Procedimiento
4.     inicializar(red, Q) //se inicializan los pesos de la red PMR y los valores
       Q de las salidas de la misma.
5.     Mientras que la red PMR se encuentre activa hacer:
6.        x ← estadoActual //se asigna el estado actual percibido por el agente
7.        a ← ejecutarRedHaciaAdelante(red) //la red PMR realiza una ac-
          tivación hacia adelante (feedforward), computando la capa oculta y poste-
          riormente la capa de salida. Se escoge una acción a como resultado.
8.        ejecutarAccion(a) //se ejecuta la acción a.
9.        y ← nuevoEstado; r ← refuerzoActual: //se establece y como
          el nuevo estado resultante de ejecutar a, y r como el refuerzo recibido por
          la acción ejecutada.
10.       actualizarQ(Q) //se actualiza la estimación del valor para Q(x, a), como
          en la ecuación 7.10.
11.       error ← calcularErrorSalida(red) //se calcula el error de la sali-
          da, error_i, como en la ecuación 7.11.
12.       retropropagar(red) //se ejecuta el algoritmo de Retropropagación y se
          actualizan los pesos de la red.
13.       actualizarQ(Q) //se actualiza la política de optimización f: f(x) ← a,
          tal que Q(x, a) = máx_{b∈a} Q(x, b).
14.    Fin Mientras
15. Fin Procedimiento
```

Algoritmo 7.2: Algoritmo de Q-learning para las redes PMR

- *Q-learning* es un algoritmo de aprendizaje capaz de aprender en tiempo de ejecución (*on-line*) a partir de ejemplos de entrenamiento individuales y aleatorios, prescindiendo del lote de ejemplares pre-determinados (patrones de entrenamiento) que se requiere en el aprendizaje supervisado.

- Tiene la capacidad para proporcionar salidas continuas respecto a entradas continuas.

- Su capacidad de generalización permite una convergencia rápida de las predicciones para todos los puntos de un espacio de estados continuos.

Como observación final, cabe destacar que tanto el mecanismo de aprendizaje supervisado como el mecanismo de aprendizaje por refuerzo en la red PMR son sistemas de lazo cerrado, puesto que emplean una señal de retroalimentación para coordinar las dinámicas de ajuste de pesos sinápticos: en el aprendizaje supervisado, el algoritmo de retropropagación emplea el error cuadrático como señal de retroalimentación que propaga la minimización de éste hacia las capas anteriores; y en el aprendizaje por refuerzo, la retroalimentación es una señal (positiva, negativa, o nula) que procesa el algoritmo de *Q-learning* para ajustar los pesos entre las conexiones sinápticas de la red.

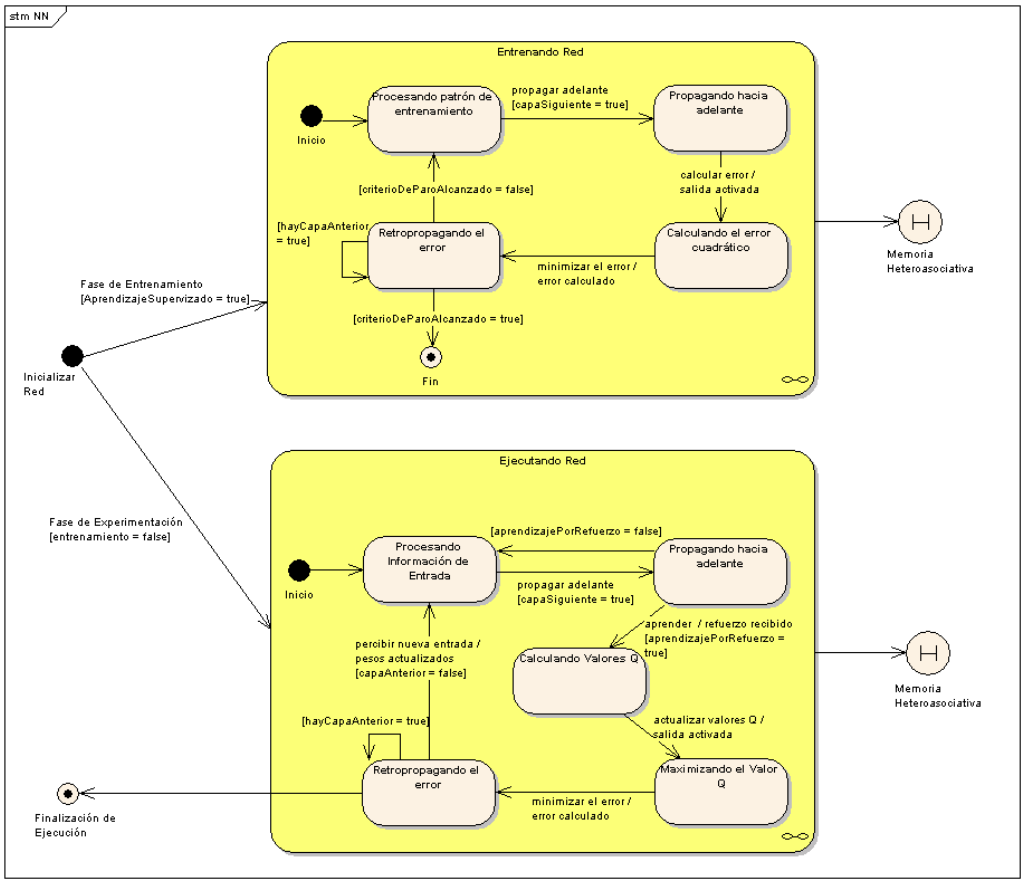


Figura 7.2: Diagrama de Transición de Estados para la red PMR.

7.3.3. Máquina de Estados para la red PMR

En la figura 7.2 se presenta un diagrama de Transición de Estados para la red PMR. El diagrama resume el proceso de ejecución en dos fases: una de Entrenamiento y otra de Experimentación, cada una representada por un super estado. En la fase de Entrenamiento[8], son presentados ante la red un conjunto de patrones sobre los cuales ésta aprende, empleando para ello un mecanismo de aprendizaje supervisado dirigido por el algoritmo de Retropropagación.

En la fase de Experimentación, la red interactúa directamente con las entradas que le proporciona la MT (i.e., perceptos, unidades de memoria declarativa, metas, etc.). En esta fase no se efectúa un aprendizaje supervisado, por el contrario, la red opera en tiempo de ejecución (*on-line*). En el caso de estar activado el aprendizaje por refuerzo, el estado ⟨`Propagación hacia adelante`⟩ dispara una transición que activa el estado ⟨`Calculando Valores Q`⟩ y éste a su vez activa el estado ⟨`Maximizar el Valor Q`⟩, donde es obtenida la diferencia temporal del error que se desea minimizar; los pesos de la red son actualizados en el estado ⟨`Retropropagación del`

[8]Este es un entrenamiento fuera de línea (*off-line*) que se realiza cada ciertos intervalos de tiempo, o ciclos cognitivos, definidos por el módulo Meta-cognitivo.

`error`⟩. En el caso contrario, la red simplemente opera entre los estados ⟨`Procesando información de entrada`⟩ y ⟨`Propagación hacia adelante`⟩, pero durante la interacción entre estos dos estados no se efectúa ninguna corrección del error ni ningún aprendizaje de nuevas entradas, es decir, la red sólo efectúa la activación hacia adelante simplificando el procesamiento, y por ende, proporcionando al *Comportamiento* una respuesta *reactiva*.

Los dos super estados producen, en un nodo de historia, un registro de su interacción interna que representa la memoria hetero-asociativa de la red, pero sólo se genera en los casos en que se ejecuta algún tipo de aprendizaje (supervisado o por refuerzo).

7.4. Módulo Inmunológico (SIA)

El módulo Inmunológico, se basa en un Sistema Inmune Artificial (de ahora en adelante SIA), el cual se inspira en principios inmunológicos observados en la naturaleza (ver secciones 2.1.3 y 2.2.4.4). En la presente tesis se propone un algoritmo basado en el modelo de red inmune conocido como *AiNET* (por su acrónimo en inglés de Artificial Immune Network)(N. de Castro y Timmis, 2002) aunque con algunas adaptaciones que se detallarán más adelante. Los criterios por los cuales se ha decidido emplear este modelo computacional inspirado biológicamente, como apoyo al procesamiento realizado por el módulo Conexionista en el nivel emergente de cada *Comportamiento*, se enuncian a continuación:

- *Reconocimiento de Patrones*: el SIA puede reconocer y clasificar diferentes patrones y generar determinadas respuestas. Se generaliza el uso de los anticuerpos, encargados de detectar la presencia de cuerpos extraños, en el reconocimiento de dichos patrones y sus respuestas asociadas.
- *Generación de Diversidad*: esta característica ocasiona que el sistema inmune pueda reconocer una variedad casi ilimitada de antígenos (entradas sensoriales externas y estados internos), haciendo que éste sea más eficiente.
- *Aprendizaje*: el sistema inmune es capaz de aprender de las experiencias, identificando la manera de atacar a los cuerpos extraños, o en el caso del SIA, de generar respuestas adecuadas a las entradas percibidas. Este mecanismo hace que se multipliquen los anticuerpos[9] más eficientes y que disminuyan los poco eficientes, para así poder preservar la integridad del sistema. También este tipo de mecanismo es conocido como expansión clonal.
- *Memoria*: este mecanismo mantiene un registro de lo que ha provocado algún tipo de beneficio o daño al sistema, esto es, el sistema tiene la capacidad de recordar y mantener un registro de las experiencias deseadas y no deseadas, pudiendo luego clonar (auto-replicar) estos anticuerpos (a través de procesos darwinianos) e incrementar la probabilidad de responder adecuadamente en estados futuros.

[9]Como se explicará más adelante, un anticuerpo representa una regla sub-simbólica que almacena una unidad de memoria hetero-asociativa

- *Detección distribuida*: el SIA puede verse como un sistema distribuido, es decir, los anticuerpos no dependen de ningún sistema centralizado para realizar sus dinámicas, y cuando perciben un antígeno (entrada sensorial o interna), ésta es transmitida y procesada a otros anticuerpos presentes en la red inmune.
- *Auto-Regulación*: el sistema regula sus dinámicas internas, de producción, destrucción, clonación y modificación de anticuerpos, esto con el fin de mantener el nivel de adaptabilidad ante los cambios del entorno y de optimizar el uso de anticuerpos que definan patrones de reconocimiento representativos.
- *Protección dinámica*: la expansión clonal permite la generación de anticuerpos de alta afinidad (llamada maduración de afinidad). Este proceso mantiene un equilibrio dinámico entre la exploración y la explotación en la inmunidad adaptativa. La protección dinámica incrementa el alcance que provee el sistema inmune a través del tiempo.
- *Detección probabilística*: la reacción cruzada en la respuesta del sistema inmune es un proceso estocástico. También la detección es aproximada. Esto es, un anticuerpo puede emparejarse con muchos tipos (estructuralmente relacionados) de antígenos, y un antígeno puede a su vez activar muchos de los anticuerpos presentes en la red inmune.
- *Tolerancia al Ruido*: el SIA es tolerante al ruido, es decir, éste no necesita hacer una detección exacta de las entradas percibidas, dado que el sistema inmune es flexible.
- *Generalización*: debido a que el SIA puede percibir entradas con cierto nivel de ruido, éste puede extrapolar respuestas para las entradas desconocidas y afinar, dinámicamente, dichas respuestas a través de un proceso adaptativo.

Considerando lo anterior, se propone un modelo computacional del SIA que opera en el nivel emergente de cada *Comportamiento*.

7.4.1. Arquitectura Híbrida del SIA

A pesar de que la concepción original de las redes AiNET estuvo orientada a la resolución de problemas de formación de grupos y compresión de datos, un concepto diferente a la aplicación de estos principios inmunológicos apunta hacia los problemas de optimización.

La naturaleza de la inclusión de este sistema como módulo de procesamiento en el nivel emergente del *Comportamiento*, tiene que ver con el reconocimiento de patrones y la clasificación, una clasificación que no está sujeta a parámetros conocidos a priori por el sistema de inferencia. Esto quiere decir que las reglas que definen la categorización de diferentes elementos presentes en el entorno no son conocidas y por ende, el mecanismo propuesto debe estar en la capacidad de inferir hipótesis y plantear respuestas que serán posteriormente reafirmadas o rechazadas de acuerdo con una retroalimentación del entorno. Este proceso permitirá generar nuevo conocimiento por medio de un esquema de aprendizaje por refuerzo, optimizando en cada paso las respuestas realizadas por el sistema en base a la información de retroalimentación.

Con el fin de dotar a la red inmune original (AINet) con un mecanismo de aprendizaje por refuerzo que permita generar unidades de memoria hetero-asociativa, se introducen conceptos de otro sistema de inferencia cognitiva basado en principios biológicos, conocido como *Sistema Clasificador de aprendizaje* (SC), propuesto por John Holland (Holland et al., 1989), y descrito en la sección 2.2.4.3.

El modelo SC posee una estructura similar a la de una Red Idiotípica de Jerne (N. de Castro y Timmis, 2002), con una población determinada de clasificadores, que no son más que reglas formadas por cadenas binarias de bits con dos partes, una condición y una acción, en analogía con las partes paratope e idiotope de un anticuerpo.

Así pues, el sistema híbrido propuesto combina: las potentes dinámicas y meta-dinámicas auto-organizativas del modelo AiNet, una máquina darwiniana basada en un algoritmo genético modificado encargada de evolucionar el acervo de anticuerpos, y un robusto mecanismo perteneciente al modelo SC (conocido con el nombre *Sistema de Asig-nación de Recompensas*) el cual articula los procesos de selección y competición entre los anticuerpos de la red inmune.

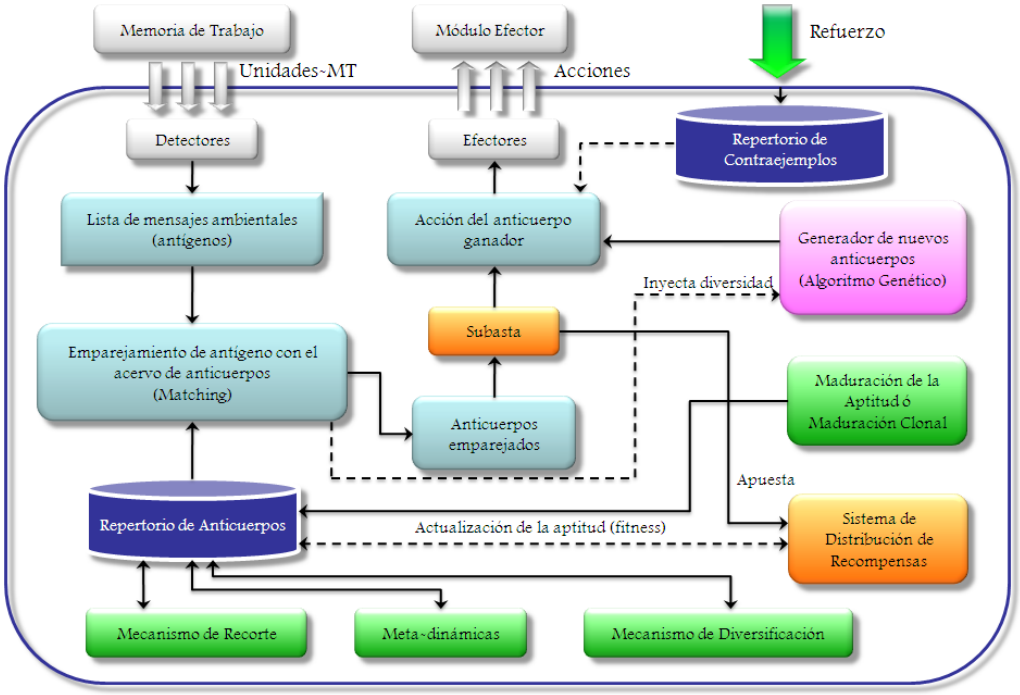


Figura 7.3: Arquitectura del Sistema Inmune Artificial.

En la figura 7.3, se presenta la arquitectura básica del SIA propuesto. Este sistema puede ser visto como un modelo de micro-cognición, puesto que posee los elementos básicos para ello: un conjunto de detectores que reciben las entradas del sistema (i.e., un conjunto de unidades activas en la MT), un conjunto de efectores que ejecutan o comunican las respuestas del sistema (al módulo Efector), un sistema de control que ejecuta las dinámicas internas de reconocimiento de patrones y selección de acciones, un mecanismo de refuerzo, un mecanismo de aprendizaje por contra-ejemplos, y un mecanismo de adaptación.

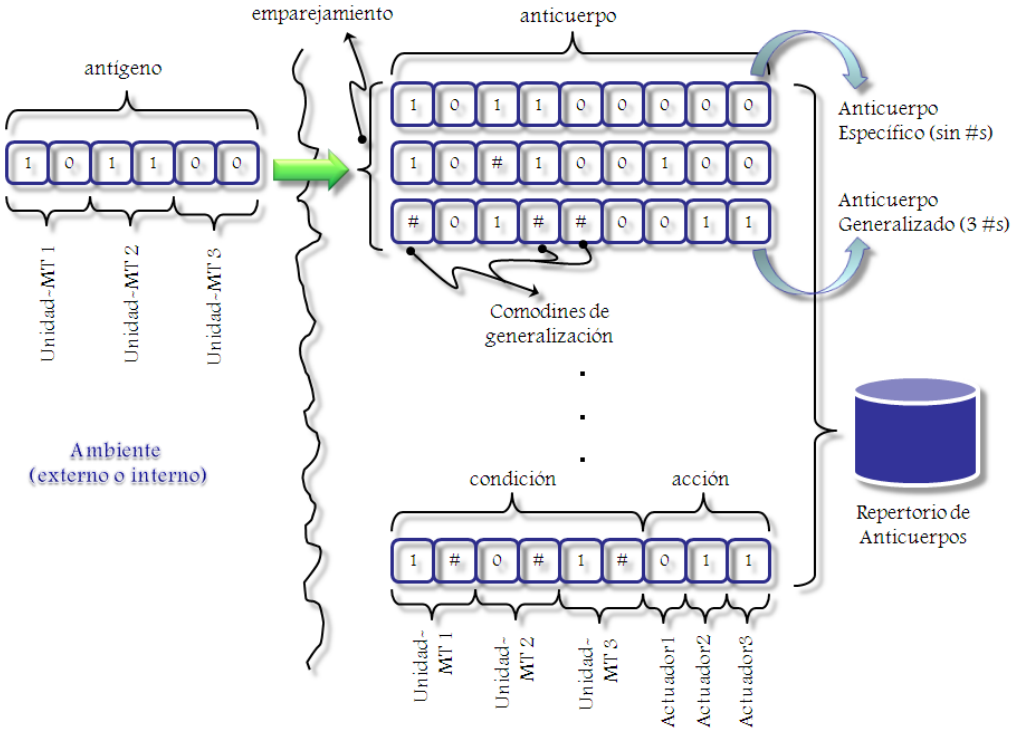


Figura 7.4: Representación de Anticuerpos y Antígenos.

En la figura 7.4 se observa además, que cada anticuerpo (con su parte paratope e idiotope) forma una regla sub-simbólica, la cual puede ser traducida a una regla simbólica y/o ser transmitida a los otros componentes del *Comportamiento*, mecanismo que se explicará al detalle en la sección 7.5.2.

En el modelo híbrido del SIA, las representaciones tanto del antígeno como de los anticuerpos son sub-simbólicas. Los antígenos se encuentran codificados en un alfabeto binario $\{0, 1\}$, mientras que los anticuerpos se codifican, en la parte condición, empleando un alfabeto ternario $\{0, 1, \#\}$, donde el caracter # cumple la función de "comodín", es decir, puede considerarse como un "0" o como un "1" según convenga, y en la parte acción, se codifica empleando un alfabeto binario $\{0, 1\}$. En la figura 7.4 se aprecia el mecanismo de emparejamiento de una estímulo sensorial, donde la entrada de un antígeno genera la activación de uno o varios anticuerpos. Como se puede observar, los anticuerpos codifican (de forma sub-simbólica) pautas de percepción y actuación.

En el ejemplo de la figura 7.4, se emparejan en su parte condición con el antígeno, un subconjunto de tres 3 anticuerpos del acervo total. El nivel de especificidad de cada anticuerpo está determinado por el número de comodines que posea, así, cuantos más comodines tenga un anticuerpo en su parte condición (paratope) más general será éste, y cuantos menos comodines tenga, su emparejamiento con los antígenos será más específico. De esta manera, cada anticuerpo define, en su parte condición, un patrón aplicable a un conjunto de situaciones observables. El número mínimo de situaciones que puede describir el paratope de cada anticuerpo es igual a "1" (conjunto unitario), lo cual

corresponde a una regla totalmente especializada (e.g., el paratope "11010"), mientras que el número máximo de situaciones que puede describir un paratope es n^2, con n igual a la longitud del paratope, lo cual corresponde a una regla totalmente generalizada que aplicaría a todas las situaciones observadas por el agente (e.g., el paratope "#####"). Por ejemplo, considérese el paratope "1#01#" de un anticuerpo a_k, la descripción por extensión del número de situaciones donde se aplicaría éste sería: $a_k = \{10010, 10011, 11010, 11011\}$[10].

Como consecuencia, los anticuerpos totalmente especializados sólo se aplican a situaciones donde todos los estados observados se corresponden necesariamente con las pautas definidas por el paratope del anticuerpo, y por tanto su activación es altamente restrictiva, mientras que en el caso contrario, los anticuerpos totalmente generalizados se aplican a todo tipo de situaciones y no crean categorías ni clasificaciones del estado observado. Intuitivamente ambos tipos de anticuerpo no son del todo deseables en el sistema y por tanto poseen un bajo nivel de aptitud (fitness)[11] que los hará menos competitivos durante el proceso que selecciona el anticuerpo que enviará su acción como respuesta inmune.

Una vez el antígeno (estado actual observado) activa el conjunto de anticuerpos que se corresponden en su parte condición con dicho antígeno, resulta necesario definir un mecanismo de selección que determine el anticuerpo que finalmente enviará su parte acción (idiotope) como respuesta al estado actual observado. Por tanto, antes de presentar el algoritmo del SIA, se describe a continuación el mecanismo híbrido que, acoplado con las técnicas basadas en la teoría inmunológica, le permite al SIA modular tanto la selección de anticuerpos como el proceso de aprendizaje por refuerzo: este es el *Sistema de Asignación de Recompensas*.

7.4.2. Algoritmo de Asignación de Recompensas

A fin de coordinar el mecanismo de aprendizaje por refuerzo del SIA, se propone emplear el algoritmo de asignación de recompensas propuesto por Holland (llamado por él, algoritmo de *Bucket Brigade*) (Holland et al., 1989). Este algoritmo basa su funcionamiento en cadenas binarias de bits llamados Clasificadores (que en el caso del SIA, serán los mismos anticuerpos), y su función es simular una cadena de servicios económicos donde cada anticuerpo es un ente que posee cierta "ganancia" o potencial económico G_i, que representa el valor de la *aptitud* del anticuerpo.

El algoritmo de asignación de recompensas consta de dos etapas. En la primera, se realiza una *subasta* entre los anticuerpos activados por el mensaje enviado desde el ambiente (antígeno), a fin de escoger el más apropiado para responder dicho mensaje. Cada anticuerpo realiza una puja P_i proporcional al valor de su ganancia (ó aptitud, G_i) que posee en ese momento. De esta forma, los anticuerpos que han acumulado una mayor

[10] Tómese en cuenta que las situaciones observadas por el agente son codificadas en un alfabeto subsimbólico (binario) en el nivel emergente del *Comportamiento*.

[11] Por ahora, conviene simplemente tener claro que la aptitud es un indicador del desempeño (bueno o malo) que ha tenido el anticuerpo durante la interacción del agente con su entorno. Sin embargo el término "aptitud" se hereda del área de los Algoritmos Genéticos, y su importancia en la orientación del proceso epigenético del SIA se discutirá más adelante.

ganancia, tendrán una mayor probabilidad de ser escogidos. Para elegir al anticuerpo ganador se emplea un método estocástico (e.g., el método de la ruleta). En la segunda etapa, en la *Casa de Subastas*, el anticuerpo activado hace un pago Pg_i correspondiente a su puja, la cual se distribuye entre aquellos anticuerpos que también tuvieron el derecho de pujar por el mensaje ambiental (antígeno).

Un anticuerpo puede además realizar un conjunto de recaudaciones (R_i) relativas a su previa actividad de envío de mensajes o debidas al refuerzo del ambiente. Además de las pujas y de las recaudaciones, un anticuerpo puede estar sujeto a uno o más impuestos, I_i, a fin de penalizar su ineficiencia. Se tiene entonces una ecuación que describe la reducción o el incremento de la ganancia de cada anticuerpo i, como sigue:

$$G_i(t+1) = G_i + R_i(t) - Pj_i(t) - I_i(t) \tag{7.12}$$

Para comprender al detalle la acumulación de ganancias netas de un anticuerpo, se deben también cuantificar sus pujas, pagos, e impuestos. Un anticuerpo realiza una puja en proporción a su ganancia G_i:

$$P_i = \alpha_{puja} \times G_i \tag{7.13}$$

donde: α_{puja} es un coeficiente de puja entre 0 y 1

Un esquema de puja más elaborado, el cual se empleará es el SIA es el siguiente:

$$P_i(t) = \alpha_{puja} \times G_i(t) + (\beta \times rp^{RPpot}) \tag{7.14}$$

donde: β es un coeficiente de puja, menor que 1, que indica el porcentaje de lo que pujará el anticuerpo en la subasta.
rp es una razón de puja determinada por el nivel de especificidad. Ésta, es una medida que indica qué tan similar o acertada es la condición de un anticuerpo con respecto a un determinado mensaje. Por ejemplo, cuando el bit n del anticuerpo es el mismo que el bit n del mensaje ambiental, se incrementa el valor de rp, de lo contrario no se incrementa.
$RPpot$ es un parámetro que controla la importancia del rp en determinar la puja de un anticuerpo (por defecto es 1).

Se podría simplemente parar aquí y seleccionar de manera determinista un ganador de la puja entre los k mejores anticuerpos (donde k es el tamaño de la lista de mensajes[12]), sin embargo esto podría generar un estado monótono donde siempre son seleccionados los mismos anticuerpos. Por lo tato, se mantendrá la subasta en presencia de un ruido aleatorio. Se calcula una puja efectiva PE por cada anticuerpo que corresponde en su parte condición con el mensaje, como la suma del valor de su puja más un factor de ruido:

$$PE_i = G_i + Rd(\sigma_{puja}) \tag{7.15}$$

[12]La lista de mensajes es una cola donde se van almacenando y procesando uno a uno los mensajes producidos por las inferencias internas del SIA.

donde: Rd es una función Gaussiana aleatoria que se aplica sobre el ruido específico de la puja σ_{puja}.

Después de incluir algo de ruido en la subasta y de seleccionar el anticuerpo que enviará su acción a los actuadores (o nuevamente a la lista de mensajes), el pago acumulado debe ser hecho a aquellos anticuerpos que fueron escogidos durante el proceso de selección[13]. Debido a que la presencia de un antígeno puede activar varios anticuerpos en cadena (i.e., la acción de un anticuerpo puede activar la condición de otro, y así sucesivamente hasta que no existan más anticuerpos que activar o hasta que se cumpla un criterio de paro), la distribución de la señal de refuerzo ambiental recompensa o penaliza todos aquellos anticuerpos que participaron en la selección de la acción ejecutada en el ambiente. Es decir, el último anticuerpo que envió su acción al entorno recibe el refuerzo, luego, una parte proporcional de esa recompensa se cede al anticuerpo que produjo el mensaje anterior que la activó, y así sucesivamente hacia atrás, siguiendo la trayectoria inversa que generó la acción final.

A cada anticuerpo se le aplica un impuesto para prevenir una "enriquecimiento" por acumulación de ganancias, y para estimular la generación de reglas (anticuerpos) más productivas. El siguiente esquema recauda un impuesto proporcional a la ganancia del anticuerpo:

$$I_i = \delta_{imp} \times G_i \tag{7.16}$$

donde: δ_{imp} es un coeficiente que se aplica sobre la ganancia para calcular el impuesto.

Después de n iteraciones de "inactividad", la ganancia de un anticuerpo puede ser calculada como sigue:

$$G_i(t+n) = G_i(t) \times (1-\delta_{imp})^n \tag{7.17}$$

De la ecuación 7.17, la vida media de un anticuerpo inactivo puede ser determinada por la siguiente ecuación:

$$n = \frac{log(1/2)}{log(1-\delta_{imp})} \tag{7.18}$$

Si la vida media es específica, la tasa de impuesto puede hallarse mediante la ecuación:

$$\delta_{imp} = 1 - (1/2))^{1/n} \tag{7.19}$$

Otro tipo de impuesto empleado en este algoritmo, es el impuesto de puja Imp_{puja}, el cual es aplicado a todos los anticuerpos que participan en un proceso de subasta a fin de penalizar los clasificadores muy generales, por ejemplo, los que pujan en cada paso pero que rara vez ganan porque tienen una baja especificidad, la cual produce pujas bajas que impiden que el anticuerpo gane y envíe su acción a los actuadores.

Estas relaciones en conjunto definen el algoritmo de asignación de recompensas usado en un gran número de sistemas cognitivos (Krichmar y Edelman, 2006). Para examinar

[13] En el caso específico del SIA, durante la Selección Clonal.

brevemente la estabilidad y efecto de este mecanismo, se simplifica la ecuación de asignación de recompensas en una donde todos los pagos e impuestos han sido reemplazados por sus ganancias equivalentes. Asumiendo que se tiene un anticuerpo activo, se obtiene la siguiente ecuación:

$$G(t+1) = G(t) - (\alpha_{puja} \times G(t)) - (\delta_{imp} \times G(t)) + R(t) - (Imp_{puja} \times \alpha_{puja} \times G(t)) \tag{7.20}$$

donde: $\alpha_{puja} \times G(t) = P_i(t)$

Finalmente, agrupando términos se obtienen las siguientes ecuaciones:

$$G(t+1) = (1-\beta) \times G(t) \times R(t) \tag{7.21}$$

Donde:

$$\beta = \alpha_{puja} + \delta_{imp} + Imp_{puja} \times \alpha_{puja} \tag{7.22}$$

A continuación, se describe el algoritmo del SIA y el punto específico donde interviene el algoritmo de Asignación de Recompensas.

7.4.3. Proceso Epigenético para el Sistema Inmune Artificial

A continuación se describe con detalle el algoritmo epigenético del SIA y la interacción entre los componentes que se presentan en la figura 7.3.

1. Se crea una población inicial aleatoria de anticuerpos. Cada anticuerpo estará representado por una cadena de bits, y estará dividido en dos partes, una condición y una acción.
2. Para cada antígeno (entrada sensorial) presentado, hacer:

   a) *Selección Clonal*: se determinan las afinidades entre el antígeno y cada anticuerpo de la población total. La medida de afinidad consiste en hacer una comparación (matching) de cada una de las posiciones (bit a bit), entre las cadenas binarias del anticuerpo (parte condición) y del antígeno. Para la posición i de ambas cadenas, se comparan los valores contenidos en esas posiciones, si los valores son iguales, se incrementa el valor de la afinidad en dos puntos, si el valor de la condición del anticuerpo en i es un comodín # se incrementa en un punto la afinidad, de lo contrario no se incrementa el valor de la afinidad. Así pues, la afinidad será la suma de equivalencias entre los valores de cada posición para las cadenas binarias del anticuerpo y del antígeno (ver figura 7.5). Esta medida de la afinidad es una adaptación de la medida conocida como distancia de Hamming (N. de Castro y Timmis, 2002). Con las afinidades calculadas, se selecciona un subgrupo de anticuerpos con alta afinidad. Este subgrupo tiene un tamaño variable puesto que se escogen aquellos anticuerpos que tengan una afinidad por encima de un umbral definido como un porcentaje de la longitud del antígeno $\times$ 2 (que es igual al valor máximo de afinidad), y este número puede variar conforme varían las estructuras de los anticuerpos.

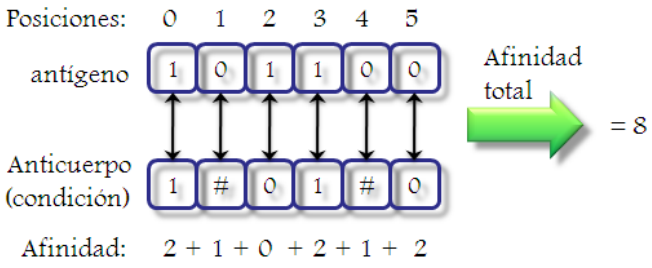


Figura 7.5: Ejemplo de medida de la afinidad empleando una adaptación del método Hamming. Una afinidad de 2 para las posiciones 0, 3 y 5 donde hay una correspondencia exacta, una afinidad de 1 en las posiciones 1 y 4 donde está un comodín, y una afinidad de 0 en la posición 2 donde no existe correspondencia.

b) *Ejecutar Pujas*: se realizan las pujas en la Casa de Subastas, es decir, se calcula la puja de cada anticuerpo en base a su afinidad con el antígeno y a su aptitud (ganancia efectiva PE). Los anticuerpos con mejor aptitud y mayor afinidad tendrán mayor probabilidad de ser escogidos para ejecutar su acción en el ambiente, sin embargo aquellos anticuerpos con menor capacidad de puja tendrán menor probabilidad de ser escogidos. Tanto este método de pujas, como la estrategia que recompensa los anticuerpos ejecutados, conforman el sistema de Asignación de Recompensas (ó Bucket Brigade) descrito anteriormente.

c) *Selección del anticuerpo ganador*: se hace una selección estocástica en base a las pujas de cada anticuerpo, y se selecciona aquél que ejecutará su acción en el ambiente. La selección estocástica de anticuerpos está determinada por el método de la ruleta (Holland et al., 1989) (véase la figura 7.6).

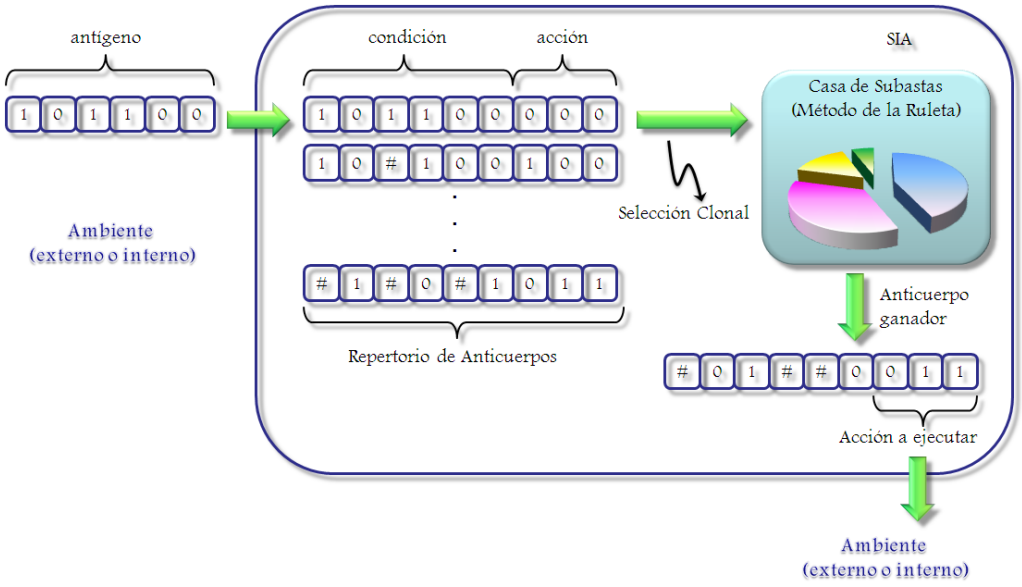


Figura 7.6: Dinámica de selección Clonal y ejecución del anticuerpo ganador.

d) *Ejecución de la acción*: el anticuerpo seleccionado envía su mensaje codificado (acción) para que sea ejecutado en el ambiente. En ese momento se cobra el valor pujado por cada anticuerpo. También se aplica un factor de reducción de aptitud para simular la muerte natural de anticuerpos. Éste factor de reducción se aplica a toda la población en cada iteración (por defecto, este factor corresponde a $fr = 0.001 \times G_{(i,t)}$).

e) *Recompensar el anticuerpo ejecutado*: de acuerdo con el refuerzo del ambiente, se premia o se castiga la aptitud del anticuerpo. Esta compensación se hace con las fórmulas descritas en el sistema de asignación de recompensas.

f) *Maduración de la Aptitud (maduración clonal en AINet)*: si el refuerzo fue positivo entonces el anticuerpo es un "padre" apto para reproducirse (véase la figura 7.7). Los anticuerpos restantes que participaron en la subasta pero no ganaron, son también madurados aunque en menor proporción (como se detallará más adelante). En esta etapa, una Máquina Darwiniana (implementada por un Algoritmo Genético) clona, recombina y muta ligeramente el anticuerpo padre, a fin de que no pierda sus características heredadas por completo. Las dinámicas de la Máquina Darwiniana empleada en esta fase de Maduración Clonal y sus respectivos operadores genéticos de clonación, recombinación y mutación, se explican en la siguiente sección.

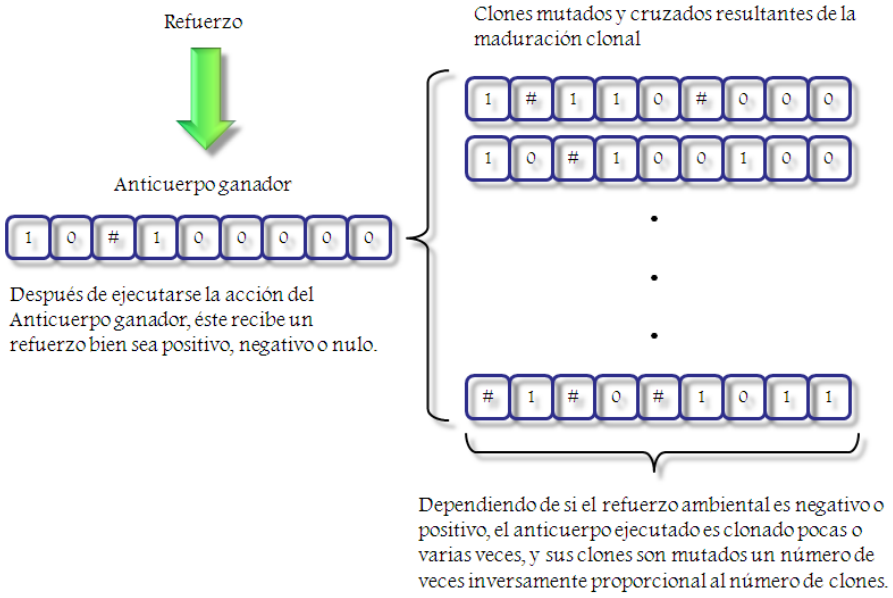


Figura 7.7: Dinámica de la Maduración de Aptitud.

g) *Memoria permanente*: si algún anticuerpo supera un umbral de Aptitud, se convierte en un anticuerpo de memoria (memoria inmune del sistema) y ya no podrá ser eliminado de la población de anticuerpos.

h) *Meta-dinámica*: con el fin de mantener dentro de un margen establecido el tamaño de la población de anticuerpos, se determinan las afinidades entre todos los anticuerpos, incluyendo los pre-seleccionados al inicio, los generados por clonación y mutación, y el resto de anticuerpos que no participaron en la

iteración actual. Aquellos anticuerpos que tengan semejantes cuya afinidad esté por debajo de un umbral de semejanza y cuya acción sea la misma, serán eliminados del acervo, teniendo en cuenta eliminar aquellos con menor aptitud. Este método se corresponde con la estrategia de "olvido" que emplea el SIA.

i) *Mecanismo de Recorte*: cada cierto número de iteraciones se revisará el tamaño del acervo total de anticuerpos, si éste supera el rango permitido, debido a la super-población producida por la maduración clonal, se seleccionará para su eliminación el conjunto de anticuerpos menos aptos. Este método también se corresponde con una estrategia de "olvido" que controla el tamaño de la memoria inmune. Para la selección de estos anticuerpos, se tendrán en cuenta las siguientes características:

- Se seleccionarán los anticuerpos con menor aptitud de manera determinista.
- Se tendrá en cuenta que los anticuerpos escogidos hayan estado en el acervo durante un número determinado de iteraciones, a fin de asegurar que han tenido la oportunidad de participar en las distintas subastas y que han permanecido un tiempo determinado dentro del acervo.

j) *Diversificación*: para cumplir con los conceptos de exploración y explotación, cada cierto número de iteraciones, se debe introducir una cantidad determinada de nuevos anticuerpos generados totalmente al azar (por una Máquina Darwiniana), a fin de explorar nuevas soluciones dentro del espacio de búsqueda. La cantidad de anticuerpos estará sujeta al límite máximo de anticuerpos permitidos en la población.

k) *Generador de Anticuerpos Ausentes*: cuando se presenta un antígeno y el sistema no posea ningún anticuerpo que sobrepase el umbral de afinidad, entonces este mecanismo generará un nuevo anticuerpo con la parte condición igual al antígeno no reconocido y con una parte condición generada aleatoriamente, y luego será introducido en la población de anticuerpos. La Aptitud de este anticuerpo será el promedio de la Aptitud de los anticuerpos de la población.

3. Continua en el paso 2 y se repite hasta que existan antígenos presentes.

En el cuadro 7.3, se presenta el algoritmo resumido que ejecuta el SIA.

7.4.3.1. Máquina Darwiniana basada en un Algoritmo Genético

El SIA requiere de una Máquina Darwiniana (MD) encargada de dirigir los procesos darwinianos en las fases de Maduración Clonal y Diversificación. Esta MD se realiza a través de un Algoritmo Genético (AG) (Goldberg, 1989) – ver sección 2.2.4.1. A continuación, se describen los criterios tomados en cuenta para el diseño del AG:

- Si el refuerzo del anticuerpo activado fue positivo, éste se clona (auto-replica) varias veces (un número entre 4 y 10) y se mutan los clones en una baja proporción (un número entre 1 y 2).

Método: ejecutarSIA

```
1.  Variables
2.     Pob          ↑↓ población original de anticuerpos.
3.     subPob       ↑↓ sub-población de anticuerpos cuya parte condición coincide con
                       una entrada dada (antígeno).
4.  Inicio Procedimiento
5.     Pob ← inicializarPoblacionAnticuerpos()
6.     Mientras que se presenten antígenos al sistema hacer:
7.        subPob ← aplicarSeleccionClonal()
8.        ejecutarPujas(subPob)
9.        x ← seleccionarAnticuerpoGanador(subPob)
10.       ejecutarAccion(x)
11.       x ← aplicarRefuerzoAnticuerpo()
12.       pobMad ← aplicarMaduracionClonal(subPob, x)
13.       almacenarMemoriaInmune(Pob)
14.       aplicarMetadinamica(Pob, subPob, pobMad)
15.       aplicarMecanismoRecorte(Pob)
16.       pob ← aplicarDiversificacion(Pob)
17.       pob ← generarAnticuerposAusentes
18.    Fin Mientras
19. Fin Procedimiento
```

Algoritmo 7.3: Algoritmo de Ejecución del SIA.

- Si el refuerzo del anticuerpo activado fue negativo, éste es clonado pocas veces (un número entre 1 y 2) y se mutan los clones en proporción mayor (un número entre 2 y 4).
- Para los demás anticuerpos preseleccionados por su buena afinidad inicial con el antígeno, su maduración puede ser así:
  - Si el anticuerpo activado tuvo un refuerzo positivo, se reproduce en una tasa baja (un número entre 1 y 2) y se muta en una proporción un poco mayor al anticuerpo ganador (un número entre 2 y 6 - hipermutación).
  - Si el anticuerpo activado tuvo un refuerzo negativo, los otros anticuerpos quedan tal como están puesto que no fueron seleccionados y tal vez puedan tener respuestas apropiadas empleando su estructura actual.

En cuanto a la fase de Diversificación en el SIA, se tienen en cuenta las siguientes consideraciones de diseño:

- El AG debe modular un conjunto de mecanismos que le permitan mantenerse en un punto de equilibrio entre la *Convergencia* (centrar la búsqueda en zonas prometedoras) y la *Diversidad* (evitar la convergencia prematura en zonas que no contienen un máximo global), o dicho en otras palabras, un equilibrio entre la explotación y la exploración.
- Se presenta el diseño de una versión adaptada del algoritmo CHC (por su acrónimo en inglés de *Cross generation elitist selection, Heterogeneous recombination, and Cataclism mutation*), el cual fue propuesto como una variante del AG clásico que

introduce un equilibrio entre convergencia y diversidad (Eshelman, 1991; Eiben y Smith, 2003). El objetivo de este algoritmo es combinar una selección elitista que preserva los mejores individuos que han aparecido hasta el momento con un operador de cruce que produce hijos muy diferentes a sus padres. El CHC introduce cuatro componentes novedosos: la Selección Elitista, el Cruce Uniforme HUX, la Prevención de Incesto, y la Reinicialización.

- *Selección Elitista*: este operador selecciona los N mejores anticuerpos entre padres e hijos (anticuerpos resultantes del cruce). De esta manera, los N mejores elementos encontrados hasta el momento permanecerán en la población actual. Esta selección elitista se relaciona con el rol que juega la memoria inmune en el SIA, donde los "mejores" anticuerpos son protegidos contra la eliminación y el olvido dentro de una memoria especial. En el cuadro 7.4 se presentan los algoritmos correspondientes a la replicación y selección elitista de anticuerpos.

Método: replicar

```
1.  Variables
2.     Pob(t-1)        ↓ población original de anticuerpos.
3.     subPob(t)       ↑ población de anticuerpos replicados.
4.  Inicio Procedimiento
5.     subPob(t,i) ← copiar( Pob(t-1, i) ) //copia todos los anticuerpos de
       Pob(t-1) en orden aleatorio.
6.  Fin Procedimiento
```

Método: seleccionarElitista

```
1.  Variables
2.     Pob(t-1)        ↓ población original de anticuerpos.
3.     subPob'(t)      ↓ población de anticuerpos recombinados.
4.     Pob(t)          ↑ nueva población de anticuerpos.
5.  Inicio Procedimiento
6.     Para cada anticuerpo ''i'' de subPob'(t) hacer:
7.        temp ← encontrarSimilar(subPob'(t,i), Pob(t-1)) //de
          todos los anticuerpos de Pob(t-1) busca el más similar a subPob'(t,i)
          utilizando como criterio la menor distancia de Hamming, esto es, encuentra
          el anticuerpo que difiera en la menor cantidad de bits.
8.        P(t,i) ← determinarMejorAptitud(subPob'(t,i), temp) //in-
          serta en P(t) el mejor anticuerpo entre subPob'(t,i) y temp (i.e., el anticuerpo
          con el nivel de aptitud más alta).
9.     Fin Para
10. Fin Procedimiento
```

Algoritmo 7.4: Algoritmos de Replicación y Selección Elitista de Anticuerpos.

- *Cruce Uniforme (HUX)*: este operador copia en los dos anticuerpos hijos todos los elementos coincidentes en ambos padres, y luego copia aleatoriamente la mitad de los elementos distintos en cada uno de los hijos, de manera que la distancia Hamming entre los hijos y entre hijos y padres sea la máxima (véase la figura 7.8 y el algoritmo 7.5).
- *Prevención de Incesto*: se forman $N/2$ parejas con los elementos de la población. Sólo se recombinan las parejas cuyos miembros difieren en un número determinado

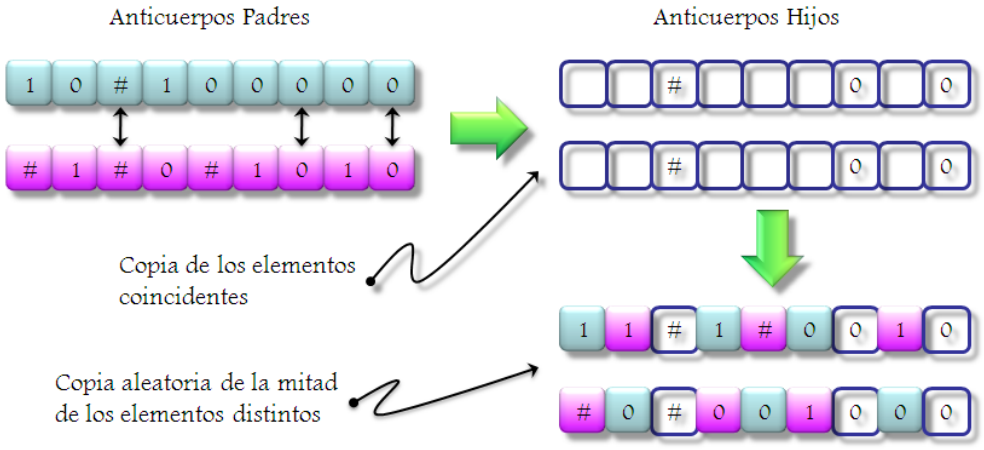


Figura 7.8: Cruce HUX.

de elementos (umbral de cruce). El umbral se inicializa en $L/4$ (L es la longitud del anticuerpo o cromosoma). Si durante un ciclo no se produce ni un sólo cruce, el umbral de cruce es reducido en 1.

Método: recombinarHUX

```
1.  Variables
2.     subPob(t)        ↓ población de anticuerpos replicados.
3.  Inicio Procedimiento
4.     Para cada una de las N/2 parejas de anticuerpos en
       subPob(t) hacer:
5.        distancia ← determinarDistanciaHamming(ac1, ac2)
6.        Si distancia/2 > umbralIncesto entonces:
7.           recombinarPareja(ac1, ac2)  //copia los elementos coincidentes de los dos
                                         padres e intercambia aleatoriamente los elementos restantes.
8.        De lo contrario
9.           eliminarPareja(ac1, ac2)  //elimina la pareja de anticuerpos de subPob(t).
10.       Fin Si
11.    Fin Para
12. Fin Procedimiento
```

Algoritmo 7.5: Algoritmo de Cruce HUX y Prevención del Incesto.

- *Reinicialización*: este operador es ejecutado cuando se ha detectado una convergencia en un máximo local. Este operador está estrechamente relacionado con el anterior, así, cuando el umbral de cruce es menor que cero, la población se reinicializa: a) si existen anticuerpos en la memoria inmune, se mantienen sin modificación y el resto de la población se genera aleatoriamente, o b) si no existen anticuerpos en la memoria inmune, se emplea el mejor elemento como plantilla (con un 35 % de variación aleatoria) y se incluye una copia suya (véase la figura 7.9 y el algoritmo 7.6).
- *Mutación*: Aunque el algoritmo original de CHC no aplica el operador de mutación, en la presenta propuesta se incluye este operador genético con una muy baja

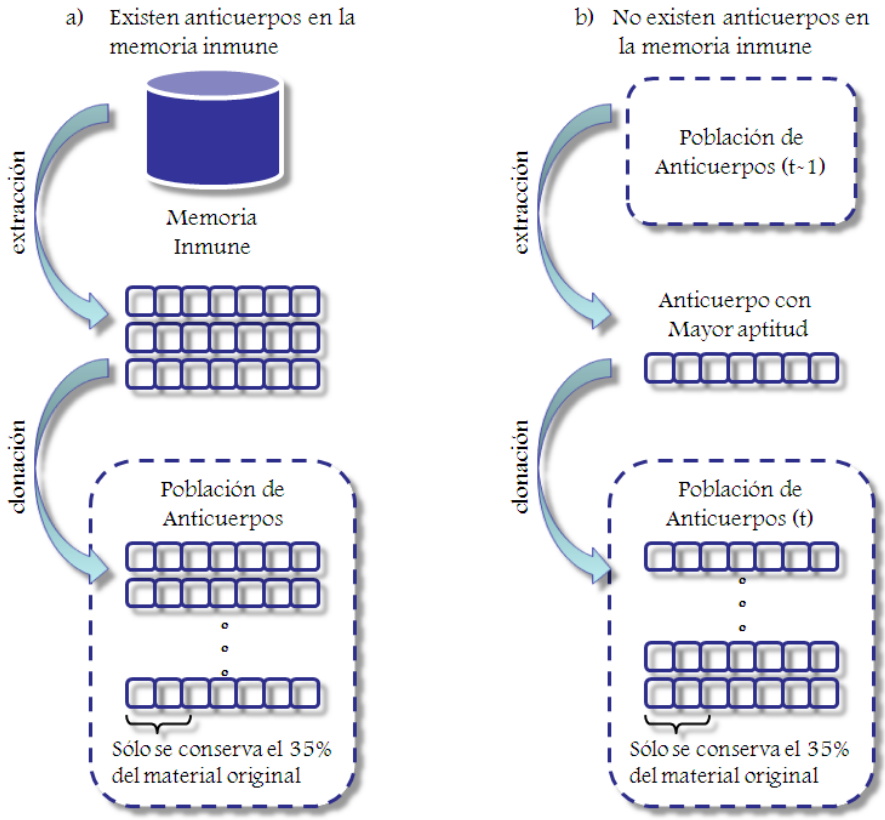


Figura 7.9: Mecanismo de Reiniciliazación en CHC.

Método: reinicializar

1. **Variables**
2. `Pob(t)` ↓ población original de anticuerpos.
3. **Inicio Procedimiento**
4. `Pob(t) ← reemplazar( M, ac1 )` //en el caso de que no hayan anticuerpos en la memoria inmune, reemplaza $Pob(t)$ con M copias del mejor anticuerpo ($ac1$) seleccionado en el ciclo cognitivo anterior ($Pob(t-1)$), de lo contrario, selecciona aleatoriamente M anticuerpos de la memoria inmune y los reemplaza en $Pob(t)$.
5. **Para** todas las copias ``i'' de anticuerpos de $Pob(t)$, excepto una, **hacer**:
6. `intercambiar(Pob(t).ac`$_i$`, elementos)` //se cambian aleatoriamente, r $\times$ L "elementos" del anticuerpo, con r un número aleatorio flotante entre 0 y 1, y L la longitud total del anticuerpo.
7. `evaluarAnticuerpo( Pob(t).ac`$_i$ `)` //se evalúa la aptitud y afinidad del anticuerpo con respecto al antígeno actual)
8. **Fin Mientras**
9. **Fin Procedimiento**

Algoritmo 7.6: Algoritmo de Reinicialización.

tasa de operatividad, con el objetivo de combinar la diversidad local (a través de la mutación en anticuerpos con un alto nivel de aptitud) junto con la diversidad

global que incorpora la reinicialización y el cruce HUX. El operador de mutación es aplicado sobre los anticuerpos resultantes de la selección elitista con una probabilidad de 0.5, y luego, se realiza una mutación aleatoria en un sólo punto del cromosoma (anticuerpo). La mutación en la parte condición del anticuerpo puede ser cualquier elemento del alfabeto ternario $\{0, 1, \#\}$, mientras que para la parte acción sólo puede ser un elemento del alfabeto binario $\{0, 1\}$.

De esta manera, los operadores: Cruce HUX, Prevención del Incesto, Mutación, y Reinicialización, garantizan una alta diversidad, mientras que el operador de Selección Elitista dirige la convergencia elevada a máximos globales. Finalmente, la estructura del algoritmo genético CHC completo es presentada en el cuadro 7.7.

Método: activarCHC

```
1.  Variables
2.     Pob(t)          ↑ población actual de anticuerpos.
3.     Pob(t-1)        ↑↓ población de anticuerpos del ciclo cognitivo anterior.
4.  Inicio Procedimiento
5.     t = 0
6.     d = L / 4
7.     inicilizar(Pob(t))
8.     evaluarAnticuerpos( Pob(t-1) )
9.     Mientras que no se satisfaga cierta condición de paro hacer:
10.       t = t + 1
11.       subPob(t) ← replicar(Pob(t-1))
12.       subPob'(t) ← recombinar(subPob(t))
13.       evaluarAnticuerpos( subPob'(t) )
14.       Pob(t) ← seleccionarElitista( subPob'(t), Pob(t-1))
15.       mutar( Pob(t) )
16.       Si Pob(t) = P(t-1) entonces
17.          d--
18.       Fin Si
19.       Si d ==  0 entonces
20.          reinicializar( Pob(t) )
21.          d = L / 4
22.       Fin Si
23.    Fin Mientras
24. Fin Procedimiento
```

Algoritmo 7.7: Algoritmo Genético CHC Modificado

Como último comentario, cabe decir que la eficiencia del Algoritmo Genético depende, por un lado, de la definición de las dinámicas entre los operadores genéticos (e.g., tipo de operadores genéticos empleados, probabilidad y frecuencia de activación de cada operador, número de generaciones, criterios de parada y de reemplazo, etc.); y, por otro, de la definición de la *Función de Aptitud* encargada de evaluar cada solución (anticuerpo) generado en cada paso del algoritmo.

La Función de Aptitud es un tipo particular de función objetivo que formula la optimalidad de una solución (i.e, un anticuerpo). Ésta, se debe maximizar y/o minimizar,

encontrando valores para los diferentes parámetros que resulten óptimos al ser reemplazados en la función objetivo. Por tanto esta función debe reflejar los aspectos más relevantes del problema, estableciéndose las condiciones que restringen los resultados proporcionados por el algoritmo. El valor de la Función de Aptitud representa la calidad de la solución. Idealmente, la función de aptitud debe correlacionar de una manera muy aproximada, la solución codificada por cada anticuerpo con la meta que se quiere alcanzar con la ejecución del *Comportamiento*. Es por este motivo que la función de aptitud debe ser definida para cada *Comportamiento* de manera independiente, y debe ser además determinada en cada concreción contextual de la arquitectura. En el capítulo 12 sobre la Experimentación se presentan las funciones de aptitud para cada *Comportamiento* propuesto.

7.4.3.2. Mecanismo inferencial del SIA

Hasta el momento, se ha descrito el funcionamiento del SIA como un sistema reactivo (sistema estímulo-respuesta) que captura mensajes ambientales (antígenos) y que produce una respuesta inmune como consecuencia. Sin embargo, el SIA puede operar también en un modo más "racional" a través del encadenamiento inferencial de múltiples anticuerpos.

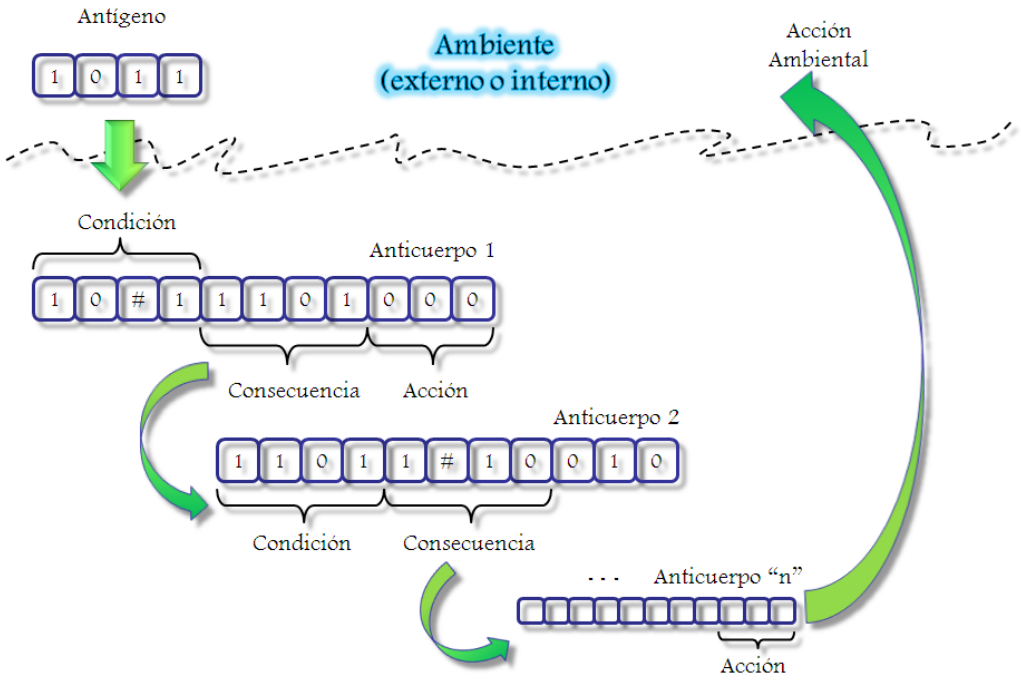


Figura 7.10: Activación en cadena de múltiples Anticuerpos.

En la figura 7.10 se observa cómo ésta interacción inferencial ocurre. Primero que todo, para que el SIA se comporte como un sistema inferencial resulta necesario extender la estructura básica de cada anticuerpo a una compuesta por tres partes: una condición, una consecuencia y una acción. El elemento nuevo, la *consecuencia*, es la parte del anticuerpo

que permite la activación de otros anticuerpos en el acervo inmune. El procedimiento es el siguiente:

1. Entra un mensaje ambiental al sistema (antígeno).
2. El SIA realiza las dinámicas típicas de selección de anticuerpos.
3. Una vez se ha seleccionado un anticuerpo, se toma su parte *consecuencia* y se realiza nuevamente el proceso de selección de anticuerpos, manteniendo un registro de la cadena de anticuerpos que han sido activados. La *consecuencia* del anticuerpo actual debe emparejar con la condición del siguiente anticuerpo como si ésta fuese el nuevo antígeno, y así sucesivamente.
4. El proceso se repite hasta que:
   a) No exista ningún anticuerpo que empareje la *consecuencia* del último anticuerpo activado, o
   b) Hasta que se genere una referencia circular, es decir, se active nuevamente el primer anticuerpo, o
   c) Hasta que se alcance un número determinado de iteraciones definidas por el diseñador.
5. La acción del último anticuerpo activado es enviada como respuesta del sistema. Se almacena la cadena inferencial producida por el encadenamiento de anticuerpos.

Este mecanismo de encadenamiento, que se ilustra en la figura 7.10, simula el funcionamiento de las redes inmunes naturales, donde los anticuerpos forman redes de inmunidad en respuesta a un antígeno presentado. Los operadores genéticos descritos en la sección 7.4.3.1 actúan de la misma manera en esta nueva estructura del anticuerpo, la única observación al respecto es que la parte *consecuencia* emplea un alfabeto ternario tal como lo hace la condición $\{0, 1, \#\}$. Cabe notar que es necesario definir para cada *Comportamiento* de la arquitectura, si el componente SIA empleará un mecanismo reactivo (estimulo-respuesta) o inferencial (como un red inmune).

7.4.3.3. Mecanismo de Aprendizaje empleando Contraejemplos

Cuando los humanos aprendemos de nuestras propias experiencias, no sólo lo hacemos guardando los recuerdos de aquellas decisiones satisfactorias que hemos tomado y cuya repetición en el futuro puede suponernos nuevamente un beneficio; sino que además guardamos los recuerdos de aquellas decisiones que, por el contrario, han desencadenado resultados perjudiciales o desfavorables (desde el punto de vista de nuestras expectativas), evitando de esta forma la ocurrencia de tales pautas conductuales en el futuro, y convirtiendo el aprendizaje en un proceso más eficiente. Así pues, con el objetivo de acelerar y hacer más eficiente el aprendizaje de pautas que lleva a cabo el SIA, se propone en la presente tesis, un mecanismo innovador que "fusiona" el proceso de selección de anticuerpos (descrito en las secciones anteriores) junto con un mecanismo de inferencia basado en contraejemplos, reafirmando una vez más la cualidad híbrida del SIA[14].

[14]Este mecanismo no es esencial para el funcionamiento del SIA, con lo cual puede ser activado o desactivado para cada concreción contextual de la arquitectura; sin embargo, cuando el mecanismo está activo, la

Considérese que las pautas correctas que debe aprender el SIA corresponden a puntos o regiones específicas de un espacio de búsqueda. El descubrimiento de tales pautas, empleando el proceso epigenético descrito en los apartados anteriores, implica la exploración y explotación de las regiones más prometedoras mediante las heurísticas propuestas (e.g., la selección clonal, las meta-dinámicas, el algoritmo de distribución de recompensas, etc.). No obstante, la eficiencia de este proceso puede incrementarse si durante el aprendizaje de nuevas pautas, no sólo se consideran aquellas que representan un conocimiento "correcto" de cómo actuar frente a determinadas situaciones, sino también un conocimiento acerca de lo "incorrecto". Así pues, la idea básica del mecanismo propuesto es proveer al SIA con la capacidad de aprender cierta cantidad de contraejemplos (i.e., ejemplos que contradicen las pautas que sí deben ser aprendidas) a partir de los errores cometidos por el propio agente, con el objetivo de acotar el espacio de búsqueda mediante la exclusión de las pautas no deseadas, lo cual, como consecuencia, aceleraría la convergencia en el aprendizaje de pautas correctas. En otras palabras, los contraejemplos son pautas de lo que el agente "no debe hacer", y se encargan de guiar el proceso de aprendizaje de las conductas deseadas.

Adicionalmente, el mecanismo de contraejemplos intenta *"generalizar"* continuamente el conjunto de pautas erróneas, de manera que puede identificar si el anticuerpo elegido coincide con un patrón ⟨`percepción-acción`⟩ incorrecto y, como consecuencia, puede pronosticar si éste obtendrá un refuerzo negativo después de su ejecución. Siendo posible la predicción de este tipo de situaciones, el SIA podrá modificar la acción del anticuerpo elegido antes de su ejecución en el entorno (mediante el operador de mutación).

El cuadro 7.8 resume el algoritmo para el mecanismo de contraejemplos. Grosso modo, la idea que subyace este algoritmo es, por un lado, la variación (a través del operador de *mutación*) de la acción del anticuerpo seleccionado cuando éste coincide con algún contraejemplo aprendido previamente y, por otro, la generalización (a través del operador de *subsumción*) de aquellos nuevos contraejemplos aprendidos que siguen un patrón similar de comportamiento. Cabe notar que la creación de un nuevo contraejemplo solamente ocurre cuando el SIA recibe un refuerzo negativo por la ejecución de la última acción, y cuando éste no existe ya en el conjunto de contraejemplos (CE).

Con el fin de ilustrar la dinámica del mecanismo propuesto, supóngase que se desea automatizar, mediante el SIA, el sistema de aspersión y riego de un cultivo hidropónico. Así pues, se consideran tres variables de entrada o sensores (i.e., pronóstico climático, humedad del ambiente, y temperatura externa); y un actuador que controla la apertura y cierre del sistema de aspersión. En la tabla 7.1 se presenta la codificación sub-simbólica de los sensores y del actuador, y en la tabla 7.2 se presentan las pautas que debe aprender el SIA (acciones correctas y contraejemplos).

Así, por ejemplo, si la entrada sensorial es ⟨`lluvioso, húmedo, frío`⟩ (`000`) se activaría el actuador ⟨`cerrar`⟩ (`0`); mientras que, si la entrada sensorial es ⟨`lluvioso, seco, calor`⟩ (`011`) se activaría el actuador ⟨`abrir`⟩ (`1`); y así sucesivamente. Supóngase ahora que se presentan al SIA una a una las pautas sensoriales de la tabla 7.2, y que éste escoge, en cada ocasión, un anticuerpo diferente cuya acción produce siempre un

convergencia en el aprendizaje es incrementada (como se demostrará en el capítulo sobre la Experimentación), haciendo al sistema más robusto.

Método: activarSIA

```
1.  Variables
2.     antigeno_t      ↓ entrada sensorial en el instante actual.
3.     antigeno_{t-1}  ↓ entrada sensorial en el instante anterior.
4.     accion_t        ↑ acción del anticuerpo resultante del proceso de selección en el
                         instante actual.
5.     accion_{t-1}    ↑ acción del anticuerpo resultante del proceso de selección en el
                         instante anterior.
6.  Inicio Procedimiento
7.     Si refuerzo <0 entonces:
8.        crearContraejemplo(antigeno_{t-1}, accion_{t-1})
9.     Fin Si
10.    accion_t ← activar(antigeno_t) //ejecuta el proceso epigenético del SIA (se-
          lección clonal, maduración, etc.)
11.    accion_t ← comprobarContraejemplos(antigeno_t, accion_t)
12. Fin Procedimiento
```

Método: crearContraejemplo

```
1.  Variables
2.     CE              conjunto de contraejemplos.
3.     antigeno_{t-1}  ↓ entrada sensorial en el instante anterior.
4.     accion_{t-1}    ↓ acción del anticuerpo resultante del proceso de selección en el
                         instante anterior.
5.  Inicio Procedimiento
6.     Si el contraejemplo no existe en CE entonces:
7.        ces ← buscarContraEjSimilares(antigeno_{t-1}, accion_{t-1})
             //busca todos aquellos contraejemplos que sólo difieran en un bit
8.        Si existen contrajemplos similares entonces:
9.           subsumir(antigeno_{t-1}, accion_{t-1}, ces) //se crea un nuevo con-
                traejemplo más general que subsume a los más específicos
10.       De lo contrario se inserta el nuevo contraejemplo en CE
11.    Fin Si
12. Fin Procedimiento
```

Método: comprobarContraejemplos

```
1.  Variables
2.     CE              conjunto de contraejemplos.
3.     antigeno_t      ↓ entrada sensorial en el instante actual.
4.     accion_t        ↑ acción del anticuerpo resultante del proceso de selección en el
                         instante actual.
5.  Inicio Procedimiento
6.     Mientras que existan contraejemplos en CE que coincidan con
       la pauta ⟨antigeno_t, accion_t⟩ hacer:
7.        mutarAccion(accion_t)
8.     Fin Mientras
9.  Fin Procedimiento
```

Algoritmo 7.8: Algoritmo de Aprendizaje por Contraejemplos

refuerzo negativo (esto con el fin de ilustrar la dinámica del mecanismo de contraejemplos de forma aislada al proceso epigenético mediante el cual el SIA aprende). También asúmase que el SIA opera reactivamente como un sistema estímulo-respuesta.

Variable (Entrada/Salida)	Valores	Codificación Sub-simbólica
Pronóstico Climático (PC)	Lluvioso	0
	Soleado	1
Humedad del Ambiente (HA)	Húmedo	0
	Seco	1
Temperatura Externa (TE)	Frío	0
	Calor	1
Actuador Aspersor (AA)	Cerrar	0
	Abrir	1

Tabla 7.1: Codificación de los tres sensores y del actuador para la automatización del sistema de aspersión.

Sensores			Actuador	Contraejemplo
PC	HA	TE		
0	0	0	0	000:1
0	0	1	0	001:1
0	1	0	0	010:1
0	1	1	1	011:0
1	0	0	0	100:1
1	0	1	1	101:0
1	1	0	1	110:0
1	1	1	1	111:0

Tabla 7.2: Correspondencia entre los sensores y las acciones que idealmente deberían enviarse al actuador. Nótese que las salidas de los contraejemplos son los inversos de los valores deseados para el actuador.

Instante t = 0: dada la entrada ⟨`lluvioso, húmedo, frío`⟩, el SIA escoge el anticuerpo "000:1", indicando que el actuador debe abrirse (⟨`abrir`⟩).

Instante t = 1: el sistema recibe un refuerzo negativo debido a que la acción ejecutada en el isntante anterior fue incorrecta y, por tanto, el SIA crea un nuevo contraejemplo: 000:1. El sistema percibe ahora la entrada ⟨`lluvioso, húmedo, calor`⟩ para la cual selecciona el anticuerpo "001:1", cuya acción (⟨`abrir`⟩) es incorrecta.

Instante t = 2: el SIA crea un segundo contraejemplo: 001:1 debido a que la acción anterior fue incorrecta. Luego, el sistema comprueba si es posible generalizar los dos contraejemplos existentes, y debido a que estos difieren en tan sólo un bit de información, se crea entonces un contraejemplo más general que subsume los dos más específi-

cos: “00#:1”. Nótese que, el bit que es diferente en los dos contraejemplos específicos es reemplazando por un comodín (#) en el contraejemplo más general. En este mismo instante el sistema percibe la entrada sensorial ⟨lluvioso, seco, frío⟩ y, como respuesta, envía la acción ⟨abrir⟩ del anticuerpo “010:1”.

Instante t = 3: debido a la acción incorrecta del instante 3, el sistema crea el contraejemplo 010:1 debido al refuerzo negativo recibido por la acción anterior. Como este contraejemplo resulta muy similar al contraejemplo identificado en el instante 1, el SIA crea entonces un contraejemplo más general que subsume a los dos más específicos: 0#0:1. En este mismo instante el sistema percibe la entrada sensorial ⟨lluvioso, seco, calor⟩ y activa la respuesta ⟨cerrar⟩ correspondiente al anticuerpo: “011:0”.

Instante t = 4: el sistema crea el contraejemplo 011:0 debido al refuerzo negativo de la última acción. Por ahora, no existen contraejemplos lo suficientemente similares a éste, con lo cual no es subsumido. El SIA procesa la entrada sensorial ⟨soleado, húmedo, frío⟩ y, a partir del anticuerpo “100:1”, envía la acción ⟨abrir⟩ al actuador.

Instante t = 5: una vez más, debido al refuerzo negativo producido por la ejecución de la acción anterior, se crea un contraejemplo: 100:1. Este nuevo contraejemplo difiere en tan sólo un bit de información con el contraejemplo del instante 1, con lo cual se crea un nuevo contraejemplo más general: #00:1. Asimismo, en este instante de tiempo el sistema recibe la lectura sensorial ⟨soleado, húmedo, calor⟩ y activa la acción del anticuerpo “101:0”, que corresponde a la acción ⟨cerrar⟩.

Instante t = 6: el SIA crea el contraejemplo 101:0 debido a la última acción incorrecta. El sistema realiza una búsqueda sobre el CE y no encuentra contraejemplos que cumplan con la condición de similitud, con lo cual el contraejemplo creado no es subsumido. Luego, el sistema recibe la entrada sensorial ⟨soleado, seco, frío⟩ y envía como respuesta al actuador la acción ⟨cerrar⟩, correspondiente al anticuerpo “110:0”.

Instante t = 7: se crea el contraejemplo correspondiente a la última acción incorrecta: 110:0. Este contraejemplo no es subsumido debido a que no existen pautas similares en el CE. Luego, es procesada la entrada sensorial ⟨soleado, seco, calor⟩ y el SIA recomienda la acción ⟨cerrar⟩ correspondiente al anticuerpo “111:0”.

Instante t = 8: se crea el contraejemplo para la última acción incorrecta: 111:0. Este nuevo contraejemplo posee una alta afinidad con tres de los contraejemplos que se encuentran en el CE, por tanto, se crean tres nuevos contraejemplos más generales: (1) el contraejemplo #11:0 que subsume los contraejemplos de los instantes 4 y 8; (2) el contraejemplo 1#1:0 que subsume los contraejemplos de los instantes 6 y 8; y (3), el contraejemplo 11#:0 que subsume los contraejemplos de los instantes 7 y 8.

La estructura jerárquica de los contraejemplos creados se muestra en la figura 7.11. Obsérvese que los contraejemplos más generales son los únicos que se encuentran activos y, por tanto, son los únicos contraejemplos contra los cuales se comparará, en cada ciclo cognitivo, el anticuerpo seleccionado por el SIA. En la medida que aparezcan nuevos contraejemplos similares a los ya existentes, serán subsumidos por contraejemplos más

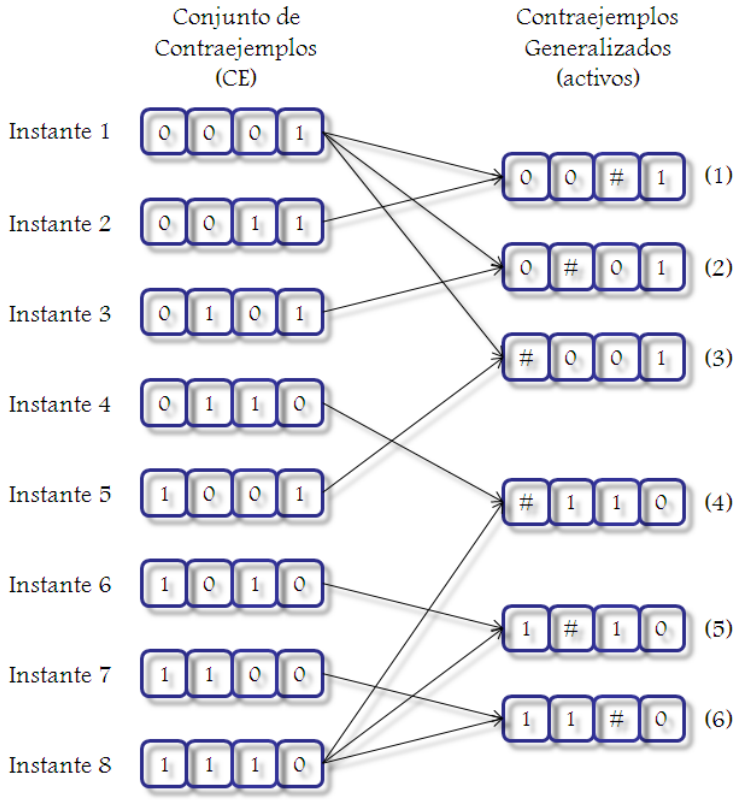


Figura 7.11: Ejemplo de una estructura jerárquica de Contraejemplos activos.

generales, reduciéndose, de esta manera, el número de comparaciones y validaciones que tendría que hacer el sistema por cada anticuerpo seleccionado. Por ejemplo, si el SIA identificase el contraejemplo específico "100:0", entonces los contraejemplos generales 5 y 6 podrían ser subsumidos por el nuevo contraejemplo general $\boxed{1\#\#:0}$, debido a que éste incluiría todos los casos particulares: "100:0", "101:0", "110:0", y "111:0"; con lo cual, cada anticuerpo solamente se validaría contra éste contraejemplo general y no contra cada una de sus instancias específicas. No obstante, si en algún momento el contraejemplo general no realiza una predicción correcta (i.e., predice un refuerzo negativo correspondiente a una acción incorrecta, pero ésta resulta ser correcta) entonces el contraejemplo general es eliminado y todas sus instancias específicas vuelven a hacerse activas.

Como se ha mencionado anteriormente, el mecanismo de contraejemplos determina cuándo la acción de un anticuerpo debe ser mutada a fin de evitar la obtención de un refuerzo negativo. Por ejemplo, supóngase que después de aprender los diferentes contraejemplos de la figura 7.11, el SIA percibe la entrada sensorial: ⟨`lluvioso, seco, calor`⟩, y activa como resultado el anticuerpo "011:0" cuya acción indica que el aspersor debe "cerrase". En ese instante, el sistema buscará esta pauta dentro del conjunto de contraejemplos aprendidos y determinará que la acción es incorrecta debido a que existe evidencia de que producirá un refuerzo negativo después de su ejecución. Entonces, el SIA procederá a mutar el único bit de la acción del anticuerpo de la siguiente manera:

"011:**1**", lo que será correcto puesto que indicará al actuador del aspersor que debe "abrirse". En el caso de que la acción tenga más de un bit, se elegirá aleatoriamente la posición que será mutada, y el proceso se repetirá mientras se encuentre un contraejemplo que coincida con la pauta mutada.

En síntesis, debido a que durante el proceso epigenético pueden producirse anticuerpos poco aptos (por acción de los operadores de mutación y recombinación), los cuales obtendrán, con mayor probabilidad, refuerzos negativos, el mecanismo de contraejemplos ofrece una alternativa para mutar dichos anticuerpos antes de que sean ejecutados, evitando así el coste que supondría ejecutar su acción en el ambiente y esperar hasta recibir un refuerzo negativo por ello.

7.4.4. Modelo de Interacción

En las figuras 7.12 y 7.13 se presentan los diagramas de secuencia simplificados para el SIA. Dentro de lo que cabe destacar en estos diagramas son los dos posibles escenarios de ejecución del SIA. En el primer escenario se generaría la población (acervo) de anticuerpos, así como las respectivas clases controladoras del sistema (figura 7.12). En este primer escenario, el sistema no recibe refuerzo ambiental, motivo por el cual el flujo de control sólo refleja la Selección Clonal y el proceso de selección de anticuerpos dirigido por el Sistema de Asignación de Recompensas. En un segundo escenario (figura 7.13), se percibe una señal de refuerzo ambiental, la cual dirige ahora las meta-dinámicas, el proceso de recorte, el mecanismo de diversificación, y el mecanismo de contraejemplos. Este segundo escenario se ejecuta indefinidamente a menos que se reinicie totalmente la población de anticuerpos, caso en el cual se ejecutaría el primer escenario nuevamente.

7.5. Módulo de Producciones

El módulo de Producciones de cada *Comportamiento* es de enfoque cognitivista, en contraste con los módulos Conexionista e Inmunológico que son de enfoque emergente. Este módulo tiene como objetivo capturar el conocimiento explícito mediante representaciones simbólicas o localistas (Clark y Karmiloff-Smith, 1993), donde cada unidad es fácilmente interpretable y tiene un claro significado conceptual, por ejemplo, una etiqueta semántica.

El conocimiento explícito (simbólico) en contraposición con el conocimiento implícito (sub-simbólico), se compone de piezas substanciales de información que pueden ser accedidas y manipuladas directamente. Este conocimiento debe ser fácilmente accesible debido a que es la base del pensamiento complejo. Mientras que el conocimiento implícito representa información sub-consciente, el conocimiento explícito representa información que se procesa en estados de mayor conciencia (Smolensky, 1988; Sun, 1994).

Por consiguiente, este módulo emplea como mecanismo de representación simbólica, un sistema basado en reglas de producción (Klahr et al., 1987). Las reglas de producción son un método procedimental de representación del conocimiento, es decir, soportan la

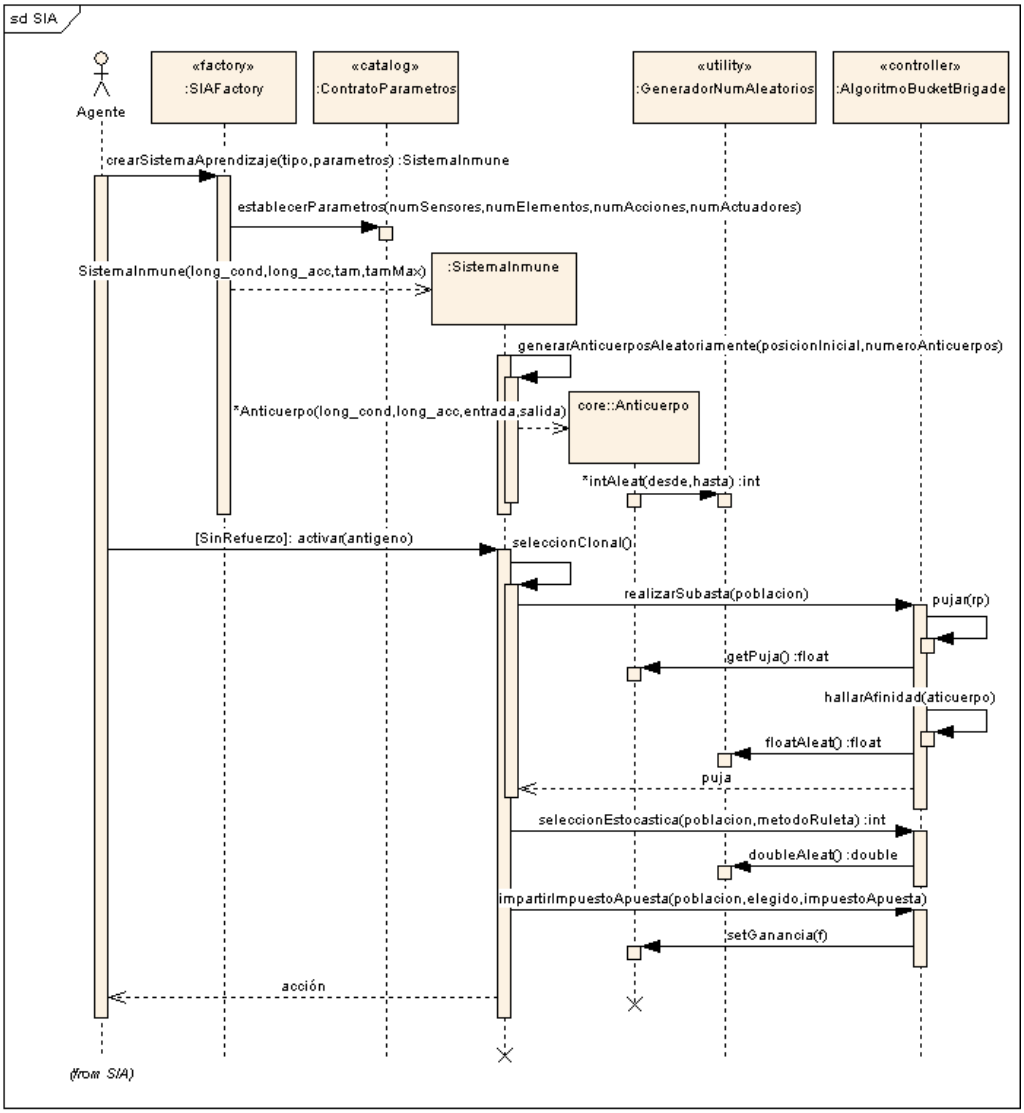


Figura 7.12: Diagrama de Secuencia para el SIA. Escenario de activación sin refuerzo.

representación de las relaciones inferenciales del dominio, en contraposición a los métodos declarativos que hacen mayor énfasis en la representación de los hechos. En las siguientes secciones se describe el modelo de representación y los tipos de estructuras basadas en reglas de producción que emplean los componentes RES y RP del *Comportamiento*.

7.5.1. Representación

De manera similar a los componentes del nivel emergente del *Comportamiento*, el módulo de producciones recibe como entrada un conjunto de pautas espaciotemporales organizadas en parejas del tipo ⟨`propiedad, valor`⟩. Estas pautas, al igual que en el nivel

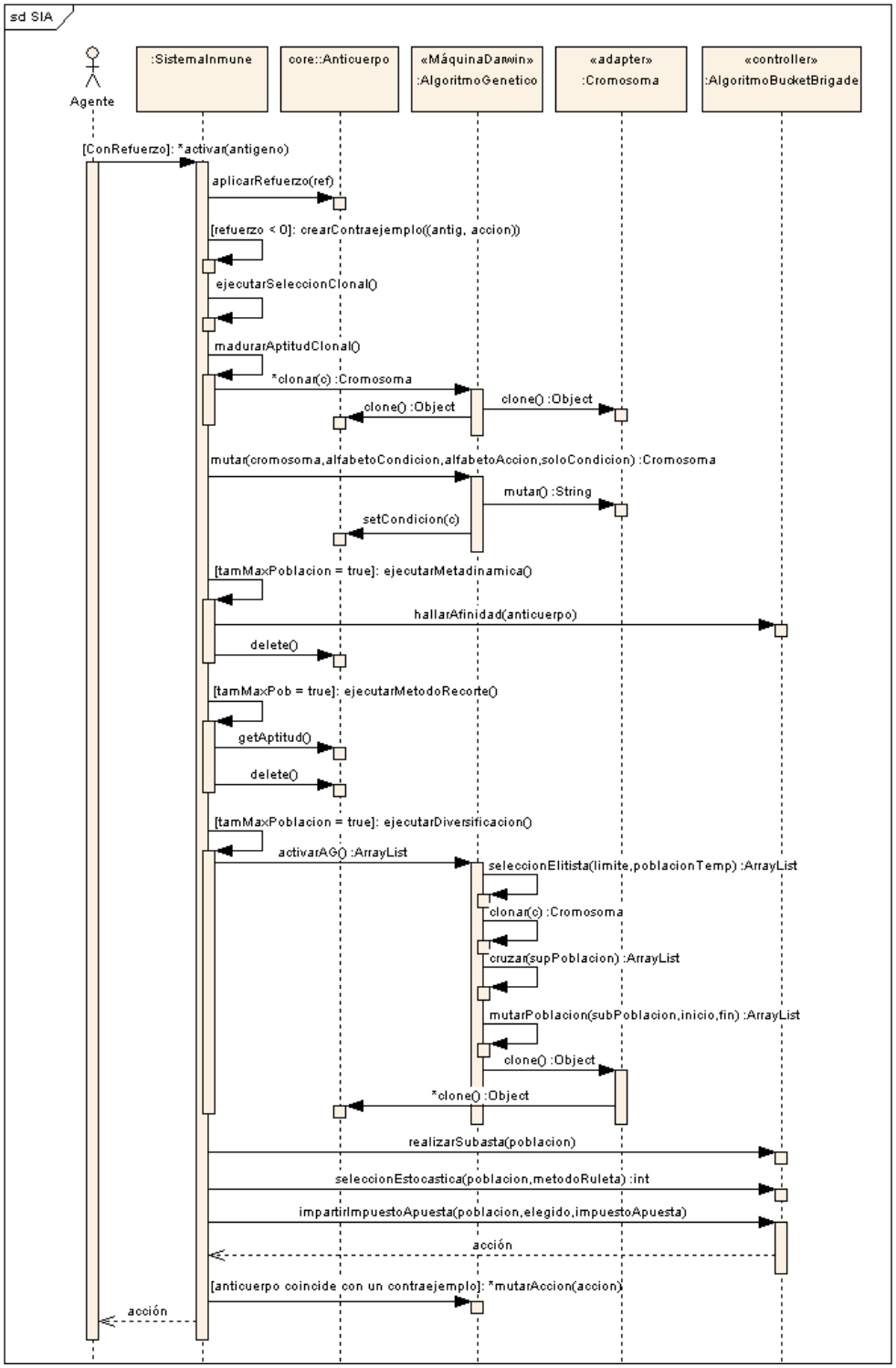


Figura 7.13: Diagrama de Secuencia para el SIA. Escenario de activación con refuerzo.

emergente, corresponden a un sub-conjunto de unidades activas en la MT que represen-

tan el estado actual del mundo (i.e., perceptos, asociaciones locales, estados anticipados, metas, etc).

Usualmente, una regla de producción es definida de la forma: "Si *antecedentes* Entonces *consecuentes*", donde los antecedentes son las condiciones que han de cumplirse para activarse la regla y, los consecuentes, las conclusiones, acciones o hipótesis de dicha regla. De manera similar a lo definido para los componentes del nivel emergente (SIA y PMR), los antecedentes de una regla de producción corresponden a un conjunto de unidades en la MT, mientras que los consecuentes envían un conjunto de acciones recomendadas al módulo Efector.

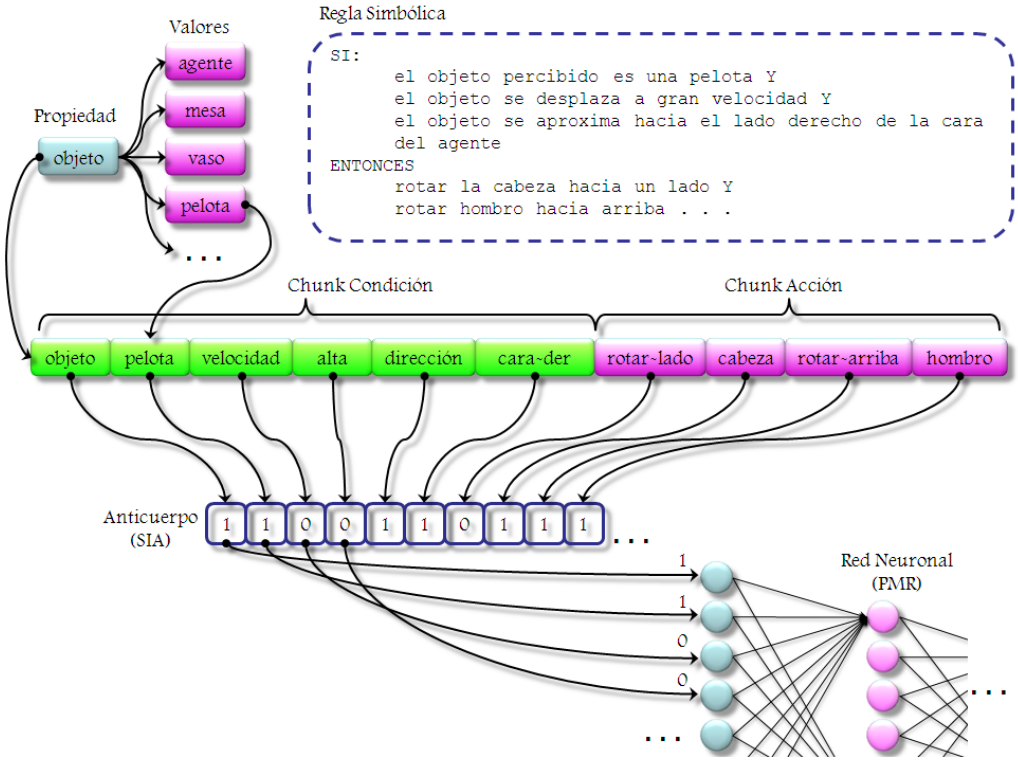


Figura 7.14: Correspondencia entre representaciones simbólicas y sub-simbólicas.

En la figura 7.14 se presenta un ejemplo de la estructura de una regla simbólica y su correspondencia con las representaciones sub-simbólicas en el nivel emergente. Las partes condición y acción de una regla simbólica encapsulan bloques de conocimiento con significado semántico, denominados *chunks* [15]. Los *chunks* también han sido propuestos como unidades significativas en sistemas cognitivos artificiales de tipo simbólico tales

[15] El término *chunk*, es empleado comúnmente por las ciencias cognitivas, en especial por la psicología cognitiva y la mnemotecnia, para referirse a las estructuras de la memoria de largo plazo que pueden ser usadas como unidades de percepción y significado, de lo cual deriva el término *chunking* que se refiere a la adquisición de tales estructuras a través de mecanismos de aprendizaje (Gobet et al., 2001; Bapi et al., 2005). Miller (Miller, 1956), psicólogo especialista en ciencia cognitiva, llamó "chunk" (trozo) a la cantidad total de información capaz de ser recibida en la memoria inmediata o de trabajo (instante de percepción), procesada por el sistema cognitivo, y almacenada en las memorias de largo plazo.

como SOAR (Lehman et al., 2006), ACT-R (Anderson, 1996), CLARION (Sun et al., 2001), entre otros.

Así pues, en METÁFORA, una regla de producción estaría conformada por dos tipos de chunks: un chunk condición y un chunk acción. Un chunk condición encapsula tanto las características de aquello que es percibido (e.g., los objetos del mundo) como los estados internos del agente, mientras que el chunk acción caracteriza una o varias acciones específicas (externas o internas) que ejecutará el agente en un instante determinado. En la figura 7.14 se presenta un ejemplo de una regla simbólica conformada por chunks. A fin de comprender la representación de una regla simbólica, supóngase la siguiente situación: el agente se encuentra observando su mundo, cuando repentinamente se aproxima un objeto hacia él a gran velocidad, situación que activaría posiblemente una regla de producción como la siguiente:

SI
⟨el objeto percibido es una pelota⟩ Y
⟨el objeto se desplaza a gran velocidad⟩ Y
⟨el objeto se aproxima hacia el lado derecho de la cara del agente⟩
ENTONCES
⟨rotar la cabeza hacia un lado⟩ Y
⟨rotar hombro hacia arriba⟩

En la figura 7.14 es posible observar cómo el chunk condición se forma a partir de la combinación específica de pautas ⟨propiedad, valor⟩, por ejemplo, la *propiedad* "objeto" tiene varios valores asociados (e.g., agente, mesa, vaso, pelota, etc.). En este caso, el objeto observado es una pelota con lo cual se crea la pauta ⟨objeto, pelota⟩. Cada una de las pautas del chunk condición de la regla sigue el mismo proceso de asociación, de manera que la agrupación del conjunto de pautas caracteriza la situación observada por el agente. Así mismo, el chunk de acción se conforma por pautas que dirigen la actuación del agente.

A través de los procesos de aprendizaje vertical (i.e., asimilación y consolidación), es posible generar una correspondencia entre cada regla simbólica en el nivel cognitivista y sus respectivas representaciones sub-simbólicas en el nivel emergente del *Comportamiento*, y viceversa. En otras palabras, el conjunto de parejas ⟨propiedad, valor⟩ que conforma el chunk condición y el chunk acción de cada regla de producción, tendrían una correspondencia con la parte condición y la parte acción de una regla heteroasociativa de representación sub-simbólica (un anticuerpo en el SIA), y a su vez, una correspondencia con un patrón de activación (entradas/salidas) en la red PMR asociada. Estos mecanismos de aprendizaje vertical serán descritos al detalle en el apartado 7.6.

En cuanto a los mecanismos que se encargan de seleccionar la regla de producción que enviará su acción recomendada al módulo Efector, se definen dos variantes: en la primera, el mecanismo realiza un emparejamiento total, es decir, que todas las parejas ⟨propiedad, valor⟩ que conforman la parte condición de la regla deben coincidir con el estado actual observado; y la segunda, el mecanismo realiza un emparejamiento

parcial, es decir, que la regla puede ser seleccionada si coincide un número mínimo requerido de pautas con respecto al estado actual observado. En el emparejamiento parcial se emplea una función de activación basada en una sumatoria ponderada de los pesos de las pautas ⟨`propiedad, valor`⟩ en el nivel emergente, como se explicará más adelante. Cabe notar, que como consecuencia, el emparejamiento parcial de reglas permite la generalización de situaciones debido a que una regla puede coincidir parcialmente con una o muchas situaciones dadas.

Como es evidente, cuando es empleado el emparejamiento parcial, el estado actual x "activará" todos los chunks de condición que están contenidos en éste, esto es, se activarán todas las reglas que en su chunk condición especifiquen un subconjunto de pautas ⟨`propiedad, valor`⟩ contenidas también en el estado x. Así pues, el emparejamiento parcial produce la activación de múltiples reglas dado un estado actual observado, con lo cual resulta necesario definir un mecanismo que arbitre la competencia entre las reglas a fin de decidir cuál de estás enviará su acción como respuesta.

El mecanismo que dirige la competición entre las reglas puede emplear una función *max*[16] o una función *softmax* como se verá más adelante. Adicionalmente, el mecanismo de competición seleccionará la regla ganadora (en cada ciclo cognitivo), en función de una medida de utilidad que determinará el nivel de activación de cada regla. En el caso de que no se disponga de una medida de utilidad, el sistema empleará un mecanismo estocástico para determinar la regla que será activada.

A continuación, se describen los dos tipos de reglas simbólicas empleadas en el módulo de Producciones: las Reglas de Extracción Sub-simbólica (RES) y las Reglas Predeterminadas (RP).

7.5.2. Reglas de Extracción Sub-simbólicas (RES)

Este tipo de reglas es creado como resultado del proceso de aprendizaje ascendente (bottom-up) donde es extraído el conocimiento sub-simbólico del nivel emergente para luego ser consolidado en el nivel cognitivista.

La idea básica del algoritmo es la siguiente: si se determina que una regla subsimbólica en el nivel emergente resulta ser satisfactoria (i.e., que satisface cierto criterio), entonces el *Comportamiento* construirá una regla simbólica, cuya acción se corresponde con aquella recomendada por el nivel emergente, y cuya condición se corresponde con el estado actual percibido por el agente. Luego, en las subsecuentes interacciones del agente con el ambiente, el *Comportamiento* refinará la regla mediante la evaluación de sus consecuencias: si la consecuencia es satisfactoria, el *Comportamiento* intentará generalizar la condición de la regla para hacerla más universal; si la consecuencia no es satisfactoria, entonces la condición de la regla debe hacerse más específica (y exclusiva para el estado actual observado). El cuadro 7.9 presenta el pseudocódigo del algoritmo.

Existe evidencia empírica y teórica a favor de este algoritmo en (Bruner et al., 1956; Haygood y Bourne, 1970; Sun et al., 2001). A continuación, se discutirán los detalles

[16]En este caso, la función selecciona la acción que maximiza una medida en el estado actual, o en otras palabras, selecciona la mejor opción para el estado actual de acuerdo con esa medida.

Método: extraerReglas

```
1.  Variables
2.     reglasRES      ↑↓ conjunto de reglas extraídas del nivel emergente.
3.     estadoActual   ↑↓ entrada sensorial percibida por el agente en el instante actual.
4.  Inicio Procedimiento
5.     Para cada estadoActual percibido por el agente hacer:
6.        Si existen reglas extraídas entonces:
7.           actualizarEstadisticas( reglasRES )
8.        Fin Si
9.        Si se cumple el criterio de extracción de reglas y no
          existe una regla que empareje el estado actual entonces:
10.          regla ← extraerRegla( estadoActual ))
11.          reglasRES ← almacenarRegla( regla ))
12.       Fin Si
13.       Si el resultado para el criterio de generalización es
          satisfactorio entonces:
14.          reglaGeneralizada = generalizar( estadoActual ) //se
             genera una regla generalizada para el estado actual dado.
15.          reglasRES ← eliminarReglas( estadoActual ) //se revisan
             todas las reglas generalizadas que emparejan el estado actual y se eli-
             minan del repositorio.
16.          reglasRES ← agregarRegla( reglaGeneralizada ) //se
             agrega la versión generalizada de esta regla al repositorio.
17.       Fin Si
18.       Si el resultado para el criterio de especialización no
          es satisfactorio entonces:
19.          reglasEspecializada = especializar( estadoActual )
             //se genera una regla especializada para el estado actual dado.
20.          reglasRES ← eliminarReglas( estadoActual ) //se revisan
             todas las reglas especializadas que emparejan el estado actual y se
             eliminan del repositorio.
21.          reglasRES ← agregarRegla( reglaEspecializada ) //se
             agrega la versión revisada (especializada) de esta regla al repositorio.
22.       Fin Si
23.    Fin Para
24.  Fin Procedimiento
```

Algoritmo 7.9: Algoritmo para la Extracción de Reglas Sub-simbólicas.

de las operaciones empleadas en el algoritmo, así cómo los criterios de medición que determinarán si un resultado es satisfactorio o no.

En cada ciclo cognitivo se examina la siguiente información: (x, y, r, a), donde x es el estado del mundo antes de que la acción a sea ejecutada, y es el nuevo estado después de que la acción a ha sido ejecutada, y r es el refuerzo recibido después de la acción a. Considérese además C como la parte condición de una regla. Tomando esto en cuenta, se definen dos estadísticas para cada regla. Por un lado, se define una "consecuencia positiva" $CP_{(x,a)}(C)$, que incrementa su valor en uno cada vez que se cumple que: las pautas de la condición C de la regla coinciden con el estado observado, la acción a es ejecutada, y el resultado es satisfactorio; y, por otro, una "consecuencia negativa" $CN_{(x,a)}(C)$, que incrementa su valor en uno cuando se cumple que: el estado observado coincide con las pautas de la condición C de la regla, la acción a es ejecutada, y el

resultado no es satisfactorio. La condición de satisfacibilidad del resultado de una regla depende de su nivel de positividad en comparación con otras reglas, procedimiento que se detallará más adelante.

Los niveles de "positividad" o "negatividad" (que son determinados respectivamente por los valores de CP y CN) pueden ser estimados tomando en cuenta un criterio basado en la retroalimentación inmediata. Esto es, el nivel de positividad es incrementado si $r > umbral_{RES}$, de lo contrario, es incrementado el nivel de negatividad. El umbral $umbral_{RES}$ debe ser determinado para cada *Comportamiento* de la concreción contextual de la arquitectura en función de las señales de refuerzo recibidas por el mismo. Por defecto, $umbral_{RES} = \max(r) \times 0.6$, esto es, el nivel de positividad será incrementado si el *Comportamiento* recibe un refuerzo que excede el 60 % del valor del refuerzo máximo. Así pues, cada estadística es actualizada de la siguiente forma:

- $CP = CP + 1$, cuando se cumple el criterio para la positividad,
- $CN = CN + 1$, cuando NO se cumple el criterio para la positividad,
- Al finalizar cada iteración, son descontados estos valores así: $CP = CP * 0.9$ y $CN = CN * 0.9$. Este descuento tiene como fin reducir las estadísticas en el tiempo, evitando así los estados estacionarios del sistema (Sun et al., 1998).

Tomando en cuenta estas estadísticas, se calcula una medida de ganancia de información (GI) para cada regla, de la siguiente manera:

$$GI(A,B) = log_2 \frac{CP_a(A) + c_1}{CP_a(A) + CN_a(A) + c_2} - log_2 \frac{CP_a(B) + c_1}{CP_a(B) + CN_a(B) + c_2} \quad (7.23)$$

donde: A y B son dos condiciones de reglas diferentes que conducen a la misma acción. a. c_1 y c_2 son dos constantes que influyen sobre el impacto de CP y CN (por defecto son c_1 = 1 y c_2 = 2)

Esencialmente, esta medida compara los niveles de positividad entre las partes condición de dos reglas diferentes A y B (empleando un estimador de Laplace[17]). Si A puede mejorar en cierto grado el nivel de B, entonces A es considerada mejor que B. En el caso de que una regla resulte ser mejor que la regla del tipo "emparejar-todos-los-estados" (e.g., una regla con la misma acción que la primera pero cuyas *propiedades* en su parte condición coinciden con todos los posibles estados observados)[18], entonces la regla es considerada "satisfactoria".

A continuación, se describen los criterios que emplea el componente RES para extraer, generalizar, y especializar las reglas de producción, basándose para ello en los criterios de positividad y satisfacibilidad del ciclo cognitivo actual (x, y, r, a):

[17]Este es un método empleado comúnmente en la Programación Lógica Inductiva, y es bien justificada en el campo empírico (Lavrac y Dzeroski, 1994)

[18]Claramente, existen tantas reglas del tipo "emparejar-todos-los-estados" como posibles acciones. Esto es, el número de reglas del tipo "emparejar-todos-los-estados" es igual a $|prop_1| \times |prop_2| \times \ldots |prop_n|$, donde $1, 2, \ldots, n$ son todas las *propiedades* de la parte acción de la regla, y $|prop_i|$ mide el número de posibles valores de la *propiedad* i.

Extracción: si el ciclo cognitivo actual es positivo (de acuerdo con el criterio de positividad actual) y si no existe ninguna regla simbólica (en el componente RP) aplicable en este ciclo, se establece una regla $C \rightarrow a$ que especifica todos los valores de las *propiedades* que conforman exactamente el estado actual x, y donde a es la acción ejecutada en el ciclo actual. Cuando se extrae una regla inicial, se verifica si existen unidades activas en la MT, de ser así, éstas pueden ser incluidas en la condición de la regla extraída (véase la figura 7.15).

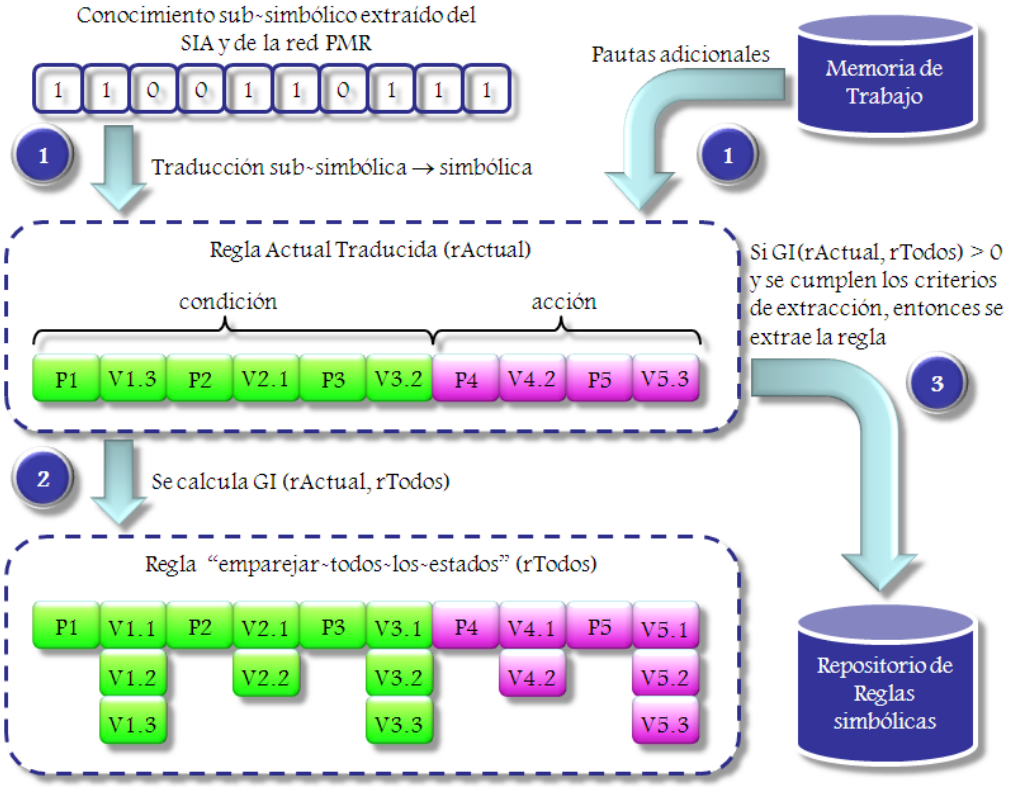


Figura 7.15: Extracción de Reglas Simbólicas. Donde $P_1, P_2, \ldots P_m$ son las *propiedades* de las partes condición y acción de cada regla, $V_{1.1}, V_{1.2} \ldots V_{2.1}, V_{2.2} \ldots V_{m.n}$ son los respectivos valores de cada *propiedad* m, y $rTodos$ corresponde a la regla "emparejar-todos-los-estados".

Cabe notar, que durante este proceso resulta necesario controlar el nivel de proliferación de reglas extraídas. Adicional al criterio de positividad, se define un parámetro p_{RES} que determina la probabilidad de que una regla sea extraída, en el caso de cumplirse el criterio para la extracción de la regla (e.g., si $r > umbral_{RES}$). Por otro lado, se define un parámetro de densidad d_{RES} que determina la frecuencia mínima de emparejamientos en la parte condición necesarios para mantener una regla en el repositorio. Por ejemplo, si el parámetro es definido en $1/n$, se requiere que cada n pasos al menos una pareja $\langle$`propiedad, valor`$\rangle$ de la parte condición de la regla, coincida con una pauta igual en el estado observado. De lo contrario, la regla será eliminada del repositorio.

Por otro lado, los operadores de *Generalización* y *Especialización* se basan en la medida de ganancia de información de la regla (ecuación 7.23). La generalización implica

la agregación de un *valor* adicional a una de las *propiedades* en la parte condición de la regla, de esta manera la regla tendrá más oportunidades de emparejar con el estado actual, mientras que la especialización implica la eliminación de uno de los valores de alguna de las *propiedades* de la condición de la regla, de manera que ésta tendrá menos oportunidades de emparejar con el estado actual.

Generalización: el proceso de generalización ocurre si se cumple que $GI(C, r_{total}) > umbral_G$, y $max_{C'}(C', C) \geq 0$, donde C es la condición de la regla actual (la cual empareja con el estado actual), r_{total} se refiere a una regla del tipo "emparejar-todos-los-estados" (es decir, una regla con la misma acción especificada por la regla original pero con una condición que empareja con cualquier estado), C' es una condición modificada tal que $C' = C$ más un *valor* (e.g., C' tiene un *valor* adicional en una de las *propiedades* de la condición). Así pues, **si la regla actual es satisfactoria y su parte condición (generalizada) la hace potencialmente mejor**, entonces hacer $C'' = max_{C'} GI(C', C)$ como la nueva condición (generalizada) de la regla. Luego de esto, se reinicializan todas las estadísticas de la regla. Cualquier regla que pueda ser reemplazada por la regla generalizada, será situada dentro de la lista de reglas hijas de la generalizada. La lista de reglas hijas tiene como fin mantener inactivas aquellas reglas que son más específicas, y que están, por consiguiente, totalmente cubiertas por la regla actual. Esto resulta útil cuando la regla generalizada deja de ser valida, momento en el cual son activadas nuevamente todas las reglas de la lista de reglas hijas.

Especialización: la especialización ocurre si se cumple que $GI(C, r_{total}) < umbral_E$, y $max_{C'}(C', C) > 0$, donde C es la condición de la regla actual (la cual empareja con el estado actual), r_{total} se refiere a una regla del tipo "emparejar-todos-los-estados", C' es una condición modificada tal que $C' = C$ menos un *valor* (e.g., C' tiene al menos un *valor* menos en una de las *propiedades* de la condición). Así pues, **si la regla actual NO es satisfactoria, pero su parte condición (especializada) la hace potencialmente mejor**, entonces hacer $C'' = max_{C'} GI(C', C)$ como la nueva condición (especializada) de la regla[19]. Luego, se reinicializan todas las estadísticas de la regla. Nótese que, a menos que alguna *propiedad* tenga más de un *valor*, no se permite la eliminación del último *valor* de una *propiedad*. La eliminación del último *valor* de cierta *propiedad* de una regla hace imposible que ésta empareje con algún estado, con lo cual, en este caso, resultaría más conveniente eliminar directamente la regla.

En la figura 7.16, se ofrece un ejemplo del proceso de extracción, generalización y especialización de reglas simbólicas. Supóngase un ambiente de vida artificial donde existe un animal virtual (animat) que interactúa con su entorno. En primera instancia, el sistema identifica una pauta cognitiva que se ha formado en el nivel emergente y que cumple cierto criterio de "eficiencia" necesario para iniciar el proceso de extracción. Este criterio de eficiencia es determinado por una estadística que mantiene la propia regla sub-simbólica (e.g., el nivel de aptitud de un anticuerpo del SIA, o el registro de un refuerzo positivo reiterado obtenido por una pauta de activación neuronal de una red PMR). Si se cumplen los criterios de positividad y de extracción, se construye una regla simbólica a partir de la sub-simbólica. En el ejemplo, esta regla corresponde con un unidad de

[19] Claramente, es necesario hacer $umbral_E \leq umbral_G$ a fin de evitar la oscilación.

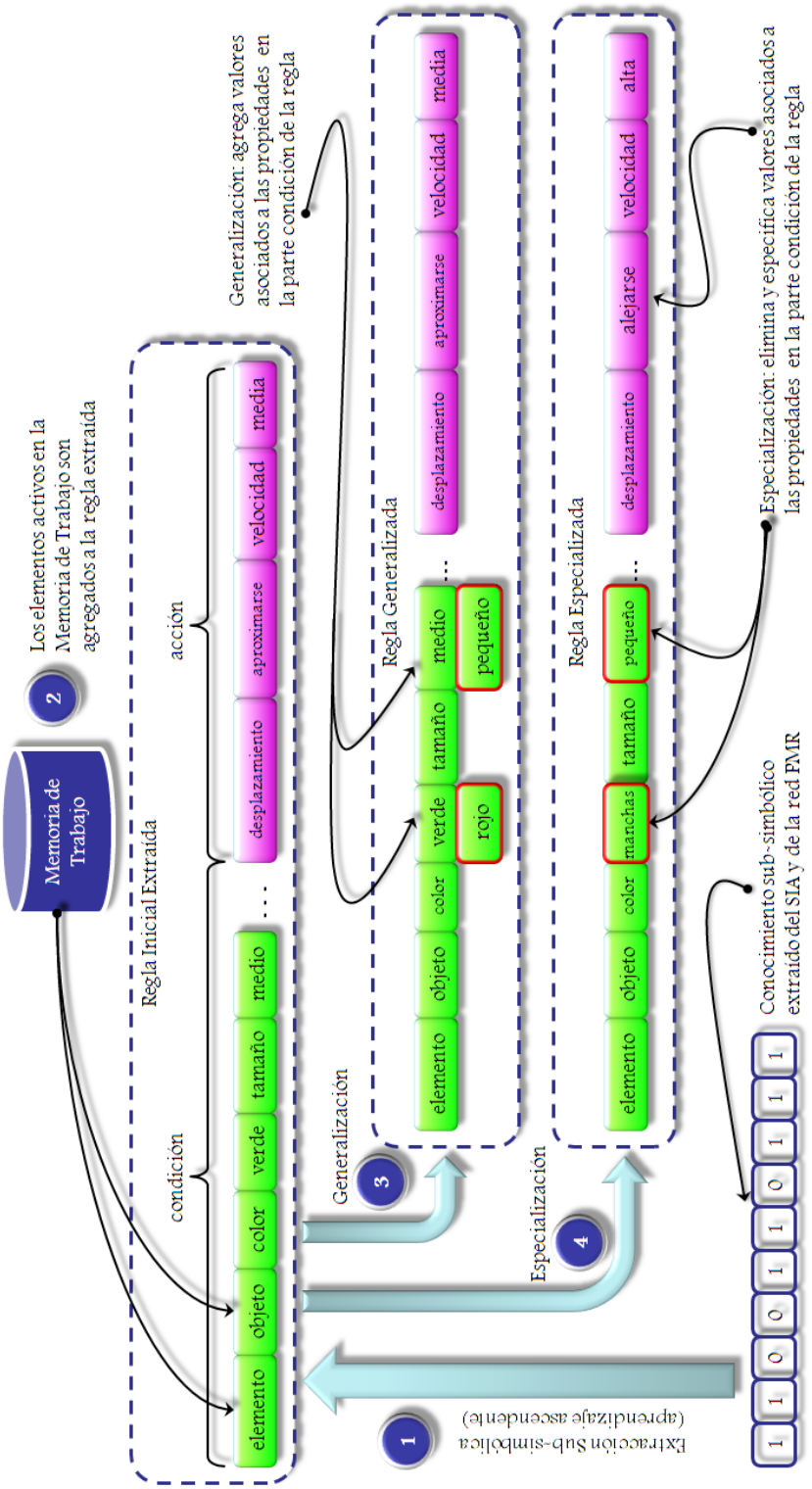


Figura 7.16: Ejemplo de Extracción, Especialización y Generalización de Reglas Sub-simbólicas.

conocimiento que el agente (animal virtual) emplea cuando percibe un objeto verde de tamaño medio, y como consecuencia, ejecuta la acción aproximarse hacia el objeto con una velocidad media.

Una vez se ha extraído la regla, se incluyen en ésta las pautas adicionales del estado actual que se encuentran en la MT y se almacena en el repositorio de reglas simbólicas. La regla extraída quedaría entonces de la forma:

SI
⟨elemento, objeto⟩ Y
⟨color, verde⟩ Y
⟨tamaño, medio⟩ Y
⟨consistencia, sólido⟩ (MT)
ENTONCES
⟨desplazamiento, aproximarse⟩ Y
⟨velocidad, media⟩

Luego de un cierto número de interacciones con el entorno, el agente comprueba que la regla extraída ha sido satisfactoria en ciertas situaciones: cuando el objeto es verde o naranja, y con un tamaño medio o pequeño, la mejor opción es aproximarse a este objeto con una velocidad media, y comprobar si éste es un objeto comestible (probablemente alimento) o algún elemento con el que pueda jugar (una pelota por ejemplo). Entonces, el agente generaliza una regla para estas distintas situaciones, la cual aplicará en sucesivas interacciones con el entorno. La nueva regla puede formalizarse de la siguiente manera:

SI
⟨elemento, objeto⟩ Y
(⟨color, verde⟩ O ⟨color, rojo⟩) Y
(⟨tamaño, medio⟩ O ⟨tamaño, pequeño⟩) Y
⟨consistencia, sólido⟩
ENTONCES
⟨desplazamiento, aproximarse⟩ Y
⟨velocidad, media⟩

Ahora, supóngase que al aplicar la regla extraída, ésta no resulta satisfactoria. En este caso, es activado el proceso de especialización de la regla: se construye una regla con cada una de las *propiedades* y valores respectivos observados en el estado actual. Para el ejemplo, asúmase que el agente se ha encontrado con un objeto de tamaño medio y de color verde con manchas que no ha resultado ser ni alimento ni un elemento de juego, sino todo lo contrario, una planta venenosa. En este caso, la acción de aproximarse para interactuar con el objeto no ha tenido una consecuencia satisfactoria, motivo por el cual la regla debe ser especializada (mediante la eliminación del valor ⟨rojo⟩ para la propiedad ⟨color⟩, y del valor ⟨medio⟩ para la propiedad ⟨tamaño⟩). Es posible además que nuevos valores sean agregados a las *propiedades* para especializar aún más la

regla. Por ejemplo, si el objeto, además de ser verde tenía manchas de color amarillo, esta característica debe ser incluida en la regla especializada. Así mismo, una nueva acción evasiva puede incluirse, tal como alejarse del objeto a una velocidad alta. De esta manera, la nueva regla especializada sería de la siguiente forma:

SI
⟨elemento, objeto⟩ Y
⟨color, verde-manchado⟩ Y
⟨tamaño, pequeño⟩ Y
⟨consistencia, sólido⟩
ENTONCES
⟨desplazamiento, alejarse⟩ Y
⟨velocidad, alta⟩

En síntesis, es posible argumentar que aunque en este tipo de aprendizaje la acumulación de las distintas estadísticas es gradual, la adquisición y refinamiento de las reglas es un proceso de "un sólo intento" que emplea una estrategia de "todo o nada". Por consiguiente, este tipo de aprendizaje es muy diferente de la sincronización gradual de los pesos en la red PMR y de la adaptación progresiva de la aptitud en los anticuerpos del SIA.

Como comentario final, cabe resaltar que el conocimiento explícito adquirido de esta forma, concierne claramente con aspectos contextuales, ecológicos, y existenciales relativos al mundo observado. En otras palabras, los conceptos y reglas aprendidas conciernen a esos y sólo esos aspectos del mundo que tienen alguna relevancia significativa para el agente durante su interacción con dicho mundo. Estas reglas no pueden ser consideradas sólo como clasificaciones absolutas de los objetos del mundo (Lave, 1988), sino que son el resultado de la interacción del agente con su entorno, manifestando siempre las regularidades encontradas en tal interacción. Por consiguiente, estas reglas no representan objetivos finales, sino un conocimiento orientado hacia el desarrollo de tareas y relacionado con la interacción entre el agente y su mundo (Heidegger, 1996).

7.5.3. Reglas Predeterminadas (RP)

Parte de las reglas simbólicas que emplea el *Comportamiento* pueden ser definidas a priori dentro de una estructura inmutable, idéntica a la explicada anteriormente para las reglas RES. La diferencia básica entre las reglas predeterminadas (de ahora en adelante RP) y las reglas RES, es que las primeras no definen procesos de aprendizaje, extracción, generalización, o especialización, sino que simplemente reflejan (de forma parcial) un conocimiento refinado respecto a las funciones que debe estar en capacidad de exhibir el *Comportamiento*. Las reglas RP representan tanto el conocimiento innato heredado genéticamente (en este caso, a través del proceso filogenético) como el conocimiento experto pre-programado por un diseñador. Alternativamente, las RPs podrían ser adquiridas por fuentes externas (e.g., otros agentes, instrucciones de un humano, etc.).

Una característica importante de las RPs es que sirven como unidades de conocimiento orientativo durante el proceso de *aprendizaje descendente* (top-down): los componentes del nivel emergente refinan su conocimiento no sólo mediante la modulación de una señal de retroalimentación, sino a través de la supervisión que provee el conocimiento innato (reglas RPs) sobre el proceso de aprendizaje implícito (sub-simbólico). En los etapas iniciales de interacción del agente con un entorno nuevo, éste basa su proceso de toma de decisiones en el conocimiento innato o experto que mantiene encapsulado en las reglas del tipo RP. Sin embargo, en la medida que el tiempo de interacción avanza, el agente empieza gradualmente a tomar decisiones basadas en los componentes del nivel emergente (i.e., SIA y PMR), los cuales le permiten descubrir y adquirir nuevo conocimiento y, si es pertinente, reformular las reglas predeterminadas que se han establecido en el nivel cognitivista. A partir de este punto, se produce el *aprendizaje ascendente* (bottom-up), donde son extraídas nuevas reglas simbólicas, como se describió anteriormente en las reglas RES. Los procesos de aprendizaje vertical serán descritos al detalle en la sección 7.8.

Una ventaja evidente de las RPs es que éstas pueden representar algo más que una simple estructura proposicional. Las RPs pueden representar estructuras más complejas, las cuales establecen correlaciones refinadas entre las pautas de entrada (condiciones) y las pautas de salida (acciones) y, como consecuencia, es posible generar también secuencias de acciones más elaboradas. La idea es esencialmente compatible con la noción de "esquemas" (Arbib, 1980; Shapiro y Schmidt, 1982; Arbib, 1998; Roy, 2005), y con el concepto de "comportamientos abstractos" (Mataric, 2001).

Un aspecto que con frecuencia es poco detallado en las diferentes arquitecturas cognitivas (e.g., SOAR (Newell et al., 1993; Lehman et al., 2006), ACT-R (Anderson, 1996), CLARION (Sun et al., 2001), entre otras), es la manera en que el sistema adquiere nuevas reglas de producción predeterminadas. Evidentemente, una argumentación válida es que éstas son definidas a priori por el diseñador de la concreción contextual de la arquitectura como unidades de conocimiento experto. Sin embargo, la especificación detallada de un mecanismo de aprendizaje que permita al agente aprender nuevas reglas innatas de forma autónoma, es un tema que no se ha tratado con la suficiente relevancia en la literatura existente (Sun et al., 2001). No obstante, en este trabajo se propone la hipótesis de que estas reglas pueden ser transmitidas y "heredadas" entre los agentes del sistema, a través de estructuras cognitivas más complejas llamadas "memes", las cuales reflejan el conocimiento experto refinado que ha adquirido cada agente durante su interacción con el entorno, y que puede ser asimilado por cada agente como un conocimiento innato fruto del proceso filogenético. Este aspecto será discutido al detalle en la sección 10.4.

7.5.4. Soporte, Utilidad y Activación de Nivel Base de las Reglas

En las secciones anteriores se definieron algunas mediciones numéricas y estadísticas asociadas a cada regla. En esta sección se describe el uso de estas mediciones en el razonamiento basado en reglas (tomando como referencia a (Sun et al., 2001)).

Definición de las Mediciones Numéricas empleadas: una de las mediciones más importantes, el *Soporte de Regla*, mide el grado de validez que soporta la conclusión de

una regla. Cuando no es considerado el emparejamiento parcial, esta medición puede ser simplemente 0 (no soportada) o 1 (soportada). Por otro lado, cuando se considera el emparejamiento parcial, se emplea una función de sumas ponderadas para calcular el soporte de cada regla. Como se ha mencionado anteriormente, la condición de una regla es representada por un chunk, el cual se encuentra asociado con pautas del tipo ⟨`propiedad, valor`⟩ en el nivel emergente. En términos generales, cuando las *propiedades* y valores de las pautas de entrada son activadas en el nivel emergente, esta activación se propaga hacia el nivel cognitivista activando todos los chunks relevantes. De esta manera, la intensidad de un chunk condición de una regla se establece como:

$$I_{ck}^{c} = \sum_{i} A_i \times W_i^{Ck} \tag{7.24}$$

donde: I_{ck}^{c} es el nivel de intensidad del chunk k. A_i es la activación de la i-ésima *propiedad* del chunk $_{Ck}$ (la cual concierne a la *propiedad* i, que especifica la magnitud con la cual es activado el *valor* con mayor nivel de activación en dicha *propiedad*). W_i^{Ck} es el peso de la *propiedad* i del chunk $_{Ck}$ (donde por defecto $W_i^{Ck} = 1/n$, con n igual al número de *propiedades* especificadas para el chunk $_{Ck}$).

Tomando como base la intensidad del chunk de condición, el soporte de la regla correspondiente se calcula como sigue:

$$S_k^r = I_{ck}^{c} \times W^r \tag{7.25}$$

donde: k indica la k-ésima regla en el nivel cognitivista, S_k^r es el soporte de la regla k. W^r es el peso de la regla (que por defecto es $W^r = 1$).

En el caso de existir múltiples reglas que satisfagan la misma conclusión, la intensidad del chunk de acción es combinada de la siguiente manera:

$$I_{cj}^{c} = max_{k \to j}(I_k^r) \tag{7.26}$$

donde: I_{cj}^{c} es la intensidad combinada. j indica la j-ésima acción en el nivel cognitivista. $k \to j$ indica todas las reglas k que conducen a la acción j.

La fórmula para el soporte de regla permite un "emparejamiento parcial" como opción: cuando uno o más valores de una *propiedad* no son activados totalmente (i.e., no alcanzan un *valor* de 1 en la activación total) o no se encuentran presentes en la *propiedad*, no obstante la regla podrá ser aplicada aunque genere una conclusión con un nivel de intensidad relativamente bajo (< 1). Como consecuencia, es posible determinar un umbral $umbral_{mi}$ que establece el nivel mínimo aceptable que debe alcanzar la intensidad de la regla para hacerse activa. Sin embargo, generalmente se requiere que la meta asociada a la condición de una regla (si existe) sea emparejada totalmente y no parcialmente (como lo proponen (Anderson y Lebiere, 1998))[20].

[20]Nótese que la descripción de emparejamiento parcial es de hecho una aproximación del mecanismo de razonamiento basado en similaridad (Sun, 1995).

Por otro lado, la *Utilidad de la Regla* mide la efectividad y la conveniencia de una regla en particular, en términos de un coste/beneficio. Esta medida de utilidad puede ser estimada mediante funciones que calculan los valores de utilidad sobre la marcha. La fórmula por defecto que se empleará es la siguiente:

$$beneficio_j = \frac{c_3 + CP_j}{c_4 + CP_j + CN_j} \tag{7.27}$$

donde: c_3 y c_4 son constantes con un valor por defecto de 1 y 2 respectivamente.

$$coste_j = \frac{\textit{tiempo-ejecución-regla-j}}{\textit{tiempo-promedio-ejecución-reglas}} \tag{7.28}$$

donde: *tiempo-ejecución-regla-j* y *tiempo-promedio-ejecución-reglas* necesitan ser estimados para cada concreción contextual de la arquitectura (como constantes o como funciones que computan dichos valores). Estos valores serán definidos en el apartado 10.5 sobre los tiempos de respuesta del sistema.

En general se tiene que la utilidad corresponde a:

$$U_j^r = beneficio_j - \upsilon \times coste_j \tag{7.29}$$

La interpretación de esta medida de utilidad es la siguiente: el beneficio de una regla es calculado en base a la tasa de emparejamiento positivo (i.e., cuántos emparejamientos positivos puede producir una regla dentro del contexto de todos los posibles emparejamientos para la regla). El coste es calculado en base a las consideraciones de tiempo de ejecución: puede ser calculado en base a los tipos de operación involucrados en sus costes predeterminados. Sin embargo, no se requiere que este coste sea una medición precisa de los ciclos de ejecución, sino que puede ser un valor constante, el cual varía de regla a regla (tomando en cuenta los tipos de operación involucrados), correspondiéndose de esta manera con el tiempo de ejecución aproximado de una regla. La utilidad de una regla es la comparación de su beneficio contra su coste (e.g., el beneficio menos el coste afectado por un factor de escalamiento υ), como es generalmente empleado (Newell et al., 1993; Anderson, 1996; Sun et al., 2001).

Adicionalmente, se requiere una medición que lleve un seguimiento de las veces que han sido empleadas las unidades de conocimiento (reglas) a lo largo del tiempo, la cual es necesaria para identificar los estímulos novedosos y/o recurrentes (i.e., los estímulos desencadenantes o disparadores[21]). Esta medición, propuesta por Anderson (Anderson, 1993b), es llamada *Activación de Nivel Base* (ANB). La ANB de una regla corresponde

[21]En psicología, un estímulo desencadenante (priming effect) es un efecto de la memoria implícita en el cual la exposición a un estímulo determinado influye sobre la respuesta a un estimulo subsecuente. Esto puede ocurrir después de la repetición de estímulos perceptuales, semánticos, o conceptuales. Por ejemplo, si una persona lee una lista de palabras la cual incluye la palabra "mesa", y luego se le pide a la persona que complete una palabra que empieza con "me", la probabilidad de que responda "mesa" es más alta que si no hubiese recibido estímulo alguno previamente para la misma palabra.

a un valor que precisa cuán novedosa es la misma (i.e., cuanto más alto sea el valor de ANB, menos novedosa y más recurrente ha sido la regla):

$$ANB_j^r = iANB_j^r + c \times \sum_{l=1}^{n} t_l^{-d} \tag{7.30}$$

donde: t_l es la l-ésima vez que se emplea la regla r. c es una constante de amplitud. d es una tasa de degradación. Los valores por defecto de los parámetros son: $c = 2$, $d = 0.5$, y $iANB_j^r$ es el valor inicial de ANB_j^r.

La ANB determina la posibilidad de necesitar una regla particular tomando como referencia la historia de su uso en experiencias anteriores (Anderson, 1993b). La ANB decae gradualmente siguiendo un comportamiento exponencial. Así pues, esta cantidad representa un *estímulo desencadenante* sobre una regla como resultado de su uso en ciclos cognitivos anteriores. El estímulo desencadenante se establece como un valor proporcional a la probabilidad de necesitar dicho estímulo (Anderson, 1993b)[22].

De manera similar, la ANB del chunk acción resultante de una regla también puede calcularse como un valor que indica cuán novedoso es el chunk:

$$ANB_j^a = iANB_j^a + c \times \sum_{l=1}^{n} t_l^{-d} \tag{7.31}$$

donde: t_l es la l-ésima vez que se activa el chunk de acción a. Los valores por defecto de los parámetros son: $c = 2$, $d = 0.5$, y $iANB_j^a$ es el valor inicial de ANB_j^a.

Esta cantidad representa la probabilidad de necesitar un chunk determinado tomando como referencia la historia de su uso en ciclos anteriores. Esta activación representa un *estímulo desencadenante* sobre el chunk resultante en función de las aplicaciones (satisfactorias) anteriores de dicho chunk (Anderson, 1993b). Como se verá en la siguiente sección, el proceso de selección de una regla simbólica está en función de su activación de nivel base, lo cual se fundamenta en los principios de activación nerviosa propuestos en la sección 6.3.3.

Intuitivamente, la ANB de una regla o de un chunk aumenta progresivamente con el transcurso del tiempo, pero debido a la tasa de degradación, ésta no crece exponencialmente sino que lo hace logarítmicamente, como se observa en la figura 7.17.

7.5.5. Mecanismo de Razonamiento

A fin de recomendar una acción apropiada, el módulo de Producciones debe evaluar los siguientes factores por cada regla (RES y RP):

1. La frecuencia de uso de la regla, lo cual determina, de alguna manera, qué tan útil puede ser ésta en el estado actual,

[22] Nótese que resulta necesario especificar una ANB inicial que represente la experiencia previa antes de que sea inicializada la interacción del agente con su entorno. Este valor inicial es con frecuencia cero, a menos de que la regla defina una ANB diferente antes de iniciarse la ejecución.

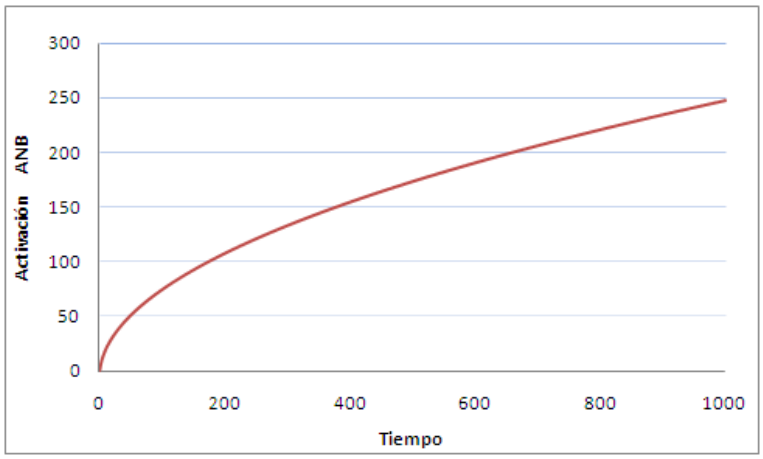


Figura 7.17: Ejemplo del comportamiento de la ANB para un chunk o una regla.

2. La intensidad con que es activada la condición de la regla, lo cual mide el nivel de soporte que tiene la acción recomendada (incluyendo una estadística de cuántos elementos son emparejados en la condición y con qué intensidad han sido activados),
3. La utilidad de la regla, la cual mide su beneficio general.

El primer factor es representado por la ANB de la regla; el segundo, por la suma ponderada que estima el soporte de la regla; y la tercera, por la medición de la utilidad. Por consiguiente, el impacto de la recomendación de una regla es determinado por: 1) las activaciones de nivel base de la regla aplicada y su chunk de acción resultante, 2) el soporte provisto por la regla aplicada, y 3) la utilidad de la regla.

Las mediciones de cada regla pueden ser empleadas en la determinación de la salida resultante del nivel cognitivista. También pueden ser empleadas en el cálculo del tiempo de respuesta de aplicar una regla. Existen muchas opciones de emplear esta tres medidas, como se argumenta en (Anderson, 1993b) y en (Sun et al., 2001). A continuación se describe la manera en que serán empleadas estas mediciones en el proceso de razonamiento llevado a cabo por el módulo de Producciones de METÁFORA.

La selección de una acción recomendada en el nivel cognitivista es determinada por la competición de todas las reglas que emparejan el estado actual observado, tomando como base el valor de la utilidad de cada regla. Como una aproximación al modelo de Luce (Luce, 2000), se propone el uso de una distribución de Boltzmann para seleccionar una recomendación de acción del conjunto de reglas cuyas condiciones son aplicables al estado actual. Esta distribución de Boltzmann es construida empleando la utilidad U_j^r de dichas reglas. Si la utilidad de la regla no es utilizada (lo cual puede configurarse en el módulo Meta-cognitivo), la distribución de Boltzmann es construida empleando el soporte de regla. De esta manera, la probabilidad de seleccionar una regla determinada i sería en un ciclo cognitivo específico:

$$p(i|x) = \frac{e^{U_i^r/\tau}}{\sum_j e^{U_i^r/\tau}} \tag{7.32}$$

donde: x es el estado actual. i y j indican las reglas que aplican para el estado actual x (siendo j un rango de reglas). τ controla el grado de aleatoriedad (temperatura) del proceso de toma de decisiones.

La interpretación del proceso de selección de Boltzmann es el siguiente: la selección es un proceso que implementa una función del tipo "softmax", la cual establece que, mientras más fuerte sea la utilidad de una regla, más rápido alcanzará un nivel asintótico. Consecuentemente, cuanto más fuerte sea la utilidad de la regla, existirá mayor probabilidad de que alcance primero el umbral de utilidad (el cual es definido para cada concreción contextual de la arquitectura). Se asume que la primera regla que alcance el umbral será seleccionada, es decir, gana la competición entre todas las reglas aplicables. Por tanto, cuanto más fuerte sea la utilidad de una regla, ésta tendrá mayor probabilidad de ser seleccionada.

En cuanto al tiempo que toma una regla en procesar una decision, éste puede ser determinado por la ANB de la regla así como por el ANB del chunk resultante (recomendación de acción). El uso de estas dos clases de ANB captura dos clases de estímulos desencadenantes: la activación de reglas y la activación de recomendaciones de acciones (Anderson, 1993b). El uso del ANB en la estimación de los tiempos de respuesta del módulo de Producciones será discutido en la sección 10.5.

Una vez se han activado las reglas de acuerdo con sus mediciones de utilidad, soporte, y ANB, se ejecuta un mecanismo de razonamiento donde las reglas activadas generan conclusiones que estimulan la activación de otras reglas. Concretamente, el mecanismo de razonamiento emplea dos reglas de inferencia: el *Modus Ponens* y el *Modus Tollens*. En el *Modus Ponens* se examina si la premisa o condición de la regla de producción es cierta y, si lo es, la conclusión pasa a formar parte de una lista de hechos. Esta regla de inferencia es de la forma: $A \rightarrow B, A \vdash B$. Por otro lado, en el *Modus Tollens* se examina si la conclusión de la regla de producción es falsa y, si lo es, se concluye que la premisa también es falsa. Esta regla de inferencia es de la forma: $A \rightarrow B, \neg B \vdash \neg A$. Luego de aplicar estas reglas de inferencia, se verifican las reglas simbólicas disparadas, se activa nuevamente el mecanismo de selección de Boltzmann, y se agregan nuevamente los hechos concluidos. El proceso de inferencia continua hasta que no sean activadas nuevas reglas. El mecanismo de razonamiento deductivo por el cual se encadenan las reglas simbólicas (RES y RP) a través de un proceso inferencial se describe en el algoritmo 7.10.

7.5.6. Modelo de Interacción

En la figura 7.18 se presenta un diagrama de secuencia para el mecanismo de extracción, generalización y especialización de reglas del tipo RES. En este diagrama se pueden distinguir seis segmentos importantes: el primer segmento, donde se precisa, por un lado, el proceso de razonamiento a través del encadenamiento inferencial de reglas y, por otro, el cálculo de las mediciones relativas a cada regla (utilidad, soporte y ANB); en el segundo segmento, se actualiza el repositorio de reglas y por cada una de éstas, el conjunto de reglas hijas respectivas; en el tercer segmento, se efectúa la extracción de reglas simbólicas (a partir de conocimiento sub-simbólico) cuando no se encuentra

Método: encadenarReglas

```
1.  Variables
2.     reglas          ↑↓ conjunto de reglas simbólicas.
3.     estadoActual    ↑↓ entradas sensoriales e internas percibidas por el agente en el
                          instante actual.
4.  Iniciar Procedimiento
5.     hechos ← estadoActual //se establece el conjunto de pautas
       ⟨propiedad, valor⟩ que conforman el estado actual, como la lista de
       hechos iniciales sobre los cuales se realizará el proceso de inferencia.
6.     hacer:
7.        Para cada una de las reglas del repertorio hacer:
8.           reglas ← seleccionarReglas( hechos ) //se determina el
             conjunto de reglas que coinciden en su parte condición con un subco-
             njunto de la lista de hechos, empleando para ello el emparejamiento
             parcial (selección de Boltzmann).
9.           hechos ← aplicarModusPonens( hechos, reglas) //se apli-
             ca la regla de inferencia del Modus Ponens sobre el conjunto de reglas
             de producción activadas.
10.          hechos ← aplicarModusTollens( hechos, reglas) //se
             aplica la regla de inferencia del Modus Tollens sobre el conjunto de
             reglas de producción activadas.
11.       Fin Para
12.    Mientras que existan nuevos hechos generados en la
       iteración actual y existan reglas activadas (en su parte
       condición), volver al paso 5.
13. Fin Procedimiento
```

Algoritmo 7.10: Algoritmo de Encadenamiento de Reglas

una regla que empareje el estado actual percibido por el agente; en el cuarto y quinto segmento se efectúan respectivamente los procesos de generalización y especialización de las reglas extraídas; finalmente, en el sexto segmento, se eliminan las reglas que ya no son válidas (es decir, aquellas reglas cuya ANB está por debajo de un umbral de eliminación y/o aquellas que no poseen valores en alguna de sus *propiedades*), luego, se fusionan las reglas (i.e., si existen reglas más generales que incluyan a otras más específicas, las específicas se agregan como reglas hijas inactivas), y por último, se actualizan las estadísticas (de positividad y negatividad) para cada regla del repositorio.

Nótese que el flujo de interacción para las reglas del tipo RP resulta similar. En este caso, aplican el primer, segundo, tercer, y sexto segmento del flujo de interacción presentado, o en otras palabras, se excluyen los segmentos relativos a los procesos de extracción, generalización y especialización de reglas RP. En el caso del modelo de interacción para las RPs intervendrían objetos del tipo *GrupoReglaRP* y *ReglaRP* en lugar de *GrupoReglaRES* y *ReglaRES* respectivamente.

7.6. Conversión Representacional

Como se ha expuesto en los apartados anteriores, el nivel cognitivista encapsula un conjunto de reglas simbólicas con un formato representacional diferente al de las reglas sub-simbólicas producidas en el nivel emergente. Como consecuencia, resulta necesario

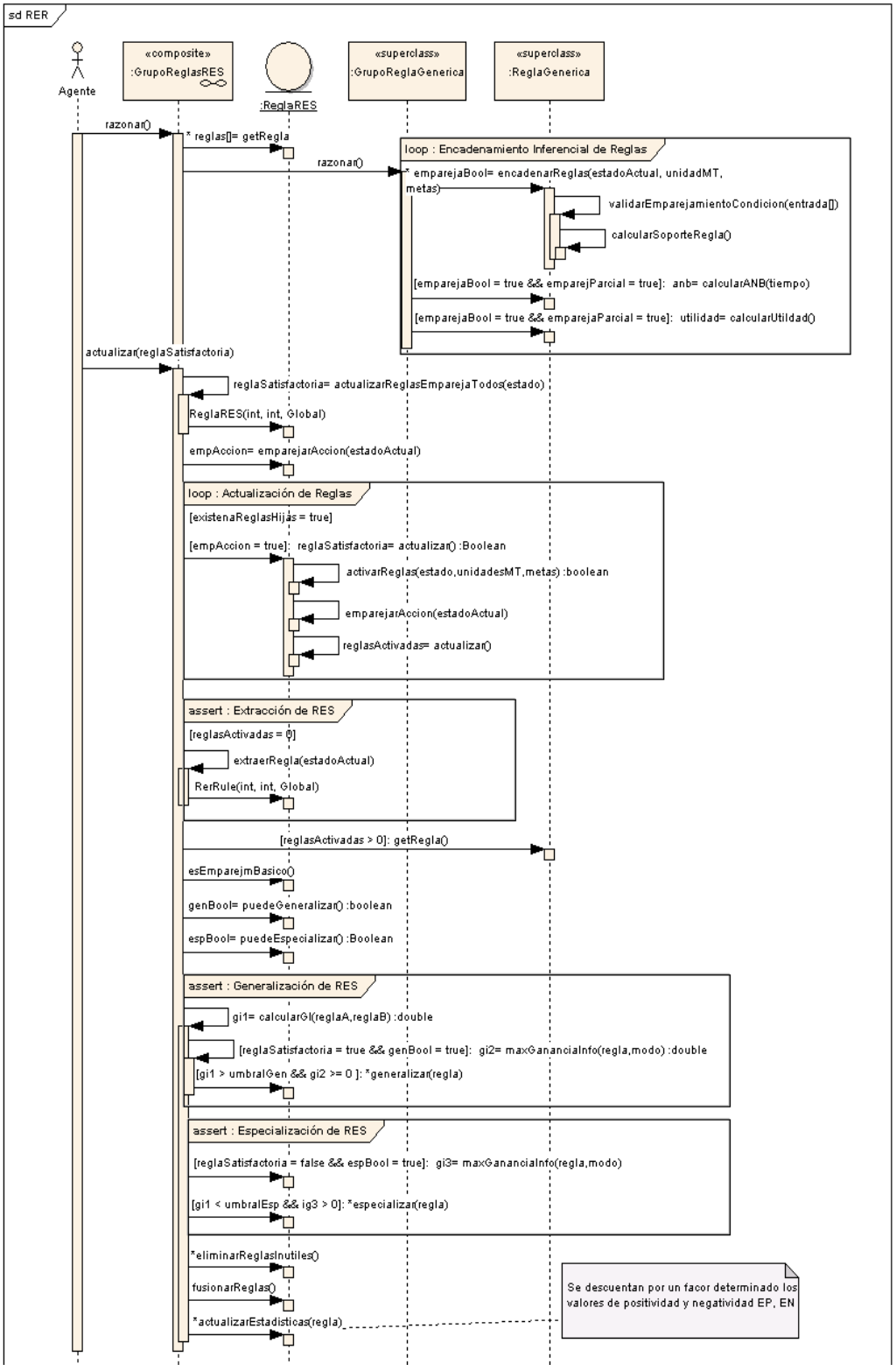


Figura 7.18: Modelo de Secuencia para las reglas RES.

establecer un mecanismo que permita realizar una conversión de dichas representaciones simbólicas a representaciones sub-simbólicas, y viceversa, a fin de disponer de un formato unificado, según convenga, que servirá de moneda común en los procesos de integración y comunicación vertical de unidades de conocimiento, llevados a cabo entre los niveles cognitivista y emergente del módulo Comportamental.

Básicamente, el proceso de conversión llevado a cabo por el módulo Efector se efectúa en tres puntos: primero, cuando se requiere convertir la información del estado actual (unidades de la MT: información sensorial interna y externa, perceptos, metas, etc.), la cual se encuentra en un formato de etiquetas semánticas (simbólicas), a un formato de representación sub-simbólica que pueda ser procesado por el nivel emergente; segundo, cuando se inicia el proceso de modulación de las acciones recomendadas por cada componente del *Comportamiento* y se requiere un formato unificado para realizar la integración de las mismas; y tercero, durante los procesos de aprendizaje vertical (ascendente y descendente).

La conversión de una regla simbólica a una regla sub-simbólica (un anticuerpo en el SIA o un patrón de activación entrada/salida en la red PMR), viene dada por una codificación binaria que se asigna a cada *valor* de la regla simbólica, y la cual depende del número total de *valores* que posea cada *propiedad*.

Esto es, dada una regla simbólica x, por cada *propiedad* de x (tanto en el chunk condición como en el chunk acción) se calcula el número total de *valores* que posea, lo que determinará el número de bits (n) que se requieren para construir la correspondiente representación sub-simbólica, y así sucesivamente con cada *propiedad* de x. Una vez determinada la longitud n para cada *propiedad*, se convierte a binario el valor posicional del *valor*. Por ejemplo, supóngase la *propiedad* $\langle$`color`$\rangle$ que posee los valores $\langle$`azul`$\rangle$, $\langle$`amarillo`$\rangle$, $\langle$`rojo`$\rangle$, $\langle$`verde`$\rangle$, $\langle$`violeta`$\rangle$, $\langle$`naranja`$\rangle$, $\langle$`blanco`$\rangle$ y $\langle$`negro`$\rangle$. En este caso, el número total de *valores* es ocho, luego de requieren tres bits para representar estos *valores* ($2^3 = 8$, con $n = 3$). La conversión a binario para cada *valor* de la *propiedad color* sería entonces:

Valor (simbólico)	Valor Posicional	Valor Binario
Azul	0	000
Amarillo	1	001
Rojo	2	010
Verde	3	011
Violeta	4	100
Naranja	5	101
Blanco	6	110
Negro	7	111

Tabla 7.3: Representacion Binaria de la *propiedad* simbólica “color”.

En la figura 7.19, se presenta un ejemplo de traducción de reglas con varias *propiedades* en el chunk condición y en el chunk acción. Considérese que existe una regla simbólica en el nivel cognitivista como la siguiente:

SI
⟨objeto-percibido, pelota⟩ Y
⟨color, rojo⟩ Y
⟨tamaño, medio⟩ Y
⟨ubicación, derecha⟩ ...
ENTONCES
⟨desplazamiento, avanzar-rápido⟩ Y
⟨rotación, derecha⟩ Y
⟨abrir, mano⟩...

Esta regla simbólica, se encuentra codificada como una secuencia de pautas ⟨propiedad, valor⟩ que forman un chunk condición y un chunk acción. La conversión sub-simbólica de cada una de estas parejas se realiza de forma independiente.

En el chunk condición del ejemplo, la *propiedad* 1 que corresponde al objeto percibido, tiene los posibles *valores*: pelota, caja, y fruta; la *propiedad* 2 que corresponde al color del objeto, tiene los posibles *valores*: azul, amarillo, rojo, verde, violeta, naranja, blanco, y negro; etc. Por tanto, para la *propiedad* 1 se requieren 2 bits, mientras que para la *propiedad* 2 se requieren 3 bits, y así sucesivamente para el resto de propiedades.

En el chunk acción, la *propiedad* 1 que corresponde al actuador que controla el desplazamiento del agente tiene los *valores*: avanzar-rápido, avanzar-normal, avanzar-lento, retroceder, y detenerse; y la *propiedad* 2 que corresponde al actuador de rotación tiene los *valores*: izquierda, derecha, y no-rotar. De esta manera, se requieren 3 bits para la *propiedad* 1 y 2 bits para la *propiedad* 2, y así sucesivamente para el resto de propiedades.

En la regla sub-simbólica, las propiedades convertidas en un formato binario son dispuestas en el mismo orden en el que se encuentran en la regla simbólica, formando así una cadena binaria de tamaño fijo. Como es de intuirse, la conversión puede realizarse en ambos sentidos: desde las representaciones simbólicas hacia las representaciones sub-simbólicas, y viceversa. En el caso de la conversión de lo sub-simbólico hacia lo simbólico, el proceso emplea el mismo mecanismo descrito anteriormente pero en orden inverso: primero, se separa la parte condición y la parte acción de la regla sub-simbólica; luego, se identifican las sub-cadenas de bits que forman cada *propiedad*[23]; finalmente, el valor binario de la subcadena se convierte a entero y éste indica el índice (valor posicional) del *valor* en la lista de *valores* de la *propiedad* en cuestión.

Debido a que cada componente del *Comportamiento* requiere diferentes tipos de representación del conocimiento y diferentes mecanismos de procesamiento interno, conviene hacer algunas observaciones acerca del diseño computacional que permite tratar a estos componentes como si su naturaleza fuese homogénea.

[23]El número de bits de cada *propiedad* depende del número total de *valores* de esta, así, si el número total de *valores* es 8, la subcadena tendrá un tamaño de 3 bits, que representan $2^3 = 8$ valores.

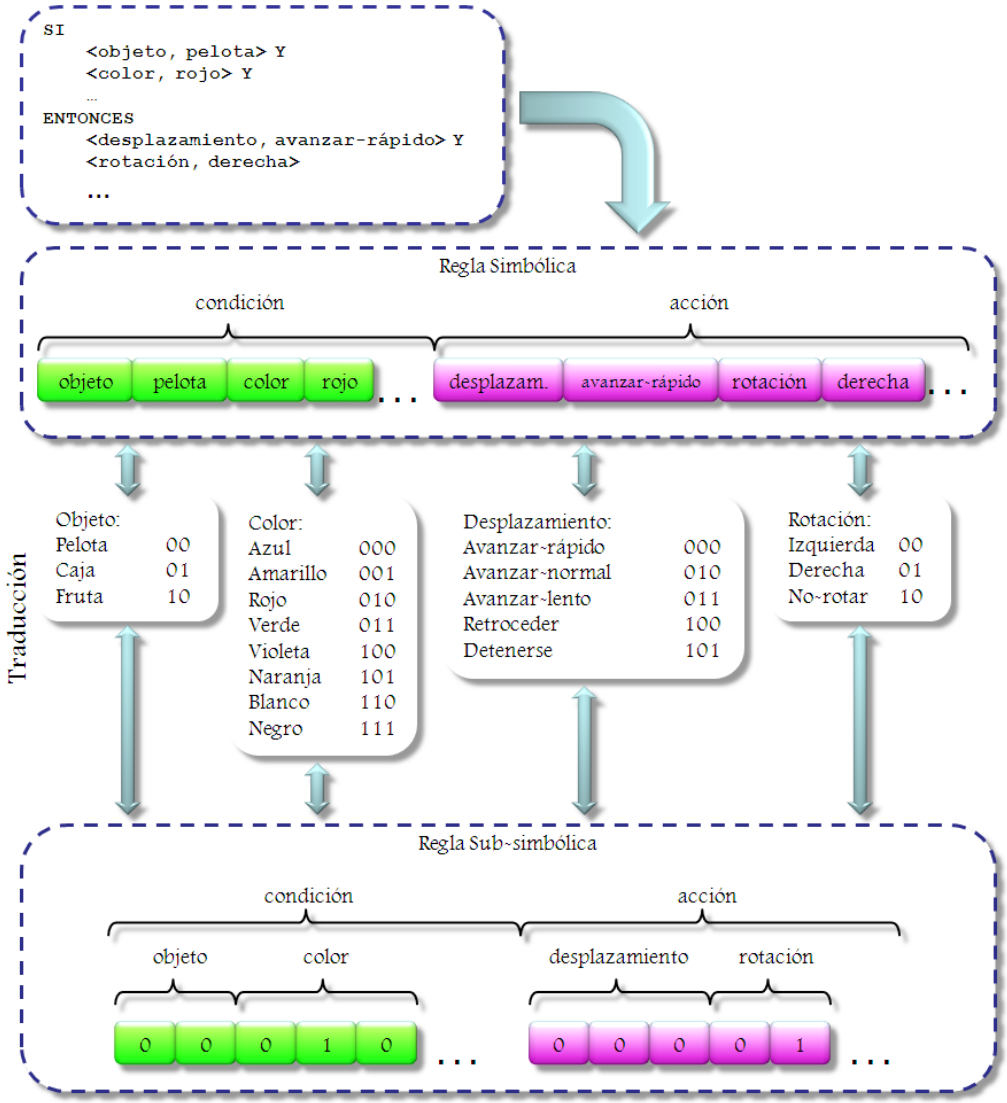


Figura 7.19: Conversión de Reglas Simbólicas ↔ Sub-simbólicas.

Para resolver la heterogeneidad de los componentes, se propone el empleo de 2 patrones de diseño: una patrón "Fachada" (Facade) que provee una interfaz única de acceso al conjunto de componentes del *Comportamiento*, y un patrón "Adaptador" (Adapter) que comunica las peticiones de la fachada con métodos concretos en cada uno de los diferentes componentes del *Comportamiento*, como se observa en la figura 7.20.

La fachada *Componente* provee gran flexibilidad y escalabilidad al permitir que nuevos componentes puedan ser "conectados" al *Comportamiento*. Adicionalmente, disminuye el acoplamiento entre la implementación del control del *Comportamiento* y la implementación específica de cada componente al reducirse el número de dependencias directas entre tales elementos del sistema.

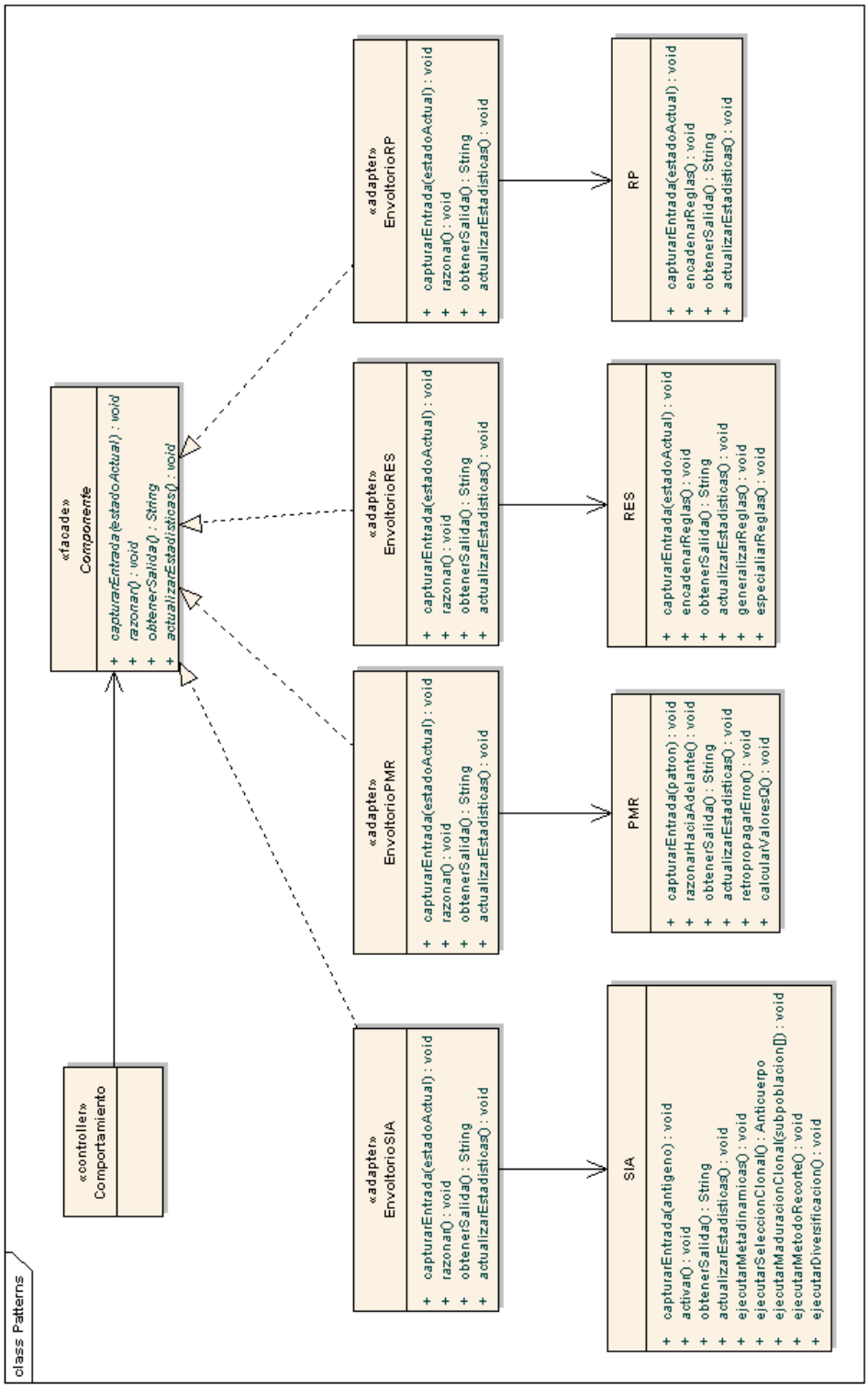


Figura 7.20: Fachada y Adaptadores para los Componentes del *Comportamiento*.

En cuanto a los adaptadores, estos permiten manipular con transparencia cada función de los diferentes componentes del *Comportamiento*, esto es, que desde la perspectiva de implementación todos los componentes exhiben 4 operaciones comunes: *capturarEntrada, razonar, obtenerSalida, y actualizarEstadisticas*, las cuales son invocadas por el *Comportamiento* sin importar qué hace internamente cada una de ellas (según el componente). Por ejemplo, el método razonar del adaptador del SIA supone la invocación de los métodos *activar*, *ejecutarMetadinamica*, *ejecutarSeleccionClonal*, *ejecutarMaduracionClonal*, *ejecutarMetodoRecorte*, y *ejecutarDiversificacion*, mientras que en el mismo método en el componente PMR activaría él método ejecutarRedHaciaAdelante, ...

Debido a que todos los componentes del *Comportamiento* producen algún tipo de regla (simbólica o sub-simbólica), y teniendo en cuenta que la red PMR no es propiamente un sistema basado en reglas, el adaptador correspondiente (*EnvoltorioPMR*) realiza una tarea adicional que consiste en construir reglas sub-simbólicas del tipo *condición* $\rightarrow$ *acción* a partir de los patrones de activación que se forman dada una entrada sensorial que entra en la red y su respectiva acción disparada. Esta regla sub-simbólica formada por el adaptador, refleja asociaciones entre configuraciones específicas de estímulos, ya sean temporales o espaciales, y respuestas específicas de la red. De esta manera, la regla sub-simbólica puede ser luego enviada al nivel cognitivista para su extracción y estructuración en una regla simbólica interpretable.

7.7. Mecanismos de Modulación e Integración de Acciones

Una vez que los componentes del *Comportamiento* (RNA, SIA, RES, y RP) han ejecutado sus dinámicas internas para obtener una recomendación de acción, éstas son enviadas al módulo Efector donde son moduladas y/o integradas en una sola acción final que ejecutará el agente en su entorno. La necesidad de modular y/o integrar estas acciones en una sola acción se debe a que existe la posibilidad de que estas acciones sean diferentes entre sí y por tanto se requiera un mecanismo que las unifique en una salida única y coherente. Una vez unificada la acción del *Comportamiento*, ésta es enviada al *Traductor* del módulo Efector donde es convertida en señales motoras que activan determinados actuadores del sistema. Es posible además que la acción unificada produzca cambios en el estado interno del agente o que sirva como entrada a un nuevo ciclo cognitivo (e.g., en el caso de la simulación interna).

Existen diversos modelos matemáticos que permiten combinar las salidas de múltiples sistemas de toma de decisiones. Por mencionar algunos de estos modelos matemáticos, los cuales se encuentran agrupados por categorías, se tienen: los métodos de Agregación (Hull et al., 1990), los métodos de Escalonamiento (Srihari et al., 1994), los métodos de esquema de votación mayoritaria (Kittler y Hatef, 1996), los métodos de Combinación Serial y Bayesiana (Rahman y Fairhurst, 1998), los métodos de Combinación en Paralelo (Rahman y Fairhurst, 1997a), los métodos de Combinación Híbridos (Rahman y Fairhurst, 1997b), entre otros.

En esta sección se presentan tres mecanismos de modulación y/o integración empleados por el módulo Efector de METÁFORA, uno de tipo estocástico, otro del tipo combinatorial, y un último de tipo de agregación por votación.

7.7.1. Selección Estocástica

En el proceso de modulación de acciones a través de la *Selección Estocástica*, existe al menos una acción de una regla (simbólica o sub-simbólica) que es aplicable en el estado actual observado. En este mecanismo, inicialmente se calcula una probabilidad de selección P_{sel} para cada componente del *Comportamiento* (RNA, SIA, RES y RP) la cual determina la probabilidad que tiene cada componente de ganar el derecho de controlar la actuación del agente en el ciclo cognitivo actual. Así pues, en cada ciclo cognitivo, el componente que controlará la actuación es seleccionado a través del método estocástico de la ruleta, en el cual la probabilidad total (es decir, 1) es dividida en cuatro regiones cuya área depende de la probabilidad de selección de cada componente, y luego es generado un número aleatorio que caerá dentro de alguna de estas cuatro regiones de área variable. De manera intuitiva, el componente con una región de mayor área, tendrá mayor probabilidad de ser seleccionado. Para cada uno de los componentes del *Comportamiento*, esta probabilidad se calcula de la siguiente manera:

$$P_{selPMR} = \frac{\beta_{PMR} \times ie_{PMR}}{\beta_{PMR} \times ie_{PMR} + \beta_{SIA} \times ie_{SIA} + \beta_{RES} \times ie_{RES} + \beta_{RP} \times ie_{RP}} \tag{7.33}$$

$$P_{selSIA} = \frac{\beta_{SIA} \times ie_{SIA}}{\beta_{SIA} \times ie_{SIA} + \beta_{PMR} \times ie_{PMR} + \beta_{RES} \times ie_{RES} + \beta_{RP} \times ie_{RP}} \tag{7.34}$$

$$P_{selRES} = \frac{\beta_{RES} \times ie_{RES}}{\beta_{RES} \times ie_{RES} + \beta_{PMR} \times ie_{PMR} + \beta_{SIA} \times ie_{SIA} + \beta_{RP} \times ie_{RP}} \tag{7.35}$$

$$P_{selRP} = \frac{\beta_{RP} \times ie_{RP}}{\beta_{RP} \times ie_{RP} + \beta_{PMR} \times ie_{PMR} + \beta_{SIA} \times ie_{SIA} + \beta_{RES} \times ie_{RES}} \tag{7.36}$$

donde: ie es un índice de éxito del componente. β es un factor que determina el peso (relevancia) del componente frente a los demás.

El índice de éxito (ie) de cada componente depende del número de aciertos y desaciertos que ha acumulado a lo largo de la interacción del agente con su ambiente, y que se ve reflejado en la sumatoria de los niveles de positividad y negatividad de cada componente (CP, CN)[24]. El ie es calculado de la siguiente forma:

$$ie_{componente} = \frac{c1 + \sum CP_{componente}}{c2 + \sum CP_{componente} + \sum CN_{componente}} \tag{7.37}$$

donde: $c1$ y $c2$ representan constantes (por defecto $c1 = 1$ y $c2 = 2$).

[24]Cabe notar que las estadísticas CP (Consecuencias Positivas o satisfactorias) y CN (Consecuencias Negativas o insatisfactorias) definidos para las reglas simbólicas, son usados en el SIA y en las redes PMR como una estadística que determina el número de veces que ha acertado o no el componente en cada ciclo cognitivo, basándose para ello en la modulación de una señal de retroalimentación.

Por otro lado, los parámetros β_{PMR}, β_{SIA}, β_{RES} y β_{RP} proveen un énfasis diferencial para cada tipo de conocimiento, reflejando una predisposición innata del componente (i.e., un valor entre 0 y 1) la cual debe ser pre-definida en el módulo Meta-cognitivo. Por ejemplo, si se desea que la naturaleza del *Comportamiento* esté orientada hacia el procesamiento sub-simbólico, el aprendizaje implícito, y la conducta reactiva, los valores para β_{PMR} y β_{SIA} serán sustancialmente más altos que los valores para β_{RES} y β_{RP}.

La elevada importancia de la distinción entre los diferentes tipos de reglas puede justificarse postulando que, dichos tipos de reglas representan diferentes "capas" de conocimiento: algunas son (casi) innatas (i.e., RP y PMR previamente entrenada), otras son basadas en plantillas de conocimiento (i.e., RES), y algunas otras son completamente empíricas –construidas a partir de las experiencias previas de interacción con el ambiente – (i.e., SIA y PMR con aprendizaje por refuerzo).

Una vez se ha decidido qué componente del *Comportamiento* posee el control de la actuación, se calcula probabilísticamente la acción recomendada que será ejecutada en el ambiente (a través de los actuadores del módulo Efector). Para esto, cada componente del *Comportamiento* define una probabilidad de selección para cada regla o acción, y luego, empleando el método de la ruleta, se escoge aleatoriamente cuál de éstas será ejecutada. Existe cierta evidencia psicológica acerca de tal activación intermitente de reglas en el cerebro humano; ver, por ejemplo, (VanLenh, 1995), (Anderson, 1993b), y (Sun et al., 2001). La figura 7.21 ilustra el mecanismo.

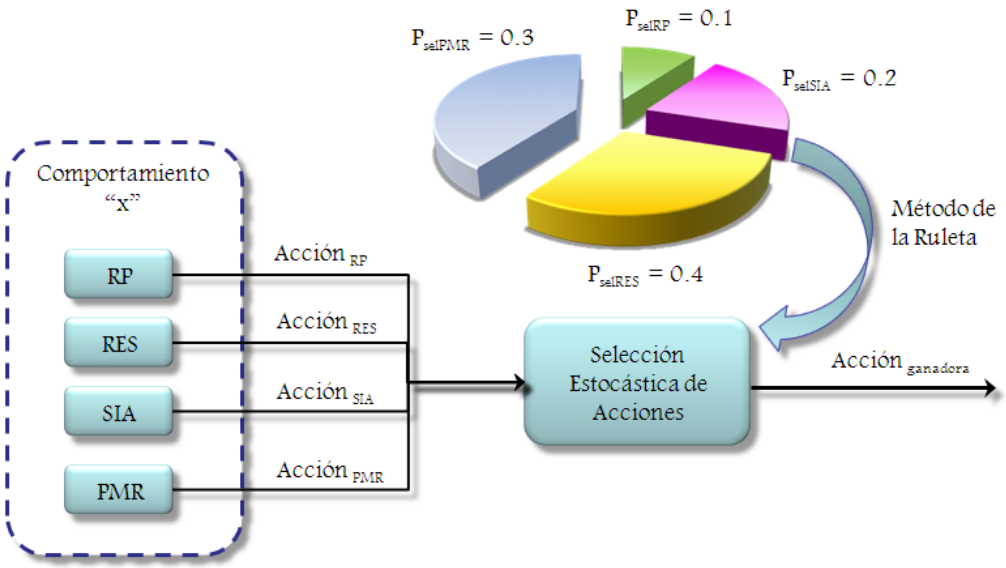


Figura 7.21: Selección Estocástica de Acciones.

En el nivel cognitivista (componentes RES y RP), la probabilidad de selección de cada regla se estima a través de un método estocástico basado en la *utilidad* de cada regla, así:

$$p(i|x) = \frac{e^{U_i^r/\tau}}{\sum_j e^{U_i^r/\tau}} \tag{7.38}$$

donde: x es el estado actual. i y j indican las reglas que emparejan el estado actual x (siendo j un rango de reglas). τ controla el grado de aleatoriedad (temperatura) del proceso de toma de decisiones.

Este método es conocido como el axioma de selección de Luce (Luce, 1959). La principal característica de este método es que la probabilidad de seleccionar una regla del sub-conjunto de reglas aplicables al estado actual observable no se encuentra afectada por la presencia ni ausencia de otras reglas en el sub-conjunto de reglas. La selección de este tipo se dice entonces que posee "independencia de las alternativas irrelevantes".

En cuanto a la selección de acciones en el nivel emergente se tiene que: (1) el componente SIA emplea una probabilidad de selección basada en la *Ganancia* y en la capacidad de *Puja* de cada anticuerpo sobre la cual es aplicado un proceso estocástico dirigido por el Sistema de Asignación de Recompensas (como se explicó en el apartado 7.4.2); y (2) las redes PMR emplean un proceso estocástico basado en una distribución de Boltzmann para los valores Q, la cual define la probabilidad de activar una determinada acción:

$$p(a|x) = \frac{e^{Q(x,a)/\tau}}{\sum_i e^{Q(x,a_i)/\tau}} \tag{7.39}$$

donde: x es el estado actual. a es una acción. τ controla el grado de aleatoriedad (temperatura) del proceso de toma de decisiones.

Si existen múltiples actuadores, el proceso de decisión basado en la distribución Boltzmann se realiza para cada actuador de forma individual, con el fin de seleccionar un sólo *valor* de cada *propiedad* que conforma la pauta de actuación.

7.7.2. Combinación Basada en Pesos

En este tipo de mecanismo, se combina la salida de los componentes del nivel cognitivista con la salida de los componentes del nivel emergente a través de un método de suma ponderada de pesos.

Los componentes del tipo RP y RES del *Comportamiento* realizan una recomendación de acción basada en la utilidad máxima del conjunto de reglas, esto es, la *propiedad* p de la acción a tiene un *valor* $v_{p,a}$ que representa la utilidad más alta del conjunto de reglas que ha emparejado el estado actual y ha recomendado la acción a.

$$v_{(p,a)} = \max_{j \to a} U_j^r \tag{7.40}$$

donde: $j \to a$ es el conjunto de reglas que emparejan el estado actual y que recomiendan la acción a.

En el nivel emergente, el *valor* para la *propiedad* p de la acción a es determinado de la siguiente manera: para la red PMR corresponde al valor "Q" de la capa de salida de la red (representado como $q_{p,a}$); y para el SIA corresponde a $g_{p,a}$, que es el *valor* de activación de la *propiedad* p de la acción a que recomienda el anticuerpo que ganó la

subasta en el sistema de asignación de recompensas. De esta manera, el *valor* combinado para la *propiedad* p de la acción a corresponde a:

$$v_{(p,a)}^{comb} = (w_{RES} \times v_{(p,a)}^{RES}) + (w_{RP} \times v_{(p,a)}^{RP}) + (w_{SIA} \times g_{(p,a)}) + (w_{PMR} \times q_{(p,a)}) \quad (7.41)$$

donde: w_{RES}, w_{RP}, w_{PMR}, y w_{SIA}, corresponden a los pesos de cada componente.

Luego de calculado el *valor* combinado, se ejecuta por cada una de las *propiedades* de la acción, un proceso de selección que determinará el *valor* final de activación de cada *propiedad*. Este proceso de selección puede ser un método estocástico basado en una distribución de Boltzmann o un método determinista basado en una función que maximiza el *valor* combinado para cada *propiedad*. A continuación se presenta un ejemplo que clarifica el procedimiento de la combinación basada en pesos.

Supóngase una acción conformada por 2 *propiedades*, cada una con tres *valores* posibles. La acción recomendada por cada componente del *Comportamiento* es traducida a una representación binaria puesto que facilita la manipulación operacional de los *valores* de cada *propiedad*. En la figura 7.22 se presenta un esquema del procedimiento de combinación. En el paso 1, son recolectadas las recomendaciones de acción de cada componente. Nótese que por cada *valor* de cada *propiedad*, se asigna un 1 o un 0 que indica si el valor está activo o no. Por ejemplo, si la propiedad 1 indicase el sentido de rotación de un robot, y sus tres posibles valores fuesen (en ese mismo orden) ⟨`rotar-der`⟩, ⟨`rotar-izq`⟩, y ⟨`no-rotar`⟩, entonces esa misma propiedad para la acción del componente RP tendría activo el valor ⟨`rotar-izq`⟩, mientras que para la acción del componente RES tendría activo el valor ⟨`rotar-der`⟩, y así sucesivamente.

En el paso 2, es multiplicado el valor de activación (determinado en el paso 1) por el peso de cada componente del *Comportamiento* (w_{PMR}, w_{SIA}, w_{RES}, y w_{RP}). En el ejemplo se han dado valores arbitrarios para estos pesos, sin embargo más adelante se detallará cómo estos pesos pueden ser establecidos para cada componente. En el caso de que múltiples *Comportamientos* puedan ser activados concurrentemente, el peso de cada componente se divide entre el número de *Comportamientos* activados y se multiplica por el valor de activación respectivo. Por ejemplo, en el caso de que se hayan activado 3 *Comportamientos*, el peso para el componente RP no sería $w_{RP} = 0.2$, sino que se actualizaría a $w_{RP} = 0.0666$ (i.e., 0.2 / 3), y este sería el nuevo peso que se emplearía para multiplicar por el valor de activación de la acción que es recomendada por el componente RP.

En el paso 3, se calcula el *valor* combinado para cada *valor* de cada *propiedad*, esto es, se realiza la sumatoria por columnas de los valores ponderados calculados en el paso 2. Se obtiene el *valor* combinado $v_{(p,a)}^{comb}$ de la acción a.

En el paso 4, se determina la acción combinada. En este punto, el mecanismo de combinación de acciones puede utilizar dos métodos como se mencionó anteriormente: un proceso estocástico basado en una distribución Boltzmann o un proceso determinista basado en una función de maximización. En la distribución Boltzmann se emplea cada *valor* combinado como una probabilidad de ser elegido posicionalmente (e.g., el primer *valor* de la *propiedad* 1 es 0.6, lo que indica que tiene una probabilidad de 0.6 de ser elegido para activar ese *valor* en la acción combinada). Nótese que los *valores* de cada

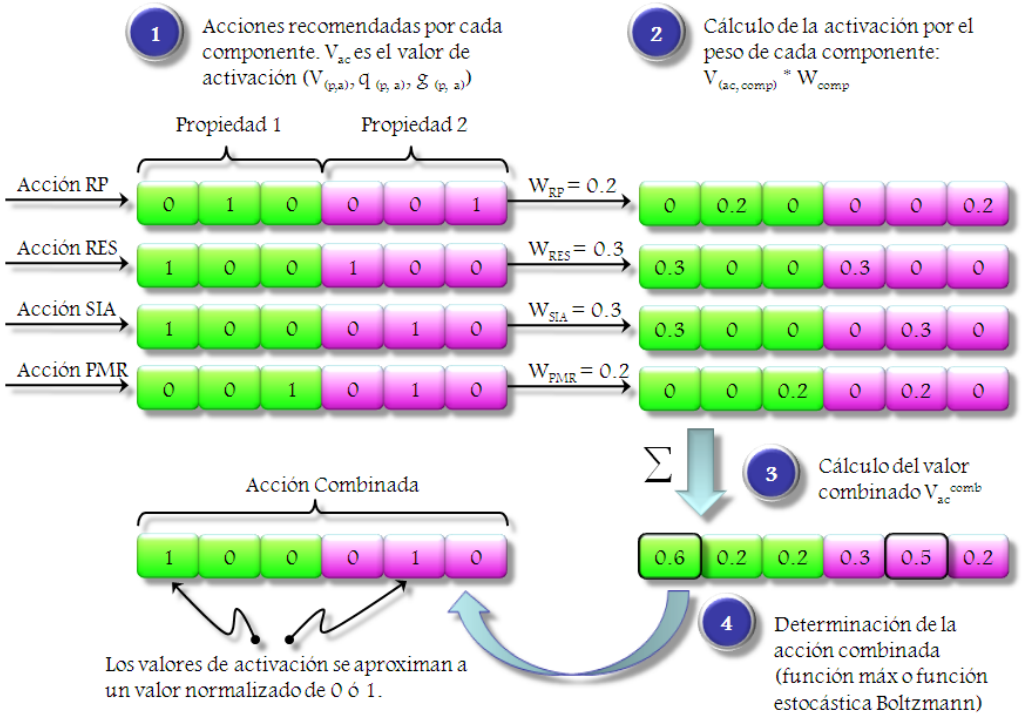


Figura 7.22: Combinación basada en Pesos.

propiedad compiten entre sí a través de su valor ponderado por ser elegidos para representar la acción combinada. En el método determinista, que es el empleado en el ejemplo de la figura 7.22, se maximiza el *valor* ponderado para cada *propiedad*, es decir, el máximo *valor* para la *propiedad* 1 es 0.6 que es el primer *valor* de la *propiedad*, y el máximo *valor* para la *propiedad* 2 es 0.5 que corresponde al segundo *valor* de dicha *propiedad*. De esta manera, la acción combinada se construye a partir de la normalización de los máximos *valores* obtenidos para cada *propiedad*. Según el ejemplo enunciado anteriormente, el valor de la propiedad 1 sería ⟨`rotación, rotar-der`⟩, y así sucesivamente.

Existe cierta evidencia (Willingham et al., 1989; Sun y Zhang, 2004) para argumentar que, en el proceso de aprendizaje de habilidades, las salidas del nivel cognitivista son enviadas al nivel emergente donde son combinadas con las salidas de éste para producir una salida combinada. Ha sido demostrado que el conocimiento explícito (en el nivel cognitivista) "influye" en la conducta experta de los humanos, pero que éste no controla directamente dicha conducta experta (Stanley et al., 1989). Alternativamente, en los procesos de razonamiento, existen algunos indicadores de que la salida del nivel emergente (sub-simbólico) es enviada al nivel cognitivista (simbólico), demostrando así que el conocimiento implícito influye en el razonamiento explícito, aunque éste tampoco se produce de forma directa (Nisbett y Wilson, 1977; Sun y Zhang, 2003).

La justificación cognitiva del proceso de combinación de acciones, se basa en que éste permite diferentes modos operacionales en los *Comportamientos* de la arquitectura: un modo que se basa sólo en el nivel emergente (el cual define un peso de 0 para los componentes del nivel cognitivista), un modo que se basa sólo en el nivel cognitivista (el cual

define un peso de 0 para los componentes del nivel emergente), y un modo combinado donde se integran las salidas de los dos niveles. Estos modos operacionales se corresponden en líneas generales con las nociones psicológicas acerca de los modos de conducta intuitivos (reactivos), modos de conducta deliberativos, y modos de conducta mezclados (teleoreactivos) (Willingham et al., 1989).

Con respecto a los pesos de cada componente empleados para calcular el *valor* combinado, estos pueden ser de dos tipos: pre-determinados o variables. Si los pesos son predeterminados, los valores w_{PMR}, w_{SIA}, w_{RES}, y w_{RP} se definen como constantes. En el caso de ser variables, se emplean unas ecuaciones de probabilidad similares a las empleadas en la selección estocástica (sección 7.7.1) las cuales se basan en un indicador de éxito ie y en las estadísticas de positividad y negatividad de cada componente, de la misma forma que se propuso en la ecuación 7.37. De esta manera, en el método variable, los pesos de los componentes son ajustados durante la interacción del agente con su entorno, indicando en términos generales, cuál es el componente que ha tenido mejor desempeño durante dicha interacción y que por tanto debe tener una mayor influencia (o probabilidad de influir) sobre las salidas combinadas de acciones futuras.

El método de combinación puede ser visto, desde una perspectiva más general y compleja, como un mecanismo de integración vertical que incluye procesos de *Rectificación Ascendente* y *Orientación Descendente*. En la Rectificación Ascendente, la salida del nivel emergente es enviada al nivel cognitivista donde es rectificada y combinada con la salida de éste último nivel, similar a como probablemente se gestan los procesos de razonamiento en el cerebro humano (Nisbett y Wilson, 1977; Stanley et al., 1989). En la Orientación Descendente, la salida del nivel cognitivista es enviada al nivel emergente donde es combinada. Este tipo de integración descendente se evidencia principalmente en situaciones de aprendizaje de destrezas, donde las acciones que provienen del razonamiento empírico (nivel emergente) toman en cuenta el conocimiento experto disponible (nivel cognitivista) para generar una actuación combinada (Willingham et al., 1989).

Una vez dadas las posibilidades de combinación de acciones descritas anteriormente, sólo resta mencionar que la manera en que este tipo de combinación es producida depende del contexto donde será aplicado cada *Comportamiento*, por tanto, resulta necesario especificar los parámetros respectivos de forma independiente para cada *Comportamiento* en el módulo Meta-cognitivo de la arquitectura.

7.7.3. Agregación por Votación

El mecanismo de agregación por votación emplea el Método de Conteo Borda, cuya utilidad en la agregación de las salidas de múltiples sistemas de aprendizaje que se ejecutan concurrentemente ha sido demostrada en (Erp y Schomaker, 2000; Jiang y Kamel, 2006; Romero y De Antonio, 2007b).

El Método de Conteo Borda, es un método de elección posicional de un único ganador, en el cual los votantes clasifican los candidatos en orden de preferencia. El método Borda determina el ganador de una elección a través del conteo del número de puntos que se otorga a cada candidato y que corresponde a la posición en la cual ha sido clasificado por cada votante. Una vez que todos lo votos han sido contados, el candidato con el

mayor número de puntos es el ganador. Los candidatos restantes se ordenan igualmente de acuerdo con el número de votos obtenidos.

A fin de entender cómo funciona el método de agregación, se presente un ejemplo. Considérese un agente que explora su entorno (un plano bidimensional donde se encuentran objetos y obstáculos) empleando para ello el *Comportamiento* ⟨`desplazarse`⟩, el cual determina en cada instante la orientación y aceleración del desplazamiento del agente dada cierta información sensorial percibida. Cada componente del *Comportamiento* ⟨`desplazarse`⟩ envía señales de salida a dos tipos de actuadores del agente: un actuador que controla la aceleración del desplazamiento del agente, y otro actuador que controla la rotación y cambio de orientación del desplazamiento. El conjunto de acciones posibles para el actuador 1 son: ⟨`avanzar, rápido`⟩, ⟨`avanzar, normal`⟩, ⟨`avanzar, lento`⟩, ⟨`retroceder, normal`⟩, y ⟨`detenerse, rápido`⟩, donde la *propiedad* del actuador corresponde al tipo tipo de desplazamiento (avanzar, retroceder, o detenerse) y el *valor* de cada *propiedad* corresponde a la tasa (semántica) de la aceleración/desaceleración (rápida, normal, lenta); y el conjunto de acciones para el actuador 2 son: ⟨`rotar, izquierda`⟩, ⟨`rotar, derecha`⟩, y ⟨`rotar, no-rotar`⟩, donde los valores de la *propiedad* indican el sentido de la rotación.

Supóngase que en el instante x, el agente identifica un obstáculo en frente de éste, y como consecuencia, los componentes del *Comportamiento* activan las recomendaciones de acción presentadas en la tabla 7.4. En este mecanismo de integración de acciones, se requiere una codificación homogénea de las salidas de cada componente[25]. Se ha escogido una codificación ordinal decimal con incrementos de 100 unidades, donde la *propiedad* que pertenece al actuador 1 tiene una codificación de 100, la *propiedad* del actuador 2 tiene una codificación de 200, y así sucesivamente. Los *valores* de cada *propiedad* tienen una codificación que se corresponde con el índice de la secuencia, así por ejemplo, el *valor* 1 tendrá una codificación de 1, el *valor* 2 una codificación de 2, y así sucesivamente. Luego, la pauta ⟨`propiedad, valor`⟩ de cada actuador, se codificaría así: *codificación-propiedad* + *codificación-valor*. Por ejemplo, la pauta ⟨`avanzar, rápido`⟩ del actuador 1 se codificaría así: 100 (avanzar) + 1 (rápido) = 101.

Componente	Acciones	Codificación
RP_1	⟨`retroceder, normal`⟩, ⟨`rotar, derecha`⟩	104, 202
RES_1	⟨`avanzar, rápido`⟩, ⟨`rotar, izquierda`⟩	101, 201
SIA_1	⟨`retroceder, normal`⟩, ⟨`rotar, izquierda`⟩	104, 201
PMR_1	⟨`detenerse, rápido`⟩, ⟨`rotar, no-rotar`⟩	105, 203

Tabla 7.4: Ejemplo de recomendaciones de acción del *Comportamiento* ⟨`desplazarse`⟩ para el estado percibido x.

[25] debido a que cada componente produce una salida en un tipo de representación distinta (simbólica o sub-simbólica), se requiere una representación unificada para facilitar la manipulación de los valores de cada acción en la agregación de una salida integrada.

Ahora, considérese un segundo *Comportamiento* ⟨evadir-obstáculos⟩ que también ha sido activado, aunque la ejecución de sus acciones posee un orden diferente (i.e., primero ejecuta la acción de rotación y luego la acción de aceleración)[26]. En la tabla 7.5

Componente	Acciones	Codificación
RP_2	⟨rotar, no-rotar⟩, ⟨detenerse, rápido⟩	203, 105
RES_2	⟨rotar, derecha⟩, ⟨avanzar, rápido⟩	202, 101
SIA_2	⟨rotar, izquierda⟩, ⟨retroceder, normal⟩	201, 104
PMR_2	⟨rotar, no-rotar⟩, ⟨avanzar, normal⟩	203, 102

Tabla 7.5: Ejemplo de recomendaciones de acción del *Comportamiento* ⟨evadir-obstáculos⟩ para el estado percibido x.

El método de conteo de votos de Borda, como se mencionó anteriormente, considera el valor posicional de cada voto (en este caso, del valor posicional u orden de prioridad de ejecución de cada acción). Así pues, arbitrariamente se ha escogido un valor de 10 puntos para las acciones que se ejecuten primero, y un valor de 5 puntos para las acciones que se ejecuten en un segundo paso. Por ejemplo, la salida del componente RP del *Comportamiento* ⟨desplazarse⟩ obtendrá los siguientes puntos: ⟨retroceder, normal⟩ = 10 puntos, ⟨rotar, derecha⟩ = 5 puntos[27]. Así pues, durante el conteo de votos para cada acción, se suma primero el número de votos obtenidos (i.e., la cantidad de repeticiones por cada acción, calculada a partir de las tablas 7.4 y 7.5) y luego se multiplican por el número de puntos correspondientes. Los resultados se resumen en la tabla 7.6.

Identificando las acciones que obtuvieron un mayor número de votos, se tiene que: para el Actuador 1 se debe ejecutar la acción ⟨retroceder, normal⟩ (104), y para el Actuador 2 se debe ejecutar la acción ⟨rotar, no-rotar⟩ (203). Como se puede observar, la nueva acción agregada es diferente a las recomendaciones de acción hechas por cada componente de los dos *Comportamientos*. Esto se debe a que el mecanismo de agregación considera los esfuerzos que realizaron las pautas de acción de forma independiente y genera una nueva salida unificada que integra las pautas con mayor votación teniendo en cuenta la prioridad de ejecución recomendada para de cada una de éstas. Cabe notar, que cada uno de estos componentes recibirá una señal de retroalimentación[28] la cual indicará que tan apropiada fue su recomendación de acción con respecto a la acción integrada. De esta forma, se irá ajustando el aprendizaje de cada componente hasta que la salida agregada se aproxime incrementalmente a la salida deseada, y las acciones de

[26] Esto con el fin de ilustrar más adelante, cómo el mecanismo de agregación determina una salida integrada que considera el orden de ejecución de las acciones. En la práctica, posiblemente estos dos *Comportamientos* (⟨desplazarse⟩ y ⟨evadir-obstáculos⟩) no serían activados simultáneamente debido a su caracter disyuntivo, sin embargo por ahora, éste resulta un buen caso para ejemplificar la integración de salidas de múltiples *Comportamientos*.

[27] Nótese que debido a que el orden de ejecución de las acciones es un aspecto importante en este tipo de integración, aquellas acciones que se recomiendan ejecutar primero obtienen una mayor cantidad de puntos en comparación con aquellas que se ejecutan después.

[28] Adicional a la señal de retroalimentación que reciben los componentes de aprendizaje por refuerzo de cada *Comportamiento* (i.e., el SIA y la red PMR), son actualizadas las estadísticas de positividad (CP) y negatividad (CN) de todos los componentes.

Acción codificada	Votos
101	$(1_{(RES_1)} \times 10) + (1_{(RES_2)} \times 5) = 15$ votos
102	$(1_{(PMR_2)} \times 5) = 5$ votos
104	$(2_{(RP_1,SIA_1)} \times 10) + (1_{(SIA_2)} \times 5) = 25$ votos
105	$(1_{(PMR_1)} \times 10) + (1_{(RP_2)} \times 5) = 15$ votos
201	$(1_{(SIA_2)} \times 10) + (2_{(RES_1,SIA_1)} \times 5) = 20$ votos
202	$(1_{(RES_2)} \times 10) + (1_{(RP_1)} \times 5) = 15$ votos
203	$(2_{(RP_2,PMR_2)} \times 10) + (1_{(PMR_1)} \times 5) = 25$ votos

Tabla 7.6: Conteo de Votos para cada salida de los Componentes.

cada componente se irán sincronizando progresivamente hasta que se produce un efecto de resonancia (como la metáfora del coro aleluya descrita en la sección 4.6). En la figura 7.23 se ofrece una ilustración del proceso.

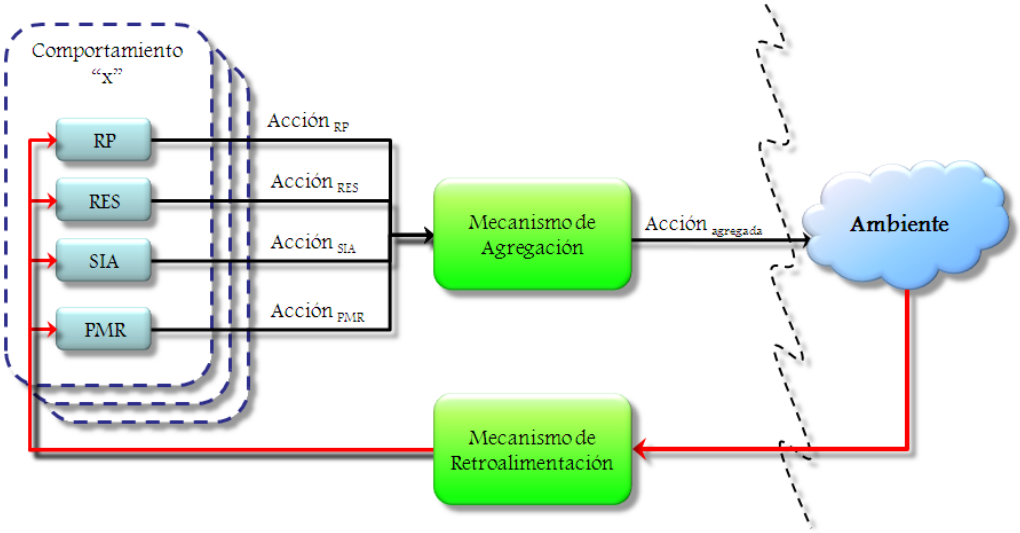


Figura 7.23: Mecanismo de Agregación y Señal de Retroalimentación.

A continuación, se enuncian algunas consideraciones importantes acerca del empleo del mecanismo de agregación de acciones:

- El mecanismo de agregación resulta ser útil cuando son activados múltiples *Comportamientos* simultáneamente y se requiere unificar las acciones de todos los componentes en una sola acción que represente las pautas de actuación más representativas (es decir, aquellas con mayoría de votos).

- Cuando se requiere un orden específico en la ejecución de los actuadores[29], resulta útil la agregación por conteo de votos que da un valor posicional (u ordinal) a cada pauta de actuación.
- En el caso de diseñarse una concreción contextual de la arquitectura donde cada uno de los componentes del *Comportamiento* no genere una sola recomendación de acción sino una secuencia de las mismas, y donde se requiera que esta secuencia de acciones tenga, por supuesto, un orden jerárquico o prioritario de ejecución de cada acción, el método de agregación resulta útil debido a que estima, no sólo la pauta de acción ganadora sino los siguientes puestos en la votación (segundo puesto, tercer puesto, etc.) con lo cual se puede establecer un orden de ejecución de cada pauta de acción con respecto a su cantidad de votos. En los casos donde una pauta de actuación tenga el mismo número de votos que otras, significa que es irrelevante el hecho de cuál de éstas se ejecute primero, por lo tanto, se emplea un mecanismo estocástico para determinar la prioridad de tales pautas con el mismo número de votos.
- Para imprimir un mayor énfasis en el proceso de votación, además del puntaje otorgado por el valor posicional de cada pauta de acción (e.g., 10 para la primera pauta, 5 para la segunda, ...), es posible incluir un factor de peso para cada componente del *Comportamiento* afectando de esta manera su cantidad de votos o puntos obtenidos. Se propone pues, un factor que está bajo la influencia del nivel de positividad o negatividad del componente de esta manera:

$$factorPeso_{comp} = \frac{c1 + \sum CP_{comp}}{c2 + \sum CP_{comp} + \sum CN_{comp}} \tag{7.42}$$

  donde: $c1$ y $c2$ representan constantes (por defecto $c1 = 1$ y $c2 = 2$).

  De esta manera, el número total de votos (puntos) obtenidos por cada pauta de acción estaría determinado por:

$$conteoVotos_i = \sum_j factorPeso_j \times valorPosicional_{ij} \tag{7.43}$$

  donde: i es la pauta de acción. j es el componente que recomienda la pauta i.

- Existe suficiente evidencia (Hull et al., 1990; Chengkai et al., 2006; Jiang y Kamel, 2006; Romero y De Antonio, 2007b) que indica que el método de agregación Borda reduce los episodios de aprendizaje y entrenamiento requeridos por los sistemas de razonamiento procedimental agregados (en este caso, cada uno de los componentes del *Comportamiento*), al ejecutar un mecanismo que genera primero una salida unificada y que luego retroalimenta cada uno de los componentes, en lugar de ejecutar individualmente cada uno de estos. En otras palabras, la ejecución individual de cada componente del *Comportamiento* implicaría un mayor tiempo y cantidad de episodios de aprendizaje hasta lograr una convergencia moderada en cada algoritmo, mientras que empleando el mecanismo de agregación

[29] Como en el ejemplo anterior, donde el *Comportamiento* ⟨`desplazarse`⟩ ejecutaba primero la señal motora que permitía acelerar o desacelerar y luego sí ejecutaba la señal motora para rotar, contrariamente al *Comportamiento* ⟨`evadir-obstáculos`⟩ que primero rotaba y luego sí aceleraba o desaceleraba.

este tiempo de convergencia es reducido, alcanzando óptimos globales con mayor rapidez y eficacia. De esta manera, se produce un aprendizaje incremental (*incremental learning*), donde la curva de aprendizaje del sistema agregado decrece de forma exponencial, en comparación con las curvas lineales o cuadráticas que producen algunos algoritmos de aprendizaje cuando son ejecutados individualmente (Chengkai et al., 2006).

- Por último cabe resaltar que, el método agregado, aunque no siempre encuentra la combinación óptima de pautas de actuación en cada ciclo cognitivo, sí elimina el riesgo de emplear como única opción un algoritmo con un desempeño deficiente y, en una escala de tiempo más amplia, la combinación y agregación de pautas de actuación es optimizada.

7.8. Aprendizaje Vertical

> *"Intelligence involves a great deal more than the ability to follow rules(which is what chess-playing program does). It is also the ability to make up the rules for oneself, when they are needed, or to learn new rules through trial and error".*
> — S. Grand, Creation. Life and how to make it, 2000.

Cada *Comportamiento* de la arquitectura define dos mecanismos verticales de aprendizaje: uno descendente y otro ascendente. El mecanismo ascendente es el que realizan las reglas del tipo RES (descrito en la sección 7.5.2), donde el conocimiento sub-simbólico que ha sido almacenado de forma implícita, es extraído y estructurado en un formato de representación explícita (una regla de producción simbólica). Por tanto, el mecanismo que resta explicar es el aprendizaje descendente.

El mecanismo de aprendizaje descendente extrae conocimiento simbólico de los componentes del nivel cognitivista y lo incorpora en un formato sub-simbólico en los componentes del nivel emergente del *Comportamiento*. El aprendizaje descendente puede llevarse a cabo de dos formas, según se configure en el módulo Meta-cognitivo:

- *Orientación*: este tipo de aprendizaje descendente resulta útil cuando no se dispone de una señal ambiental de retroalimentación que guíe el aprendizaje por refuerzo en el SIA y en la red PMR. En este caso, se produce una señal de refuerzo interna, tal que, si la acción recomendada por el componente emergente (SIA o red PMR) se corresponde con el chunk de acción de la regla simbólica que se activó para la misma entrada sensorial[30], el refuerzo es positivo, de lo contrario, el refuerzo es negativo.

  Es posible también determinar un refuerzo variable que dependa del nivel de afinidad entre la acción del componente emergente y el chunk de acción del componente

[30]Este proceso de aprendizaje descendente por orientación sólo es posible en los casos en que exista una regla simbólica del tipo RP (o una del tipo RES, pero sólo si ésta ha sido refinada bien sea por procesos de generalización o especialización), que haya sido activada en el estado percibido del ciclo cognitivo anterior, es decir, en el ciclo donde se activó el anticuerpo ganador en el SIA y los valores máximos de Q para la salida de la red PMR, los cuales se desean retroalimentar en el paso actual de ejecución.

RP. Esto es, se define un umbral $umbral_{orienta}$ que determina el número mínimo de coincidencias de pautas de acción del tipo $\langle$`propiedad, valor`$\rangle$ entre el chunk de acción del componente RP y la acción del componente emergente, para que pueda haber un refuerzo positivo. De esta manera, el número de coincidencias determina la señal de refuerzo: un número de coincidencias que sobrepase el $umbral_{orienta}$ se corresponde con un refuerzo positivo, un número de coincidencias por debajo del $umbral_{orienta}$ se corresponde con un refuerzo negativo, y si el número de coincidencias es igual a $umbral_{orienta}$ se corresponde con un refuerzo nulo. La intensidad de la señal de refuerzo se determina en función del número de coincidencias.

- *Asimilación*: en este tipo de aprendizaje descendente se incorpora nuevo conocimiento (de origen simbólico) de forma directa en el componente emergente. Este conocimiento es puesto a prueba en ciclos cognitivos posteriores a fin de validar su utilidad.

  En la red PMR, el conocimiento simbólico extraído se incluye como un conjunto nuevo de patrones de entrenamiento en la red, esto es, se entrena la red PMR con los nuevos patrones de activación durante un número limitado de iteraciones[31], y se efectúa una retroalimentación que bien puede ser a través del mecanismo de retropropagación o mediante la actualización de los valores Q (*Q-learning*).

  En el componente SIA, la regla simbólica extraída es convertida a un formato subsimbólico y agregada como un nuevo anticuerpo en el acervo inmune. El nivel de ganancia o aptitud del nuevo anticuerpo corresponde a la aptitud promedio de toda la población de anticuerpos en la memoria inmune (memoria elitista donde se encuentran los anticuerpos más aptos), con lo cual se garantiza que el nuevo anticuerpo tendrá un buen nivel para competir en las dinámicas dirigidas por el sistema de asignación de recompensas.

  La asimilación también puede darse entre los componentes del nivel emergente (SIA y PMR). En este caso, no se requiere una conversión específica de formato puesto que ambos componentes emplean representaciones sub-simbólicas.

La frecuencia de activación de cada uno de estos tipos de aprendizaje vertical y sus variantes, depende de la configuración establecida en el módulo Meta-cognitivo. Es posible que, para concreciones contextuales de la arquitectura en entornos donde se requiera un mayor procesamiento deliberativo, la activación de los mecanismos de aprendizaje vertical se realice en cada ciclo cognitivo (con el fin de asegurar una mayor consistencia y sincronización en las respuestas a costa de un mayor tiempo de procesamiento). En contraposición, para una concreción contextual con mayor énfasis reactivo, probablemente se requiera que los mecanismos de aprendizaje vertical se lleven a cabo cada cierto número de ciclos cognitivos (evitando que se afecte significativamente el tiempo de procesamiento que precisa la sincronización).

[31] Es posible determinar un criterio de paro del entrenamiento relacionado con un mínimo del error alcanzado, no obstante, si esto supera un número de iteraciones máximas o un tiempo máximo, el entrenamiento es detenido. Este criterio de paro debe ser determinado de forma individual para cada concreción contextual de la arquitectura.

7.9. Discusión

En el presente capítulo, se han propuesto los *Comportamientos* como unidades robustas y flexibles de procesamiento. En un sentido funcional, los *Comportamientos* no solamente representan la manera en que el conocimiento procedimental es almacenado en forma de reglas simbólicas y sub-simbólicas, sino que además son componentes híbridos e integradores de diversos mecanismos, tales como el reconocimiento de patrones, la representación y generalización de conocimiento, el razonamiento, el aprendizaje, el descubrimiento de pautas, la auto-organización, y la adaptación, entre otros.

A continuación se exponen algunas reflexiones acerca del diseño de los diferentes componentes del *Comportamiento*, las cuales podrían ser tomadas en cuenta en futuras mejoras de la arquitectura.

Sobre el módulo Conexionista (Redes Neuronales):

En la presente tesis se han propuesto dos modelos de Redes Neuronales (el perceptrón multicapa con retropropagación del error y el perceptrón multicapa con un mecanismo de aprendizaje Q-learning), cuyo entrenamiento es guiado, respectivamente, por dos tipos de aprendizaje: *supervisado* y *por refuerzo*. En el campo investigativo del aprendizaje automático (machine learning) existe un tercer tipo de aprendizaje de gran utilidad: el aprendizaje *no supervisado*. En este tipo de aprendizaje, todo el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formado tan sólo por las entradas al sistema, donde no se tiene información a priori sobre las categorías de tales ejemplos ni se dispone de una señal de retroalimentación que dirija el aprendizaje. Si el aprendizaje supervisado es equivalente a un aprendizaje guiado por un un profesor, entonces el aprendizaje no supervisado es análogo a un aprendizaje por descubrimiento llevado a cabo por un autodidacta. En el área de las redes neuronales artificiales, el modelo de aprendizaje no supervisado más estudiado (e indudablemente el más inspirado en el funcionamiento del cerebro) son los *Mapas Auto-organizados* o *Redes de Kohonen* (Kohonen, 2001).

En este tipo de redes, el entrenamiento se realiza presentando solamente un conjunto de entradas. Como resultado de este aprendizaje no supervisado, la red debe descubrir patrones o características significativas en los datos de entrada, e incorporarlas a sus pesos a través de un proceso de auto-organización. En los mapas auto-organizados los puntos cercanos en el espacio de entrada activan neuronas cercanas en el espacio de salida, y viceversa. Esto implica que cuando una neurona aprende un patrón específico, sus neuronas vecinas aprenden patrones similares. Algunos campos de aplicación reales en los que se ha demostrado su eficacia (Haykin, 1999), así como aspectos interesantes que potenciarían aún más el procesamiento llevado a cabo por cada *Comportamiento* de METÁFORA, incluyen tareas como la clasificación, la reducción de dimensiones, el análisis de similitud de patrones de entrada, la adquisición de conocimiento, y la extracción de rasgos, entre otros.

Otro aspecto que podría incluirse en la implementación de las redes neuronales de cada *Comportamiento*, es un mecanismo de diseño evolutivo de la arquitectura de la red. En la literatura especializada (Koza y Rice, 1991; Maniezzo, 1994; Rovithakis et al., 2004), se han documentado múltiples aproximaciones que han intentado abordar este tema mediante algoritmos genéticos que evolucionan aspectos parciales de las redes neuronales,

tales como los pesos, los umbrales, o la topología. Sin embargo, en la última década, se han propuesto diversas aproximaciones que intentan inducir, mediante algoritmos evolutivos, la estructura total de la red (Kenneth y Miikkulainen, 2002; Ferreira, 2004). La aplicación de los algoritmos evolutivos en la optimización de redes neuronales artificiales se justifica por el menor coste computacional, en comparación con los métodos de prueba y error, y la mayor robustez frente a los métodos constructivos/destructivos (Castillo et al., 2001).

Durante el desarrollo de la presente tesis, se planteó (en borrador) el diseño de un mecanismo evolutivo basado en Programación de Expresiones Genéticas (Ferreira, 2001), a través del cual sería posible codificar linealmente (en forma de cromosomas) toda la arquitectura de cada red neuronal, esto es, la topología (e.g., número de capas, número de neuronas por capa, neuronas recurrentes, etc.), los pesos sinápticos, y los umbrales y funciones de transferencia para cada neurona. Este mecanismo evolutivo permitiría aplicar fácilmente un conjunto de operadores genéticos (selección, recombinación, mutación, etc.) sobre tales representaciones cromosómicas, cuya aptitud (fitness) sería luego evaluada (dependiendo del tipo de red neuronal) mediante la política de aprendizaje respectiva: retropropagación, Q-learning, o aprendizaje competitivo (en el caso de los mapas auto-organizados). Sin embargo, debido principalmente a ciertas restricciones relativas al alcance y duración de la tesis, y a la complejidad inherente al tema, el borrador de este diseño no pudo ser validado (a través de una implementación concreta o mediante técnicas formales de validación), ni refinado (como resultado de ejecutar un proceso sistemático de pruebas). No obstante, se espera poder continuar ahondando en este aspecto en trabajos futuros, y poder así validar su viabilidad como mecanismo de adaptación del módulo conexionista.

Sobre el módulo de Producciones:

Como se puede deducir a partir del diseño del módulo de Producciones, el descubrimiento de nuevas reglas simbólicas depende en gran medida del nivel emergente, esto es, las reglas de producción en el nivel cognitivista solamente pueden crear nuevas reglas más especializadas o generalizadas, a partir de las reglas sub-simbólicas halladas por los componentes PMR y SIA en el nivel emergente. Así pues, el módulo de Producciones, por sí solo, no tiene la capacidad de crear nuevas reglas a partir de la interacción del agente con su entorno y, como consecuencia, este módulo sirve más como un refinador de reglas que como un descubridor de las mismas. La principal desventaja de esta condición es que si el agente tuviese que emplear únicamente el nivel cognitivista para crear modelos de su entorno (i.e., si se desconectasen los componentes del nivel emergente), éstos serían representaciones constreñidas a las reglas que se han definido de forma predeterminada y, por tanto, el sistema no tendría la capacidad de "aprender" nuevas pautas.

En este orden de ideas, resulta casi inminente la incorporación de un mecanismo de aprendizaje de reglas simbólicas. Probablemente, una de las áreas del aprendizaje automático más idóneas para buscar una solución a este problema es la *inducción de reglas*. Este mecanismo de aprendizaje extrae colecciones de reglas formales o patrones a partir de un conjunto de observaciones. Las aproximaciones más comunes que emplean este tipo de aprendizaje, por mencionar algunas, incluyen la generación de reglas a partir de árboles de decisión (Quinlan, 1987), los espacios de versiones (Dubois y Mohamed,

2002), la poda incremental del error reducido (algoritmo IREP) (Furnkranz, 1994), la poda incremental repetida para reducir el error (algoritmo RIPPER) (Cohen, 1995), el aprendizaje de reglas empleando taxonomías (algoritmo TRIPPER) (Vasile et al., 2004), entre otros. Por otro lado, existen aproximaciones para el descubrimiento de reglas empleando jerarquías difusas (Ba-Alwi y Bharadwaj, 2005), e incluso aproximaciones basadas en programación genética (Al-Maqaleh y Bharadwaj, 2005). Dado el amplio espectro de posibilidades, resultaría necesario evaluar, por un lado, qué alternativa de aprendizaje encajaría mejor con el tipo de representación de reglas simbólicas propuesto en esta tesis y, por otro, cuál de ellas implicaría un menor coste computacional de implementación.

Un segundo aspecto que podría tenerse en cuenta en extensiones futuras del módulo de Producciones, es la inclusión de una plataforma que soporte diversas técnicas para la ingeniería del conocimiento (e.g., diferentes mecanismos de representación, inferencia y razonamiento). La razón principal para esto es que en la actualidad existen herramientas disponibles, la mayoría de ellas de uso libre (por lo menos para fines académicos), que facilitan el desarrollo de Sistemas Expertos o SBC (sistemas basados en conocimiento), cuya funcionalidad incrementaría la robustez y complejidad del módulo de Producciones propuesto en esta tesis.

Cabe notar que la decisión inicial de no utilizar una herramienta de este tipo se fundamentó en la idea de proveer un mecanismo representacional que permitiese la fácil comunicación e integración con otros módulos que empleasen conocimiento simbólico (e.g., el módulo Declarativo), o con módulos que empleasen conocimiento sub-simbólico (e.g., los componentes del nivel emergente). Así pues, el modelo propuesto en esta tesis permite convertir fácilmente una representación simbólica a una sub-simbólica, y viceversa, evitando complicadas traducciones que tendrían que hacerse de lo contrario. Sin embargo, el proceso de razonamiento llevado a cabo por este sistema podría potenciarse si se usaran, además del encadenamiento hacia adelante, otras técnicas de encadenamiento e inferencia (e.g., el encadenamiento hacia atrás, el razonamiento bajo incertidumbre, la representación de las reglas como máquinas de estados –RETE– etc.). En otras palabras, con el módulo de Producciones propuesto se sacrifica parte de la robustez del proceso de razonamiento, mientras se gana en unificación representacional al sugerirse la regla simbólica como una unidad básica de trasmisión y conversión.

Concretamente, se propone validar la integración del módulo de Producciones de METÁFORA con la plataforma JESS[32]. JESS es un Sistema Basado en Reglas de Producción, implementado en tecnología Java, considerado como uno de los motores de reglas más rápidos disponibles. JESS permite el encadenamiento de reglas hacia adelante, emparejando hechos de la Memoria Activa con los antecedentes de cada regla; y hacia atrás, emparejando hipótesis de la Memoria Activa con los consecuentes de cada regla. Adicionalmente, emplea un algoritmo llamado RETE que mejora la eficacia del emparejamiento de reglas en el Motor de Inferencias.

En cuanto a los aspectos tecnológicos, JESS puede manipular y razonar directamente sobre objetos de Java. Por tanto, se pueden crear objetos, hacer invocaciones a métodos, o ejecutar interfaces y APIs de terceros desarrolladas para Java. Esto representa una ventaja puesto que, como se describirá en el capítulo sobre la Experimentación, para la

[32] `http://www.jessrules.com/jess/`

concreción contextual de la arquitectura se escogió el lenguaje Java y la plataforma multi-agente JADE, debido a todas las fortalezas que esto supone frente a otros lenguajes de programación orientados a objetos.

Por todas estas características, tanto funcionales como técnicas, se presupone que la integración con la plataforma JESS incrementaría la eficiencia y la potencia del módulo de Producciones.

Sobre el módulo Comportamental en general:

Finalmente, se hace una observación general sobre el tratamiento que podría dar el módulo Comportamental a la incertidumbre presente en la información sensorial percibida, y por ende, en los procesos de razonamiento y actuación.

Como se ha podido observar a lo largo de este capítulo, los diferentes componentes de cada *Comportamiento* emplean mecanismos de *"razonamiento exacto"* en lugar de emplear mecanismos de *"razonamiento aproximado"* (similar a como lo hace nuestro cerebro para resolver los problemas cotidianos). Esto es, que los diferentes procesos de inferencia se realizan sobre el supuesto de que todos los hechos observados representan valores precisos. Por ejemplo, considérese la siguiente regla simbólica:

SI
⟨un balón se aproxima ''velozmente'' hacia mi⟩ Y
⟨su trayectoria pasa ''muy cerca'' de mi cara⟩
ENTONCES
⟨debo agacharme ''rápidamente''⟩

En este caso, los cuantificadores "velozmente", "muy cerca", y "rápidamente" son información imprecisa y ambigua (i.e., no indican una medida que pueda ser cuantificable, con lo cual, lo que para un agente puede parecer "muy cerca", para otro puede parecer "lejos"). Para evitar este problema de incertidumbre, en la presente tesis, se optó por definir un método de conversión que emplea umbrales exactos a fin de determinar cuándo una variable observada toma uno u otro valor lingüístico, así, por ejemplo, el percepto "distancia de aproximación a un objeto" podría tomar los valores: ⟨muy−−cerca⟩ si $distancia < 0.2m$, ⟨cerca⟩ si $0.2m \geq distancia < 1.0m$, ⟨lejos⟩ si $1.0m \geq distancia < 5.0m$, y ⟨muy-lejos⟩ si $distancia \geq 5.0m$. Sin embargo, en la vida real estos intervalos no están definidos de manera tan estricta, sus límites pueden variar de un contexto a otro, o simplemente pueden estar bajo la influencia de experiencias subjetivas anteriores del agente.

Por ejemplo, en el supuesto de que nuestros ojos pudiesen medir con tal precisión la distancia hasta donde está situado nuestro coche, no sería razonable considerar que éste pase de estar "cerca" a "lejos", y viceversa, por el simple hecho de acerarnos o apartarnos un centímetro de él (e.g., cuando se sobrepasa la distancia de $1.0m$ a $1.01m$). Esto implica que cada uno de los valores de las propiedades de un percepto puede pertenecer (parcialmente) a un conjunto difuso de datos (puesto que sus fronteras no son exactas) y, simultáneamente, pertenecer al complemento de dicho conjunto.

Por tanto, resulta necesario definir un mecanismo que convierta los valores reales (e.g., las mediciones de los sensores del agente) en etiquetas lingüísticas o valores difusos, y viceversa. Un modelo matemático que permite este tipo de tratamiento de la incertidumbre es la *lógica difusa* y los *conjuntos borrosos* propuestos por Zadeh (Zadeh, 1983). La lógica difusa es una alternativa a la lógica clásica que pretende introducir un grado de vaguedad en las cosas que evalúa. En comparación con la lógica convencional, la lógica difusa permite trabajar con información que no es exacta para poder definir evaluaciones convencionales, contrario a la lógica clásica que permite trabajar solamente con información definida y precisa. Un aspecto importante de la lógica difusa es que permite incorporar el concepto de incertidumbre y/o confianza mediante la integración de dos factores: la imprecisión en la medición y la subjetividad que caracteriza al control lingüístico. De esta manera, no sólo se podría tratar la incertidumbre producida por el ruido en la información percibida (e.g., debido a la falta de precisión en la calibración de los sensores del agente), sino que además permitiría el procesamiento de las etiquetas lingüísticas a través del razonamiento aproximado.

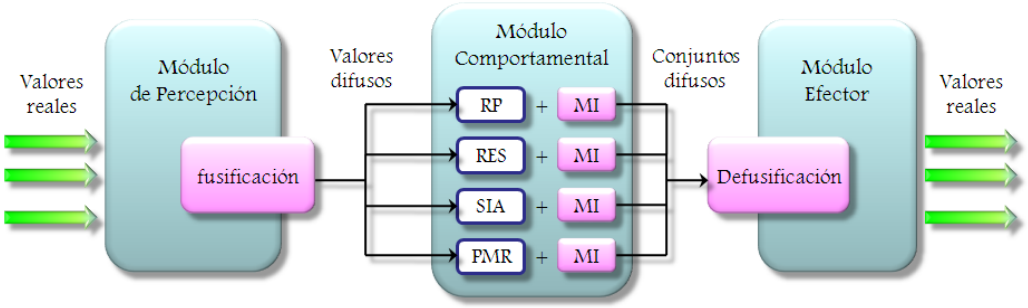


Figura 7.24: Inclusión de un Sistema de Lógica Difusa en METÁFORA.

Una posible solución a este problema puede ser la inclusión de un Sistema de Lógica Difusa en la arquitectura propuesta para METÁFORA (como se ilustra en la figura 7.24). Tomando como referencia a (Duarte, 1997), este sistema constaría de: (1) un mecanismo de *fusificación* (*fuzzification*, en inglés), situado en el módulo de Percepción, que se encargaría de convertir las entradas reales en conjuntos difusos; (2) una *base de conocimiento* compuesta por reglas difusas, las cuales serían generadas por cada uno de los componentes del *Comportamiento* (RP, RES, SIA, y PMR), aunque, a diferencia de lo planteado en este capítulo, cada regla difusa tendría una clase y un grado de certeza asociado a la clasificación; (3) un *Motor de Inferencia* (MI) que determinaría, a través de un mecanismo de razonamiento difuso, qué reglas pueden ser activadas o disparadas; y (4) un mecanismo de *defusificación* (*defuzzification*, en inglés), situado en el módulo Efector, que convertiría los valores difusos generados durante el proceso de inferencia a valores numéricos concretos que serían procesados por los respectivos actuadores del agente.

Capítulo 8

La Puesta en Escena: Ejecución de los Procesos Subconscientes

"The brain is massively parallel – many things are happening at the same time – largely unconscious in its details, and widely decentralized in any task".
— B.J. Baars, In the Theatre of Consciousness, 1997.

Retomando la metáfora de Baars sobre el Teatro de la Consciencia (descrita en el apartado 4.6.1), en el presente capítulo se exponen los procesos que la arquitectura lleva a cabo, en gran parte, de manera subconsciente y en paralelo.

Por un lado, se describen los procesos subconscientes que según la metáfora de Baars corresponden a la *Audiencia* del teatro: los sistemas de memoria declarativa de largo plazo, los automatismos, la estructura de metas, los motivadores, y el estado anticipado; y, por otro, se describe el *Escenario* del teatro, representado por la Memoria de Trabajo, donde los contenidos subconscientes permanecen a la espera de ser convertidos, por el mecanismo de Orquestación de la Consciencia, en procesos conscientes. Todos estos procesos subconscientes reunidos y operantes en la memoria de trabajo, confeccionan la puesta en escena donde la actuación consciente tendrá lugar.

Debido a la naturaleza miscelánea de este capítulo, se expone cada tema de manera muy puntual, dejando quizás algunas cuestiones abiertas (las cuales podrán ser profundizadas en trabajos futuros), pero estableciendo claramente los elementos que sirven como contexto para el acceso selectivo de los contenidos conscientes, tema que se explicará en el capítulo 9.

8.1. Construcción de Perceptos

El módulo de Percepción, es el encargado de procesar los estímulos externos capturados a través del conjunto de sensores del agente, de manera que dichos estímulos puedan ser fácilmente manipulables y referenciables por otros módulos de la arquitectura. El módulo de Percepción incluye un *Interprete* el cual se encarga de dos funciones esenciales:

1. *La traducción de estímulos*: dado un conjunto de vectores, donde cada uno almacena los estímulos percibidos por un tipo específico de receptor o sensor del agente, el *Interprete* se encarga de procesar y estructurar dichos estímulos en pautas de la forma ⟨`propiedad, valor`⟩. Cada *propiedad* corresponde a una característica específica percibida por el agente a través de los sensores, la cual tiene asociado un conjunto de *valores* posibles (e.g., la *propiedad altura* puede definir el conjunto de valores *baja*, *media*, *alta*, ...).
2. *La construcción de Perceptos*: una vez el *Interprete* ha traducido los datos percibidos por los sensores, estos son enviados al módulo Declarativo donde el *Razonador* ejecutará un mecanismo de inferencia que retornará un conjunto de asociaciones semánticas (mecanismo que será descrito en la sección 8.2.2.1), las cuales aportan información adicional sobre el significado de los datos percibidos. El *Interprete* construye, a partir de los estímulos externos y de la información semántica extraída, un conjunto de estructuras llamadas *Perceptos*, las cuales representan una dimensión interpretativa de la información percibida.

El mecanismo de traducción del *Interprete* es dependiente del contexto. Esto quiere decir que la conversión a *perceptos* de los datos sin procesar capturados por los sensores requiere de procedimientos distintos para cada concreción contextual de la arquitectura. Por ejemplo, supóngase un agente que percibe la ubicación de dos objetos en un espacio bidimensional a través de las coordenadas cartesianas que captura un sensor de posicionamiento (e.g., gps, odometría, etc.). El agente podría calcular la distancia entre los dos objetos empleando dichos datos sin procesar, y a partir de ésta, construir un percepto que determinase si los dos objetos están uno cerca del otro. Sin embargo, si lo que se requiere es que el agente se aproxime a uno de dichos objetos, el *Interprete* tendrá que estimar adicionalmente la orientación relativa del objeto con respecto al agente (y no solamente sus coordenadas en el plano), de manera tal que el agente pueda calcular un ángulo de rotación y un desplazamiento que le permita acercarse gradualmente al objeto. En este caso, se requeriría la definición de procedimientos adicionales en el mecanismo de traducción con el fin de construir los *perceptos* que indicasen la orientación de los objetos con respecto al agente.

Un *percepto* se define entonces como:

$$Percepto_i = estimulos_i \cup significado_i \tag{8.1}$$

donde: $estimulos_i$ es el vector de estímulos percibidos por un conjunto de sensores específicos. $significado_i$ es el conjunto de asociaciones semánticas producidas por el *Razonador*.

Tomando como referencia los estudios de Tanaka (Tanaka y Sagi, 1998), las interpretaciones perceptuales serán *acumulativas* y *volátiles*, esto es, que el significado completo de un estímulo específico se establecerá a través de la construcción gradual de perceptos producida durante ciclos cognitivos continuos, y que la disponibilidad de tales perceptos depende de un nivel de *intensidad* que decae con el transcurrir del tiempo. Por tanto, debido a que los *perceptos* interpretados en cada ciclo cognitivo son almacenados "temporalmente" en un espacio de memoria perceptual, la intensidad de estos se verá afectada por una función de degradación que determinará el tiempo que permanecerán disponibles

en dicho espacio de memoria. La función que determina el nivel de intensidad para un percepto y su tasa de degradación en el tiempo, se propone a continuación:

$$I_{perc} = \frac{k}{log_2\ nc} \tag{8.2}$$

donde: nc es el número de ciclos cognitivos transcurridos desde la última vez que fue percibido el percepto (con $nc > 1$). k es una constante empírica con un valor por defecto de 5.5, la cual se ha determinado en función del tiempo que tardaría en desvanecer la intensidad de cada percepto.

En términos prácticos, la intensidad de un percepto permite al agente el mantener temporalmente un recuerdo (e.g., de un objeto percibido) a pesar de que éste deje de estar presente dentro el campo perceptual. Por ejemplo, si el percepto ⟨`objeto, manzana`⟩ previamente construido por el agente deja de ser percibido debido a un cambio en el foco de atención, la información relativa al objeto manzana se mantendrá disponible en la memoria perceptual durante cierto tiempo, de tal manera que el agente podrá seguir realizando inferencias sobre esta información.

Cuando el nivel de intensidad de un percepto se encuentre por debajo de un umbral determinado, el percepto dejará de ser válido, a menos que el percepto reciba una continua activación por parte de los estímulos sensoriales.

$$I_{perc} > umbral_{perc} \tag{8.3}$$

donde: $umbral_{perc}$ tiene un valor empírico por defecto de 1.0.

De la ecuación 8.2, se obtiene que la intensidad I_{perc} de un percepto está por debajo de $umbral_{perc}$ cuando $nc = 45$, por lo tanto, un percepto que ha dejado de recibir activación por parte de los estímulos sensoriales durante más de 45 ciclos cognitivos es desestimado por el agente. Esto traducido en términos temporales corresponde a un intervalo entre 4.5 y 9.0 segundos (en el supuesto de que cada ciclo cognitivo dure entre 100 y 200 ms. respectivamente), lo cual concuerda con los estudios sobre la volatilidad de los perceptos en el sistema perceptual humano (Byrne y Anderson, 1998).

Por ejemplo, supóngase que el agente es un robot que está equipado con cuatro sensores de proximidad situados en la parte delantera, y que cada uno de estos sensores estima, a través de una señal de propagación acústica (sonar), la distancia de colisión (en metros) hasta otros objetos. En un instante determinado, la medición realizada por los cuatro sensores podría ser la siguiente: *[1.32, 0.85, 0.87, 1.23]*, lo cual, convertido por el *Interprete* en perceptos sería (suponiendo que una distancia de aproximación entre 1.0 y 1.5 metros implica que al agente está "cerca" del obstáculo, y una distancia inferior a 1.0 metros implica que el agente se encuentra a una distancia de aproximación "muy-cerca" del obstáculo): Percepto 1 = ⟨`distancia-obstáculo, cerca`⟩, Percepto 2 = ⟨`distancia-obstáculo, muy-cerca`⟩, Percepto 3 = ⟨`distancia--obstáculo, muy-cerca`⟩, y Percepto 4 = ⟨`distancia-obstáculo, cerca`⟩.

A partir de estos perceptos, y con ayuda del razonador semántico (el cual será descrito en la sección 8.2.2), el *Interprete* podría inferir un nuevo percepto que indicase que el

objeto observado es un obstáculo con una forma curvada (esto debido a las distancias percibidas), como se observa a continuación:

Percepto 5:

⟨objeto, obstáculo⟩
⟨proximidad, muy-cerca⟩
⟨forma, curvada⟩

Una vez construidos los perceptos, son almacenados en la Memoria de trabajo donde podrán ser accedidos por otros módulos de METÁFORA.

8.2. Recuperación y Almacenamiento del Conocimiento Declarativo

8.2.1. Inspiración del Modelo Declarativo

En la metáfora del Teatro de la Consciencia de Baars, la *Audiencia* o público, está conformada por un conjunto vasto de procesos subconscientes[1], los cuales proveen la contextualización que da lugar a la activación de los procesos conscientes. Estos procesos subconscientes están relacionados con la memoria autobiográfica, la memoria declarativa, la memoria semántica, el aprendizaje subconsciente, los automatismos, y el lenguaje entre otros. En esta sección, se presta especial atención al conocimiento declarativo mantenido por METÁFORA y a los mecanismos de aprendizaje relacionados. Los automatismos por su parte, corresponden a la ejecución automática de ciertos *Comportamientos*, los cuales se describen en la sección 8.3.

El módulo Declarativo almacena y recupera, de diferentes formas, el conocimiento general que tiene el agente acerca del mundo. En la literatura, esto es comúnmente referido como *conocimiento declarativo* (Lehman et al., 2006; Anderson, 1996; Sun et al., 2001; Franklin et al., 2005), cuya principal función es almacenar información acerca de los objetos, hechos y sus relaciones. En analogía con el cerebro humano, este módulo representa la memoria de largo plazo declarativa, funcionalidad desempeñada por el lóbulo temporal medial, que incluye la región hipocampal y las cortezas perirrinal, entorrinal y parahipocampal adyacentes (Squire et al., 2004).

La Memoria Declarativa se compone de dos tipos de memoria: la Semántica y la Episódica o Autobiográfica. La distinción de *Memoria Semántica* (MS), da cuenta de un almacén de conocimientos acerca de los significados conceptuales que etiquetan los objetos del mundo percibido por el agente y las relaciones entre estos significados, mientras que la *Memoria Episódica* (ME), representa eventos o sucesos que reflejan detalles de la situación experimentada por el agente (el "qué", "cuándo" y "dónde") y no solamente

[1]Esto es, los espectadores que procesan en "silencio" (de manera subconsciente para el agente) la actuación realizada por los actores.

acerca del significado. Tener en cuenta estas dos subdivisiones de la Memoria Declarativa es importante para entender de qué modo la información está representada y es recuperada diferencialmente por METÁFORA.

La organización de los contenidos en la ME está sujeta a parámetros espacio-temporales; esto es, los eventos que se recuperan de la memoria representan los momentos y lugares en que se presentaron (Tulving, 2002). Sin embargo, la información representada en la MS sigue una pauta conceptual, de manera que las relaciones entre los conceptos se organizan en función de su significado (Dodd, 2005). Otra característica que diferencia ambos tipos de representación se refiere a que los eventos almacenados en la ME son aquellos que han sido explícitamente codificados, mientras que la MS posee una capacidad inferencial y es capaz de manejar y generar nueva información que nunca se haya aprendido explícitamente, pero que se halla implícita en sus contenidos (entender el significado de un nuevo concepto utilizando conceptos ya conocidos). Podría decirse entonces que el conocimiento declarativo en su totalidad conforma el sistema de *creencias* que el agente emplea para describir el estado más probable del entorno.

De manera similar al aspecto híbrido de representación del conocimiento definido para los *Comportamientos* en el capítulo 7, el módulo Declarativo define dos niveles de representación: el simbólico (nivel cognitivista) y el sub-simbólico (nivel emergente). Estos dos niveles definen cada uno, mecanismos diferentes para la extracción de conocimiento, el razonamiento y el aprendizaje; características que motivan la complementariedad entre estos dos niveles.

8.2.2. Conocimiento Semántico

8.2.2.1. Memoria Semántica

La Memoria Semántica (MS) corresponde a una "red semántica", de representación simbólica, cuyos nodos están conformados por chunks. En la arquitectura propuesta, cada chunk, representado por un conjunto de parejas $\langle$`propiedad, valor`$\rangle$, se encargará de codificar grupos de co-ocurrencias de características observadas. Se ha escogido las "redes semánticas" basadas en chunks debido a su simplicidad de representación y a la facilidad que proporciona durante el proceso de razonamiento[2].

Los chunks están conectados entre sí a través de asociaciones uni y bi-direccionales, formando así *reglas asociativas*. De tal manera, la activación de un chunk puede a su vez activar una cadena de chunks mediante la dispersión de activación que fluye a través de las reglas asociativas.

Chunks: la forma básica de un chunk es la siguiente: *chunk-id:* $(prop_1, val_1)$ $(prop_2, val_2) \ldots (prop_n, val_n)$, donde *prop* denota una propiedad particular y *val* es el

[2]En comparación con otros dos modelos analizados: el modelo de Dependencia Conceptual refina la normalización de los elementos de las redes semánticas aunque hace más compleja su representación, y el modelo de Frames (Marcos), permite la representación de conocimiento estereotipado aunque posee problemas para representar conocimiento no taxonómico (Russell et al., 2003).

valor correspondiente a la propiedad. Por ejemplo, el chunk *Mesa-1: (tamaño, grande)(color, blanco)(número-de-patas, cuatro)*, denota una mesa grande de cuatro patas y de color blanco.

Cuando las pautas ⟨`propiedad, valor`⟩ de un chunk son activadas, simultáneamente las pautas ⟨`propiedad, valor`⟩ asociadas en el nivel sub-simbólico (Red de Memoria Asociativa – RMA – la cual se describirá en la siguiente sección) también son activadas. Lo mismo sucede en el sentido contrario. La idea principal es que los valores de las propiedades en la RMA sirvan para representar las características implícitas del chunk, y que el chunk a su vez sirva para etiquetar como un todo este conjunto de características. De esta manera, el chunk funciona como un concepto o trozo de conocimiento que puede ser empleado durante el proceso de razonamiento.

La red semántica define un parámetro de densidad (d_c) que determina la frecuencia mínima de invocación (e.g., codificación, extracción, o activación) necesaria para mantener activo un chunk dentro de la red. Por ejemplo, si $d_c = 1/100$, entonces debería existir al menos una invocación de un chunk por cada 100 pasos de ejecución o de lo contrario se eliminará el chunk. Este parámetro aplica a todos los chunks excepto a aquellos que conforman la memoria episódica. Nótese que, si un chunk es eliminado, todas las reglas asociativas con las que está relacionado serán eliminadas también.

Reglas Asociativas: las reglas asociativas son tuplas del tipo *condición-relación- conclusión*, donde *condición* y *conclusión* son chunks, y *relación* puede ser una de las siguientes:

- *Ocurrencia*: es un vínculo entre un chunk particular y la clase a la que pertenece. Etiqueta: PERTENECE-A.
- *Generalización*: esta relación denota que un chunk es un caso particular de otro chunk de naturaleza más general. Etiqueta: ES-UN.
- *Agregación*: es un vínculo entre un chunk y los chunks que son parte de él. Etiqueta: ES-PARTE-DE y TIENE-UN.
- *Propiedad*: es un vínculo entre un chunk y sus características. La etiqueta de esta relación varía en función del tipo de propiedad.
- *Acciones*: representan vínculos de carácter dinámico. La etiqueta de esta relación varía en función del tipo de acción.

La *condición* de una regla asociativa se conforma de uno o varios chunks, mientras que su *conclusión* de un chunk solamente. Estos chunks están representados como nodos en la red semántica (nivel cognitivista), mientras que sus propiedades y valores respectivos están representados como nodos de características individuales en la RMA (nivel emergente). Por tanto, dada la representación de un chunk, una regla asociativa resulta ser simple, natural y uniforme (i.e, cada regla asociativa representa una correspondencia desde un conjunto de parejas ⟨`propiedad, valor`⟩ hacia otro, en una manera explícita pero sin una estructura demasiado compleja).

Similar al caso de los chunks, existe un parámetro de densidad relativo a la regla asociativa (d_a) que determina la frecuencia mínima de invocación (e.g., codificación, extracción, o activación) necesaria para mantener una regla asociativa dentro de la red.

Intensidad de un Chunk Los chunks pueden ser "activados", bien como el resultado de aplicar una regla asociativa que conecta un chunk activo con otro inactivo o latente, o como el resultado de recibir una entrada ambiental (e.g., un estímulo sensorial percibido por el agente). Específicamente, un chunk puede ser activado de la siguiente manera:

- Todos los chunks cuyas pautas $\langle$`propiedad, valor`$\rangle$ constituyan un sub-conjunto de las pautas $\langle$`propiedad, valor`$\rangle$ de la entrada ambiental (estímulos sensoriales), son "activados" con una intensidad de "1" cuando la entrada es recibida,
- Una regla asociativa puede "activar" un chunk a través de la dispersión de activación,
- Las acciones producidas por los *Comportamientos* pueden "activar" un chunk, y
- Al final de la correlación asociativa realizada en el nivel emergente del módulo Declarativo, se realiza un proceso de activación ascendente (bottom-up) que puede "activar" un chunk.

Combinando estos diferentes tipos de activación, se tiene que la intensidad de un chunk está determinada por:

$$I_k^c = \max_x(I_k^{c,x}) \tag{8.4}$$

donde: x denota un tipo específico de activación, y máx es una función de maximización que selecciona la intensidad más alta entre todos los tipos de activación posibles.

Dado que pueden existir conexiones bi-direccionales entre los chunks, resulta necesario evitar aquellos "bucles de retroalimentación" positiva que conducen a la saturación de la intensidad de los chunks involucrados. La posibilidad de presentarse un bucle de retroalimentación positiva es reducida debido al hecho que la intensidad de activación que dispersa un chunk hacia otros es atenuada por la restricción: $\sum_k I_k \leq 1$. La intensidad puede regresar al chunk origen (debido a un enlace bi-direccional o a un bucle de retroalimentación indirecto), pero ésta siempre regresará de forma atenuada.

Activación de Nivel Base: Cada chunk tiene una activación de nivel base (ANB) similar a la ANB para los chunks de acción en el módulo Comportamental, la cual es determinada por la siguiente ecuación:

$$ANB_j^{ms} = iANB_j^{ms} + c \times \sum_{l=1}^{n} t_l^{-d} \tag{8.5}$$

donde: l indica la l-ésima vez que fue activado el chunk de la memoria semántica ms. t_l indica el tiempo transcurrido desde la l-ésima vez que fue activado el chunk ms (medida en segundos). c es una constante de amplitud, y d es una tasa de degradación. Los valores por defecto de los parámetros son: $c = 1$, $d = 0.7$, y $iANB_j^{ms}$ es el valor inicial de ANB_j^{ms}.

Análogamente, cada regla asociativa tiene una ANB similar a la definida para las reglas simbólicas de los componentes RP y RES en el módulo Comportamental, determinada por la siguiente ecuación:

$$ANB_j^{ra} = iANB_j^{ra} + c \times \sum_{l=1}^{n} t_l^{-d} \tag{8.6}$$

donde: t_l es la l-ésima vez que se emplea la regla asociativa ra. t_l indica el tiempo transcurrido desde la l-ésima vez que fue empleada la regla ra (medida en segundos). Los valores por defecto de los parámetros son: $c = 1$, $d = 0.7$. $iANB_j^{ra}$ es el valor inicial de ANB_j^{ra}.

8.2.2.2. Memoria Asociativa

Situadas en el nivel emergente del módulo Declarativo, y relacionadas con la MS en el nivel cognitivista, se encuentran las Redes de Memorias Asociativas (RMA) o memorias de direccionamiento por contenido. Se entiende por memoria asociativa, el almacenamiento y recuperación de información por asociación con otras informaciones. En relación con este tipo de habilidades, se dice que la mente humana es asociativa dado que podemos recordar, mediante asociaciones, lo que hemos aprendido: objetos, seres vivos, conceptos e ideas abstractas, etc., a pesar de que éstas puedan estar afectadas por el "ruido" (como en el caso de los rostros con lentes o las melodías incompletas).

Computacionalmente, este tipo de memorias son bien representadas por redes neuronales artificiales, específicamente las redes de Hopfield y las redes retropropagación (backpropagation). Rumelhart (Rumelhart y McClelland, 1986), ha aportado evidencia suficiente para demostrar que las redes de retropropagación resultan ser útiles para implementar este tipo de memoria. La red RMA propuesta para el módulo Declarativo, resulta similar en su arquitectura con la red neuronal Perceptrón Multicapa con Retropropagación propuesta para el módulo Comportamental. No obstante, a diferencia de la red PMR, el modelo conexionista de la RMA almacena conocimiento asociativo no orientado a la producción de acciones. El modelo matemático y computacional para la RMA es similar al descrito para la red PMR con aprendizaje por retropropagación (sección 7.3.2.1), motivo por el cual no se detallará acá.

Para la RMA, se proponen dos tipos de entrenamiento que conducen a la formación de nuevas asociaciones implícitas. En el primer tipo de entrenamiento, un conjunto de pautas del tipo $\langle$`propiedad, valor`$\rangle$ es presentado a la red de manera auto-asociativa:

tanto las entradas como las salidas deseadas son de la misma naturaleza[3]. Luego de varios ciclos de entrenamiento, la red puede crear un representación interna (en la capa oculta) de tales conjuntos de pautas, y por ende, puede ser capaz de extraer las tendencias principales de dichas entradas. Uno de los propósitos del uso de este tipo de memorias en METÁFORA, es proveer al agente de un mecanismo para razonar sobre situaciones bajo incertidumbre o con información incompleta. Por ejemplo, cuando se presenta a la RMA información parcial acerca del estado observado por el agente, la red es capaz de "predecir" el estado más probable basándose en dicha información parcial.

En el otro tipo de entrenamiento, la red es capaz de aprender a crear hetero-asociaciones entre patrones. En este caso, el conjunto de pautas ⟨`propiedad, valor`⟩ de entrada es de naturaleza diferente al conjunto de pautas de salida con las que se entrena la red. Después de varios ciclos de entrenamiento, la red es capaz de generar asociaciones entre una pauta de entrada y otra de salida. Este tipo de aprendizaje resulta útil, cuando se establecen conexiones entre hechos (e.g., cuando se relaciona la información climática de una región con sus productos agrícolas, o cuando se relaciona una persona con sus intereses, hobbies, estudios, etc.).

La RMA sirve además como un localizador de conocimiento que ha sido empleado repetida y frecuentemente. El emparejamiento continuo de una entrada y una salida determinada, provocará que el razonamiento sobre cierto conocimiento se ejecute de manera casi automática. Por ejemplo, esto puede aplicarse al reconocimiento de palabras: un lector experimentado no necesita pensar conscientemente acerca de cada palabra en una oración y recordar qué es.

8.2.2.3. Mecanismo de Razonamiento

El *Razonador* del módulo Declarativo, se encarga de inferir conocimiento semántico acerca del estado percibido por el agente, a través de la dispersión de activación entre las reglas asociativas de la MS.

El proceso de razonamiento inicia, cuando el módulo de Percepción envía al módulo Declarativo un conjunto de pautas ⟨`propiedad, valor`⟩ que representan los estímulos percibidos por el agente sobre su entorno. Estas pautas producen la activación de un conjunto de chunks en la MS (combinando la activación producida en la RMA a través del proceso ascendente junto con la activación externa que recibe directamente la MS, es decir, la intensidad combinada del chunk, como se explicó en la sección 8.2.2.1).

Una vez se han activado los chunks de la MS, se procede a activar todas aquellas reglas asociativas que sean aplicables, esto es, no hay competición ni proceso de selección entre dichas reglas. Como resultado, se produce un nuevo conjunto de chunks inferidos. Estos nuevos chunks activan nuevos patrones en la RMA a través del proceso de activación descendente. Una vez se han integrado los dos niveles de representación, se procede a repetir el procedimiento un número determinado de iteraciones o hasta que no se puedan obtener nuevas conclusiones (chunks) del proceso de inferencia.

[3]Esto quiere decir, por ejemplo, que si se presentan como entradas a la red las propiedades de *color* y *forma* de un objeto determinado, las salidas estarán asociadas también a propiedades de *color* y *forma*, aunque posiblemente con valores diferentes.

El proceso de razonamiento inductivo emplea dos medidas: la activación de nivel base (ANB) y el Soporte de Regla. la ANB resulta útil para determinar los tiempos de respuesta mientras que el Soporte de Regla es quien propaga la activación del chunk condición al chunk conclusión. El chunk conclusión es activado si su intensidad sobrepasa un umbral de activación $umbral_{asoc}$ que por defecto es 0.8 (el nivel máximo de intensidad para un chunk es 1.0). El Soporte de Regla es calculado para cada regla asociativa de la siguiente manera:

$$S_j^a = \sum_i I_i^c \times W_i^a \tag{8.7}$$

donde: j indica la j-ésima regla en la MS. S_j^a es el soporte de regla para j. I_i^c es el nivel de intensidad del i-ésimo chunk de la condición de la regla. i representa todos los chunks en la condición de la regla j. W_i^a es el peso del i-ésimo chunk de la condición de la regla j (que por defecto es $1/n$, donde n es el número de chunks de la condición de la regla).

El chunk conclusión de cada regla asociativa tiene un nivel de intensidad que es determinado por el soporte de todas las reglas relevantes que influyen en su activación:

$$I_{ck}^{c,a} = \max_{j \to ck} I_j^a \tag{8.8}$$

donde: $I_{ck}^{c,a}$ es la intensidad del chunk conclusión ck. $j \to ck$ representa todas las reglas j que conducen a la activación de ck.

En el cuadro 8.1 se presenta el algoritmo para el proceso de razonamiento semántico.

Método: RazonarMemoriaSemantica

```
1.  Variables
2.     chunks          ↑↓ conjunto de chunks semánticos.
3.     rma             ↑↓ Red de Memoria Asociativa del nivel emergente.
4.  Inicio Procedimiento
5.     Mientras que se encuentren nuevos chunks conclusión o se
       alcance un criterio de paro hacer:
6.        procesarActivacionRMA(rma) //ejecuta un paso hacia adelante de la red
          neuronal semántica RMA.
7.        calcularIntensidadCombinada(chunks, rma)
8.        Para cada chunk ''ck'' activado en la MS hacer:
9.           S ← calcularSoporteRegla(ck_condicion)
10.          I_ck^{c,a} ← calcularIntensidadChunkConclusion(ck_condicion, S)
11.          chunks ← inferirNuevoChunk(ck_conclusion, I_ck^{c,a}) //valida si se
             cumple que I_ck^{c,a} > umbral_asoc.
12.       Fin Para
13.    Fin Mientras
14. Fin Procedimiento
```

Algoritmo 8.1: Mecanismo de Razonamiento Semántico.

En la figura 8.1 se presenta un ejemplo del mecanismo de razonamiento semántico. En el nivel emergente, el estado percibido que entra a la RMA activa la pauta ⟨objeto, mesa⟩. Esta activación es enviada al nivel cognitivista a través del proceso de integración ascendente donde es luego activado el chunk *Mesa* (integrando también la activación

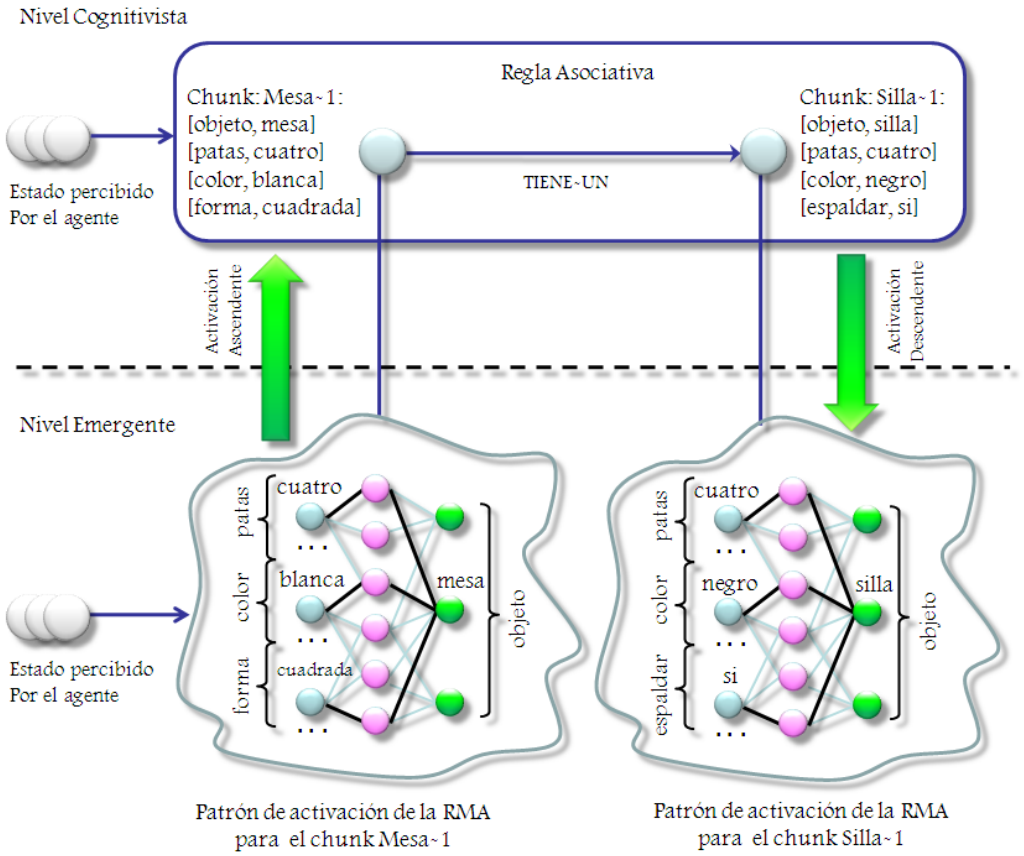


Figura 8.1: Mecanismo de Razonamiento Semántico.

externa que recibe el chunk de la MS por parte del estado percibido). Aplicando el *mecanismo de razonamiento*, se infiere que el chunk *Mesa* está relacionado con el chunk *Silla* a través de una relación semántica de agregación. Esta activación del chunk *Silla* en el nivel cognitivista causa la activación del patrón neuronal para el mismo objeto en el nivel emergente (i.e., en la RMA), lo cual es coordinado por el proceso de integración descendente.

8.2.3. Conocimiento Autobiográfico

8.2.3.1. Memoria Episódica

La Memoria Episódica (ME) es una memoria especial del módulo Declarativo que almacena las experiencias recientes del agente de diferentes formas. Principalmente, la ME almacena experiencias orientadas a la ejecución de acciones en forma de *componentes* episódicos que encapsulan un conjunto de estímulos, respuestas y sus respectivas consecuencias, junto con un registro de tiempo que indica cuándo acontecieron tales eventos. Adicionalmente, la ME puede almacenar conocimiento no orientado a acciones en forma

de chunks y reglas asociativas (similar a cómo lo hace la MS), con la diferencia de que se incluye además un registro de tiempo.

En el modelo propuesto, se emplearán dos tipos de chunks episódicos: (1) cada estado observado por el agente es almacenado en la ME como un chunk y es etiquetado con un registro de tiempo del momento en que ha ocurrido, y (2) cada paso ejecutado por el agente, entendido como un todo (i.e, conformado por la siguiente tupla: el estado observado, la acción ejecutada en dicho estado, el próximo estado, y el refuerzo inmediato obtenido por la ejecución de la acción, ⟨`estado, acción, próximo-estado, refuerzo`⟩), es almacenado en la ME como un chunk junto con un registro de tiempo del momento en que ha ocurrido. Cabe destacar que estos chunks están relacionados con experiencias anteriores que ocurren en una dimensión temporal (pautas espaciotemporales), lo que por naturaleza los hace episódicos.

Como consecuencia, la activación de (1) cada chunk y regla de acción (producida por el módulo Comportamental), (2) cada percepto almacenado en la MT[4], y (3) cada nueva asociación inferida (producida por el *Razonador* de la memoria semántica), puede generar un nuevo *componente* episódico en la ME. Esto es, cada vez que una entidad (i.e., un percepto, un chunk, o una regla) sea "activada" con un cierto nivel de intensidad (que sobrepase un umbral $umbral_{me}$), se procede inmediatamente a construir un *componente* episódico con la misma estructura de la entidad pero con un registro de tiempo agregado. Adicionalmente, el componente episódico puede agregar información sobre el estado interno del agente, concretamente acerca del estado de sus *motivadores*[5], sólo si estos no se encuentran en un equilibrio homeostático[6].

La información en la ME es filtrada de acuerdo con un valor que indica cuán novedosa es ésta. Para esto, se emplea un valor de activación de nivel base (ANB) asociado a cada *componente* de la ME. El mecanismo para calcular la ANB para cada *componente* es el mismo que el empleado para los otros tipos de chunks, excepto por sus parámetros que pueden variar. Asimismo, la ANB decaerá gradualmente con el transcurso del tiempo. Eventualmente, cuando la ANB de un *componente* de la ME cae por debajo de cierto umbral ($umbral_{elim}$), el *componente* es eliminado de la ME. La recuperación del conocimiento episódico consiste en hacer activos todos aquellos *componentes* que tengan una ANB que sobrepase el umbral de activación $umbral_{acti}$. Se tiene entonces que:

$$umbral_{acti} > umbral_{elim} \tag{8.9}$$

donde: $umbral_{acti} = 0.7$ y $umbral_{elim} = 0.3$ como valores por defecto.

Los *componentes* en la ME no están sujetos a un parámetro de densidad como los chunks y reglas asociativas en la MS, debido a que su "duración" en el módulo Declara-

[4]En la extracción de conocimiento de la ME se emplean aquellos perceptos que se encuentran almacenados y activos en la MT como resultado del proceso de interpretación de estímulos llevado a cabo durante los ciclos cognitivos previos. Cada percepto es tratado como un chunk que encapsula un conjunto de pautas ⟨`propiedad, valor`⟩.

[5]Esto debido a que el modelo motivacional de METÁFORA está basado en el diseño de motivadores (drives). En un modelo extendido de METÁFORA podría incluirse otros elementos como las emociones, estados de ánimo, gestos corporales, etc.

[6]Esto es, sólo si la intensidad del motivador es lo suficientemente relevante como para influir en la situación actual del agente. Para más detalles ver la sección 8.4

tivo está determinada sólo por su ANB. En el momento en que un *componente* de la ME alcance una ANB igual a "1" es almacenado en la memoria de largo plazo, esto es, que no se verá afectado posteriormente por el mecanismo de eliminación. Desde un punto de vista cognitivo, esto implica, por un lado, una *Memoria Episódica Transitoria* que almacena eventos relevantes para el agente pero sólo durante cierto intervalo de tiempo, el cual es determinado por la degradación de la ABN del *componente*[7]; y, por otro, representa una *Memoria Episódica de Largo Plazo* la cual consolida el conocimiento episódico transitorio que satisface cierto requisito de relevancia para el agente (ANB = 1), y que por tanto debe ser almacenado indefinidamente.

Parte de esta memoria puede ser usada para soportar el aprendizaje en el nivel Comportamental, debido a que provee escenarios adicionales para el entrenamiento del agente. El conocimiento episódico almacenado en *componentes* de la forma ⟨`estado, acción, próximo-estado, refuerzo`⟩, puede ser empleado para simular experiencias ante las que se podría encontrar el agente. Esto, por supuesto, permite el aprendizaje y refinamiento de nuevas habilidades (*Comportamientos*) en el agente. La justificación cognitiva de lo anterior es que la ME sirve como un proceso de consolidación de conocimiento que mejora el desempeño del aprendizaje procedimental (McClelland, 1995).

En la figura 8.2 se presenta un ejemplo de construcción de componentes episódicos. Supóngase que el agente en un estado inicial de tiempo (Instante 0) observa una manzana que se encuentra sobre una mesa, y como consecuencia decide aproximarse a ésta (suponiendo que esta acción satisface una de las metas del agente, como por ejemplo "alimentarse"). En ese instante de tiempo, se crea un *componente* episódico (Componente Episódico 1) que almacena el estado observado así como la acción tomada por el agente, junto con un registro de tiempo.

En un instante de tiempo posterior (Instante 2), luego de haberse aproximado lo suficiente a la mesa, el agente percibe que sobre la manzana se encuentra otra fruta, una pera. Para poder coger la manzana tiene primero que retirar la pera de encima, acción que es tomada por alguno de los *Comportamientos* del agente. Una vez construido el *componente* episódico que representa la situación actual (Componente Episódico 2), el módulo Declarativo actualiza la información del *componente* creado en el instante inmediatamente anterior, asignándole un valor al refuerzo (que en este caso es un refuerzo positivo por la acción ejecutada de aproximarse hasta la manzana) y conectando el atributo "próximo estado" del *componente episódico 1* con el estado actual observado (Estado 2) que forma parte del *Componente Episódico 2*. El proceso continúa de manera sucesiva, construyendo una secuencia histórica de componentes episódicos conectados a través de relaciones espacio-temporales.

Estos *componentes* del ejemplo conformarían los nodos de la ME transitoria. No obstante, si la experiencia resultara ser lo suficientemente relevante para el agente (i.e, si los chunks que conforman cada *componente* lograran acumular una ABN igual a 1), el componente sería almacenado como un nodo en la ME de largo plazo.

[7] Según Conway (Conway, 2001), la tasa de degradación de las unidades almacenadas en la Memoria Episódica Transitoria en los humanos, puede medirse en términos de horas o incluso días.

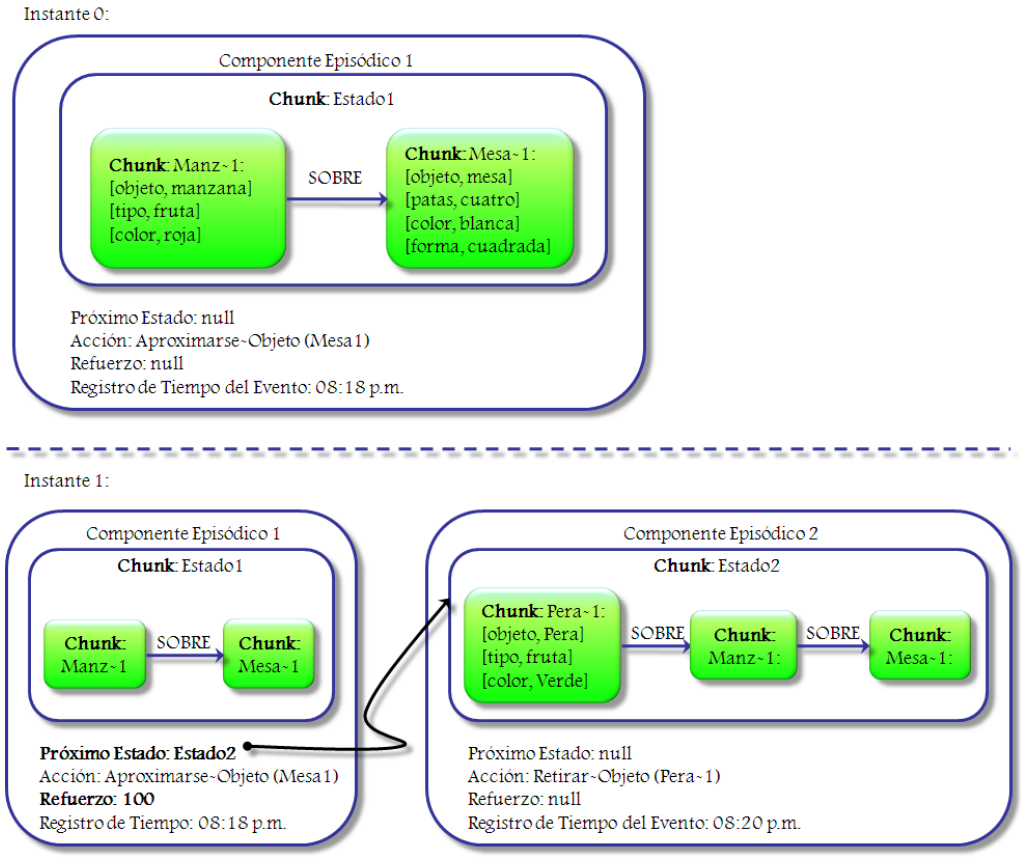


Figura 8.2: Construcción de Componentes Episódicos.

8.2.3.2. Memoria Episódica Abstracta

La representación correspondiente de la ME en el nivel emergente es la Memoria Episódica Abstracta (MEA). A diferencia de la ME que almacena experiencias específicas, la MEA compendia información respecto a los eventos generales experimentados en el pasado por el agente, de tal manera que puedan ser fácilmente accesibles sin la necesidad de examinar los detalles de cada paso individual.

La MEA es implementada a través de una Red neuronal de Frecuencia de Estado (RFE). Esta RFE correlaciona parejas ⟨`estado, acción`⟩ con una distribución de frecuencia de estados sucesivos, así como con un distribución de frecuencia del refuerzo inmediato. En este tipo de red neuronal, los estados no son codificados en parejas del tipo ⟨`propiedad, valor`⟩ (como se ha venido haciendo en otros módulos de la arquitectura), sino que se propone una representación "unitaria" con el propósito de que las redes produzcan salidas de frecuencias de transición de estados: cada estado es representado por un nodo individual (tanto en la capa de entrada como en la capa de salida de la red) y las conexiones sinápticas de la red representan las transiciones entre los mismos. En la medida que estos estados son experimentados por el agente, se identifican uno a uno y se representan en cada una de las redes como nodos unitarios de información general.

De manera análoga, las acciones también deber ser codificadas empleando una representación unitaria por la misma razón que los estados[8]. Las acciones son identificadas una a una en la medida que son desempeñadas por el agente. La señal de refuerzo es distribuida entre cierto número de intervalos, donde cada intervalo es codificado de manera individual. De esta manera, se obtiene una distribución de frecuencia del refuerzo.

Las redes neuronales RFE son redes de retropropagación que son entrenadas con un algoritmo alternativo de aprendizaje: el algoritmo de aprendizaje por retropropagación probabilística (Bartlett y Downs, 1992), el cual resulta más apropiado para el aprendizaje de distribuciones de frecuencias. El entrenamiento de la red se basa en datos episódicos de la ME, como se describe en la sección 8.2.3.3.

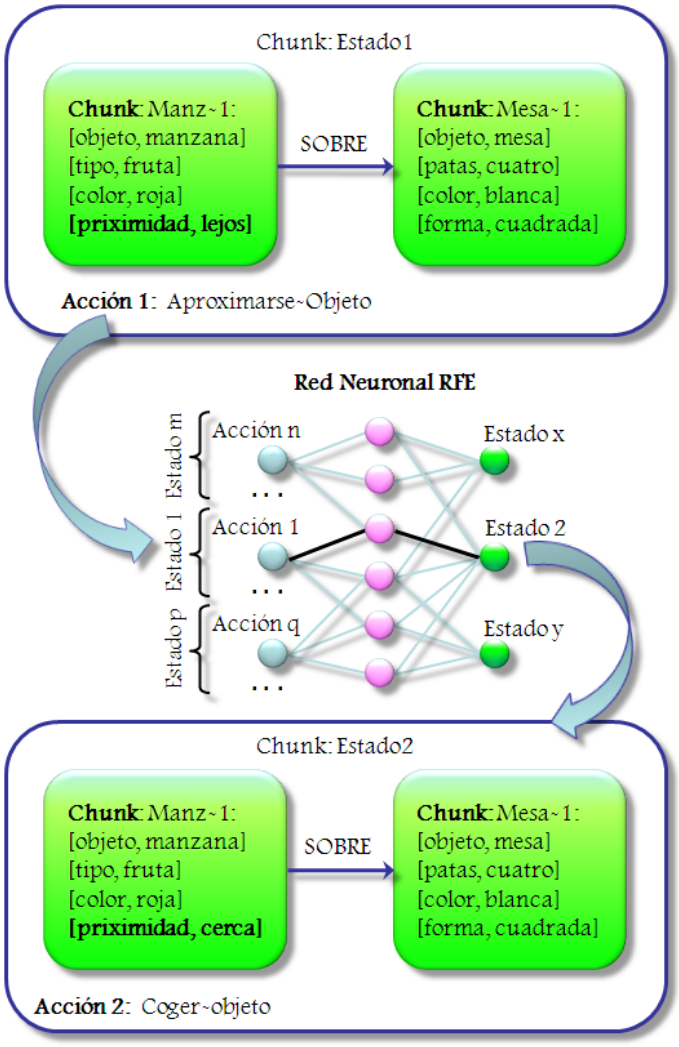


Figura 8.3: Memoria Episódica Abstracta.

En la figura 8.3, se presenta un ejemplo para una RFE. Supóngase dos estados observados por el agente los cuales se encuentran almacenados en forma de chunks en la

[8]El número máximo de estados y acciones requiere ser especificado a priori.

ME del nivel cognitivista (Estado 1 y Estado 2). En el Estado 1, el agente percibe una manzana sobre una mesa, la cual se encuentra lejos del alcance de éste; y en el Estado 2 el agente percibe la misma manzana pero ahora a su alcance, luego de haber ejecutado la acción de aproximarse a la mesa (Acción 1). La RFE es entrenada de tal manera que las pautas ⟨`estado, acción`⟩ representen los nodos de la capa de entrada de la red, y los nodos de la capa de salida representen los estados siguientes. De este modo, cuando se cumple que ⟨`Estado1, Acción1`⟩, la RFE activa como respuesta el estado ⟨`Estado2`⟩. Luego, el ⟨`Estado2`⟩, junto con alguna acción ejecutada por el agente, estimulará otro patrón de activación de la red que producirá como respuesta otro estado diferente percibido por el agente. Como consecuencia, es posible generar una secuencia de estados que se conectan entre sí a través de las acciones ejecutadas por el agente.

Al igual que la ME, la MEA puede ser usada para generar experiencias simuladas en las cuales se entrenan los *Comportamientos* del módulo Comportamental. El proceso consiste primero en seleccionar aleatoriamente un estado y, luego, en muestrear la distribución de acciones, la distribución de estados sucesivos, y la distribución del refuerzo basada en el estado actual. La experiencia simulada generada (una tupla de la forma ⟨`estado, acción, próximo-estado, refuerzo`⟩) puede ser usada para entrenar el módulo Comportamental, como si ésta fuese una experiencia real percibida por el agente.

Esta memoria puede ser usada para otros propósitos más complejos. Por ejemplo, puede utilizarse para mejorar el proceso de aprendizaje guiado por *Q-learning* (ver sección 8.2.3.3), o durante el proceso de extracción de planes y simulación interna (descrito en la sección 10.3).

8.2.3.3. Aprendizaje Episódico

Específicamente, para las redes RFE de la Memoria Episódica se emplea una red neuronal de dos capas (sin capa oculta de procesamiento). La regla de aprendizaje acumulativo fuera de línea (i.e, cuando el aprendizaje se efectúa como un proceso subconsciente) es (Bartlett y Downs, 1992):

$$\Delta w_{jk} = \alpha \sum_{i} (d_j(x_i) - h_j(x_i)) \times x_{jk}(x_i) \tag{8.10}$$

donde: i es el número de neuronas en la capa de entrada. x_i es la entrada a la red. $d_j(x_i)$ es la salida deseada para x_i. $h_j(x_i)$ es la salida actual (que representa la frecuencia). $x_{jk}(x_i)$ es la k-ésima entrada conectada con la j-ésima unidad de salida desde x_i. α es una tasa de aprendizaje. Δw_{jk} son los pesos actualizados entre las neuronas de la capa de entrada y la capa de salida.

La versión en línea del aprendizaje (i.e, cuando el aprendizaje se efectúa como un proceso consciente) es (Bartlett y Downs, 1992):

$$\Delta w_{jk} = \alpha (d_j(x_i) - h_j(x_i)) x_{jk}(x_i) \tag{8.11}$$

Las redes RFE pueden ser entrenadas empleando chunks de la ME (tuplas del tipo ⟨`estado, acción, próximo-estado, refuerzo`⟩), los cuales proveen: la

entrada x_i y la salida deseada $d_j(x_i)$ requeridas para el entrenamiento. Adicionalmente, por cada chunk se define un valor que determina la probabilidad de emplear dicho chunk durante el entrenamiento de la MEA. Luego, en cada paso, se escoge aleatoriamente un número específico de chunks (de la ME) y se entrena cada una de las redes RFE[9].

Por otro lado, cabe resaltar la función de la MEA en la mejora del desempeño del algoritmo *Q-learning* (específicamente, para el componente PMR del módulo Comportamental), para lo cual se utilizan las estimaciones de frecuencia generadas por cada red RFE. En el caso de emplearse el componente MEA, la ecuación para el algoritmo *Q-learning* puede ser reemplazada entonces por la siguiente:

$$\Delta Q_{x,a} = \alpha(r + \gamma \sum_{y} Pr(x, a, y) \times e(y) - Q(x, a)) \tag{8.12}$$

donde: $Pr(x, a, y)$ es la probabilidad de transición que va desde el estado actual (x, a) hasta el estado resultante y después de ejecutar la acción a (el cual es obtenido de la RFE). La sumatoria es realizada sobre todos los estados próximos posibles.

Existe evidencia que soporta que esta mejora sobre el algoritmo de *Q-learning* puede acelerar el aprendizaje y mejorar la convergencia (Barto et al., 1995; Bertsekas y Tsitsiklis, 1996; Sun et al., 2001).

8.2.4. Integración de los dos Niveles de Representación

Un chunk del nivel cognitivista puede ser "activado" tanto por una fuente interna (e.g., otro chunk que dispersa energía de activación a través de una regla asociativa) como por la activación ascendente producida desde el nivel emergente, con lo cual se requiere calcular una intensidad combinada que integre las dos fuentes de activación. En el caso de la activación ascendente (desde el nivel emergente hacia el nivel cognitivista) se activan aquellos valores de las propiedades de los chunks cuyos valores de las propiedades en la red neuronal estén activados con un nivel de intensidad igual a "1.0".

Para calcular la intensidad total de un chunk, se requiere combinar el resultado de activación de la red neuronal (a través de la activación ascendente) con el resultado de activación del nivel cognitivista (bien sea por aquellos chunks activados por una entrada externa o por los chunks resultantes del proceso de inferencia que actúa sobre las reglas asociativas). En este caso, se emplea la función de maximización:

$$I_i^c = \text{máx}(I_i^{c,nc}, I_i^{c,ne}) \tag{8.13}$$

donde: $I_i^{c,ne}$ es la intensidad de activación ascendente producida por el nivel emergente. $I_i^{c,nc}$ es la intensidad del chunk i en el nivel cognitivista (que es la intensidad total producida por una activación externa o como resultado de la activación de una regla asociativa). I_i^c es la intensidad final combinada del chunk i.

[9] En el diseño de la concreción contextual de la arquitectura, es posible definir una red REF por cada tipo de tarea que desempeñará el agente, o por cada sub-dominio específico de conocimiento, o por cada *Comportamiento* del módulo Comportamental.

En el caso de que la activación se produzca en el nivel cognitivista pero no en el nivel emergente, se genera una activación descendente, donde los valores de las propiedades de los chunks activados por el nivel cognitivista causan la activación de los valores en la capa de salida de las redes neuronales en el nivel emergente. En este caso, los valores de los chunks conclusión activados sirven como "expertos" que guían, a través de un aprendizaje supervisado (aprendizaje por retropropagación), el entrenamiento de la red neuronal hasta que ésta logre "aprender" los nuevos patrones de activación (de manera similar al proceso descrito en la sección 7.8).

8.2.5. Aprendizaje Vertical

Los mecanismos de aprendizaje expuestos en esta sección son similares a los descritos para el módulo Comportamental en el capítulo 7, razón por la cual no se hará nuevamente una explicación detallada de tales procesos.

Los dos tipos de aprendizaje vertical del módulo Declarativo son: el *Aprendizaje Descendente*, y el *Aprendizaje Ascendente*. En el Aprendizaje Descendente, el conocimiento en el nivel cognitivista representado de forma explícita, es asimilado en el nivel emergente como nuevo conocimiento implícito. Este proceso es realizado de forma similar al descrito en el módulo Comportamental (ver sección 7.8).

Por otro lado, el Aprendizaje Ascendente se encarga de extraer conocimiento implícito del nivel emergente (RMA y MEA) que luego es consolidado en el nivel cognitivista (MS y ME) como chunks de representación explícita. Este proceso también resulta similar al descrito en el módulo Comportamental en el componente de Extracción de Reglas Sub-simbólicas (ver sección 7.5.2).

8.3. Activación de Comportamientos Automáticos

Según la metáfora del Teatro de la Consciencia de Baars, gran parte de los procesos subconscientes consisten en tareas automáticas complejas, que son accionadas por eventos desencadenantes conscientes. Esto quiere decir, que los eventos en el foco de atención (los *Comportamientos* activos) pueden disparar automáticamente eventos complejos que tienen lugar en la audiencia (procesos subconscientes).

En el capítulo 7 se habló de los *Comportamientos* como componentes híbridos que dirigen la actuación consciente del agente. No obstante, estos *Comportamientos* pueden además activarse como procesos subconscientes – automatismos[10] – que desempeñan tareas automáticas (rutinas) las cuales han sido previamente aprendidas y entrenadas por el agente.

Por ejemplo, supóngase que el agente debe buscar una dirección de un lugar en una ciudad determinada, proceso que lleva a cabo de forma consciente a través del *Comportamiento* ⟨`buscar-dirección`⟩. Sin embargo, mientras el agente cumple con esta ta-

[10] Se define por Automatismo un proceso que consume un mínimo de recursos atencionales, el cual puede ocurrir en paralelo con otro proceso cognitivo. Esto implica que el automatismo no está restringido al actual foco de atención.

rea, existen otros *Comportamientos* que ejecuta paralelamente de manera subconsciente, los cuales son accionados por el mecanismo de atención: ⟨`mantener-equilibrio`⟩, ⟨`caminar`⟩, ⟨`subir-escaleras`⟩, etc.

A continuación, se enuncian las consideraciones necesarias con respecto a la activación de los *Comportamientos* automáticos:

- Los *Comportamientos* automáticos pueden ejecutarse paralelamente a la ejecución de un *Comportamiento* consciente, sólo si no se comparten recursos. Esto es, el conjunto de actuadores definidos por cada *Comportamiento subconsciente* debe ser totalmente diferente al conjunto de actuadores definido por el *Comportamiento consciente* en ejecución. Se tiene entonces que:

$$actuadores_i \cap actuadores_j = 0 \tag{8.14}$$

  donde: i es el comportamiento que se ejecuta de forma consciente.
  j es el comportamiento que se ejecuta de forma subconsciente.

  Esta consideración resulta útil, por ejemplo, en el caso de que los actuadores del *Comportamiento* consciente estén controlando el movimiento de las piernas del agente a fin de producir un desplazamiento (caminar), y se desee evitar un conflicto con cualquier otro *Comportamiento* subconsciente que produzca algún tipo de control sobre el mismo conjunto de actuadores (saltar, acuclillarse, etc.).

- En el caso de que uno o más *Comportamientos* automáticos presenten conflicto al tratar de controlar algún actuador en común con el *Comportamiento* consciente ($actuadores_i \cap actuadores_j \neq 0$), estos no serán activados de manera concurrente con el *Comportamiento* consciente aunque sí participarán en el proceso de formación de coaliciones y competencia por el foco de atención que se lleva a cabo en la zona subconsciente de la Memoria de Trabajo (i.e, en el área de la MT que no está siendo iluminada por el foco de atención selectiva, proceso que se detallará en la sección 9.2).

- Los *Comportamientos* automáticos sólo ejecutarán alguno de los dos componentes de naturaleza emergente (la RNA o el SIA) sin llevar a cabo la integración de acciones. Esto es debido a que los componentes RNA y SIA resultan más apropiados para tratar situaciones donde la recuperación de conocimiento y la actuación se hace sin esfuerzo consciente, es decir, de manera implícita (como se discutió en la sección 7.1).

8.4. Regulación Homeostática de los Motivadores

8.4.1. Inspiración del Modelo basado en Motivadores

Esta sección describe los componentes básicos del módulo Motivacional de METÁFORA encargados de dirigir el proceso de activación de metas, la activación de *Comportamientos*, y la generación de la señal de retroalimentación interna empleada en la modulación

del aprendizaje. En este punto, cabe enfatizar que el modelo motivacional propuesto corresponde con un modelo inspirado en *motivadores*[11], los cuales soportan el proceso de activación de metas y comportamientos en METÁFORA. No obstante, este modelo puede ser extendido en trabajos futuros, a uno más complejo que incluya una valoración de las emociones, los estados de ánimo, la expresión corporal y facial, los rasgos de personalidad, etc., del agente.

Algunos teóricos en el campo de estudio de las emociones, argumentan que un conjunto selecto de emociones primarias o motivadores han sido otorgados a través del proceso evolutivo, debido a que prueban la habilidad que posee una criatura para generar respuestas adaptativas ante el vasto rango de demandas y oportunidades que debe afrontar en su vida cotidiana (Izard, 1998; Ekman, 1992). Las emociones tales como la ira, el fastidio, el miedo, la alegría, la tristeza, y la sorpresa, son con frecuencia definidas como primarias (o motivadores) desde las perspectivas evolutiva, conductual, y cultural (Ekman y Oster, 1982). Cada motivador, además de servir como una función en particular (con frecuencia biológica o social) que prepara y motiva a la criatura para responder de forma adaptativa, sirve como un mecanismo de refuerzo importante para el aprendizaje de nuevo comportamiento (ver sección 10.2).

El concepto en sí de motivador es entendido pues, como un factor que ayuda a la diferenciación de las reacciones emocionales, consiguiendo que distintos individuos manifiesten reacciones emocionales diversas en situaciones similares, en función de sus motivaciones variables. Grosso modo, el conjunto de motivadores representan los estímulos que incitan la actuación del agente. Cada uno de estos motivadores tiene asociado uno o varios controladores que filtran la percepción procedente del entorno, ajustando el valor de ciertas variables controladas, asociadas al motivador.

Si el valor de estas variables difiere del deseado para las mismas, se emite una señal de alarma que llega a los motivadores. A partir de la combinación de todas las señales que alarman a cada motivador acerca de posibles perjuicios para sus objetivos, el agente selecciona la meta más adecuada a perseguir (y como consecuencia, activa los *Comportamientos* asociados a dicha meta), de manera que se neutralicen o minimicen los efectos negativos de lo percibido sobre sus variables controladas. Además, la producción de estos *Comportamientos* tiene un efecto de retroalimentación directo sobre las variables controladas de los motivadores.

Así pues, el diseño de los motivadores en METÁFORA se inspira en las perspectivas etológicas de los procesos análogos observados en los animales (McFarland y Bosser, 1993; Breazeal, 2000), aunque el modelo propuesto, es una versión simplificada e idealizada de tales procesos.

8.4.2. Mecanismo de Regulación de un Motivador

El mecanismo de regulación de los motivadores es dirigido por un proceso de naturaleza homeostática, es decir, el comportamiento del agente deberá orientar continuamente su

[11] Algunos otros autores hacen referencia al mismo concepto pero empleando términos como emociones primarias o básicas (Plutchik, 1991; Izard, 1998; Ekman, 1992), o motivadores (drives) (Frijda, 1994; Breazeal, 2000; Dennett, 1987).

actuación hacia la conservación de los niveles de intensidad de cada motivador dentro de una zona de equilibrio o reposo. El cambio de intensidad en el motivador refleja las necesidades actuales del agente y la urgencia que tiene de suplirlas. Cada motivador define un nivel deseado y los límites aceptables de operación que se pueden realizar alrededor de dicho nivel, conformando así un rango que se denominará la *región homeostática* del motivador. Mientras que un motivador determinado se encuentre fuera de su región homeostática surgirá una necesidad que deberá ser atendida por el agente. En METÁFORA, el mantenimiento de cada uno de los motivadores dentro de su región homeostática es un proceso oscilante que se lleva a cabo de manera indefinida, y por tanto, el comportamiento global del agente se enfocará hacia la auto-regulación continua de los motivadores cuya intensidad se encuentra fuera de la región homeostática, y hacia el cumplimiento de las *metas* que se disparan como consecuencia de la saturación o la carencia de estimulación de tales motivadores.

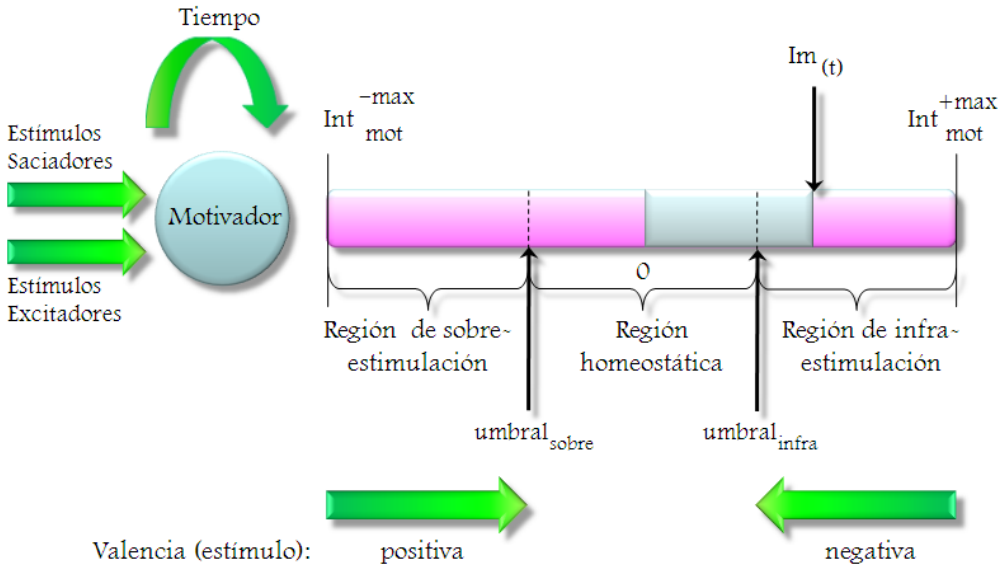


Figura 8.4: Modelo de Regulación Homeostática de un Motivador.

Cada motivador en METÁFORA es modelado como una estructura compuesta de tres regiones, como se presenta en la figura 8.4. Adicionalmente, cada motivador tiene asociadas tres clases de entradas las cuales alimentan su comportamiento oscilante: una entrada temporal, un conjunto de estímulos excitadores, y un conjunto de estímulos saciadores. Esto implica que la intensidad de un motivador puede incrementar con el paso del tiempo, con la presencia de estímulos excitadores, y con la ausencia de estímulos saciadores; o por el contrario, puede disminuir con la ausencia de estímulos excitadores y con la presencia de estímulos saciadores.[12].

La intensidad de un motivador puede variar dentro del intervalo $[Int_{mot}^{-max}, Int_{mot}^{+max}]$, cuyos valores límite corresponden por defecto a -100 y 100 respectivamente. Si la intensidad actual del motivador ($Im_{(t)}$) se encuentra dentro del intervalo $[umbral_{sobre}$,

[12]Este comportamiento oscilatorio de los motivadores es análogo, por ejemplo, a los motivadores de hambre y fatiga presentes en los animales (Breazeal, 2003).

$umbral_{infra}$], entonces el motivador se halla en la región homeostática; si se encuentra dentro del intervalo ($umbral_{infra}$, Int_{mot}^{+max}] el motivador se halla en la región de *infra-estimulación*; y si se encuentra dentro del intervalo [Int_{mot}^{-max}, $umbral_{sobre}$) el motivador se halla en la región de *sobre-estimulación*. Por defecto, los umbrales [$umbral_{sobre}$ y $umbral_{infra}$] tienen un valor de -30 y 30 respectivamente.

En general, el motivador se mantiene dentro de su región homeostática cuando sus estímulos excitadores y saciadores se producen con una intensidad y frecuencia apropiadas. En ausencia de un estímulo saciador (o cuando su intensidad es demasiado baja), y/o en presencia de un estímulo excitador, la intensidad del motivador aumentará hasta sobrepasar el umbral positivo de la región homeostática ($umbral_{infra}$), momento en el que el motivador entrará en la región de infra-estimulación. Contrariamente, en ausencia del estímulo excitador y/o cuando se produce un estímulo saciador con cierta intensidad y frecuencia, la intensidad del motivador se reducirá hasta exceder el umbral negativo de la región homeostática ($umbral_{sobre}$), lo que llevará al motivador a una región de sobre-estimulación. Cuando el motivador se halla en la región homeostática refleja un estado *"placentero"* para el agente, mientras que si se encuentra en las regiones de infra-estimulación o sobre-estimulación se produce un efecto contrario, es decir, un estado *"desagradable"* para el agente. Por tanto, los procesos de regulación homeostática del motivador intentan simular, a pequeña escala, la *búsqueda del placer* y la *evasión del dolor* que bien caracterizan el comportamiento de los organismos biológicos.

Se tiene entonces que un motivador se compone de: (1) un conjunto de *estímulos excitadores* cuya presencia determina el aumento de la intensidad del motivador; (2) un conjunto de *estímulos saciadores* cuya presencia determina la disminución de la intensidad del motivador; (3) un conjunto de disparadores denominado *disparadoresRIE*, los cuales se accionan cuando el motivador se encuentra en la región de infra-estimulación; y (4) un conjunto de disparadores denominado *dis-paradoresRSE*, los cuales se accionan cuando el motivador se halla en la región de sobre-estimulación. Concretamente, un *estímulo saciador/excitador* se corresponde con un percepto específico asociado al motivador[13], mientras que un *disparador* representa una condición que se activa cuando se exceden los umbrales de infra-estimulación o sobre-estimulación, generando como consecuencia un *"estado interno"* del agente que es almacenado en la Memoria de Trabajo y que puede ser luego empleado en la activación de nuevas *metas* (como se discutirá en la sección 8.5). Formalmente, un motivador se conforma por una tupla como la siguiente:

$$motivador = \langle \texttt{EE, ES, DRIF, DRSE} \rangle \tag{8.15}$$

donde: EE es el conjunto de estímulos excitadores. ES es el conjunto de estímulos saciadores. $DRIF$ es el conjunto de disparadores que se accionan en la región de infra-estimulación. $DRSE$ es el conjunto de disparadores que se accionan en la región de sobre-estimulación.

En cuanto a la Intensidad de un motivador, ésta se ve afectada por dos factores:

[13]Estos perceptos pueden representar información tanto externa (ambiental) como interna (e.g., el nivel de desgaste físico en un animal/personaje virtual, o el nivel de energía en un agente robótico).

1. *La Intensidad del estímulo saciador/excitador*: este factor "variable" determina la magnitud del estímulo cuando es percibido. Esta magnitud, corresponde a una fracción del nivel de activación base (ANB) del percepto asociado al motivador.
2. *La Relevancia del estímulo*: este factor "constante" determina la importancia de un estímulo (saciador o excitador) en términos de un peso. Adicionalmente, la relevancia asocia una *"valencia"* al estímulo del motivador tal que, los estímulos excitadores poseen una valencia positiva (ya que tienden a aumentar la intensidad del motivador), mientras que los estímulos saciadores poseen una valencia negativa (debido a que estos tienden a disminuir la intensidad del motivador) .

A continuación, se presenta la ecuación que estima la intensidad del motivador i en el instante de tiempo t:

$$Im_{(i,t)} = \begin{cases} Im_{(i,t-1)} + k1, & \text{Si no se recibe ningún estímulo} \\ Im_{(i,t-1)} - \sum_{j}^{n} I_j^{es}(t) \times k2 \times R_j^{es}, & \text{De lo contrario} \end{cases} \tag{8.16}$$

donde: $Im_{(i,t)}$ y $Im_{(i,t-1)}$ son los niveles de intensidad del motivador i en los instantes de tiempo t y $t-1$. $k1$ es una constante positiva que incrementa la intensidad del motivador con el paso de los ciclos cognitivos y en ausencia de estímulos (por defecto $k1 = 0.5$). n es la cantidad total de estímulos asociados al motivador. $I_j^{es}(t)$ es la intensidad del j-ésimo estímulo saciador (correspondiente a la ANB del percepto j). $k2$ es una constante positiva que determina una fracción de la ANB de todos los estímulos saciadores (por defecto es 1). R_j^{es} es la relevancia del estímulo saciador que corresponde a un valor decimal entre -1.0 y 1.0.

A continuación se describe un ejemplo muy concreto que ilustra el mecanismo de auto-regulación homeóstatica de un motivador[14]. En la tabla 8.1 se esquematiza el ejemplo. Supóngase que el agente cognitivo es un animal virtual (animat) que explora su entorno en búsqueda de comida, y que los motivadores que debe satisfacer son el ⟨`Hambre`⟩, la ⟨`Fatiga`⟩, la ⟨`Curiosidad`⟩, etc. Por ahora, considérese únicamente el motivador ⟨`Hambre`⟩ que define: un estímulo excitador (⟨`Nivel-Energía, Bajo`⟩) que se percibe cuando los niveles de energía del animat están por debajo de cierto umbral, y un estímulo saciador que se percibe cuando la acción ejecutada por el agente es "comer" (⟨`Acción-Ejecutada, Comer`⟩). Las relevancias para cada uno de estos motivadores son $R_1^{es} = 0.5$ y $R_2^{es} = -0.5$ respectivamente.

Inicialmente, el animat dispone de reservas energéticas suficientes que le permiten moverse y explorar su entorno circundante. Con el paso del tiempo y debido a la energía consumida durante los desplazamientos, la reserva energética del animat empieza a agotarse gradualmente hasta que el agente percibe el estímulo excitador ⟨`Nivel--Energía, Bajo`⟩. Este estímulo empieza a ser más frecuente y de mayor magnitud hasta que la intensidad del motivador ⟨`Hambre`⟩ alcanza la región de infra-estimulación,

[14]Téngase en cuenta que este mismo procedimiento aplica independientemente para cada motivador definido para el agente, y considérese además que la descripción de las dinámicas relativas con la activación de metas, la atención, y la ejecución de *Comportamientos* se han omitido con el fin de simplificar el ejemplo.

Ciclo Cognitivo	Intensidad ($Im_{(i,t)}$)	Descripción	Estado Interno generado
0	0	Estímulo Excitador → RH	
50	20		
100	40		
150	60	Estímulo Excitador → RIE	⟨Estado-Hambre, Con-Hambre⟩
200	50	Estímulo Excitador - Estímulo Saciador → RH	
250	30		
300	10		
350	-10		
400	-30		
470	-55	Estímulo Saciador → RSE	⟨Estado-Hambre, Sin-Hambre⟩
550	-60	Paso del tiempo → RSE	⟨Estado-Hambre, Sin-Hambre⟩
600	-50	Paso del tiempo → RH	
650	-30		
870	60	Estímulo Excitador → RIE	⟨Estado-Hambre, Con-Hambre⟩

Tabla 8.1: Ejemplo de Regulación homeostática para el motivador Hambre. RH: Región Homeostática, RIE: Región de Infra-estimulación, y RSE: Región de Sobre-estimulación.

y como consecuencia, se dispara un nuevo estado interno del agente: ⟨Estado-Hambre, Con-Hambre⟩. A partir de ese momento (ciclo cognitivo 150), el generador de metas identifica una necesidad interna del agente que debe ser satisfecha por una nueva meta, con lo cual se genera la meta ⟨Recuperar-Nivel-Energía⟩. A continuación, el agente orquesta una serie de dinámicas de activación entre los *Comportamientos* hasta que finalmente se activa el *Comportamiento* ⟨Buscar-Comida⟩.

Una vez el agente encuentra comida se activa el *Comportamiento* ⟨Comer⟩, el cual hace que el nivel de energía del agente aumente progresivamente. Después de un rato, la intensidad del motivador empieza a situarse en la región homeostática debido a que la intensidad del estímulo excitador ⟨Nivel-Energía, Bajo⟩ disminuye en la medida que la energía del animat se acerca a un nivel deseado (entre los ciclos cognitivos 200 y 400). Paralelamente, el estímulo saciador ⟨Acción-Ejecutada, Comer⟩ incrementa su intensidad y frecuencia debido a la activación continua del *Comportamiento* ⟨Comer⟩, y como consecuencia la intensidad del motivador desciende hasta la región de sobre-estimulación donde es disparado el estado interno ⟨Estado-Hambre, Sin-Hambre⟩, indicando que el animat ha ingerido ya suficiente comida hasta la hartura (ciclo cognitivo 470).

El agente interrumpe el *Comportamiento* ⟨Comer⟩ para otorgar el control a otro *Comportamiento* (e.g., ⟨Explorar-Entorno⟩). En este momento no se percibe ningún tipo de estímulo, con lo cual la intensidad del motivador se incrementa paulatinamente con el paso del tiempo hasta alcanzar nuevamente la región homeostática (entre los ciclos cognitivos 600 y 650). Si el animat se ha mantenido activo durante este tiempo, su ni-

vel de energía se reducirá más rápidamente que si hubiese estado en reposo, alcanzando nuevamente la región de infra-estimulación y activando el estado interno ⟨`Estado-Hambre, Con-Hambre`⟩ (ciclo cognitivo 870). Así pues, se reanuda nuevamente el proceso oscilante de regulación.

Cabe notar por último, que el mecanismo que dispara los diferentes estados de un motivador (cuando su intensidad está fuera de la región homesóstatica) se corresponden con el proceso de acumulación de energía de activación descrito en la sección 6.3.3 sobre las dinámicas del sistema nervioso sintético.

8.5. Generación de la Estructura de Metas

8.5.1. Inspiración del Modelo de Metas

La *Estructura de Metas* (EM) es empleada para dirigir el proceso deliberativo del agente: le permite organizar sus acciones de una manera prioritaria, coherente y ordenada, en relación a sus objetivos y necesidades.

En términos de la metáfora del teatro de la consciencia, la EM desempeña una función ejecutiva que establece el contexto que guía los eventos conscientes acontecidos en el escenario (la MT). Según esta metáfora, la estructura de metas cumple la función análoga a las personas que se encuentran tras bambalinas (i.e., el director, el dramaturgo, el coreógrafo, etc.), quienes determinan hacia donde debe ser apuntado el reflector (i.e., el foco de atención selectiva) durante la puesta en escena. Tales funciones ejecutivas, se encuentran localizadas aparentemente en la corteza frontal del cerebro (Fuster et al., 2000).

La EM provee una manera de representar los diferentes objetivos que debe alcanzar el agente en el corto, medio y largo plazo, bajo la influencia del conjunto de *motivadores* activos en la Memoria de Trabajo. Concretamente, los motivadores disparan nuevos estados internos, los cuales satisfacen el conjunto de pre-condiciones necesarias para activarse una meta específica.

La idea práctica de la EM, es proporcionar un contexto a partir del cual se ejecuta el proceso de selección de *Comportamientos*. Los *Comportamientos* son unidades de procesamiento orientadas a la culminación de objetivos, motivo por el cual las metas son un elemento clave que ayuda a determinar el grado de cooperación y competencia entre todos los *Comportamientos* definidos para el agente. Por ejemplo, un *Comportamiento* puede cooperar con otro si existe cierta dependencia entre las metas que persigue cada uno (e.g., si el *Comportamiento* A persigue la meta $m1$, y el *Comportamiento* B persigue la sub-meta $m2$, entonces A cooperará con B a fin de lograr $m1$); por otro lado, un *Comportamiento* competirá contra otro si persiguen metas conflictivas o de propósito opuesto.

8.5.2. La Estructura de Metas

La EM contiene, básicamente, dos tipos de metas: un conjunto de *metas globales* y un conjunto de *metas específicas* (sub-metas). Las *metas globales* definen un tipo de objetivo a largo plazo que refleja el estado ideal del sistema. Por otro lado, las *metas específicas* definen objetivos al corto y medio plazo, que resultan ser una descomposición jerárquica de las metas globales.

Las *metas globales* pueden ser generadas a través de tres mecanismos diferentes: (1) pueden ser predefinidas durante el diseño de la concreción contextual de la arquitectura, (2) pueden ser recibidas de forma explícita por el agente desde una fuente externa (e.g., otro agente o un humano), y (3) pueden ser generadas como consecuencia de las dinámicas de regulación homeostática de uno o varios motivadores, como se explicó en el apartado 8.4. Por otro lado, las *metas específicas* pueden ser generadas por dos mecanismos: (1) por la influencia de los motivadores – como en las metas globales –, y (2) como un producto de las complejas dinámicas que lleva a cabo el mecanismo de orquestación de la consciencia (véase el capítulo 9), es decir, en forma de una post-condición que debe ser satisfecha después de la ejecución de un *Comportamiento*.

La EM emplea una *"lista de elementos de acceso aleatorio"* donde son almacenadas las metas. Cada meta (global o específica) es almacenada en la EM en tuplas de la forma ⟨`precond, postcond, ANB`⟩. El atributo `precond` se refiere al conjunto de pre-condiciones que deben ser ciertas para que se active la meta[15], el atributo `postcond` establece un conjunto de post-condiciones que deben cumplirse para que la meta se considere satisfecha, y el atributo `ANB` (Activación de Nivel Base) determina el nivel de activación de la meta (el cual es degradado progresivamente con el paso del tiempo). La ANB de cada meta está definida por la siguiente ecuación:

$$ANB_i^m = iANB_i^m + c \times \sum_{l=1}^{n} t_l^{-d} \tag{8.17}$$

donde: i indica un elemento de la EM (una meta global o específica). l indica la l-ésima vez que fue propuesta la meta m como la meta actual (y por ende agregada a la EM). t_l indica el tiempo transcurrido desde la l-ésima vez que fue agregada la meta a la EM (medida en segundos). c es una constante de amplitud con un valor por defecto $c = 5$. d es una tasa de degradación con un valor por defecto $d = 0.3$. $iANB_i^m$ es el valor inicial de ANB_j^m, que normalmente se establece en 0.

Las metas almacenadas en la EM compiten entre sí por convertirse en la(s) meta(s) activa(s), esto es, las metas vigentes que dirigen la activación de los *Comportamientos*. Sólo las metas que tengan una ANB superior a cierto umbral ($ANB_i^m > umbral_{meta}$), participarán en el proceso de competición llevado a cabo en el módulo de Orquestación de la Consciencia. Intuitivamente, las metas que reciban una mayor activación (e.g., aquellas que hayan sido propuestas como la meta actual en un ciclo cognitivo reciente y/o de

[15]Estas pre-condiciones pueden ser estados internos generados por los motivadores (cuando estos se encuentran fuera de la región homeostática), o estímulos sensoriales externos.

forma reiterada por alguno de los mecanismos de generación de metas), tendrán mayor probabilidad de superar el umbral de activación, pudiéndose determinar así qué metas deben permanecer activas en la EM. El mecanismo de Orquestación de la Consciencia (capítulo 9) decide qué meta gana la competición, mientras que las demás metas continúan en constante competición por alcanzar el foco de atención.

A continuación, se enuncian las consideraciones relativas a la generación de la Estructura de Metas:

- De todo el conjunto de metas que se encuentra en la EM, sólo el subconjunto de metas que exceda el umbral de activación $umbral_{meta}$ podrá mantenerse como el conjunto de *metas activas*. Una de estas metas activas será elegida por el mecanismo de Orquestación de la Consciencia para convertirse en una meta consciente que disparará la activación de uno o varios *Comportamientos*, mientras que el resto de metas activas serán satisfechas por procesos subconscientes, o simplemente se mantendrán en continua competición hasta convertirse en la meta consciente.

  No obstante, nuevas metas pueden incluirse o excluirse de este subconjunto de metas activas. Una meta se excluye del subconjunto de metas activas, como es de suponerse, cuando su ANB se encuentra por debajo del umbral $umbral_{meta}$. Por otro lado, una nueva meta es incluida en el subconjunto de metas activas cuando: (1) su ANB supera el umbral $umbral_{meta}$, esto es, cuando es propuesta por el módulo de Orquestación de la Consciencia como una meta actual y su ANB es alta debido a que no se ve afectada aún por la función de degradación basada en el tiempo transcurrido; y (2) cuando recibe activación adicional al ser propuesta (nuevamente) por alguno de los mecanismos de generación de metas como la meta actual. De esta manera, la realización de las metas activas que participan en los procesos conscientes puede conmutarse dinámicamente durante el proceso de toma de decisiones.

- Las metas pueden ser de dos tipos, según su duración en la EM: *únicas* cuando sólo son alcanzadas una vez por el agente, y *permanentes* cuando pueden ser reactivadas y satisfechas continuamente.

- Las *metas únicas*, una vez se convierten en activas y son satisfechas por uno o varios *Comportamientos* (esto quiere decir, que las post-condiciones de la meta son totalmente satisfechas), son eliminadas de la EM para ser luego incluidas en una nueva estructura: la lista de *metas resueltas*.

- Las *metas permanentes* en la EM (metas que se deben conservar durante todo el ciclo de vida del agente, como por ejemplo las metas ⟨`alimentarse`⟩, ⟨`garantizar-supervivencia`⟩, etc.), no son eliminadas de la EM aunque sean satisfechas por el agente. Simplemente, una vez satisfechas por el agente, se mantienen como metas latentes en espera de su "reactivación" por parte de los motivadores o de algún *Comportamiento* que tenga como objetivo el logro de dicha meta.

8.6. Generación del Estado Anticipado

8.6.1. Inspiración del Modelo Anticipatorio

> *"La previsión o anticipación de los fenómenos y el dominio que tengamos sobre ellos dependen del conocimiento de sus secuencias, y no de los conceptos que podamos habernos formado acerca de su origen o de su naturaleza íntima".*
> — John Stuart Mill, Auguste Comte and Positivism, 1865.

En la metáfora de Baars sobre el Teatro de la Consciencia, se argumenta que una de las características destacables de la experiencia consciente es cómo ésta puede disparar la activación de contextos subconscientes que permiten la subsecuente interpretación de eventos conscientes futuros. En términos de Baars, esto sería equivalente a la función que pueden tener algunos actores durante el acto escénico, quienes pueden predecir los eventos que acontecerán en la siguiente escena dadas las circunstancias actuales, como las hechiceras en *Macbeth*.

Este proceso de predecir los eventos inmediatamente futuros, es lo que se denomina en METÁFORA la *generación del Estado Anticipado*, función que es desempeñada por el módulo de Prospección y Anticipación de la arquitectura. El estado anticipado puede considerarse como la *expectación* que tiene el agente, en términos de un valor probabilístico, de que suceda (o no) algún cambio en el ambiente percibido en el instante de tiempo inmediatamente siguiente, como consecuencia de una acción realizada por el agente.

Mientras que el rol del proceso anticipatorio que afecta la deliberación, la memoria, la atención, el comportamiento y otras facetas de la cognición ha sido estudiado en profundidad en el área de la psicología cognitiva, la neurofisiología y la etología, la literatura sobre los mecanismos explícitos que pueden generar conductas anticipatorias en el área de los agentes artificiales es considerablemente escasa y dispersa (Blank et al., 2005; Butz et al., 2002b; Kunde, 2001; Rosen, 1985; Schubotz y von Cramon, 2001; Butz et al., 2007). No obstante, una aproximación interesante que intenta representar, a través de un modelo computacional, la teoría cognitiva sobre el aprendizaje anticipatorio, es el Sistema Clasificador Anticipatorio propuesto en (Stolzmann, 1998) y extendido luego en (Butz et al., 2002a). El mecanismo encargado de gestionar las dinámicas del módulo de Prospección y Anticipación de METÁFORA, será una adaptación del modelo extendido de dicho sistema.

Los Sistemas Clasificadores Anticipatorios (SA) combinan el modelo computacional de los Sistemas Clasificadores de Aprendizaje (SC) propuestos por Holland (Holland et al., 1989) (y descritos en la sección 2.2.4.3), junto con la teoría cognitiva del control del comportamiento anticipado (Hoffmann, 1993). El resultado es un sistema evolutivo que construye un modelo ambiental predictivo y generalizado. En este sistema, la estructura de las reglas (clasificadores)[16] del tipo estímulo-respuesta típicas de un SC, es extendida con una parte *efecto* que representa el estado anticipado que resulta como consecuencia

[16]Recuérdese que un clasificador es una regla de producción del tipo <condición>:<acción> representada por un alfabeto sub-simbólico, normalmente binario (0 y 1) o ternario (0, 1, y #, donde # representa un comodín).

de la acción que el agente ejecuta en su ambiente en cada ciclo cognitivo. Esta parte *efecto* del clasificador, junto con un proceso de aprendizaje por refuerzo llamado Proceso de Aprendizaje Anticipatorio (PAA)(Hoffmann, 1993), permiten que el SA pueda aprender, de forma latente, una representación interna de su entorno.

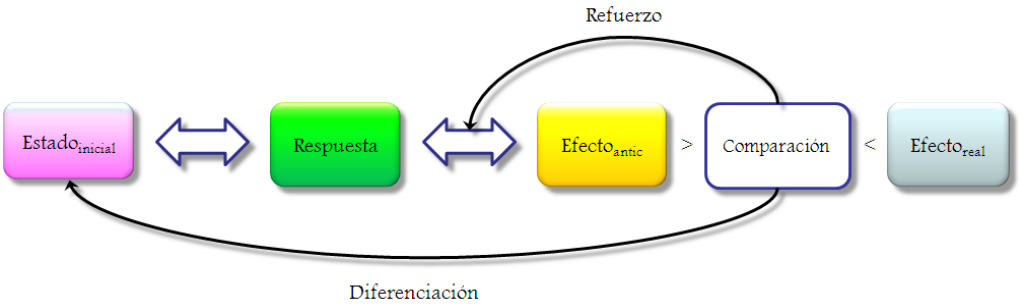


Figura 8.5: Proceso de Aprendizaje Anticipatorio.

La psicología cognitiva reconoce que el aprendizaje solamente puede tener lugar si existe una necesidad presente que requiera ser satisfecha. Debido al mecanismo de aprendizaje latente que se observa en los animales superiores, se supone que toda anticipación surge de una necesidad. La teoría del control del comportamiento anticipado evolucionó como consecuencia de esta conclusión (Hoffmann, 1993).

Esta teoría supone que: (1) cualquier acto conductual o respuesta está acompañada de una anticipación o predicción de sus consecuencias; (2) las anticipaciones de los efectos ($Efecto_{antic}$), son comparadas con los efectos reales ($Efecto_{real}$); y (3) la relación entre la respuesta y la anticipación es recompensada cuando las anticipaciones han sido correctas, y es penalizada en caso contrario. La figura 8.5 presenta la idea básica de esta teoría. Como se verá más adelante, el SA implementa esta idea de aprendizaje a través del proceso de aprendizaje anticipado (PAA) en combinación con el modelo computacional de los sistemas clasificadores de aprendizaje (SC).

En consecuencia, el módulo de Prospección y Anticipación estará provisto de habilidades para: predecir la consecuencia de las acciones del agente, construir un modelo de eventos futuros, controlar la percepción anticipando dichos estímulos futuros, orientar el mecanismo de atención selectiva (véase la sección 9.2.1), y guiar el proceso de simulación interna y extracción de planes (véase la sección 10.3.3).

8.6.2. Sistema Clasificador Anticipatorio

Como se explicó anteriormente, el Sistema Clasificador Anticipatorio (SA) tiene sus orígenes en los Sistemas Clasificadores de Aprendizaje (SC) y en la psicología cognitiva. La estructura básica de un SA resulta bastante similar a la de un SC, con algunas diferencias que intentan reflejar el proceso de aprendizaje anticipatorio.

Comúnmente, un SA se conforma por los siguientes componentes: una estructura de conocimiento (de ahora en adelante denominada *lista de clasificadores*), un mecanismo de selección de acciones, un proceso de generación de reglas (creación de clasificadores),

un mecanismo de aprendizaje anticipatorio, un mecanismo de aprendizaje ambiental, y un proceso de generalización de reglas. A continuación se enuncian las consideraciones que se tuvieron en cuenta para incorporar el SA dentro del módulo de Prospección de METÁFORA:

- El proceso de creación de la lista de clasificadores del SA no es una responsabilidad que atañe únicamente a éste sino que es una labor influenciada por el módulo Comportamental. Concretamente, el módulo Comportamental, en cada ciclo cognitivo, envía al SA la acción activada (i.e., la acción que ha *integrado* el módulo Efector para el estado actual), la cual es luego *incorporada* por el SA en la lista de clasificadores. El módulo Comportamental envía la acción integrada en un formato de representación sub-simbólica (es decir, en un alfabeto binario: 0, 1). Luego, el SA crea un clasificador (regla) de la forma <condición>:<acción>, donde la condición representa el estado actual observado y la acción corresponde a la acción enviada por el módulo Comportamental. Finalmente, se incluye elementos adicionales requeridos por el PAA.
- El mecanismo de aprendizaje ambiental (a través del cual se dirige el aprendizaje de asociaciones del tipo estímulo-respuesta) es una responsabilidad del módulo Comportamental, mientras que el mecanismo de aprendizaje anticipatorio es una responsabilidad innata del SA.
- Al igual que los *Comportamientos* en el módulo Comportamental, el módulo de Prospección y Anticipación dispone de un conjunto de SAs, cada uno orientado al desarrollo de una habilidad específica del agente. Es decir, por cada *Comportamiento* en el módulo Comportamental existirá un SA asociado en el módulo de Prospección y Anticipación. Esto con el fin de mantener la consistencia del conocimiento adquirido por cada par *Comportamiento*/SA.

Esencialmente, un SC típico consiste de una lista de reglas de producción simples (clasificadores) de la forma "SI estado x ENTONCES acción y"[17], las cuales están cada una asociada con un valor de aptitud basado en el éxito histórico que ha tenido dicha regla durante su interacción con el ambiente. Los SA difieren de la idea original de un SC en que cada clasificador no está asociado solamente con el estado actual percibido sino que además se encuentra asociado con el estado siguiente (evento futuro). Por tanto, los clasificadores del SA tienen la forma de "SI estado x_0 ENTONCES ejecutar acción y Y anticipar el próximo estado x_1".

Adicionalmente, existen dos valores asociados a cada clasificador: uno es la calidad de la predicción del clasificador y el otro es la retroalimentación o refuerzo ambiental. La calidad de la predicción es una medida de la habilidad del clasificador para predecir el próximo estado después de que el agente ejecute su acción en el estado actual. El refuerzo ambiental toma el mismo rol que la medida de aptitud en un SC tradicional, es decir, una medida del desempeño del clasificador basada en un objetivo ambiental del sistema. La adaptación de estas medidas es independiente, pero pueden ser combinadas de diversas maneras a fin de dirigir la selección y adaptación de los clasificadores.

[17] Teniendo en cuenta que ésta es una regla simbólica que representa un clasificador del tipo <condición>:<acción>, el cual viene dado generalmente en una representación sub-simbólica como por ejemplo: <100101>:<110>.

El módulo de Prospección de METÁFORA emplea la estructura representacional definida por (Butz et al., 2007) para un clasificador. De esto modo, cada clasificador en la lista de clasificadores consiste de las siguientes partes:

- Una parte *condición* C, que especifica los pre-requisitos que deben ser percibidos para que se cumpla la activación de la regla (clasificador).
- Una parte *acción* A, que especifica la acción que será ejecutada.
- Una parte consecuencia o *efecto* E, que predice los cambios perceptivos que serán causados en el ambiente cuando se ejecute la acción A dada la condición C.
- Una parte *marcador* M, que especifica los valores de cada propiedad percibida de todas aquellas situaciones donde el clasificador no realizó la predicción (anticipación) correctamente.
- Una medida de *calidad* Q que mide la precisión de las anticipaciones.
- Una medida de *refuerzo* R que predice la señal de retroalimentación ambiental (o interna) después de ejecutar A.

Las partes *condición* y *efecto* del clasificador son representadas por una cadena de valores discretos de longitud finita, tal que $C, E \in \{s_1, \ldots, s_m, \#\}^L$, donde $S = \{s_1, \ldots, s_m\}$ es el conjunto de valores observados (en el caso de C, corresponden a los valores del estado actual observado, y en E son valores generados y refinados por el proceso de aprendizaje anticipatorio). L es el número de valores discretos que pertenecen a $S \cup \{\#\}$, donde el símbolo $\#$ representa en C un comodín que puede ser reemplazado por cualquier valor de la propiedad observada (1 ó 0), y en E representa la predicción de que dicho valor "no cambiará" en el siguiente estado una vez ejecutada la acción A.

La parte *acción* es representada por cualquiera de las n posibles acciones ejecutables en el ambiente, correspondiéndose con la acción que envía el módulo Comportamental. El *marcador* tiene la siguiente estructura: $M \in (s_1, \ldots, s_L)$, donde cada $s_i = \{1 \ldots L\}$ es un subconjunto de S que contiene los *valores* de la i-ésima *propiedad* de las situaciones donde el clasificador no realizó una predicción correcta. La figura 8.6 presenta un ejemplo de la estructura de un clasificador del SA propuesto.

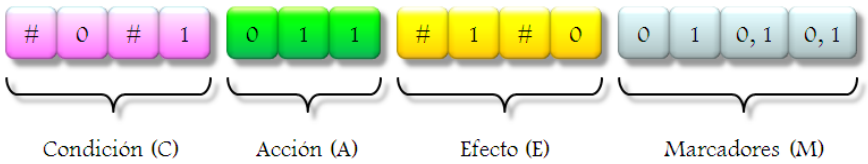


Figura 8.6: Estructura de un clasificador en el SA. Este clasificador empareja todas las situaciones que tienen un "0" en la segunda posición y un "1" en la quinta, y predice que después de ejecutar la acción "011" el estado del ambiente cambiará a un "1" en la segunda posición y a un "0" en la quinta, mientras que los demás valores se mantienen sin cambio. Los marcadores especifican que el clasificador no anticipó correctamente ciertos estados.

Las medidas Q y R son dos valores escalares donde $Q \in [0, 1]$ y $R \in \Re$. Una vez la calidad Q de un clasificador supere el umbral $umbral_{conf}$, éste se etiqueta como un "clasificador confiable" y se convierte en parte del modelo ambiental interno generado por el agente. Si su calidad está por debajo del umbral $umbral_{conf}$ es considerado inadecuado y como consecuencia es eliminado. Todas estas partes del SA son explicadas al detalle en la siguiente sección.

Antes de iniciar la descripción de la estructura del SA, resulta conveniente distinguir las dos fases que ejecuta el sistema:

1. Primero, se ejecuta una fase de aprendizaje donde se lleva a cabo el aprendizaje anticipatorio que refina el conjunto de clasificadores del SA. En esta fase, las predicciones (que cumplen el rol de las *expectativas* del agente) son evolucionadas a través de un proceso cíclico de especialización y generalización, a fin de predecir con un alto nivel de confiabilidad un conjunto de situaciones. Las predicciones generadas en esta fase pueden ser empleadas por otros procesos subconscientes durante la actividad cognitiva del agente.

   Durante esta fase, eel SA provee información necesaria sobre los cambios significativos que puedan acontecer en un estado futuro del entorno, información a partir de la cual es posible que los *Comportamientos* (conscientes y subconscientes – automatismos) produzcan una respuesta *reactiva* en el agente.

2. En una fase posterior, cuando se supone que el agente ha adquirido cierto conocimiento sobre su entorno que le permite crear un modelo anticipado de los posibles eventos futuros, el SA sirve a los procesos de simulación interna mediante la validación de los planes generados. En esta fase, el SA servirá además como una herramienta para la creación de secuencias posibles de eventos, a través de una línea causal de interacciones del tipo percepción/actuación. Este proceso será descrito más detalladamente en la sección 10.3.3.

8.6.3. Proceso Epigenético para el Mecanismo de Predicción

El mecanismo de predicción realizado por el SA consta de 3 procesos: la *Incorporación de Acciones desde el Módulo Comportamental*, la *Especialización de las Anticipaciones a través del PAA*, y la *Generalización de las Anticipaciones a través de Generalización Genética*.

El proceso de Incorporación de Acciones, como se explicó anteriormente, consiste en recibir (en cada ciclo cognitivo) la acción propuesta por el módulo Comportamental, la cual es luego transformada, junto con el estado actual percibido, en un clasificador. Este nuevo clasificador es agregado a la lista de clasificadores del SA donde además son incluidos elementos adicionales, tales como una parte Efecto, una parte Marcadores, una medida de calidad y una medida del refuerzo esperado.

En las siguientes secciones se describe el algoritmo global llevado a cabo por el SA y se detallan los procesos de especialización y generalización de anticipaciones.

8.6.3.1. Algoritmo Anticipatorio

A continuación se presentan los pasos que sigue el algoritmo global del SA para predecir el estado anticipado del sistema, y en la figura 8.7 se presenta un ejemplo de las dinámicas moduladas por el SA:

1. *Percepción en el instante* t: el SA recibe el conjunto de perceptos que se encuentran aún vigentes en la memoria de trabajo en el instante t. Adicionalmente, el SA recibe la acción a_{mc} resultante del proceso decisorio llevado a cabo en el módulo Comportamental.
2. *Selección de Clasificadores*: en este paso el SA busca dentro de su lista de clasificadores (P_t), uno que empareje con el estado percibido. Inicialmente esta lista de clasificadores sólo poseerá clasificadores totalmente generalizados (como se describirá más adelante). El procedimiento es el siguiente:

   a) Se realiza una búsqueda sobre la lista de clasificadores en busca de los clasificadores que emparejen el estado actual (i.e., el conjunto de perceptos activos en la MT). Estos clasificadores resultantes de la búsqueda son agregados a una nueva lista de emparejamiento M_t.

   b) En la sub-lista de emparejamiento M_t se busca que al menos un clasificador tenga la misma acción que la que ha sido producida por el módulo Comportamental (a_{mc}) para el instante t. Si no existe ningún clasificador con esta acción, se procede a crear un nuevo clasificador cuya condición está formada por comodines (#) y cuya acción es a_{mc}. Igualmente, la parte *efecto* del clasificador es rellenada con comodines (#), lo que indica que la regla predice un estado siguiente totalmente generalizado pero el cual será refinado y especializado en ciclos sucesivos a través del aprendizaje anticipatorio (este proceso es detallado en el apartado siguiente). Inicialmente, los valores de los marcadores son indeterminados (#) y los valores para la calidad Q y el refuerzo R son definidos en 0.5. Posteriormente, el clasificador creado es agregado a la lista de clasificadores y a M_t.

3. *Activación del Clasificador Ganador*: si en la lista de emparejamiento M_t (formada en el paso anterior) existe un sólo clasificador, se propone como el clasificador ganador, en caso contrario, se ejecuta un proceso de competición que simula una subasta entre los clasificadores de la lista de emparejamiento. Este proceso es similar al descrito en la sección 7.4.2 para el algoritmo de distribución de recompensas del SIA: básicamente, cada clasificador efectúa una puja que determina su probabilidad de ganar la subasta, y luego, a través del método de selección de la ruleta (*roulette wheel*), se selecciona el clasificador ganador, es decir, el clasificador que mejor representa el estado anticipado.

   Intuitivamente, los clasificadores con mayor calidad de predicción, mayor cantidad de refuerzos positivos acumulados, y mayor nivel de especificidad, tendrán mayor probabilidad de ser elegidos por el método de la ruleta.

4. *Creación de la sub-lista de Aprendizaje*: se crea una sub-lista de clasificadores A_t a partir de la lista de emparejamiento M_t. Esta sub-lista contiene todos los

clasificadores de M_t cuya parte *acción* sea igual a la parte *acción* del clasificador ganador determinado en el paso anterior.

5. *Almacenamiento en la Memoria de Trabajo*: la parte consecuencia o *Efecto* del clasificador activado es almacenada en la MT como el nuevo estado anticipado. Una vez está en la MT, el estado anticipado puede ser accedido por otros módulos de la arquitectura (la memoria episódica, los automatismos, etc.).
6. *Percepción en el instante* $t+1$: el sistema recibe los nuevos perceptos del estado del sistema en el instante $t+1$. Adicionalmente, se recibe la acción determinada por el módulo Comportamental en el instante $t+1$.
7. *Aprendizaje por Refuerzo*: en este paso se retroalimentan todos los clasificadores en M_t cuyas acciones sean iguales a la acción determinada por el módulo Comportamental en t. El valor de la retroalimentación es calculado en función del refuerzo ambiental recibido por ejecutar la acción a_{mc}.
8. *PAA*: se inicia un proceso de aprendizaje anticipado sobre la sub-lista A_t. En este paso son creados nuevos clasificadores que especializan el conocimiento encapsulado por los clasificadores en A_t. Este proceso es descrito en la sección 8.6.3.2.
9. *Exploración y generalización*: en este paso, a través de una máquina Darwiniana (un Algoritmo Genético), se procede a generar nuevos clasificadores que generalizan el conocimiento representado por la lista de clasificadores P_t. Este proceso es descrito en la sección 8.6.3.3.

Respecto al proceso de selección de clasificadores enunciado en el paso 3, éste se hace en base al nivel de puja de cada clasificador, el cual se calcula así (Holley, 2004):

$$puja_{cl} = \frac{1}{Q_t} \times Q_{cl} \times 2^{Spec_{cl}} \times R_{cl} \tag{8.18}$$

donde: Q_t es una función que calcula todas las calidades de predicción de los clasificadores en M_t. Q_{cl} es la calidad de predicción del clasificador cl. R_{cl} es el refuerzo acumulado por el clasificador cl. $Spec_{cl}$ es una función de especificidad que determina la cantidad de bits en la parte condición del clasificador que diferentes al símbolo comodín $\#$.

y donde Q_t corresponde a:

$$Q_t = \sum_{i=0}^{n} (Q_i \times 2^{Spec_i} \times R_i) \tag{8.19}$$

En cuanto al aprendizaje por refuerzo del paso 7 del algoritmo, se adapta la idea del algoritmo *Q-learning* descrito en la sección 7.3.2.2. Así pues, a fin de aprender una política óptima en el SA, la predicción de la recompensa (refuerzo positivo) r para cada clasificador de la sub-lista de Aprendizaje A_t, en el instante de tiempo t, es actualizada de acuerdo con la siguiente ecuación (Métivier y Lattaud, 2003):

$$R_{(cl,t)} = (1 - b_r) \times R_{(cl,t-1)} + b_r \times (\gamma \times \max_{cl \in M_{t+1}} (q_{cl} \times r_{cl}) + \rho(t)) \tag{8.20}$$

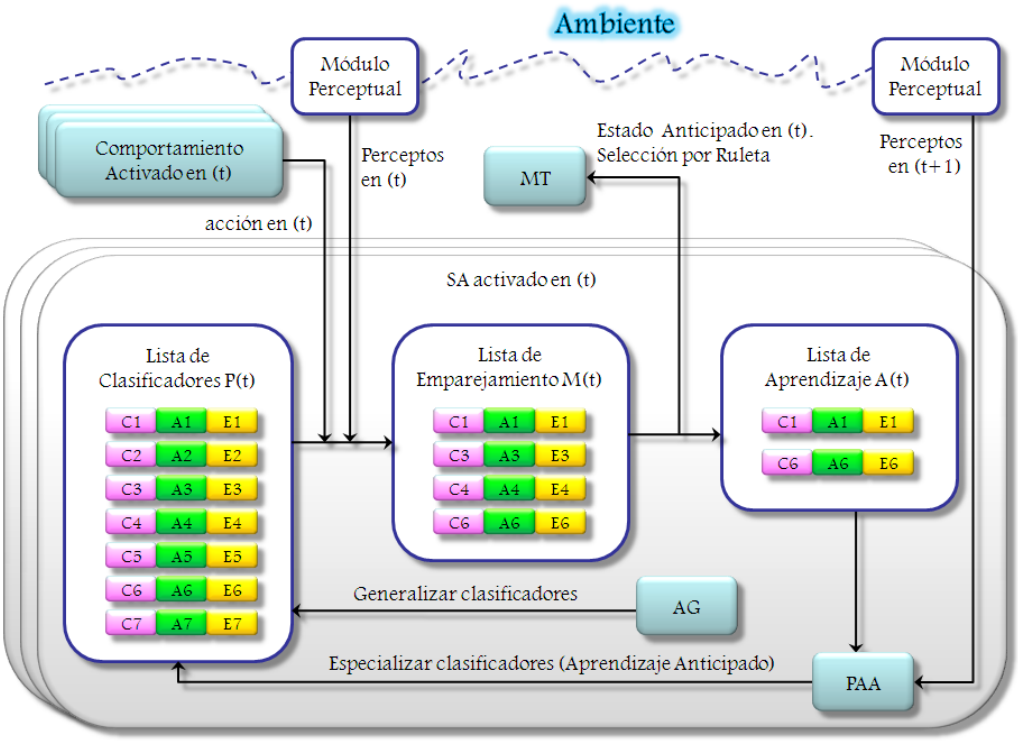


Figura 8.7: Sistema Clasificador Anticipatorio. Con $C1, C2, \ldots Cn$, $A1, A2, \ldots An$, y $E1, E2, \ldots En$, las partes condición, acción y efecto respectivamente del clasificador n. MT: Memoria de Trabajo, AG: Algoritmo Genético, y PAA: Proceso de Aprendizaje Anticipatorio. Cada *Comportamiento* del módulo Comportamental tiene asociado un SA específico en el instante t.

donde: $b_r \in [0, 1]$ es la tasa de aprendizaje (por defecto es 0.05). $R_{(cl,t-1)}$ es el refuerzo acumulado por el clasificador cl en el instante $t-1$. $\gamma \in [0, 1]$ es un factor de descuento similar al empleado en el algoritmo de *Q-learning* (por defecto es 0.95). M_{t+1} es la sub-lista de emparejamiento en el instante $t+1$. $\rho(t)$ es el refuerzo escalar recibido en el instante t.

En el cuadro 8.2 se presenta el pseudo-código para el algoritmo global anticipatorio.

8.6.3.2. Especialización: Proceso de Aprendizaje Anticipatorio

Como se ha mencionado anteriormente, el PAA se deriva de la teoría de control del comportamiento anticipado (Hoffmann, 1993). Esta teoría ha sido trasladada al contexto de los sistemas artificiales a través de los SA propuestos por Stolzmann (Stolzmann, 1998).

Método: EjecutarSA

```
1.  Variables
2.     A_{t-1}        ↑↓ Sublista de clasificadores cuya acción es igual a la acción del
                      clasificador ganador en el ciclo cognitivo anterior.
3.     R_{t-1}        ↑↓ Señal de refuerzo obtenida por la acción ejecutada en el ciclo
                      cognitivo anterior.
4.  Iniciar Procedimiento
5.     t ← 0
6.     percibirSituacion()
7.     Hacer:
8.        Si t >0 entonces
9.           ejecutarAprendizajeRefuerzo(A_{t-1}, R_{t-1})
10.          ejecutarAprendizajeAnticipatorio(A_{t-1})
11.       Fin Si
12.       M_t ← seleccionarClasificador()
13.       cl ← activacionClasificadorGanador()
14.       A_t = generarSublistaAprendizaje()
15.       enviarEstadoAnticipadoMT(cl)
16.       R_t = percibirRefuerzo(cl)
17.    Mientras que el agente perciba su entorno
18. Fin Procedimiento
```

Algoritmo 8.2: Algoritmo Global del Sistema Clasificador Anticipatorio.

El PAA inicia con una base de conocimiento general: por cada acción posible[18] se crea un clasificador donde sus partes condición (C) y efecto (E) se componen únicamente de comodines ($\#$)[19]. Por tanto, antes de iniciar el algoritmo de predicción, el SA contiene una lista de n clasificadores: uno por cada acción posible. Este conjunto n de clasificadores originales poseen un estado particular en el sistema: no pueden ser eliminados ni pueden ser marcados.

El PAA siempre ocurre en la sub-lista de aprendizaje A_t creada en el instante t. El PAA emplea dos vectores de datos: $\sigma(t)$ que representa el conjunto de perceptos que coinciden con la parte condición C de todos los clasificadores en A_t, y $\sigma(t+1)$ que es la situación percibida (como un conjunto de perceptos) después de que la acción propuesta por los clasificadores de A_t ha sido ejecutada[20]. Por cada clasificador de A_t, el PAA distingue los siguientes casos:

[18]Esto quiere decir, todas las combinaciones de *valores* posibles para el conjunto de *propiedades* que conforman la acción. Por ejemplo, si la acción se compone de tres *propiedades* (cada una representando un actuador específico) y cada una de estas propiedades puede admitir dos *valores* específicos (e.g., activado y desactivado), el conjunto total de acciones (combinando los distintos valores) es: $2^3 = 8$ acciones diferentes.

[19]Recuérdese que, de manera similar a la representación de la parte condición de los anticuerpos en el Sistema Inmune Artificial (descrita en la sección 7.4.1), los comodines ($\#$) son empleados para generalizar situaciones donde estos pueden ser reemplazados por cualquier valor percibido posible (en este caso, por un valor binario: 0, 1).

[20]En este punto cabe resaltar que, todos los clasificadores en la sub-lista A_t poseen la misma acción, que a su vez corresponde a la acción enviada por el módulo Comportamental en t.

- *El caso Inoperante*: en este caso, la situación no presenta cambios entre el instante t y el instante $t+1$, lo que significa que $\sigma(t)$ y $\sigma(t+1)$ se mantienen idénticos. En esta caso, la calidad Q del clasificador es disminuida y en la parte de los marcadores se agrega cada uno de los valores percibidos en $\sigma(t)$. La justificación que da la psicología cognitiva sobre este caso es que cada acción que ejecuta el individuo debe causar algún cambio en las percepciones futuras, al menos a nivel proprioceptivo. La calidad del clasificador es disminuida empleando la *regla delta* propuesta por Widrow y Hoff (Widrow y Hoff, 1960):

$$Q_{cl} = Q_{cl} \times (1 - b_q) \tag{8.21}$$

  donde: Q_{cl} es la calidad del clasificador cl. $b_q \in [0,1]$ es la tasa de aprendizaje (por defecto es 0.05).

  La calidad de todos los clasificadores es inicializada en 0.5 y si ésta desciende por debajo del umbral $umbral_{qmin}$ (por defecto es 0.1) el clasificador es eliminado de la lista de clasificadores.

  - *El caso corregible*: el clasificador es especializado en su parte condición y efecto de manera que representen una anticipación correcta con la pareja situación-consecuencia ($\sigma(t)$, $\sigma(t+1)$). Este caso representa los clasificadores cuya parte efecto contienen un símbolo comodín # en al menos uno de los atributos que cambia entre $\sigma(t)$ y $\sigma(t+1)$. El PAA genera un nuevo clasificador más específico en los *puntos cambiantes*: cada comodín # (tanto en la parte C como en la parte E) es reemplazado por el valor específico que define $\sigma(t)$ para C, y $\sigma(t+1)$ para E. La calidad del nuevo clasificador es igual a la calidad del clasificador padre, excepto si ésta es menor de 0.5, caso en el cual se dejaría en 0.5 para darle la oportunidad al nuevo clasificador de competir con los demás.

    Por ejemplo, el clasificador $\langle$##10##:1:##01##$\rangle$[21] sería incorrecto en el siguiente caso: $\sigma(t)$ = 11101 y $\sigma(t+1)$ = 11001 (el quinto elemento del estado percibido $\sigma(t+1)$ no permaneció igual a $\sigma(t)$ en la misma posición, tal como lo había predicho la parte efecto del clasificador). Por tanto, el PAA crea un nuevo clasificador especializado ($\langle$##10**1**#:1:##01**0**#$\rangle$) el cual es agregado a la lista de clasificadores.
  - *El caso incorregible*: en esta caso el clasificador no puede hacerse más específico. Esto significa que el valor en la posición específica de la parte E ha sido previamente especializado y difiere con $\sigma(t+1)$ en la misma posición. En este caso no se crea un nuevo clasificador.

- *El caso esperado*: en este caso el clasificador se anticipa al estado $\sigma(t+1)$ correctamente. En esencia, cada atributo que sea un comodín (#) en la parte efecto E del clasificador se mantiene igual al valor correspondiente en $\sigma(t)$, mientras que los valores específicos en E son iguales a los valores específicos correspondientes en $\sigma(t+1)$. En este caso, si el clasificador no procrea nuevos clasificadores (i.e, debido a que el clasificador no puede hacerse más específico, éste no genera nuevos

[21] Donde ##10## corresponde a la condición, 1 corresponde a la acción, y ##01## corresponde al efecto del clasificador.

clasificadores especializados), entonces la calidad del clasificador es incrementada empleando la regla delta de Widrow y Hoff (Widrow y Hoff, 1960):

$$Q_{cl} = Q_{cl} \times (1 - b_q) + b_q \tag{8.22}$$

donde: Q_{cl} es la calidad del clasificador cl. $b_q \in [0, 1]$ es la tasa de aprendizaje (por defecto es 0.05).

Si la calidad de un clasificador sobrepasa el umbral $umbral_{qmax}$ (por defecto es 0.9) el clasificador es considerado "confiable". Por consiguiente, el modelo interno que construye el agente de su entorno es representado por todos aquellos clasificadores confiables.

Por otro lado, si el clasificador está marcado y es posible detectar las diferencias entre el marcador y la situación en la cual está siendo aplicado el clasificador, se crea un nuevo clasificador que especifica parte o todas aquellas diferencias en la parte condición. Recuérdese que la parte marcadores M del clasificador especifica las características del estado observado donde el clasificador no ha realizado la anticipación correctamente. Por tanto, una distinción en dichos estados percibidos conduce a la creación de clasificadores que son cada vez más y más específicos a la situación donde la anticipación es correcta. Por ejemplo, el clasificador $\langle$##10##:1:##01##$\rangle$ genera una anticipación correcta para la siguiente pareja situación-consecuencia: $\sigma(t)$ = 111011, $\sigma(t+1)$ = 110111. Suponiendo que el clasificador contiene los siguientes marcadores: {{1, 0}, 1, 1, 0, 1, {1, 0}}, se crearía y agregaría a la lista de clasificadores el siguiente clasificador: $\langle$1#10#1:1:1#01#1$\rangle$.

El sistema considera que todos los clasificadores son distintos, es decir, cuando se crea un nuevo clasificador y éste ya existe en la lista de clasificadores, el clasificador nuevo no es agregado a la lista y en su lugar la calidad del clasificador existente es incrementada. A continuación se presenta un ejemplo de la dinámica del PAA.

Supóngase que el agente es un robot que se encarga de levantar cajas. El robot está equipado con una pinza (actuador) que le permite levantar las cajas, y con 3 sensores codificados en binario así: el primer sensor indica si el robot tiene agarrada la caja con las pinzas (1: las pinzas están cerradas sujetando la caja, 0: las pinzas están abiertas), el segundo sensor indica la situación de la caja (1: la caja está sobre la mesa, 0: la caja está en el suelo), y los dos últimos sensores representan el peso de la caja (00: muy liviana, 01: liviana, 10: pesada, 11: muy pesada). La única acción que interesa en este ejemplo es "situar la caja sobre la mesa", la cual para efectos de simplificación del ejemplo estará codificada con un valor de 1. En un estado inicial, el SA empezará con un sólo clasificador totalmente generalizado:

Clasificador	Condición	Acción	Efecto	Marcador	Calidad
cl_1	####	1	####	{#, #, #, #}	0.50

Siendo el estado actual percibido $\sigma(t) = 1000$ (i.e., las pinzas sujetan una caja liviana), el robot emplea la regla general que le permite de manera exitosa, dejar la caja sobre

la mesa, produciéndose un *caso inesperado* debido a que el estado futuro ha cambiado (i.e., el estado de la pinza, y la ubicación de la caja: 0100), contrariamente a la predicción realizada por el clasificador general donde el estado futuro se mantiene sin cambios (####). Intuitivamente, el clasificador cl_1 debe reducir su probabilidad de ser activado en ciclos posteriores para la misma situación (puesto que es demasiado general). Por tanto, se disminuye la calidad del clasificador cl_1 y se modifica la parte marcadores del mismo: $M_{cl_1} = \{1, 0, 0, 0\}$[22]. Teniendo en cuenta que $\sigma(t) = 1000$ y $\sigma(t+1) = 0100$ (i.e., la caja es situada sobre la mesa), es posible apreciar que los bits que cambian de un estado al siguiente son los dos primeros. Aleatoriamente, se escoge una de estas dos posiciones y se especializa en un nuevo clasificador que se crea a partir de cl_1, por ejemplo:

Clasificador	Condición	Acción	Efecto	Marcador	Calidad
cl_1	####	1	####	{1, 0, 0, 0}	0.45
cl_2	#0##	1	#1##	{#, #, #, #}	0.50

Es importante resaltar que la idea del SA no es crear un clasificador totalmente especializado que sólo restrinja su activación a un estado observado específico (e.g., en el caso de que cl_2 fuese $\langle 1000 : 1 : 0100 \rangle$), sino que contrariamente la idea es crear, de forma "gradual", clasificadores que puedan generalizar un conjunto de situaciones.

Considérese que sólo las cajas livianas (01) y muy livianas (00) son las únicas que puede levantar el robot. Ahora, asúmase que el nuevo estado percibido es $\sigma(t) = 1001$, por tanto se agregarían a la lista de emparejamiento (M_t) los dos clasificadores en la lista de clasificadores (cl_1 y cl_2) que coinciden con el estado observado. Se estima la puja de cada uno de los dos clasificadores y se escoge uno de ellos (a través del método de la ruleta). Supóngase que el ganador es cl_2. En esta ocasión se produce un *caso inesperado corregible* (debido a que el robot logra situar la caja sobre la mesa, i.e., $\sigma(t+1) = 0101$, mientras que la predicción del clasificador indica que el estado futuro es igual a $E_{cl_2} = \#1\# = 1101$). Por tanto, la calidad de cl_2 es disminuida y su parte marcadores es actualizada. Igualmente, se actualizan la calidad y los marcadores de cl_1. A partir de los marcadores de cl_2 (el clasificador ganador) se crea un nuevo clasificador más especializado:

Clasificador	Condición	Acción	Efecto	Marcador	Calidad
cl_1	####	1	####	{1, 0, 0, {0, 1}}	0.40
cl_2	#0##	1	#1##	{1, #, 0, 1}	0.45
cl_3	10##	1	01##	{#, #, #, #}	0.50

Ahora imagínese que el robot intenta levantar una caja pesada ($\sigma(t) = 1010$). En este caso emparejarían los tres clasificadores en la lista de clasificadores (cl_1, cl_2, y cl_3). Cada uno de estos clasificadores hace una puja, y después de aplicar el método de la ruleta

[22]Nótese que los marcadores corresponden al estado observado $\sigma(t)$ para el cual el robot no efectuó correctamente la predicción

asúmase que se selecciona el clasificador cl_3. Debido a que el robot no puede levantar la caja (i.e., el nuevo estado percibido es $\sigma(t+1) = 1010$, a diferencia de la predicción del clasificador cl_3 que corresponde a $E_{cl_3} = 01\#\# = 0110$)[23], se produce un *caso inesperado corregible*. Por tanto, se disminuye la calidad de cl_3 y se actualizan sus marcadores. Igualmente, todos los demás clasificadores en la lista de emparejamiento son actualizados: su parte marcadores es especificada y su calidad es incrementada o disminuida (dependiendo de si la predicción realizada por cada uno fue correcta o incorrecta). Luego, a partir de los marcadores de cl_3 se crea un nuevo clasificador más especializado:

Clasificador	Condición	Acción	Efecto	Marcador	Calidad
cl_1	####	1	####	{1, 0, {0, 1}, {0, 1}}	0.35
cl_2	#0##	1	#1##	{1, #, {0, 1}, {0, 1}}	0.40
cl_3	10##	1	01##	$\{\#, \#, 1, 0\}$	0.45
cl_4	100#	1	010#	$\{\#, \#, \#, \#\}$	0.50

Este ejemplo revela la idea principal de los marcadores (en este caso, los marcadores de cl_3) al indicar que los dos últimos bits de la condición de los próximos clasificadores creados resultan ser relevantes para la ejecución exitosa de la acción, no obstante, la diferencia sustancial del tercer bit es más fuerte que la diferencia ocasional en el cuarto bit (tanto en la parte condición como en la parte efecto), y por ende el tercer bit es especificado en la condición de cl_4[24]. Cabe notar también, cómo los marcadores conservan un registro histórico de aquellas situaciones donde los clasificadores no han creado una anticipación correcta del estado futuro.

Finalmente, el resultado es un clasificador (cl_4) especializado y refinado, que refleja la situación en la que el robot puede levantar cajas livianas y muy livianas y situarlas luego sobre una mesa. En sucesivos ciclos cognitivos, el SA podría obtener un clasificador como este: $cl_n = \langle 101\#\texttt{:}1\texttt{:}101\# \rangle$, el cual generaría un caso inoperante del sistema (aunque plausible), que indicaría que el robot no podría levantar cajas pesadas y, como consecuencia, el estado futuro no sufriría cambios (i.e., $\sigma(t) = \sigma(t+1)$).

En este punto cabe destacar una extensión que se propone al SA original presentado por (Butz et al., 2002a): en cada paso de aprendizaje del PAA, contrariamente a la propuesta original que sólo actualiza el clasificador seleccionado por el método de la ruleta, se sugiere la actualización de todos los clasificadores de la lista de aprendizaje (A_t) tanto en su parte marcadores (M) como en su medida de calidad (Q). Esto con el fin de asegurar la convergencia y reducir la curva de aprendizaje del sistema. Esto resulta factible ya que en cada estado actual del sistema se obtiene una entrada perceptual que confirma si las predicciones hechas por los clasificadores de la lista de aprendizaje en el estado

[23] Nótese que, la parte efecto de cl_3 indica que el primer y segundo bit del estado observado ($\sigma(t) = 1010$) cambiará (en el estado futuro anticipado) a "01", mientras que los demás bits se mantienen igual, por eso este clasificador predice que el estado futuro será (erróneamente) $\sigma(t+1) = 0110$.

[24] Esta conclusión se obtiene después de varios ciclos cognitivos a través de los cuales las calidades y los refuerzos de cada clasificador son ajustados. En el ejemplo, se asume un proceso de descubrimiento de los clasificadores relativamente directo, esto con el fin de simplificar el procedimiento y de presentar los resultados globales que puede obtener el SA a través del PAA

inmediatamente anterior, han sido correctas o no. Se produce entonces un aprendizaje en paralelo que es retroalimentado por cada estado actual observado.

8.6.3.3. Generalización: Máquina Darwiniana para el SA

A través del proceso gradual de refinamiento que ejecuta el PAA son generados nuevos clasificadores que representan situaciones cada vez más específicas. En algunas ocasiones, el PAA especifica atributos irrelevantes en la parte condición del clasificador, lo que causa un efecto de *"hiper-especialización"*. Desde este punto de vista, el PAA puede considerarse como el mecanismo del SA que ejerce una presión de especialización sobre la lista de clasificadores. Como una contramedida, se propone la inclusión de un Algoritmo Genético (AG) a fin de ejercer un efecto contrario en el SA: la presión de generalización.

La idea básica del AG (como Máquina Darwiniana), es inyectar cierto grado de generalización en la parte condición de los clasificadores de la lista de aprendizaje (A_t) del sistema. El proceso de evolución de los clasificadores en A_t está determinado por la medida de precisión de cada clasificador (la aptitud), que en este caso, corresponde a la medida de la calidad Q de cada clasificador. Aquellos clasificadores con mayor calidad tendrán mayor probabilidad de ser seleccionados para su reproducción (i.e, replicación, recombinación, y mutación). En la figura 8.8 se presenta un esquema de la interacción entre las presiones de especialización y generalización ejercidas por el PAA y el GA, respectivamente, sobre la lista de clasificadores.

En cuanto al diseño de la Máquina Darwiniana (i.e., el Algoritmo Genético) que modula el proceso de generalización en el SA, se tomaron en cuenta las siguientes consideraciones:

- El PAA debe ser aplicado antes de ejecutar el Algoritmo Genético (AG) con el fin de garantizar que cada clasificador obtenga su máximo nivel de calidad Q.
- Se establece una condición de activación para el AG determinada por un registro de tiempo. Para esto, cada clasificador almacena un registro de tiempo t_{ag} que representa la última vez que el clasificador estuvo involucrado en una ejecución del AG[25]. El AG es aplicado si el valor del registro de tiempo actual (i.e., la hora de sistema en el instante actual) menos el promedio de todos los registros de tiempo de los clasificadores de la lista de aprendizaje, sobrepasa un umbral $umbral_{ag}$. Esto es:
$$t_{ag} - (\sum_{j \in A_t}^{n} \frac{t_{ag_j}}{n}) > umbral_{ag} \tag{8.23}$$
donde: $umbral_{ag}$ tiene un valor por defecto igual a 10.000.
- En el momento en que es activado el AG, se seleccio.an dos de los mejores clasificadores mediante el método de la ruleta. A fin de favorecer la selección de aquellos clasificadores con mayor nivel de aptitud (i.e., con una calidad más alta), se aplica una función de potencia a cada clasificador en función de su calidad, como por

[25]Este registro de tiempo corresponde a la hora actual del sistema medida en unidades de milisegundos.

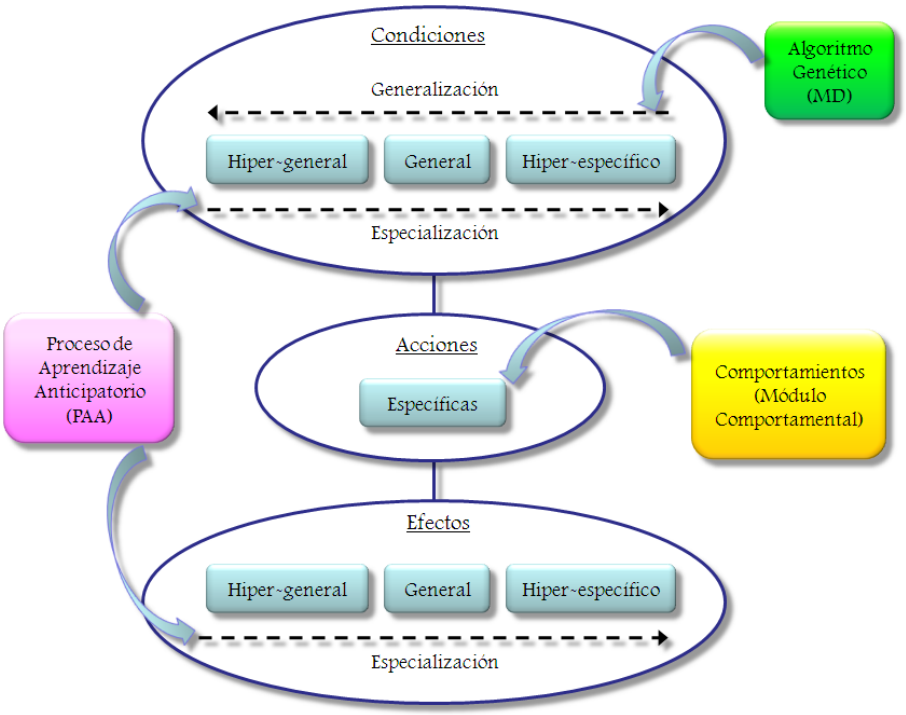


Figura 8.8: Procesos de Generalización y Especialización en el SA. El PAA tiende a especializar las partes condición y efecto de cada clasificador, mientras que el AG tiende a generalizar la parte condición de cada clasificador. Puesto que la idea es crear anticipaciones con cierto nivel de precisión (en relación a situaciones específicas), no resulta conveniente ejercer una presión de generalización sobre la parte efecto de los clasificadores. La parte acción de cada clasificador es definida (especializada) mediante el proceso de aprendizaje que coordina cada *Comportamiento* asociado al SA.

ejemplo q^3. Estos dos clasificadores "padres" se cruzan (recombinan) y generan dos nuevos clasificadores "hijos".

- Con una probabilidad μ, cada atributo especializado de la parte condición del clasificador hijo, es mutado por un símbolo comodín (haciéndolo más general). Por defecto $\mu = 0.5$.

- Si la parte efecto (predicción) de los dos clasificadores hijos es idéntica, se recombinan entre sí aplicando una recombinación en dos puntos. En este caso, tanto la predicción del refuerzo (R) como la calidad (Q) del clasificador es estimada como el promedio del valor correspondiente que tiene cada uno de los clasificadores padres.

- La calidad Q de cada clasificador hijo es multiplicada por 0.5 a fin de prevenir errores de hiper-generalización.

- Los nuevos clasificadores hijos son insertados en la lista de clasificadores sólo si la condición no es completamente general (i.e., no está conformada solamente por comodines).

- Cada clasificador almacena una medida adicional ($card$) que representa la cardinalidad del mismo. Antes de ser almacenado el clasificador hijo en la lista de clasificadores, se busca en la lista si existe uno con una condición igual a la de éste. Si existe, el clasificador hijo no es agregado a la lista de clasificadores y es incrementada en 1 la cardinalidad $card$ del clasificador que ya se encuentra en la lista. Esto resulta similar al proceso descrito por Wilson en los Sistemas Clasificadores Extendidos (Wilson, 1995).

- Si el tamaño de la lista de aprendizaje A_t sobrepasa el umbral $umbral_{la}$, se elimina el exceso de clasificadores. El proceso de eliminación emplea una selección por torneo, con una tasa equivalente a 1/3 del tamaño de la lista de aprendizaje. El torneo está basado en la calidad de cada clasificador. Si la calidad difiere en más de 0.1, el clasificador con menor calidad es eliminado, de lo contrario, es eliminado el clasificador que esté más especializado. Para más detalles acerca del proceso de selección por torneo, el lector puede consultar a (Goldberg y Deb, 1991).

Debido a que las dos presiones opuestas de especialización y generalización (ejercidas por el PAA y el AG respectivamente) pueden provocar que entre medias se pierdan clasificadores "valiosos" (es decir, clasificadores cuyo nivel de generalización es el máximo posible para abstraer un conjunto de situaciones), se requiere por tanto definir un mecanismo de subsumción. Este mecanismo de subsumción de clasificadores fue inicialmente propuesto por Wilson (Wilson, 1995). La idea básica de la subsumción es que una vez es encontrado un clasificador preciso, con un nivel máximo de generalización, éste absorbe todos aquellos clasificadores que sean más específicos que él. Concretamente, un clasificador cl_{sub} subsume un clasificador nuevo cl_{nv} si:

1. El clasificador ha alcanzado el nivel de confiabilidad, es decir, esta etiquetado como "confiable". Un clasificador cl_{sub} es considerado confiable si su calidad Q_{sub} sobrepasa el umbral de confiabilidad $umbral_{conf}$ (por defecto es 0.9).
2. cl_{sub} debe ser un clasificador experimentado. Para determinar el nivel de experiencia de cada clasificador se emplea una medida adicional exp que almacena el número de veces que el clasificador ha participado en el PAA. cl_{sub} se considera experimentado si sobrepasa el umbral $umbral_{exp}$ (por defecto $umbral_{exp} = 20$).
3. cl_{sub} no debe estar marcado. Este criterio garantiza que cl_{sub} ha generado anticipaciones correctas hasta el momento, y no posee elementos en la parte marcadores.
4. cl_{sub} es más general que cl_{nv}. En este criterio se considera la generalidad sintáctica de las condiciones de los dos clasificadores. Por tanto, cl_{sub} es más general que cl_{nv}, si cl_{nv} tiene menos comodines (#) que cl_{sub}.
5. cl_{sub} tiene la misma acción que cl_{nv}.
6. cl_{sub} anticipa el mismo estado que cl_{nv}.

Cuando el clasificador cl_{nv} es subsumido, no es insertado en la lista de clasificadores. Además, si la subsumción ocurre en el PAA, la calidad de cl_{sub} es incrementada empleando la ecuación 8.22. Por otro lado, si la subsumción ocurre en el el AG, la cardinalidad de cl_{sub} es incrementada en 1. Por consiguiente, sólo el AG puede crear clasificadores

Método: activarAG

```
1.  Variables
2.     A_{t-1}          ↑↓ Sublista de clasificadores cuya acción es igual a la acción del
                        clasificador ganador en el ciclo cognitivo anterior.
3.  Inicio Procedimiento
4.     Si se cumple la condición de activación del AG entonces:
5.        Mientras que no se haya generado un número k de nuevos
          clasificadores hacer:
6.           padres ← seleccionarPadres(A_t)
7.           hijos ← cruzarPadres(padres)
8.           hijos ← mutarHijos(hijos, μ)
9.           Si hijos[0].condicion = hijos[1].condicion entonces:
10.             hijos ← recombinarHijos(hijos)
11.             calcularRefuerzoPromedio(hijos)
12.             calcularCalidadPromedio(hijos)
13.          Fin Si
14.          Para cada hijo i creado hacer:
15.             hijos ← reducirCalidad(hijo.i) //se reduce en un 0.5.
16.             esGen ← validarGeneralidad(hijo.i) //valida si el clasificador hi-
                jo no es completamente general (su parte condición no está conformada
                sólo por comodines).
17.             esExs ← validarExistencia(hijo.i, cl) //comprueba si el cla-
                sificador hijo no se encuentra ya en la lista de clasificadores. Si se encuen-
                tra en la lista de clasificadores, cl se hace igual al clasificador hijo, de lo
                contrario, nulo.
18.             Si esGen = true Y esExs = false entonces:
19.                listaClasificadores ← agregar(hijo.i)
20.             Sino
21.                subsume ← validarSubsumcion(cl, hijo.i)
22.                Si subsume = true entonces:
23.                   subsumir(cl, hijos)
24.                Fin Si
25.                aumentarCardinalidad(listaClasificadores.cl)
26.             Fin Si
27.          Fin Para
28.          Si el tamaño de A_t excede el tamaño máximo entonces:
29.             eliminarClasificadores(A_t)
30.          Fin Si
31.       Fin Mientras que
32.    Fin Si
33. Fin Procedimiento
```

Algoritmo 8.3: Algoritmo Genético para el Sistema Clasificador Anticipatorio.

idénticos incrementando sus cardinalidades mientras que el PAA se mantiene firme con su principio de no crear clasificadores idénticos. Este método asegura que el PAA no generará clasificadores hiper-especializados, una vez se encuentra un clasificador preciso y con el máximo nivel de generalidad posible para abstraer un conjunto de situaciones. Este proceso desencadena en un ciclo continuo, donde la presión de especialización ejercida por el PAA produce clasificadores hiper-especializados, que luego se convierten en clasificadores más generales a través del proceso evolutivo y de generalización que lleva a cabo el AG. El efecto de "tirar hacia su propio lado" producido por cada uno de estos

mecanismos, convierte el proceso global en un modelo cíclico oscilatorio [26]. En el cuadro 8.3 se presenta el pseudo-código para el Algoritmo Genético del SA.

Como comentario adicional, cabe notar que la función de aptitud que evalúa el desempeño de cada clasificador del SA se basa en dos términos: (1) el refuerzo o retroalimentación recibida por el *Comportamiento* asociado al SA, en relación a la acción ejecutada en el ciclo cognitivo inmediatamente anterior (como se describirá en la sección 10.2); y (2) un refuerzo basado en la precisión de la anticipación, la cual se evalúa en el instante inmediatamente posterior a la predicción (como se describió en la ecuación 8.20).

En cuanto al modo de operación del Sistema de Prospección y Anticipación (SPA), éste define dos fases: una fase de aprendizaje y una fase de ejecución. Durante la fase de aprendizaje, ejecutada en las fases iniciales de interacción del agente con un entorno nuevo, el SPA genera un estado anticipado por cada situación observada por el agente, cuya validez es luego comprobada por la información sensorial observada en el siguiente ciclo cognitivo. Como se puede apreciar en la figura 8.9.a, el propósito de esta fase de operación es refinar, durante la ejecución de varios ciclos cognitivos y mediante la acción de la señal de refuerzo, las predicciones construidas por el SPA. Los estados anticipados generados en esta fase no cumplen una función de apoyo al proceso de toma de decisiones, sin embargo, sirven de entrenamiento para la ejecución de la segunda fase.

En la fase de ejecución, el sistema emplea el conocimiento anticipatorio adquirido previamente por el SPA como soporte al proceso deliberativo de toma de decisiones. En la figura 8.9.b se observa que el SPA genera un estado anticipado que reemplaza la entrada sensorial enviada por el módulo de Percepción. Este bucle que se forma entre los módulos MT, MC, ME, y SPA, genera un lazo de simulación interna cuyo propósito es validar la causalidad de una secuencia de acciones a partir de una única entrada sensorial inicial. Finalmente, cuando la simulación interna no requiere de más ciclos, el módulo Efector envía una acción final a los actuadores del agente. Obsérvese que durante esta fase no son reforzados los estados anticipados propuestos por el SPA, ya que el agente no recibe una señal de retroalimentación inmediata por las acciones intermedias, aunque sí lo hace después de haber ejecutado la acción final. Este mecanismo de simulación interna será descrito al detalle en la sección 10.3.3.

8.6.3.4. Modelo de Interacción del Sistema Anticipatorio

En la figura 8.10 se presenta el diagrama de interacción de objetos donde participan todas las dinámicas descritas en las secciones anteriores para el Sistema Anticipatorio.

[26] Como se mencionó anteriormente en la sección 6.4.4, esta función es desempeñada por el bloque de construcción *Oscilador* del SA.

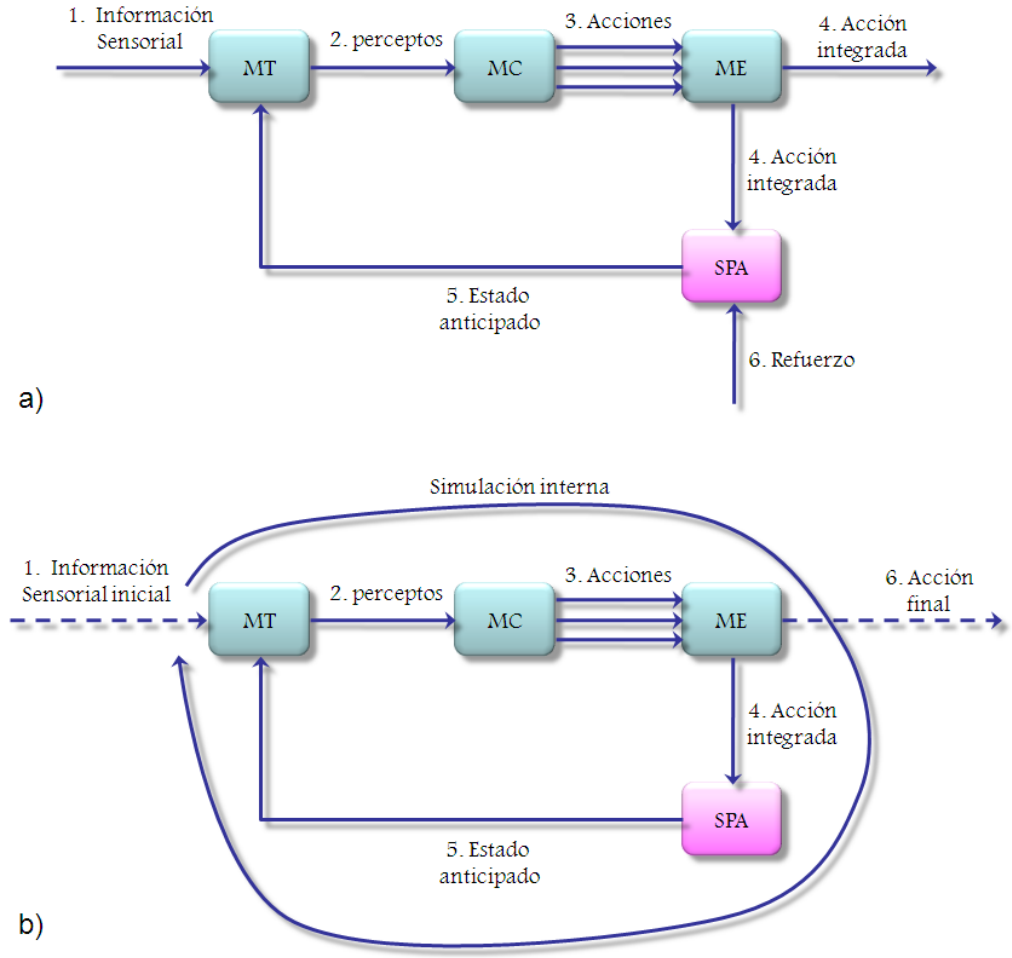


Figura 8.9: Fases de operación del SPA. MT: Memoria de Trabajo. MC: Módulo Comportamental. ME: Módulo Efector. SPA: Sistema de Prospección y Anticipación. a) Fase de Aprendizaje. b) Fase de Ejecución.

8.7. Administración de la Memoria de Trabajo

8.7.1. Inspiración del Modelo

La memoria de trabajo (MT) o memoria operativa, es el "espacio mental" de trabajo, necesario para el recuerdo episódico y semántico, para el pensamiento y la toma de decisiones, para la comprensión del lenguaje y el cálculo mental, y en general, para todas las actividades cognitivas que requieren atención y procesamiento controlado. Esta memoria está constituida por procesos y representaciones activados de forma temporal, "implicados en el control, la regulación y el mantenimiento activo de información relevante para una tarea, al servicio de la cognición compleja". Neurofisiológicamente, está relacionada con la corteza prefrontal dorsolateral la cual desempeña un papel crucial en el almacenamiento temporal (Baddeley, 1991).

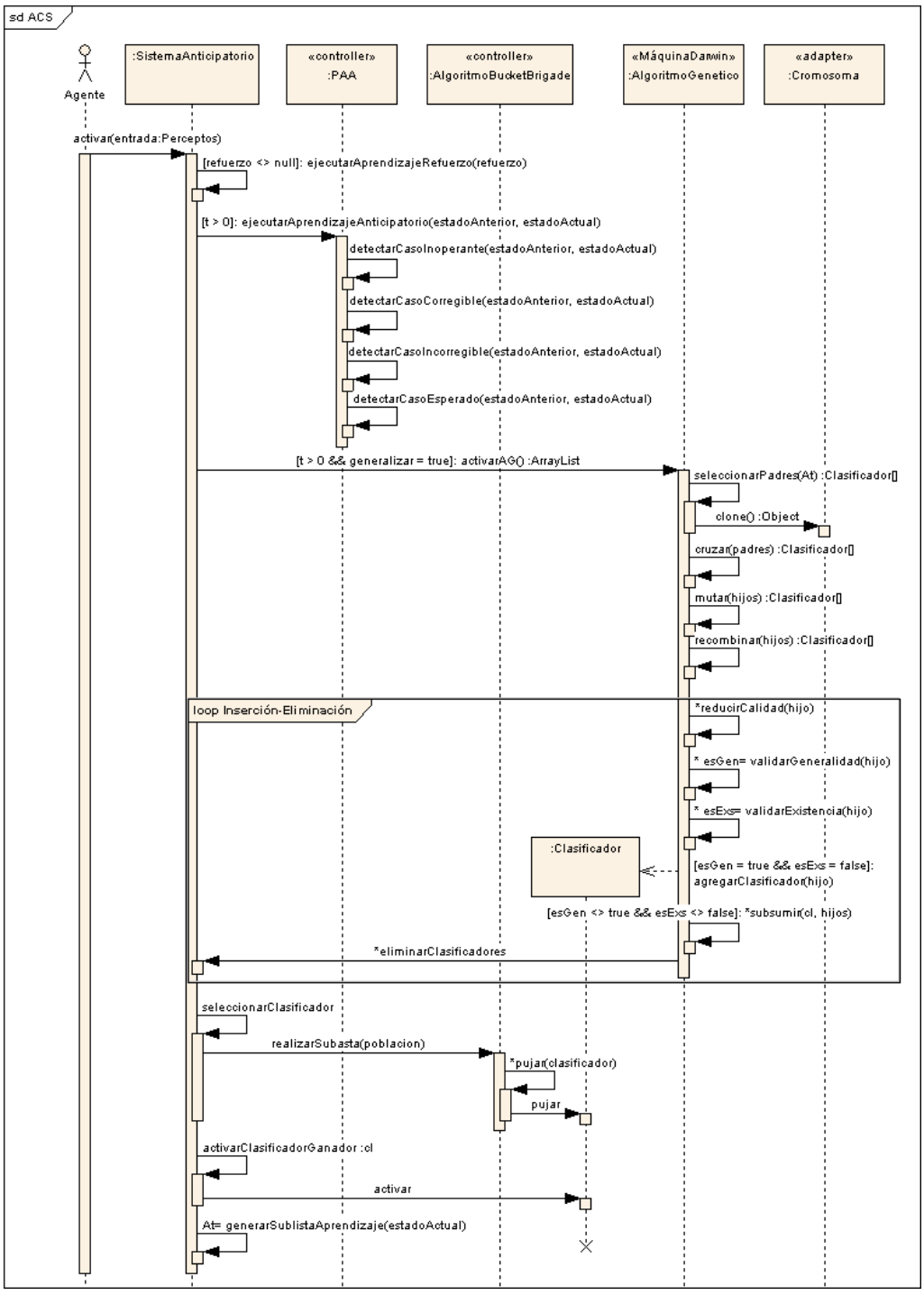


Figura 8.10: Diagrama de Interacción para el Sistema Anticipatorio.

Desde el punto de vista de la metáfora de Baars, la MT es el escenario donde los actores declaman sus papeles y el público puede verlos y escucharlos. Baars propone la MT como un "Área de Trabajo Global", teoría que derivó de la integración de los

trabajos anteriores realizados sobre cognición, por autores como Allan Newell (Newell, 1990), Herbert A. Simon (Newell y Simon, 1976), John Anderson (Anderson y Lebiere, 1998), entre otros.

La MT es una memoria de corto plazo que provee un acceso global a los contenidos subconscientes (e.g., las memorias autobiográficas y declarativas, los automatismos, los sistemas de reconocimiento de objetos y de situaciones, las funciones del lenguaje, entre muchos otros). Por tanto, la MT puede acceder, de forma global y paralela, a un vasto número de procesos subconscientes, aunque su capacidad de mantenerlos dentro de un área consciente, sea notablemente limitada y operacionalmente secuencial (Miller, 1956), en analogía con la parte del escenario donde el foco de luz ilumina sólo unos cuantos actores, quienes dicen su discurso uno a la vez, mientras los demás permanecen en la sombra a la espera de su turno para interpretar su papel.

8.7.2. Modelo Computacional de la MT

La Memoria de Trabajo le conciernen las siguientes funciones dentro del módulo de Orquestación de la Consciencia:

1. *Efectuar el Almacenamiento de Información*: la MT se encarga de almacenar tanto información ambiental capturada por los sensores y interpretada por el módulo de Percepción (en forma de perceptos), así como información del estado interno del agente (i.e., metas actuales, acciones ejecutadas por los *Comportamientos* y automatismos, recuerdos extraídos de las memorias declarativas, y estados anticipados).
2. *Controlar la Capacidad Limitada de Almacenamiento*: la MT modula el desvanecimiento gradual de la información que ha almacenado a fin de mantener únicamente aquello que sea relevante para el agente. Esto se debe principalmente a la cualidad limitada que tiene la MT de almacenar información (lo que acelera y facilita el acceso de los contenidos dentro de la misma), y que según los estudios en el área de las ciencias cognitivas, se corresponde con una capacidad de 7 ± 2 unidades o chunks de información (Miller, 1956).
3. *Actualizar el Nuevo Conocimiento Adquirido*: la MT se encarga de activar los procesos que consolidan el nuevo conocimiento dentro de las memorias de largo plazo.

En la figura 8.11 se presenta un esquema de interacción de la MT con los procesos subconscientes de METÁFORA. A continuación se describe brevemente este proceso de interacción en la MT.

1. La MT recibe los perceptos que han sido creados por el módulo de Percepción. Antes de agregar un nuevo percepto, la MT valida si éste ya ha sido previamente almacenado (en ciclos cognitivos anteriores), de ser así, incrementa su intensidad de activación, de lo contrario, lo agrega en la memoria con una intensidad por defecto. Recuérdese que los perceptos son construidos por el módulo de Percepción a

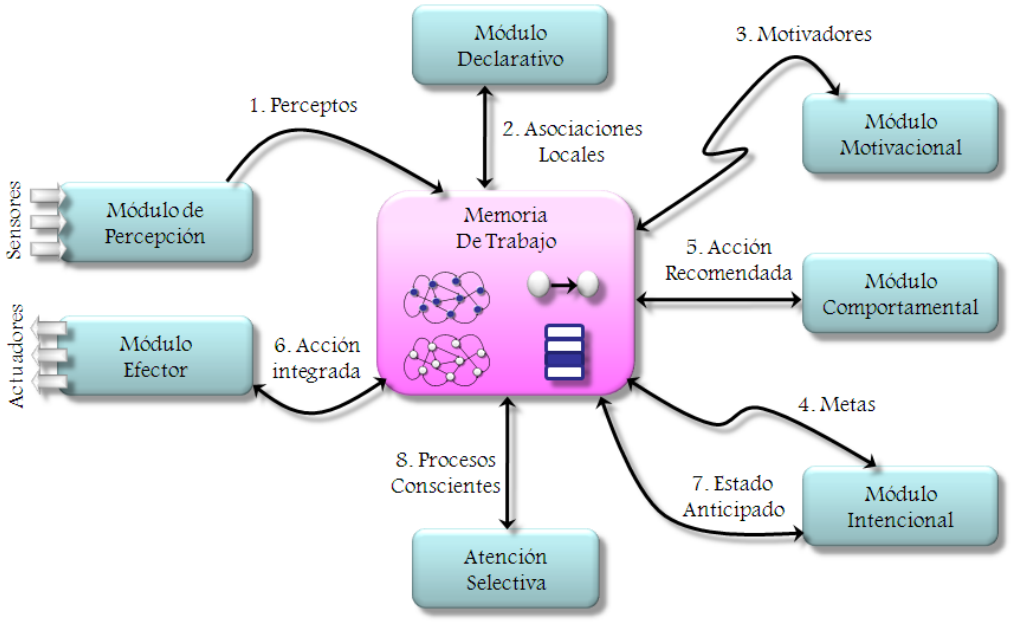


Figura 8.11: Interacción con la Memoria de Trabajo.

partir de la información sensorial percibida y de las relaciones semánticas inferidas por el Razonador en el módulo Declarativo.

2. La MT extrae conocimiento de la memoria declarativa de largo plazo a partir de los perceptos que se encuentran activos en el estado actual. Una vez el módulo Declarativo recibe los perceptos enviados por la MT, los traduce en forma de chunks y los emplea luego para extraer conocimiento de las memorias episódicas (simbólicas y sub-simbólicas). Estos nuevos chunks extraídos de la memoria declarativa de largo plazo representan asociaciones locales que son almacenadas en la MT. Adicionalmente, el conocimiento acerca de las situaciones nuevas que experimenta el agente junto con las relaciones semánticas entre los objetos que hacen parte de dichas situaciones, es consolidado en las memorias declarativas de largo plazo (semántica y episódica).

3. El módulo Motivacional envía a la MT, el conjunto de motivadores activos (es decir, aquellos motivadores que superen cierto nivel de intensidad).

4. La estructura de metas en el módulo Intencional determina qué metas (o sub-metas) se encuentran activas en el estado actual. Luego, estas metas activas son enviadas a la MT para su almacenamiento temporal.

5. El módulo Comportamental recibe la información almacenada en la MT que se encuentra aún vigente (i.e, las unidades de la MT que aún tienen una activación que sobrepasa cierto umbral), y ejecuta el proceso de razonamiento que da lugar a un conjunto de acciones recomendadas (una por cada componente del *Comportamiento* activo). Estas acciones son almacenadas en la MT.

6. La MT envía al módulo Efector el conjunto de acciones recomendadas por el módulo Comportamental, con el objetivo de que sean moduladas e integradas en una sola salida que ejecutará el agente en su entorno.

7. El módulo Intencional, a través del Sistema de Prospección y Anticipación (SPA), genera un nuevo estado anticipado que representa la *expectativa* que tiene el agente acerca del cambio que producirá en el entorno la ejecución de la acción integrada en el paso anterior. Esta predicción sobre el estado anticipado puede ser empleada como una entrada adicional que permite a los automatismos del módulo Comportamental, generar una respuesta reactiva ante el entorno cambiante.

8. El mecanismo de atención selectiva determina qué contenidos de la MT se harán conscientes y cuales se mantendrán temporalmente en un estado subconsciente. Los contenidos conscientes dirigirán la activación de *Comportamientos* en el siguiente ciclo cognitivo.

Como se observa en la figura 8.11, la MT está conformada por un conjunto de elementos (denominados *unidades-MT*) los cuales se encuentran separados en *espacios de memoria* diferentes, de acuerdo con la naturaleza de los mismos (i.e., perceptos, metas, acciones, asociaciones locales, estados anticipados, automatismos activos, etc.). Esto resulta consistente con la teoría de Baddeley (Baddeley, 1991) quien argumenta que la MT se compone de espacios de memoria independientes de capacidad limitada, donde son almacenados diversos tipos de información.

En METÁFORA, una *unidad-MT*, es un chunk de memoria representado de la forma familiar: ⟨`propiedad, valor`⟩. La cantidad de *unidades-MT* para cada espacio de memoria se determina de acuerdo con las características del dominio del problema. Por defecto, se emplea una capacidad de 7 elementos por cada espacio de memoria[27]. Así mismo, la capacidad total de la MT es limitada y está supeditada a la suma de las diferentes capacidades para cada espacio de memoria.

Cada *unidad-MT* posee una activación de nivel base (ANB) la cual determina el tiempo que permanecerá activa la unidad en la MT antes de ser eliminada. Esta activación puede ser incrementada en cada ciclo cognitivo con la presencia continua de los mismos elementos que conforman tales unidades (i.e., perceptos, metas, chunks episódicos, etc.), o disminuida por la ausencia de los mismos. La ecuación que describe la activación de nivel base para cada unidad-MT en función de una tasa de desvanecimiento gradual, es la siguiente:

$$ANB_i^{mt} = iANB_i^{mt} + c \times \sum_{l=1}^{n} t_l^{-d} \tag{8.24}$$

[27] Según estudios sobre la MT, existe cierta evidencia (Miller, 1956; Baddeley, 1991; Calvin, 2001; Halgren et al., 2002; Frank et al., 2000) para creer que la capacidad limitada de este tipo de memoria temporal corresponde a un valor de 7 ± 2. Esto depende, en los humanos, de muchos factores tales como la edad, la raza, entre otros.

donde: i indica una unidad de la memoria de trabajo. $iANB_i^{mt}$ es el valor de la activación inicial (por defecto es 0). l es la l-ésima vez que se activa (o re-activa) el mismo chunk i en la memoria de trabajo mt. t es el tiempo transcurrido desde la última actualización de l. d es una tasa de degradación. c es una constante de amplitud. Los valores por defecto de los parámetros son: $d = 0.8, c = 3$.

La activación de nivel base representa la probabilidad de necesitar una *unidad-MT* en particular basándose en su historia pasada (Anderson, 1993b). Por tanto, el desvanecimiento asintótico del nivel de activación de cada *unidad-MT* resulta directamente proporcional a la probabilidad de necesitar dicha unidad-MT.

Cuando la ANB de una unidad-MT se encuentra por debajo de un umbral $umbral_{elim}$, la unidad-MT es eliminada de la memoria de trabajo. Por el contrario, cuando la unidad-MT sobrepasa un umbral $umbral_{act}$ es activada con una intensidad de "1" y es mantenida dentro de la región de los contenidos conscientes (los contenidos que están bajo el foco de atención).

Esta imagen de la memoria de trabajo es, de algún modo, similar a la teoría de Baddeley (Baddeley, 1991) en términos de: (1) la existencia de múltiples bucles de activación que forman gradualmente las unidades-MT, (2) la formación de unidades temporales de memoria que sirven para el ensayo y la simulación interna de actuaciones, y (3) la separación de espacios temporales de memoria según su naturaleza. También coincide con la perspectiva de Anderson (Anderson, 1993b) concerniente a la propagación y degradación de la activación en las unidades de la MT. Con respecto a la teoría del Área de Trabajo Global de Baars y Franklin (Franklin et al., 2005), resulta similar en cuanto a que la MT es un mecanismo de acceso global de los procesos subconscientes. Finalmente, contrasta con la idea de Sun (Sun et al., 2001) quien propone (en su arquitectura cognitiva *Clarion*) que la MT sólo puede actuar durante el proceso de toma de decisiones y que ésta no se encuentra en capacidad de proveer un acceso global a los contenidos.

8.8. Discusión

En este capítulo se ha presentado un conjunto variado de procesos que operan de forma subconsciente, los cuales contextualizan la escena donde tendrá lugar la actuación de los procesos conscientes del agente. A continuación, se mencionan los aportes más significativos de este capítulo y se discuten asimismo algunos aspectos que podrían ser tenidos en cuenta en posteriores mejoras de la arquitectura.

En la sección 8.6.3 se discutió el mecanismo epigenético que dirige el proceso de predicción en el módulo de Prospección y Anticipación de la arquitectura. En estas líneas, merece la pena hacer hincapié sobre el cómo, a partir de información sub-simbólica arbitraria, es producido un conjunto de estructuras anticipatorias (*expectativas* del agente) que predicen el estado futuro en el cual se encontrará el sistema dada una acción ejecutada por el agente. De forma consistente con las hipótesis propuestas en la sección 5.8, a partir de un conjunto de estructuras cognitivas cuya representación sub-simbólica inicial es arbitraria, son generadas nuevas estructuras predictivas "confiables" como fruto del proceso evolutivo que articula una Máquina Darwiniana. Una vez alcanzado cierto nivel

evolutivo de tales estructuras predictivas, éstas pueden emplearse durante la gestación de dos procesos deliberativos: (1) en el mecanismo de atención selectiva, las expectativas ejercen una influencia importante sobre la orientación del foco de atención y por ende de la percepción del agente (véase la sección 9.2.1); (2) en el proceso de simulación interna donde son probados diferentes escenarios alternativos de actuación antes de que sean ejecutados por el agente en el mundo en el que se encuentra inmerso (véase la sección 10.3.3).

Otro aspecto importante que cabe resaltar sobre el desarrollo de este capítulo, es el modelado híbrido de las memorias declarativas. A pesar del esfuerzo "doble" que requiere el modelado tanto del nivel cognitivista como del nivel emergente de estas mermorias, se obtiene el beneficio del aspecto complementario: mientras el nivel emergente dirige el aprendizaje implícito de nuevo conocimiento, el nivel cognitivista extrae conocimiento sub-simbólico que luego es representado como conocimiento explícito comunicable, y éste último a su vez, "supervisa" el aprendizaje y los procesos de generalización llevados a cabo por las redes neuronales del nivel emergente. Así pues, la integración vertical de estos dos niveles, similar al mecanismo descrito para los *Comportamientos*, provee un alto grado de consistencia y complementariedad entre los dos tipos de representaciones.

Es evidente que algunos de los apartados tratados en este capítulo puedan ser especificados con un mayor nivel de detalle o incluso ser reemplazados por modelos computacionales existentes que hayan demostrado un mayor nivel de sofisticación y eficacia. Concretamente:

- En el modelo cognitivo de Franklin y Baars (Franklin et al., 2005) se propone una división taxonómica interesante sobre los distintos sistemas de memoria que se pueden identificar en los humanos. Según este modelo resulta necesario definir, además de las memorias declarativas y procedimentales de largo plazo y de la memoria de trabajo, una Memoria Sensorial y una Memoria Perceptual. La memoria sensorial registra, sin un procesamiento relativamente sofisticado, los estímulos sensoriales entrantes. Esta memoria provee un área de trabajo común donde es integrada la información multimodal captada por los sentidos: visual, auditiva, táctil, etc. Por otro lado, la memoria perceptual juega un rol importante en el reconocimiento, categorización e interpretación de los objetos percibidos y sus relaciones.

  En el diseño de METÁFORA, estas dos memorias no están representadas como espacios de almacenamiento independientes sino que se emplea un modelo de representación integrado: los perceptos, los cuales representan la información sensorial que ha sido interpretada como resultado de las inferencias realizadas sobre la memoria semántica. La propuesta de Franklin y Baars se fundamenta en la idea de que estos tipos de memoria requieren distintos espacios de almacenamiento donde se lleven a cabo procesos diferenciados de extracción y aprendizaje.

  Resultaría interesante incluir en posteriores versiones de METÁFORA, un mecanismo de *aprendizaje por imitación* basado en las hipótesis sugeridas en el área de la neurociencia sobre estos dos tipos de memoria: estudios a través de resonancias magnéticas demuestran que la circuitería neuronal en el neocórtex prefrontal-temporal (Memoria Sensorial) y en la corteza perinasal (Memoria Perceptual) son

las principales áreas del cerebro de los mamíferos que intervienen, junto con la región hipocampal, en los procesos donde son recreados, onomatopéyicamente, secuencias de sonidos y movimientos (Alain et al., 1998; Davachi et al., 2003).

- Con el fin de establecer en METÁFORA un mecanismo de representación unificado para las memorias declarativas, la memoria procedimental (en el nivel cognitivista de los *Comportamientos*) y la memoria de trabajo, se propone el *chunk* como unidad de almacenamiento de conocimiento. No obstante, a pesar de los beneficios que provee el emplear una representación unificada (e.g., los chunks pueden ser fácilmente convertibles de un tipo de memoria a otro: *chunk-procedimental* $\rightarrow$ *chunk-episódico* $\rightarrow$ *chunk-MT*, etc.), existen modelos computacionales actualmente desarrollados que optimizan este tipo de tareas.

  Por ejemplo, el sistema *slipnet* (Hofstadter y Mitchell, 1994) es una vasta red de nodos semánticos de activación por propagación, que podría administrar de manera optimizada, los procesos de extracción y asimilación de conceptos en la memoria semántica y en la memoria perceptual de la arquitectura. En cuanto a la memoria declarativa simbólica, ésta podría ser representada empleando una memoria de acceso aleatorio como la que emplea la Red Kanerva (Sparse Distributed Memory) (Kanerva, 1988). La Red Kanerva puede ser empleada para almacenar y recuperar grandes cantidades de datos de manera eficiente, sin requerirse que los datos de entrada sean completamente precisos ni requerirse tampoco la definición exacta de aquella información que se desea recuperar. A pesar de que éste no es un modelo exacto de cómo funciona la memoria humana, sí que comparte suficientes características como para sugerir que la memoria humana funciona de manera similar (Kanerva, 1988). Por último, un tema que atrae mucho a los investigadores en el área de los agentes inteligentes y el cual resultaría interesante considerar en futuras mejoras de METÁFORA, es el modelado de creencias a través de semánticas descriptivas u ontologías, cuya cualidad principal es la definición de un vocabulario común que facilita la comunicación y el intercambio de información entre diferentes sistemas y entidades.

  Lo que se quiere hacer realmente hincapié en estas líneas, es que la decisión de "no" emplear tecnologías existentes en el modelado de las memorias declarativas del nivel cognitivista, recae en la necesidad de definir una representación unificada que permita integrar la heterogeneidad de dichas memorias, y de esta manera, se generen unidades de conocimiento que puedan ser comprensibles y manipulables por el resto de los módulos de la arquitectura. Así pues, se ha decidido sacrificar, en este caso en particular, el desempeño y la velocidad de recuperación de la información a favor de mantener la homogeneidad representacional del conocimiento a través de los chunks.

- En el apartado 8.2.2.3 se trató el mecanismo de razonamiento semántico el cual emplea un proceso inferencial por el cual son activados conceptos relacionados con un conjunto de estímulos percibidos. Debido a que el mecanismo propuesto en METÁFORA funciona bajo el supuesto de que la información percibida es completa y suficiente, resultaría conveniente incorporar otros mecanismos complementarios de razonamiento los cuales pudiesen realizar inferencias bajo incertidumbre. Un claro ejemplo de esto podría ser el *razonamiento basado en similitudes* (similarity-

based reasoning) (Hullermeier, 2001), un tipo de razonamiento por aproximación que asocia la afinidad que existe entre los estímulos percibidos y los chunks de la memoria para realizar inferencias por inducción. Otro ejemplo es el *razonamiento causal* (Kemp et al., 2006), que a diferencia del anterior, se enfoca en las relaciones entre propiedades y no en las relaciones de similitud entre objetos. Por último, se sugiere que una aproximación complementaria donde se combinasen múltiples enfoques de razonamiento produciría un estado inferido más enriquecido y aproximado a la realidad.

- En cuanto al modelo motivacional de METÁFORA descrito en la sección 8.4, se propuso una aproximación basada únicamente en *emociones básicas* o *motivadores*. Este modelo simplificado resulta útil en cuanto a que refleja las características necesarias para comprender cómo el comportamiento global del agente está bajo la influencia de las motivaciones del mismo, no obstante, sería interesante incorporar en el diseño de METÁFORA un módulo motivacional más robusto, basado en las investigaciones actuales sobre *Agentes Emocionales*, donde el modelado de los sentimientos y las emociones artificiales influirían y dirigirían los procesos cognitivos en el agente, tales como la racionalidad, la toma de decisiones, el aprendizaje de nuevo conocimiento significativo o el olvido de éste, la anticipación basada en expectativas y la planificación deliberativa de acciones futuras. Por tanto, la aproximación actual del módulo Motivacional de METÁFORA podría ser mejorada por un modelo extendido donde se tengan en cuenta los *Estados Afectivos y de Ánimo*, los *Estados Físicos*, los *Rasgos de la Personalidad*, las *Actitudes*, y las *Emociones* del agente, con el propósito de hacer más "creíble" la actuación de éste en relación a la comunicación con otros agentes artificiales o con humanos. Algunos modelos que pueden ser tomados en cuenta son: KISMET (Breazeal, 2003), Cognitiva (Imbert, 2005), Cathexis (Velásquez, 1996), y Empathy (Catucci et al., 2004).

Capítulo 9

Orquestación Adaptativa de la Experiencia Consciente

*"... **theatre metaphor** gives us the easiest way to think about* consciousness*. A theatre combines very limited events taking place on stage with a vast audience, just as* consciousness *involves limited information that creates access to a vast number of unconsciousness sources of knowledge. **Consciousness** seems to be the publicity organ of the brain. It is a facility for accessing, disseminating and exchanging information, and for exercising global coordination and control".*
— B. Baars, In the theatre of Consciousness, 1997.

En el presente capítulo se confecciona un modelo computacional para la orquestación adaptativa de los procesos que operan en un nivel subconsciente de la arquitectura (descritos en los capítulos 7 y 8), los cuales compiten por participar en cada experiencia consciente del agente.

Cabe notar que en este capítulo se adopta una perspectiva intermedia entre la descripción fenomenológica y neurofisiológica de la *consciencia*, la cual resulta ser la posición más apta para capturar las características fundamentales que se quieren reflejar en el módulo de Orquestación de la Consciencia de METÁFORA. Adicionalmente, el sentido que se da en esta tesis al término de *consciencia*, se debe advertir desde el principio, es el de *accesibilidad*, y no el más básico de consciencia sensorial (cualitativa), ni el de consciencia de orden superior (donde surge la noción del yo y la memoria simbólica en forma de lenguaje). A todo este nivel de descripción constreñido y reduccionista es lo se denominará de ahora en adelante la *consciencia sintética* del agente.

Como se mencionó en capítulos anteriores, METÁFORA define múltiples niveles jerárquicos de organización. El nivel de organización descrito en el presente capítulo se soporta sobre componentes subyacentes auto-organizados, los *Comportamientos*, que a su vez se conforman de sub-componentes de menor escala organizativa (las redes neuronales, los sistemas inmunes, y las reglas de producción). Así pues, la orquestación de la consciencia sintética emerge como un nuevo nivel de organización, o mejor aún, de auto-organización si se considera que éste nivel funciona como una máquina autopoiética que

auto-regula sus propias dinámicas internas de producción, transformación y destrucción de componentes. Este nuevo nivel de organización a su vez formará la base para niveles cognitivos de mayor complejidad (i.e., la planificación, la resolución de problemas, ... y en una perspectiva más ambiciosa, la imaginación, la creatividad, etc.).

Durante el desarrollo de esta capítulo se presenta un modelo computacional para la modulación de la consciencia sintética que consta de dos partes: inicialmente, se propone un modelo para la orquestación de *Comportamientos* basado en un mecanismo de redes de dispersión de energía de activación; y posteriormente se propone, como uno de los aportes principales de esta tesis, un modelo evolutivo que refina la estructura y conectividad de tales redes de dispersión de activación a través de un desarrollo ontogenético basado en procesos darwinianos. Como resultado, la arquitectura es dotada con cierta capacidad auto-organizativa a través de la cual, el agente, desarrolla adaptativamente (a lo largo de su ciclo de vida) nuevas estructuras cognitivas que le permiten articular los procesos de la actuación consciente y subconsciente.

9.1. Inspiración del Modelo de Consciencia Sintética

En el ámbito del estudio científico, se han propuesto diversas hipótesis que describen de manera interesante, desde diferentes perspectivas, la fenomenología de la *consciencia*. De esta manera, algunas de las las teorías que han tenido mayor aceptación dentro de la comunidad científica, se podrían distinguir en dos categorías: las *Teorías Neurológicas* y las *Teorías Psicológicas*.

Desde la perspectiva de las teorías neurológicas, una de las más destacadas es la de (Edelman, 1987), la cual se apoya en la teoría darwinista y se enmarca en la llamada TNGS (por su acrónimo en inglés: Theory of Neuronal Group Selection). En primer lugar, Edelman hace una distinción fundamental entre una conciencia primaria y una conciencia de orden superior.

La *consciencia primaria*, la cual servirá de inspiración en el desarrollo del módulo de Orquestación de la Consciencia de METÁFORA, consiste en la capacidad que posee el individuo de ser consciente del mundo que le rodea y de formar imágenes mentales del momento presente, integrando aspectos del pasado y del futuro como partes de esa escena correlacionada. Los mecanismos neurofisiológicos que harían posible esta hipótesis, están soportados por un sistema de valor, compuesto por el troncoencéfalo y el sistema límbico, y por un sistema tálamo-cortical, que comprende principalmente la corteza y los núcleos talámicos. Por otro lado, la *consciencia de orden superior*, otorga la capacidad de construir modelos del "yo" como una entidad separada del entorno. El modelado de este tipo de consciencia superior, está fuera del alcance de este trabajo de tesis, aunque podría ser contemplada como una línea de profundización dentro del trabajo investigativo futuro.

Por otro lado, desde la perspectiva psicológica, la consciencia es mejor descrita en el ámbito de lo metafórico. Las metáforas más relevantes sobre la consciencia fueron mencionadas en la sección 4.6.1. La metáfora en la cual se inspira el diseño del presente módulo es la *metáfora del teatro de la consciencia* propuesta por Baars (Baars, 1997).

Para Baars, la conciencia es el órgano de publicidad de la mente. Un altavoz que vocea a la sociedad de la mente, a la audiencia del teatro, un mensaje. Para comprender de mejor forma la definición de Baars se ofrece el siguiente ejemplo: "Cuando se pierde un niño en la playa, por megafonía avisan a todo el mundo: a la policía, a los servicios de salvamento y a los bañistas. No es una comunicación selectiva, sino una difusión masiva de la información. La megafonía describe a un niño de tres años, rubio y con bañador rojo. El público responde al mensaje en función de su conocimiento particular. Alguien avisa, un daltónico, yo vi pasar a un niño de tres años, pero ignoro el color de su bañador. Un ciego afirma haber oído llorar a un niño solitario. Alguien cree recordar a una cabellera rubia que se movía detrás de una valla no muy alta". Todos son indicios que, interconectados, permiten localizar al niño. Así funciona la mente humana, cada módulo cerebral hace su trabajo (detecta los colores, detecta las formas...) y la consciencia permite que trabajen todos juntos por un objetivo común, aunque no lo sepan, coordinando todo este trabajo que no se produce de forma secuencial sino en paralelo.

A pesar de que la consciencia humana puede acceder a un extenso número de contenidos subconscientes de forma paralela, ésta es limitada y sólo puede procesar un flujo secuencial de información. Así por ejemplo, un conductor experimentado puede conducir un coche de forma cuasi-automática y a la vez procesar en paralelo otros procesos subconscientes (como respirar, controlar la temperatura corporal, recuperar recuerdos de una trayectoria en particular, etc.), mientras que sostiene conscientemente un diálogo interno sobre "qué va a preparar para cenar". En el momento en que el conductor deba enfrentarse a una ruta de desvío por reparaciones en la calzada, un nuevo proceso consciente emergerá entre aquellos subconscientes para hacerse con el control ("recalcular nueva trayectoria para llegar al destino") y relegar así, al nivel de lo subconsciente, el pensamiento que se encontraba bajo el foco de atención justo en el momento anterior ("¿qué voy a preparar para cenar?"). De manera similar, en el módulo de Orquestación de la Consciencia de METÁFORA se pretende simular los procesos de competición y cooperación que ocurren en paralelo entre los contenidos subconscientes, cuyo fin es producir un flujo de activación secuencial de *Comportamientos*, los cuales a su vez ejecutan acciones externas o internas.

Este efecto de *cuello de botella* por el cual un extenso número de contenidos subconscientes compiten y forman coaliciones entre sí, pasando la frontera de lo subconsciente y situándose secuencialmente en un primer plano (i.e, la experiencia consciente), es modulado por el mecanismo de *Atención Selectiva* de METÁFORA. Retomando la metáfora de Baars, los contenidos de la consciencia pueden ser guiados de forma voluntaria y espontánea, como el foco de luz que ilumina el escenario, que en este caso corresponde a la memoria de trabajo. Recuérdese que la MT no es completamente consciente (como se explicó en la sección 8.7), sólo es parte de la experiencia consciente la región bajo el foco de atención: *"el teatro (la consciencia) posee un foco de luz muy potente (la atención), donde sólo los eventos que se encuentran en el área iluminada del escenario (la MT) son los únicos que se mantienen de forma consciente"*.

La atención desempeña un importante papel en el proceso cognitivo, tal es así que han sido múltiples los esfuerzos realizados por muchos autores por definirla, estudiarla, y delimitar su estado entre los procesos psicológicos. Reátegui (Reátegui y Sattler, 1999) señala que la atención es un proceso discriminativo y complejo que acompaña todo el

procesamiento cognitivo, además de ser la responsable de filtrar la información y asignar los recursos para permitir la adaptación interna del sistema cognitivo, en relación a las demandas externas. Otros autores como Rosselló (Rosselló-Mir, 1999), Ruíz (Ruiz Vargas y Botella, 1987), y García (García Sevilla, 1997), consideran que la atención es un mecanismo que pone en marcha los procesos que intervienen en el procesamiento de la información y facilita el trabajo de todos los procesos cognitivos ejerciendo un control sobre ellos.

Se puede decir entonces, que la atención selectiva es la actividad que genera y controla los procesos y mecanismos mediante los cuales el agente procesa tan sólo una parte de toda la información percibida, y como consecuencia, el agente sólo da respuesta a aquellos aspectos del ambiente que considere interesantes e importantes. Así pues, se puede argumentar que la selectividad atencional de METÁFORA implica dos aspectos: (1) la selección del conjunto de contenidos subconscientes en la MT (i.e., perceptos, metas, motivadores, anticipaciones, recuerdos, etc.), y (2) la selección de los procesos y respuestas que se van a ejecutar (i.e., los *Comportamientos*). En ambos casos existe una clara función adaptativa. En el primero de ellos, porque el procesamiento selectivo de la información permite que no se produzca una sobrecarga del sistema cognitivo ante la gran complejidad de la información de entrada (de acuerdo con los principios de (Broadbent, 1958)); y en la segunda, porque la selección de los *Comportamientos* más adecuados, una vez procesada la información, evita que se produzca una parálisis del sistema cognitivo cuando el entorno requiere que tengan lugar respuestas incompatibles de forma simultánea (en concordancia con los postulados de (Posner, 1978; Shallice, 1972)).

En la figura 9.1, se presenta un esquema integrativo entre el proceso de orquestación de la consciencia sintética, los procesos subconscientes, los *Comportamientos*, y la memoria de trabajo. El escenario (la MT) recibe información de todos los procesos que se ejecutan subconscientemente (perceptos, recuerdos de las memorias declarativas, metas, anticipaciones, etc.), pero sólo los eventos que son iluminados por el foco de atención se encuentran completamente conscientes. Cada actor (*Comportamiento*) ejecuta su actuación, de forma secuencial, bajo la influencia de los operadores contextuales (las metas globales y los planes deliberativos del agente), los cuales se corresponden con la funcionalidad de las personas que trabajan tras bambalinas (el director, el dramaturgo, etc.). Dado el estado actual del agente, el foco de atención selecciona los actores más relevantes sobre el escenario, y una vez iluminados, sus locuciones (i.e., las acciones recomendadas por cada *Comportamiento*) son distribuidas a la audiencia, la cual se compone de todas las rutinas subconscientes y repositorios de conocimiento (memorias declarativas) del agente.

En este punto merece la pena hacer una importante observación. En la figura 9.1 se encuentran dos mecanismos inmersos: una *entrada convergente* y una *salida divergente* de información. Sobre el escenario convergen los *Comportamientos* (actores) y sus discursos, las metas, los planes deliberativos, la información sensorial percibida (perceptos), y el estado anticipado. En cada instante, cada acción recomendada por el *Comportamiento* activo refleja dicha convergencia de entrada. Una vez el agente ejecuta (interna o externamente) la acción recomendada, ésta se dispersa entre la audiencia, provocando efectos desencadenantes en estados futuros del sistema. La acción es transmitida globalmente, aunque es interpretada de forma local por cada uno de los miembros de la

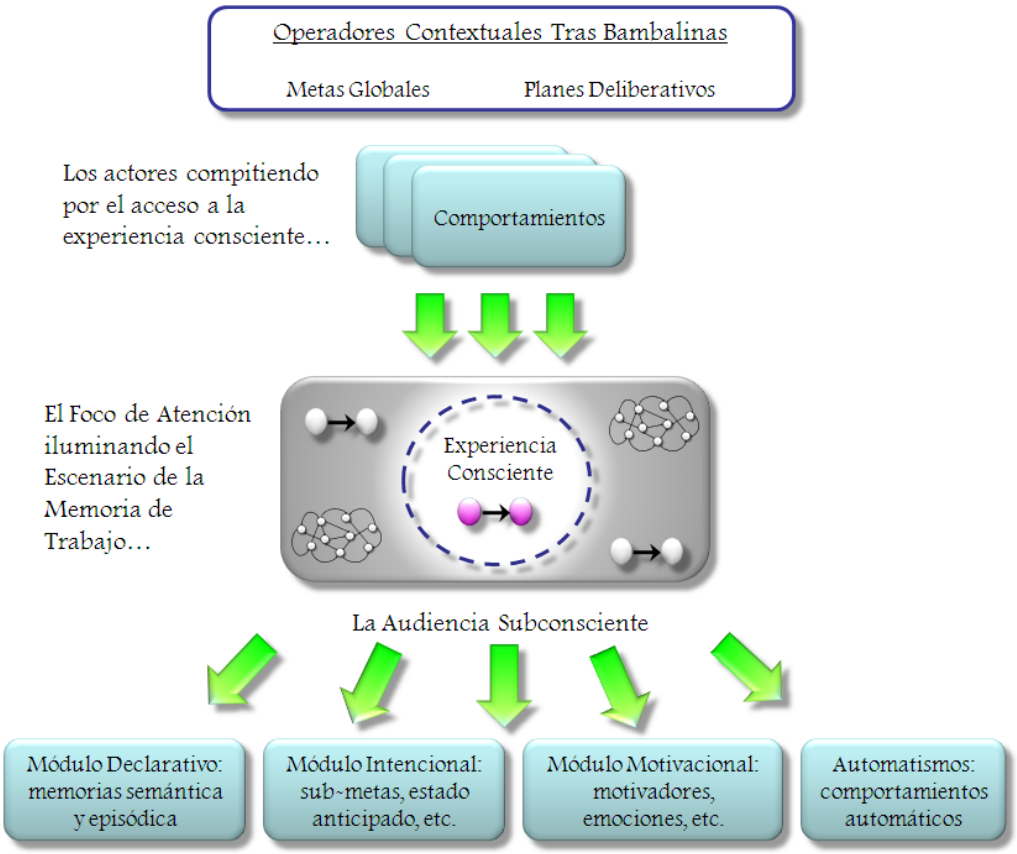


Figura 9.1: Metáfora del Teatro aplicada al Modelo Cognitivo de METÁFORA.

audiencia. El mecanismo de Orquestación de la Consciencia, en su rol de director, se encuentra "escuchando" tras bastidores la actuación de cada *Comportamiento*, y de esta manera "ajusta" adaptativamente la escena en el siguiente estado del sistema. En resumidas cuentas, existe una convergencia masiva de información en la MT (escenario), pero una vez esta información se reúne para dar forma a una nueva experiencia consciente, es dispersada de forma divergente a toda la audiencia.

Desde otra perspectiva, tomando como base las ideas de Edelman (Edelman, 1987) y de Calvin (Calvin, 2001), se propone un modelo donde se pretende incorporar algunos aspectos clave de la consciencia humana, como la capacidad modeladora de la realidad, tanto espacial como temporal, y los procesos implicados en la toma de decisiones. Se propone además, un modelo computacional que podría ser una visión mecanicista sobre cómo surgen, a través de un proceso ontogenético, nuevas estructuras cognitivas de orden superior que representan un conjunto de redes de interacción entre los *Comportamientos* del agente.

En esencia, este capítulo propone cómo el mecanismo de Orquestación de la Consciencia podría funcionar como una Máquina Darwiniana, dirigiendo evolutivamente los constantes cambios de enfoque de la consciencia e incluso aquellos procesos subconscientes que con tanta frecuencia saltan al primer plano sin ser advertidos. Muestra cómo

el agente podría crear múltiples copias de estructuras cognitivas de orden superior (las Redes de Comportamientos) las cuales estarían compitiendo constantemente entre sí por ser la entidad que modula la activación de *Comportamientos*, y presenta además, la manera en que el agente, a partir de estas estructuras cognitivas, podría adquirir cierta capacidad para simular internamente futuras acciones en el mundo real (o virtual), una capacidad que es la esencia de la inteligencia conjeturadora del agente.

9.2. Modulación Adaptativa del Foco de Atención

En esta sección se describe el mecanismo de Atención Selectiva encargado de filtrar la información almacenada en la MT y de articular el proceso adaptativo de activación de *Comportamientos*.

9.2.1. Descripción de la Maquinaria Atencional

> *"Se ha dicho recientemente que la "percepción puede considerarse básicamente como la modificación de una anticipación". Siempre es un proceso activo, condicionado por nuestras expectativas y adaptado a las situaciones. En lugar de hablar de ver y saber, haríamos mejor en hablar de ver y advertir. Sólo advertimos algo cuando estamos buscando algo, y buscamos cuando nuestra atención se despierta a causa de algún desequilibrio, de una diferencia entre nuestras expectativas y el mensaje que nos llega. No podemos tener en cuenta todo lo que vemos en una habitación, pero advertimos si algo ha cambiado".*
> — E.M. Gombrich, Art and Illusion, 1960.

El mecanismo atencional de METÁFORA se compone de un conjunto de elementos que convenientemente se han denominado *Máquinas de Atención* (MAT). Una MAT es un componente especializado que se encarga de dirigir el foco de atención del agente hacia el logro de una meta global en concreto, empleando un subconjunto determinado de *Comportamientos* y un subconjunto filtrado de elementos de la MT (unidades-MT). Formalmente, una MAT se define como la siguiente tupla:

$$MAT_i = \langle \texttt{Meta}_i, \texttt{ RC}_i, \texttt{ Est}^*, \texttt{ GFA} \rangle \tag{9.1}$$

donde: $Meta_i$ es una meta global asociada a la MAT_i. RC_i es una red de *Comportamientos* contextuales cuya interacción permite que se alcance $Meta_i$. Est^* es un subconjunto de unidades-MT que representa una porción filtrada del estado actual observado por el agente sobre la cual se centra la atención. GFA es un Gestor del Foco de Atención, encargado de dirigir las dinámicas de activación entre los *Comportamientos* contextuales.

Como se ha mencionado en secciones anteriores (véase 6.3.1.2), las *Redes de Comportamientos* (RC) son estructuras cognitivas que se encuentran en el nivel de organización inmediatamente superior a los *Comportamientos*[1]. Por ahora, lo que conviene tener

[1]Recuérdese que la arquitectura se compone de múltiples estructuras cognitivas jerarquizadas con cierto nivel de auto-organización. En cada nivel de organización, las estructuras cognitivas forman bucles retroa-

claro es que cada RC asociada a una MAT en particular es una estructura cognitiva que define un subconjunto de *Comportamientos* contextuales que están relacionados entre sí a través de enlaces de activación e inhibición.

Un *Comportamiento* contextual, a diferencia de los *Comportamientos* genéricos a los que se ha hecho referencia hasta el momento, es una estructura que encapsula un *Comportamiento* genérico junto con dos listas de elementos especializadas: una lista de elementos que "se espera" estén presentes en el campo perceptual del agente una vez finalizada la ejecución del *Comportamiento*, y una lista de elementos que "no se espera" estén presentes después de la ejecución del *Comportamiento*. Los *Comportamientos* contextuales serán descritos al detalle en la siguiente sección.

Por su parte, el GFA es un mecanismo de enfoque atencional, el cual coordina las dinámicas de activación e inhibición entre los *Comportamientos* de cada RC. El mecanismo consiste en dispersar un flujo de energía entre los elementos de la red (los *Comportamientos*). Este flujo de energía puede ser positivo, en términos de un flujo que estimula la activación de ciertos *Comportamientos*, o puede ser negativo, en términos de un flujo inhibitorio que evita que los *Comportamientos* asociados se activen. En síntesis, el GFA de una MAT *"modula"* el flujo continuo de energía en la RC asociada, articulando la competencia entre *Comportamientos* (a través de la dispersión de energía negativa inhibitoria) y la cooperación entre los mismos (a través de la dispersión de energía positiva excitativa), con el objetivo de que alguno de estos se haga con el control del foco de atención y a su vez evite que los otros lo hagan.

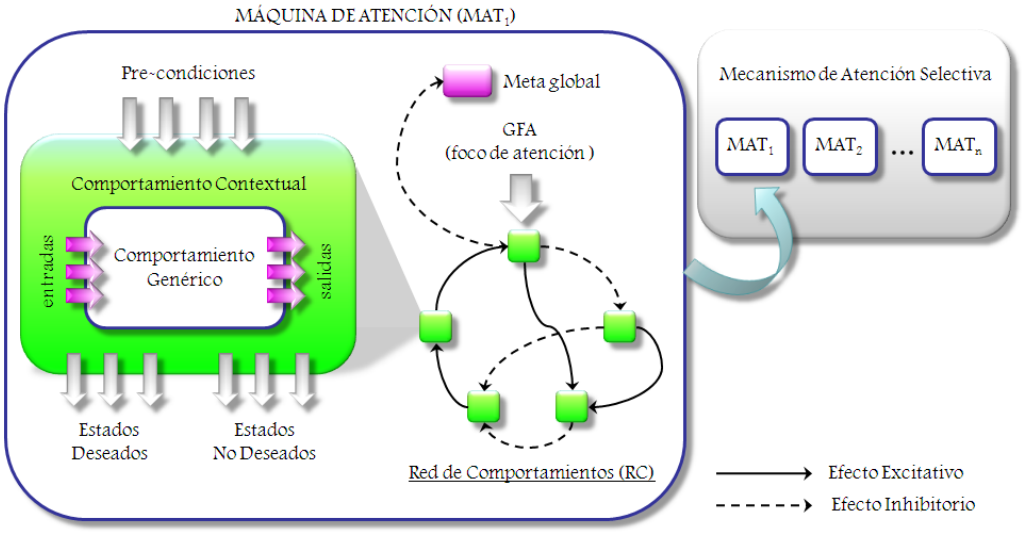


Figura 9.2: Estructura de una Máquina de Atención.

limentados que generan nuevas estructuras más complejas en el siguiente nivel. Así por ejemplo, las reglas sub-simbólicas producidas por las redes neuronales y los sistemas inmunes son las estructuras cognitivas más básicas. El conjunto total de tales reglas forma una nueva estructura más compleja, el *Comportamiento*, cuya interacción con otros *Comportamientos* forma nuevas estructuras cognitivas de mayor complejidad, las Redes de Comportamientos, y la interacción de éstas a su vez genera los Planes...

Una representación gráfica de los componentes de una MAT se presenta en la figura 9.2. Obsérvese cómo la lista de efectos deseados y efectos no deseados en cada *Comportamiento* contextual, produce efectos excitativos e inhibitorios, respectivamente, hacia los otros *Comportamientos* contextuales que conforman la RC.

Adicionalmente a la estructura descrita, cada MAT define un conjunto de características que se corresponden con algunos de los factores más relevantes de la atención identificados desde la perspectiva psicológica:

- *Amplitud*: según Posner (Posner, 1980), la cantidad de información a la que se puede atender simultáneamente no es fija sino variable. Ésta depende de la complejidad de la tarea y de la organización de la información percibida. En este sentido, la amplitud es definida por cada *Comportamiento* contextual a través de un conjunto finito de pre-condiciones que se presuponen deben estar presentes en el estado observado por el agente para que el *Comportamiento* pueda ser activado.
- *Selectividad*: debido a que la atención del agente puede ser selectiva, cada *Comportamiento* genérico define un vector de entradas que filtran el conjunto de estímulos percibidos. Cuando un *Comportamiento* es activado, filtra la información percibida por el agente al procesar sólo un subconjunto de los estímulos entrantes y al descartar aquellos irrelevantes. No obstante, la memoria de trabajo continuamente construye el estado actual empleando toda la información percibida, con lo cual la MAT puede re-dirigir la atención hacia otro *Comportamiento*.
- *Oscilación*: son períodos involuntarios de distracción (en el orden de segundos) a los que está supeditada la atención (Rubenstein, 1982). Esta capacidad para oscilar la atención puede ser considerada como un tipo de flexibilidad que se manifiesta en situaciones diversas, especialmente en las que el agente tiene que re-orientar la atención de forma apropiada porque se ha producido una distracción o porque resulta necesario que atienda a varios estímulos a la vez. En este sentido, el Gestor del Foco de la Atención (GFA) define ciertas dinámicas de dispersión de activación (las cuales serán descritas al detalle en la siguiente sección) que permiten realizar desplazamientos de la atención. Los cambios de atención se producen como consecuencia de un flujo continuo de activación entre los *Comportamientos* de la RC. Este cambio en el foco de atención puede ser producido principalmente por tres factores:

  1. Detección de estímulos novedosos: los estímulos más novedosos o inusuales atraen más la atención. La concepción de hasta qué punto un estímulo es novedoso para un agente, depende de la experiencia previa de dicho agente. Como se verá más adelante, cada RC asociada a una Máquina de Atención recibe una activación del estado percibido. Cuando son percibidos nuevos objetos, estos producen una mayor activación a la red.
  2. Detección de cambios físicos en los objetos: cuando alguno de los atributos de los objetos percibidos cambia (e.g., de tamaño, color o posición), se produce una mayor activación de entrada para estos en la RC.

3. Detección de elementos relevantes: cuando el agente percibe un objeto que forma parte de sus expectativas o de alguna meta específica[2] , éste produce una mayor activación de entrada a la RC.

- *Concentración*: según Kahneman (Kahneman, 1973) la concentración es la capacidad que tiene el individuo de mantener la atención sostenida sobre un estímulo o sobre algún aspecto del mismo, con preferencia sobre otros estímulos. Este aspecto de ignorar cierta información, resulta útil en aquellas situaciones en que haya una sobrecarga de estímulos percibidos, y el agente deba mantener la atención en la ejecución de una tarea. A menos que se produzca una oscilación de la atención, el GFA mantendrá, por un intervalo de tiempo definido, la atención sobre el *Comportamiento* activado. Esto implica que durante dicho intervalo de tiempo, no se efectuarán las dinámicas de dispersión de activación en la RC ni habrán cambios en el foco de atención. Según estudios empíricos documentados (Posner, 1980), este intervalo de atención sostenida corresponde a cortos períodos de tiempo del orden de segundos, que se prolongan continuamente hasta unas pocas horas. No obstante, entre dichos intervalos cortos de tiempo es posible que se produzcan oscilaciones rápidas que luego convergen nuevamente a los estados de mantenimiento sostenido de la atención. Tomando como base la evidencia empírica, se define por defecto un intervalo de concentración para cada GFA de 2 segundos.

El mecanismo de Atención Selectiva de METÁFORA, define pues un conjunto de MATs que se corresponde una con cada meta global definida en el dominio del problema. En las siguientes secciones, se describe al detalle el modelo algorítmico y matemático que subyace a las dinámicas de dispersión de activación propias de cada RC.

9.2.2. Mecanismo para la Orientación del Foco de Atención

A continuación se describe el mecanismo por el cual, el GFA, orienta el foco de atención a través de un algoritmo de dispersión de energía de activación entre los nodos de la Red de Comportamientos. Concretamente, el GFA modula los procesos de competencia y cooperación entre los nodos (*Comportamientos*) de la RC, lo que produce como resultado la activación de un *Comportamiento* ejecutable el cual se hace con el foco de atención del agente durante cierto intervalo de tiempo establecido por la Máquina de Atención (referido como el nivel de concentración del agente).

En cuanto al algoritmo que describe las dinámicas de competencia y cooperación entre *Comportamientos* coordinadas por el GFA, se propone una adaptación al modelo original ideado por Pattie Maes (Maes, 1989) para la selección de acciones en agentes inteligentes. Las mejoras incorporadas en este modelo serán mencionadas a lo largo de esta sección y de la siguiente.

Como se mencionó en la sección anterior, un *Comportamiento* genérico difiere de un *Comportamiento* contextual en que este último, además de incluir una instancia del

[2]Por ejemplo, en el caso de las expectativas, el "estado anticipado" puede predecir la presencia de cierto objeto en un estado futuro del sistema. De la misma forma, un objeto percibido es considerado relevante si forma parte de la intensión de la meta, como por ejemplo, cuando la meta del agente es "buscar alimento" y éste lo encuentra, se dice entonces que el agente ha detectado un elemento relevante.

primero, posee un conjunto de *expectativas* refinadas y especializadas acerca de lo que pueda suceder después de ser ejecutado el *Comportamiento*. Estas *expectativas* se definen en términos de dos listas, una de "efectos deseados" y otra de "efectos no deseados", las cuales especifican las proposiciones que se esperan sean ciertas o no, respectivamente, después de la ejecución del *Comportamiento*. De ahora en adelante, se referirá al término *Comportamiento* para hacer mención a un *Comportamiento* contextual con las características antes descritas. En este punto, considérese que el *Comportamiento* x se conforma únicamente por los siguientes elementos[3]:

$$x = \langle \mathtt{P}_x\mathtt{,}\ \ \mathtt{ED}_x\mathtt{,}\ \ \mathtt{END}_x\mathtt{,}\ \ \alpha_x \rangle \tag{9.2}$$

donde: P_x es una lista de precondiciones que deben satisfacerse antes de activar el *Comportamiento*. ED_x y END_x representan las expectativas del agente después de la actuación de x, en términos de una *lista de efectos deseados* y una *lista de efectos no deseados* respectivamente. Adicionalmente, cada *Comportamiento* define un nivel de activación α.

Los *Comportamientos* están relacionados entre si a través de tres tipos de enlaces: *enlaces sucesores*, *enlaces predecesores*, y *enlaces de conflicto*. Considérese los *Comportamientos* $x = (P_x, ED_x, END_x, \alpha_x)$, e $y = (P_y, ED_y, END_y, \alpha_y)$, se tiene entonces que:

- Existe un *enlace sucesor* entre x e y, por cada proposición q que forma parte de la *lista de efectos deseados* de x y que también forma parte de la lista de precondiciones de y. Formalmente, existe un enlace sucesor de x hacia y, por cada proposición $q \in ED_x \cap P_y$.
- Existe un *enlace predecesor* de x hacia y si existe un *enlace sucesor* de y hacia x. Formalmente, se tiene que: $q \in P_x \cap ED_y$.
- Así mismo, existe un *enlace de conflicto* de x hacia y por cada proposición q que forma parte de la *lista de efectos no deseados* de y y que también forma parte de la lista de precondiciones de x. Formalmente, se tiene que: $q \in P_x \cap END_y$.

Por tanto, la idea intuitiva del modelo es que cada *Comportamiento* emplee este tipo de enlaces para excitar e inhibir otros *Comportamientos* de tal forma que, después de cierto tiempo, la energía de activación se acumule en aquellos *Comportamientos* que representan la "mejor" decisión que pueda tomar el agente dado el estado actual y el conjunto de metas. En el momento en que el nivel de activación del *Comportamiento* exceda cierto umbral y cumpla con las condiciones para ser ejecutable, éste se convierte en un *Comportamiento* activo que obtiene el derecho de ejecutar sus acciones en el ambiente. El patrón de dispersión de la activación entre los *Comportamientos*, así como la entrada de nueva energía de activación dentro de la red, son determinados por el estado actual del entorno y las metas globales del agente:

[3]En la sección 7.2 se propuso una estructura que consideraba además las entradas y las salidas de un *Comportamiento*, no obstante, a fin de simplificar la descripción del proceso de dispersión de energía de activación, considérese solamente los elementos de la ecuación 9.2

- *Activación recibida desde el Estado*: existe una entrada de activación que proviene tanto del estado ambiental como del estado interno del agente, que se dispersa entre los *Comportamientos* que emparejan parcialmente el estado actual (formado por las unidades de la MT). Se dice que un *Comportamiento* empareja parcialmente el estado actual si al menos una de sus precondiciones es verdadera. La activación recibida desde el estado puede estar influida por la detección de elementos novedosos en el campo perceptual, y por el estado anticipado generado por el módulo de Anticipación.
- *Activación recibida desde las Metas*: este tipo de activación producida por las metas del agente, incrementa el nivel de activación de aquellos *Comportamientos* que prometen alcanzar una de las metas globales o específicas del agente. Se dice que un *Comportamiento* está en capacidad de alcanzar una meta global/específica si la meta aparece como uno de los elementos de la *lista de efectos deseados* que define éste. Nótese que pueden existir dos tipos de metas: aquellas que sólo se alcanzan una vez y aquellas que deben ser alcanzadas continuamente (metas permanentes), como se describió en la sección 8.5.
- *Inhibición recibida desde las Metas Protegidas*: adicionalmente, existe cierta inhibición externa (o eliminación de la activación) recibida desde las metas que ya han sido alcanzadas por el agente y que por tanto deber ser protegidas contra la posibilidad de regresar a un estado de "no alcanzadas". Estas "metas protegidas" eliminan parte de la activación que proviene de los *Comportamientos* que intentan "deshacerlas". Se dice que un *Comportamiento* intenta deshacer una de las metas protegidas cuando la meta protegida es un elemento de la *lista de efectos no deseados* que define éste.

Estos procesos son continuos: existe un flujo constante de energía de activación hacia los *Comportamientos* que emparejan parcialmente el estado actual y hacia los *Comportamientos* que alcanzan alguna de las metas. Existe además una disminución continua del nivel de activación de los *Comportamientos* que deshacen las metas protegidas. El estado actual del ambiente y las metas pueden cambiar de forma impredecible en cualquier momento. Si esto sucede, la entrada de activación externa fluye automáticamente haca otros *Comportamientos*. Además del flujo de activación que ejerce el estado actual y las metas sobre los *Comportamientos*, estos pueden también producir flujos de activación (excitación) y de inhibición entre sí. Esta dispersión de la activación entre los *Comportamientos* se realiza a través de los enlaces de la siguiente forma:

- *Activación de Comportamientos Sucesores*: un *Comportamiento* x que es ejecutable dispersa su energía de activación hacia adelante. Éste incrementa (en una fracción de su propio nivel de activación) la activación de aquellos *Comportamientos* sucesores y para los que la proposición común $q \in ED_x \cap P_y$ no es verdadera. Intuitivamente, se desea que estos *Comportamientos* sucesores adquieran una mayor intensidad en su activación, puesto que están próximos a convertirse en ejecutables debido a que la mayoría de sus precondiciones serán satisfechas después de que x sea activado. Formalmente, el *Comportamiento* ejecutable x dispersa energía de activación hacia adelante a través de sus enlaces sucesores para los cuales la proposición que los define ($q \in ED_x$) sea falsa.

- *Activación de Comportamientos Predecesores*: un *Comportamiento* x que no es ejecutable dispersa su energía de activación hacia atrás. Éste incrementa (en una fracción de su propio nivel de activación) la activación de aquellos predecesores y para los que la proposición $q \in P_x \cap ED_y$ es verdadera. Intuitivamente, un *Comportamiento* no ejecutable dispersa energía de activación hacia aquellos que "prometen" hacer verdaderas sus precondiciones que aún no lo son, de tal forma que el *Comportamiento* pueda convertirse en ejecutable posteriormente. Formalmente, el *Comportamiento* x que no es ejecutable, dispersa energía de activación hacia atrás a través de aquellos enlaces predecesores para los que la proposición que los define ($q \in P_x$) sea falsa.
- *Inhibición de Comportamientos con Conflicto*: cada *Comportamiento* x (ejecutable o no) disminuye (por una fracción de su propio nivel de activación) el nivel de activación de aquellos *Comportamientos* y para los que la proposición común $q \in P_x \cap END_y$ es verdadera. Intuitivamente, un *Comportamiento* x intentará prevenir la activación de aquellos *Comportamientos* que deshacen las precondiciones de x que ya son verdaderas para el estado actual. En el caso de que existan conflictos mutuos entre dos *Comportamientos*, sólo aquél con el nivel de activación más alto podrá inhibir al otro. Esto previene el fenómeno por el cual los *Comportamientos* más relevantes se eliminan entre sí. Formalmente, el *Comportamiento* x substrae (libera) energía de activación a través de todos sus enlaces de conflicto para los que la proposición que los define ($q \in P_x$) sea verdadera, excepto para aquellos enlaces para los que existe un enlace de conflicto inverso que sea más potente (es decir, que tenga un mayor nivel de activación).

Es posible deducir que, la energía de activación producida por los enlaces sucesores y predecesores corresponde al mecanismo a través del cual *cooperan* entre sí los *Comportamientos*, mientras que la energía activación que se substrae a través de los enlaces de conflicto corresponde al mecanismo mediante el cual *compiten* entre sí los mismos. En resumidas cuentas, en cada ciclo del algoritmo de dispersión se realizan las siguientes operaciones computacionales para todo el conjunto de *Comportamientos* en la RC (véase el algoritmo 9.1).

Algunas de las propiedades de especial interés observables en el algoritmo son: la secuencia de *Comportamientos* que resulta de los procesos de activación, la optimalidad de esta secuencia (la cual es estimada por una función dependiente del dominio), y la velocidad con la cual es construida tal secuencia (en términos del número de pasos que requiere un *Comportamiento* para convertirse en activo en relación al número total de pasos de ejecución del sistema). El algoritmo de la Red de Comportamientos define cuatro *parámetros globales* que permiten ajustar las dinámicas de dispersión de activación y que por tanto, influyen en el proceso de selección de *Comportamientos* que coordina el GFA:

1. θ, es el umbral mínimo que debe alcanzar un *Comportamiento* para convertirse en activo. θ es disminuido en un 10 % cada vez que no resulte elegido ningún *Comportamiento* en el ciclo de activación. El valor de θ es restaurado a su valor inicial una vez resulte seleccionado un *Comportamiento* en el ciclo de activación.
2. ϕ, la cantidad de energía de activación que una proposición (verdadera) inyecta a la red.

Método: dispersarActivacion

```
1.  Variables
2.     rc              ↓↑ Red de Comportamientos.
3.  Inicio Procedimiento
4.     Mientras que no exista un Comportamiento ejecutable hacer:
5.        Para cada Comportamiento comp de rc hacer:
6.           comp ← calcularEntradaEnergia() //se calcula el impacto que
                tienen sobre el nivel de activación de cada Comportamiento, el estado,
                las metas, y las metas protegidas.
7.           comp ← calcularDispersion() //se calcula el nivel de excitación
                y de inhibición que dispersa cada Comportamiento a través de los enla-
                ces sucesores, predecesores, y de conflicto.
8.           comp ← calcularDegradacion() //se calcula una función de de-
                gradación que asegura que el nivel de activación global se mantenga
                constante.
9.           comp ← validarActivacion() //determina si existe un Compor-
                tamiento que cumpla con todas las condiciones para su activación
10.       Fin Para.
11.    Fin Mientras que
12.  Fin Procedimiento
```

Método: validarActivacion

```
1.  Variables
2.     rc              ↓↑ Red de Comportamientos.
3.     listaComps      ↑ lista de comportamientos que compiten por el foco de atención.
4.  Inicio Procedimiento
5.     Para cada Comportamiento comp de la rc hacer:
6.        listaComps ← validarEsEjecutable(comp) //valida si el Compor-
             tamiento es ejecutable (todas sus precondiciones son verdaderas).
7.        listaComps ← validarSobrepasaUmbral(listaComps) //valida si
             el Comportamiento sobrepasa el umbral de activación de la RC.
8.     Fin Para
9.     listaComps ← determinarActivacionMasAlta(listaComps) //de-
          termina qué Comportamiento (s) tiene(n) el nivel de activación más alto de
          todos los Comportamientos que cumplen con las dos condiciones anteriores y
          que están en listaComps.
10.    comp ← seleccionarComportamiento(listaComps) //cuando dos
          Comportamientos cumplen con las tres condiciones anteriores, es decir, son
          igualmente relevantes, uno de ellos es escogido de forma aleatoria. El nivel
          de activación del Comportamiento que ha sido activado, se reinicializa a 0 (de
          no ser así, un mismo Comportamiento se activaría de forma ininterrumpida
          por cierto número de ciclos, no siendo esto lo más apropiado pues evitaría la
          activación de otros posibles Comportamientos igualmente aptos). Si ninguno
          de los Comportamientos de la red cumple con validarEsEjecutable y
          validarSobrepasaUmbral, el umbral es reducido en un 10 % y se reinicia el
          ciclo.
11.  Fin Procedimiento
```

Algoritmo 9.1: Algoritmo de Dispersión de Activación para la Red de Comportamientos.

3. γ, la cantidad de energía de activación que una meta inyecta a la red.

4. δ, la cantidad de energía de activación que una meta protegida libera de la red.

Estos parámetros definen además, la cantidad de energía de activación que los *Comportamientos* dispersan hacia adelante, dispersan hacia atrás, o liberan de la red. Más precisamente, por cada proposición falsa en la lista de precondiciones de un *Comportamiento* no ejecutable, éste dispersa a sus predecesores una cantidad α de su energía de activación. Por cada proposición falsa en su lista de efectos deseados, un *Comportamiento* ejecutable dispersa a sus sucesores cierta cantidad ($\alpha\frac{\phi}{\gamma}$) de su energía de activación. Por cada proposición verdadera en su lista de precondiciones, un *Comportamiento* substrae (libera) cierta cantidad ($\alpha\frac{\delta}{\gamma}$) de energía de activación a través de sus enlaces de conflicto con otros *Comportamientos*.

Estos factores han sido determinados de tal manera que, la dispersión interna de activación debería tener semánticas (o efectos relativos) similares a las entradas/salidas producidas por el estado actual y las metas. Así, la tasa de relación entre la entrada del estado, la entrada de las metas, y la salida de las metas protegidas, se corresponde con la tasa de relación entre la entrada de los predecesores, la entrada de los sucesores, y la salida de los *Comportamientos* con los cuales existe conflicto.

Intuitivamente, se desea que las precondiciones que aún no son verdaderas sean consideradas como "sub-metas", los efectos que están a punto de ser verdaderos sean considerados como "predicciones" o "expectativas", y las precondiciones que son verdaderas sean consideradas como "metas protegidas".

9.2.3. Modelo Matemático Subyacente a la Dispersión de Activación

Esta sección presenta el modelo matemático que soporta el proceso de dispersión de activación modulado por el GFA. Considérese los siguientes elementos:

- Un conjunto de *Comportamientos* $1 \ldots n$,
- Un conjunto de proposiciones q,
- Una función $S(t)$, que retorna las proposiciones observadas que se suponen ciertas en un instante de tiempo t; siendo S el estado actual del ambiente percibido por el agente[4],
- Una función $G(t)$, que representa las metas actuales del agente en el instante de tiempo t; siendo G un proceso independiente que lleva a cabo la Estructura de Metas del módulo Intencional,
- Una función $R(t)$, que retorna las proposiciones que representan las metas logradas por el agente en un instante de tiempo t,
- Una función $esEjecutable(i, t)$, que retorna 1 si el *Comportamiento* i es ejecutable en el instante de tiempo t, es decir, si todas sus precondiciones se encuentran contenidas en $S(t)$, ó 0 en caso contrario,
- Una función $M(j)$, que retorna el conjunto de *Comportamientos* que contienen como precondición la proposición j, es decir, los *Comportamientos* x para los cuales $j \in P_x$,

[4]Recuérdese que el estado actual percibido por el agente se compone del conjunto de elementos activos en la memoria de trabajo, es decir, aquellas unidades-MT cuya activación sobrepasa el umbral establecido.

- Una función $A(j)$, que retorna el conjunto de *Comportamientos* que satisfacen la proposición j, es decir, los *Comportamientos* x para los cuales $j \in ED_x$,
- Una función $U(j)$, que retorna el conjunto de *Comportamientos* que hacen falsa la proposición j, es decir, los *Comportamientos* x para los cuales $j \in END_x$,

Dado un *Comportamiento* $x = (P_x, ED_x, END_x, \alpha_x)$, la activación de entrada generada por el estado actual observado en el instante t es:

$$actBasicaEstadoActual(x,t) = \sum_j \phi \frac{1}{\#M(j)} \times \frac{1}{\#P_x} \tag{9.3}$$

donde: $j \in S(t) \cap P_x$, y donde $\#$ indica la cardinalidad del conjunto.

Con el fin de incluir los aspectos relacionados con el proceso de oscilación y cambio del foco de atención de la MAT, se propone una adaptación al modelo original de Maes el cual incorpora tres entradas adicionales de energía de activación a la RC: activación por estímulos novedosos, activación por cambios físicos, y activación por elementos relevantes (descritos en la sección 9.2.1).

$$actEstimuloNovedoso(x,t) = \sum_j \phi \frac{1}{\#N(j)} \times \frac{1}{\#P_x} \tag{9.4}$$

donde: $j \in S(t) \cap P_x$. j es una proposición que determina un estado novedoso, es decir, una proposición que está en $S(t)$ pero que no lo estaba en $S(t-1)$. $N(j)$ es una función que retorna el conjunto de *Comportamientos* que contienen como precondición la proposición j, esto es, los *Comportamientos* x para los cuales $j \in P_x$. El símbolo $\#$ indica la cardinalidad del conjunto.

$$actCambioFisico(x,t) = \sum_j \phi \frac{1}{\#F(j)} \times \frac{1}{\#P_x} \tag{9.5}$$

donde: $j \in S(t) \cap P_x$. j es una proposición que refleja si ha tenido lugar un cambio físico en algún objeto percibido durante el tiempo transcurrido entre $S(t-1)$ y $S(t)$. $F(j)$ es una función que retorna el conjunto de *Comportamientos* que contienen como precondición la proposición j, esto es, los *Comportamientos* x para los cuales $j \in P_x$. El símbolo $\#$ indica la cardinalidad del conjunto.

$$actElementoRelevante(x,t) = \sum_j \phi \frac{1}{\#V(j)} \times \frac{1}{\#P_x} \tag{9.6}$$

donde: $j \in S(t) \cap P_x$. j es una proposición observable en $S(t)$ que corresponde a un elemento del estado anticipado, a una expectativa, o a una sub-meta. $V(j)$ es una función que retorna el conjunto de *Comportamientos* que contienen como precondición la proposición j, esto es, los *Comportamientos* x para los cuales $j \in P_x$. El símbolo $\#$ indica la cardinalidad del conjunto.

Estas entradas adicionales de energía de activación, propuestas en la presente tesis, permiten que el proceso de re-orientación del foco de atención llevado a cabo por cada MAT del agente, esté bajo la influencia de factores relativos a la oscilación atencional. Por tanto, la activación integrada que corresponde al estado actual percibido sería:

$$\begin{aligned} actIntegradaEstadoActual(x,t) &= actBasicaEstadoActual(x,t) + actEstimuloNovedoso(x,t) + \\ & actCambioFisico(x,t) + actElementoRelevante(x,t) \end{aligned} \tag{9.7}$$

Por otro lado, el flujo de activación que entra al *Comportamiento* x generado por las metas en el instante t corresponde a:

$$activacionMetas(x,t) = \sum_j \gamma \frac{1}{\#A(j)} \times \frac{1}{\#ED_x} \tag{9.8}$$

donde: $j \in G(t) \cap ED_x$.

La liberación de energía de activación para el *Comportamiento* x generada por las metas que se encuentran protegidas en el instante t es:

$$liberarMetasP(x,t) = \sum_j \delta \frac{1}{\#U(j)} \times \frac{1}{\#END_x} \tag{9.9}$$

donde: $j \in R(t) \cap END_x$.

En cuanto a la dispersión de activación, la siguiente función determina que, el *Comportamiento* $x = (P_x, ED_x, END_x, \alpha_x)$, dispersa hacia atrás un flujo de activación hacia el *Comportamiento* $y = (P_y, ED_y, END_y, \alpha_y)$, si se cumple que $j \notin S(t) \wedge j \in P_x \cap ED_y$.

$$dispersarAT(x,y,t) = \begin{cases} \sum_j \alpha_x(t-1) \frac{1}{\#A(j)} \times \frac{1}{\#ED_y} & \text{si } esEjecutable(x,t) = 0 \\ 0 & \text{si } esEjecutable(x,t) = 1 \end{cases} \tag{9.10}$$

Por otro lado, la siguiente función describe cómo el *Comportamiento* x dispersa un flujo de activación hacia adelante al *Comportamiento* y, si se cumple que $j \notin S(t) \wedge j \in ED_x \cap P_y$.

$$dispersarAD(x,y,t) = \begin{cases} \sum_j \alpha_x(t-1) \frac{\phi}{\gamma} \times \frac{1}{\#M(j)} \times \frac{1}{\#P_y} & \text{si } esEjecutable(x,t) = 1 \\ 0 & \text{si } esEjecutable(x,t) = 0 \end{cases} \tag{9.11}$$

La función que especifica la cantidad de energía que el *Comportamiento* x libera del *Comportamiento* y cuando se cumple que, $j \in P_x \cap END_y$ (i.e, la energía liberada a

través de los enlaces de conflicto) corresponde a:

$$liberarConflictos(x,y,t) = \begin{cases} 0, & \text{si } (\alpha_x(t-1) < \alpha_y(t-1)) \wedge (\exists i \in S(t) \cap P_y \cap END_x) \\ \text{máx}\left(\sum_j \alpha_x(t-1)\frac{\delta}{\gamma} \times \frac{1}{\#U(j)} \times \frac{1}{\#END_y}, \alpha_y(t-1)\right), & \text{de lo contrario} \end{cases} \tag{9.12}$$

Adicionalmente, es definida una función de degradación cuyo propósito es mantener constante el nivel de activación global de la Red de Comportamientos:

$$degradar(y) = \rho_y \times \alpha_y, \text{ tal que } \sum_y^n \alpha_y(t) = n \times \pi \tag{9.13}$$

donde: ρ_y es un factor de degradación del nivel de activación α_y del *Comportamiento* y. n es el número de *Comportamientos* de la RC. π es el nivel promedio de activación en la RC.

Por consiguiente, el nivel de activación del *Comportamiento* y en el instante de tiempo t se define como:

$$\begin{aligned} \alpha_{y,0} &= 0 \\ \alpha_{y,t} &= degradar(\alpha(y,t-1)(1 - \text{activo}(y,t-1) \\ &\quad + actIntegradaEstadoActual(y,t) + activacionMetas(y,t) \\ &\quad + \textstyle\sum_{x,z}(dispersarAT(x,y,t) + dispersarAD(x,y,t) \\ &\quad - liberarMetasP(y,t) - liberarConflictos(z,y,t))) \end{aligned} \tag{9.14}$$

donde: x representa todos los *Comportamientos* de la red. z son todos los *Comportamientos* de la red excepto el *Comportamiento* y, con $t > 0$.

Finalmente, el *Comportamiento* que se activa en el instante t corresponde a:

$$\begin{aligned} activo(t,i) = 1, \quad \text{si} \begin{cases} \alpha(i,t) > \theta & (1) \\ esEjecutable(i,t) = 1 & (2) \\ \forall i \text{ que cumple } (1) \wedge (2)\text{: } \alpha(i,t) \geq \alpha(j,t) \end{cases} \\ activo(t,i) = 0, \quad \text{de lo contrario} \end{aligned} \tag{9.15}$$

9.2.4. Caso de Estudio: Orquestación Adaptativa del Comportamiento de un Animal Virtual (Animat)

En esta sección se presenta un ejemplo ilustrativo que describe (cualitativamente) las dinámicas que articula el mecanismo de Atención Selectiva.

Supóngase que el agente cognitivo es un animal virtual (animat), más concretamente, un ratón que deambula por un espacio bidimensional donde se encuentran rocas (obstáculos), alimento, y depredadores. El mecanismo de atención selectiva del ratón virtual

tiene que modular la activación de dos MATs, las cuales se encargan cada una de lograr las dos metas globales del agente: ⟨Mantenerse-Con-Energía⟩ (MAT_1) y ⟨Mantenerse-A-Salvo⟩ (MAT_2). Cada una de estas MATs se compone de una Red de Comportamientos que define la estructura de activación/inhibición para los *Comportamientos* que la constituyen. La RC para la MAT_1 se compone de los *Comportamientos* ⟨Buscar-Comida⟩, ⟨Comer⟩, ⟨Explorar-Entorno⟩, y ⟨Dormir⟩; y la RC para la MAT_2 se compone de los *Comportamientos* ⟨Huir-Depredador⟩, ⟨Explorar--Entorno⟩, ⟨Evadir-Obstáculo⟩ y ⟨Dormir⟩. La estructura para cada *Comportamiento* se describe a continuación:

Comportamiento: Buscar-Comida:

Precondiciones:	⟨Estado-Hambre, Con-Hambre⟩ ⟨Ubicación-Comida, Sin-Comida-Cerca⟩ ⟨Ubicación-Depredadores, Sin-Depredadores-Cerca⟩
Efectos Deseados:	⟨Ubicación-Comida, Comida-Cerca⟩
Efectos No Deseados:	⟨Estado-Curiosidad, Aburrido⟩ ⟨Ubicación-Comida, Sin-Comida-Cerca⟩

Comportamiento: Comer:

Precondiciones:	⟨Estado-Hambre, Con-Hambre⟩ ⟨Ubicación-Comida, Comida-Cerca⟩ ⟨Ubicación-Depredadores, Sin-Depredadores-Cerca⟩
Efectos Deseados:	⟨Estado-Hambre, Saciado⟩ ⟨Meta, Mantenerse-Con-Energía⟩
Efectos No Deseados:	⟨Estado-Curiosidad, Curioso⟩ ⟨Estado-Hambre, Con-Hambre⟩

Comportamiento: Dormir:

Precondiciones:	⟨Estado-Hambre, Saciado⟩ ⟨Estado-Curiosidad, Aburrido⟩ ⟨Ubicación-Depredadores, Sin-Depredadores-Cerca⟩
Efectos Deseados:	⟨Estado-Energía, Con-Energía⟩ ⟨Meta, Mantenerse-Con-Energía⟩
Efectos No Deseados:	⟨Estado-Curiosidad, Aburrido⟩

Comportamiento: Huir-Depredador:

Precondiciones:	⟨Estado-Miedo, Miedoso⟩ ⟨Estado-Energía, Con-Energía⟩ ⟨Ubicación-Depredadores, Con-Depredadores-Cerca⟩
Efectos Deseados:	⟨Ubicación-Depredadores, Sin-Depredadores-Cerca⟩ ⟨Meta, Mantenerse-A-Salvo⟩
Efectos No Deseados:	⟨Ubicación-Depredadores, Con-Depredadores-Cerca⟩ ⟨Estado-Sueño, Somnoliento⟩ ⟨Estado-Hambre, Con-Hambre⟩

Comportamiento: Explorar-Entorno:	
Precondiciones:	⟨Estado-Curiosidad, Curioso⟩ ⟨Obstáculos, Sin-Obstáculos-Cerca⟩ ⟨Ubicación-Depredadores, Sin-Depredadores-Cerca⟩ ⟨Estado-Hambre, Sin-Hambre⟩
Efectos Deseados:	⟨Obstáculos, Con-Obstáculos-Cerca⟩ ⟨Ubicación-Otros-Ratones, Otros-Ratones-Cerca⟩
Efectos No Deseados:	⟨Estado-Curiosidad, Aburrido⟩ ⟨Estado-Sueño, Somnoliento⟩

Comportamiento: Evadir-Obstáculos:	
Precondiciones:	⟨Obstáculos, Con-Obstáculos-Cerca⟩
Efectos Deseados:	⟨Obstáculos, Sin-Obstáculos-Cerca⟩ ⟨Meta, Mantenerse-A-Salvo⟩
Efectos No Deseados:	⟨Obstáculos, Con-Obstáculos-Cerca⟩

En la figura 9.3 se esquematizan las MATs del agente, con sus respectivas metas, *Comportamientos* y unidades-MT asociadas. En esta figura se presentan los enlaces sucesores y de conflicto que estimulan la excitación e inhibición, respectivamente, de los *Comportamientos* del ratón, aunque por motivos de simplificación del gráfico se omiten los enlaces predecesores puesto que estos son los mismos enlaces sucesores pero en sentido contrario.

Imagínese que inicialmente el ratón tiene activa la MAT_1 que promete alcanzar la meta ⟨Mantenerse-Con-Energía⟩, y que el estado inicial percibido es: $S(t)_0 = \{$*Con-hambre, Sin-Comida-Cerca, Sin-Depredadores-Cerca, Sin-obstáculos-Cerca*$\}$.

Este estado inicial dispersa energía de activación hacia los *Comportamientos* ⟨Buscar-Comida⟩, ⟨Comer⟩ y ⟨Dormir⟩, los cuales compiten entre sí por obtener el foco de atención. Debido a que las precondiciones ⟨Comida-Cerca⟩ y ⟨Aburrido⟩ son falsas, los *Comportamientos* ⟨Comer⟩ y ⟨Dormir⟩ no reciben la suficiente activación y como consecuencia ⟨Buscar-Comida⟩ es el *Comportamiento* que gana la competencia y se hace con el control consciente del ratón. En este caso hay que notar que ⟨Buscar-Comida⟩ no sólo recibe activación del estado percibido, sino que además recibe activación de ⟨Comer⟩ a través de un enlace predecesor. De esta manera, ⟨Comer⟩ coopera con ⟨Buscar-Comida⟩ debido a que este último "promete" satisfacer las condiciones necesarias para que se active el primero, en otras palabras, el efecto deseado después de la ejecución de ⟨Buscar-Comida⟩ es hacer verdadera la precondición ⟨Comida-Cerca⟩ del *Comportamiento* ⟨Comer⟩, motivo por el cual este último dispersa activación hacia atrás (a través del enlace predecesor) con el fin de establecer una coalición con el *Comportamiento* ⟨Buscar-Comida⟩ la cual le garantizará su activación en un estado futuro.

En su búsqueda de comida, el ratón halla un buen trozo de queso frente a él, lo que hace que se cumpla la precondición ⟨Comida-Cerca⟩. Esta activación del estado junto con la activación recibida por la meta ⟨Mantenerse-Con-Energía⟩ producen

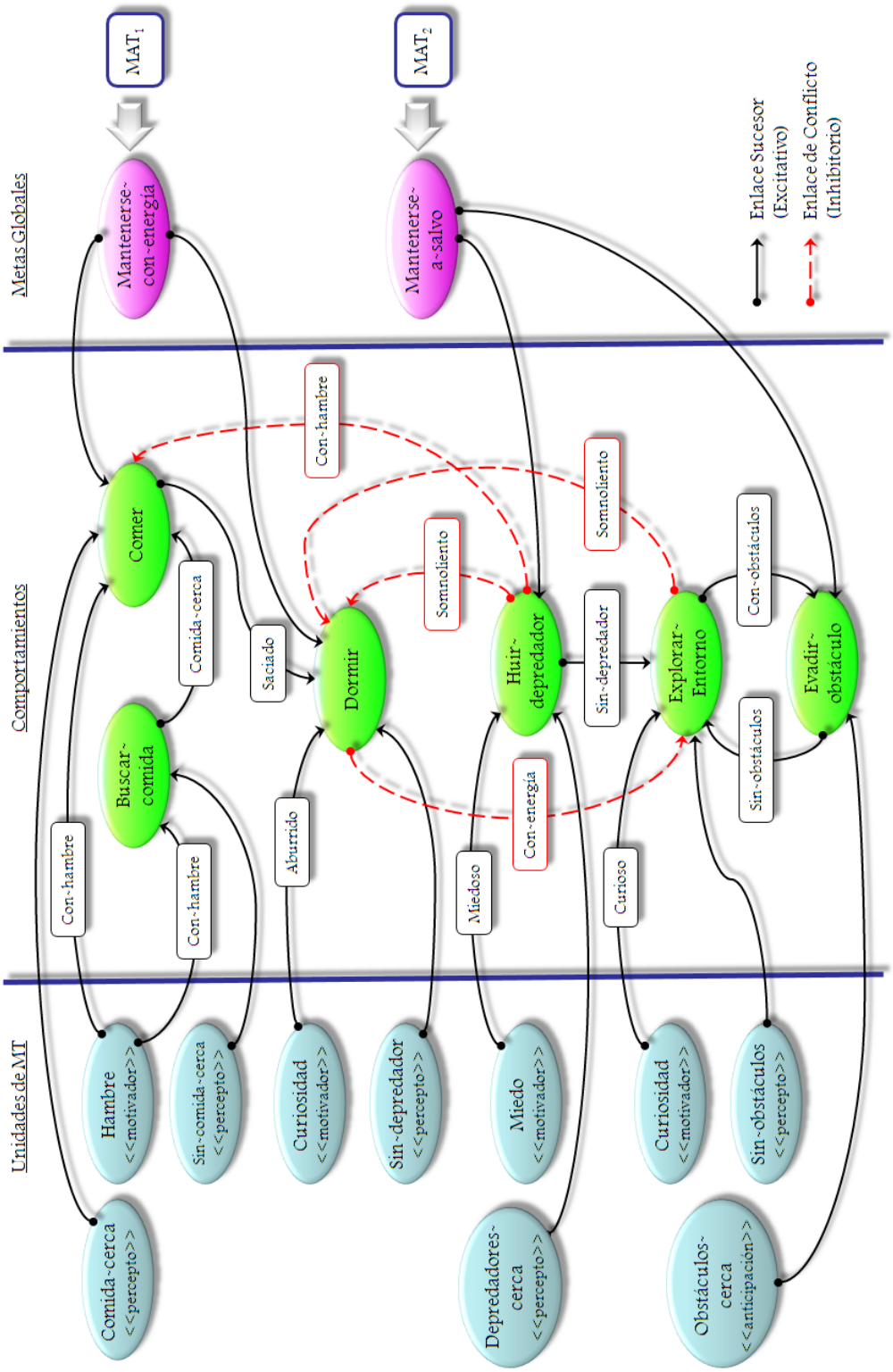


Figura 9.3: Dispersión de Activación en las Redes de Comportamientos del Animat.

la activación del *Comportamiento* ⟨Comer⟩, el cual se mantendrá activo mientras no haya otro *Comportamiento* competidor que tenga un nivel de activación superior a éste. Durante el tiempo que se mantenga activo ⟨Comer⟩, éste intentará satisfacer todas sus *expectativas*, procurando que las proposiciones en su lista de efectos deseados se hagan verdaderas (i.e., ⟨Saciado⟩) y a la vez que las proposiciones en su lista de efectos no deseados se hagan falsas (i.e., ⟨Con-Hambre⟩).

Pasado el tiempo, el ratón ha comido lo suficiente y el nivel de activación del motivador ⟨Hambre⟩ se sitúa en la región de sobre-estimulación, produciendo el estado ⟨Saciado⟩. Debido a que el *Comportamiento* ⟨Comer⟩ tiene en su lista de efectos deseados la proposición ⟨Saciado⟩, que ahora es una proposición verdadera, se considera entonces que el *Comportamiento* ha cumplido con sus sub-metas y por tanto el GFA de MAT_1 puede proceder a la activación de otro *Comportamiento* diferente. Durante los múltiples ciclos cognitivos que han ocurrido desde que el ratón empezó a comer, el *Comportamiento* ⟨Comer⟩ ha estado dispersando energía de activación hacia ⟨Dormir⟩ a través de un enlace sucesor, lo que sugiere que probablemente luego de comer, el ratón se decantará por tomar una siesta. Sin embargo, ⟨Dormir⟩ no se hace con el foco de atención sino hasta que reciba la activación que produce el motivador ⟨Curiosidad⟩ en su estado de sobre-estimulación (i.e., ⟨Aburrido⟩), el cual por ahora no se ha disparado. Ahora, suponiendo que el motivador ⟨Curiosidad⟩ se encuentra en una región de infra-estimulación (i.e., ⟨Curioso⟩), el *Comportamiento* ⟨Explorar-Entorno⟩ se hace con el foco de atención, ganándole la competición al *Comportamiento* ⟨Dormir⟩. Por tanto, ahora el ratón empieza a deambular por su entorno circundante, mientras continúan las dinámicas de competición y cooperación entre *Comportamientos* ...

Considérese ahora que un nuevo percepto entra en escena: ⟨Con-Depredadores-Cerca⟩. La situación de amenaza que percibe el ratón (probablemente al divisar un gato que se aproxima hacia él) aumenta la intensidad del motivador ⟨Miedo⟩ llevándola hasta la región de infra-estimulación que genera el estado ⟨Miedoso⟩. A su vez, este motivador dispara la activación de la meta ⟨Mantenerse-A-Salvo⟩, y como consecuencia MAT_2 es la máquina de atención que coordina ahora el mecanismo atencional del ratón (de lo que se deduce que en el nuevo estado actual del agente, la meta ⟨Mantenerse-A-Salvo⟩ tiene una mayor relevancia que la meta ⟨Mantenerse-Con-Energía⟩). De esta manera, siendo verdaderas todas las precondiciones requeridas para la activación del *Comportamiento* ⟨Huir-Depredador⟩ (i.e, ⟨Con-Depredadores-Cerca⟩, ⟨Miedoso⟩, y ⟨Mantenerse-A-Salvo⟩), éste se hace con el control consciente del ratón e inicia una espectacular huida!!!

En este punto cabe resaltar la función inhibidora que produce ⟨Huir-Depredador⟩ sobre otros *Comportamientos*: debido a que durante la huida del ratón es deseable que *Comportamientos* tales como ⟨Dormir⟩ o ⟨Comer⟩ no se hagan con el foco de atención (pues no resultaría conveniente para el ratón empezar a comer o tomar una siesta mientras que un gato está merodeando cerca), estos reciben una dispersión de energía inhibitoria a través de los enlaces de conflicto de ⟨Huir-Depredador⟩ que evita que se activen (véanse los enlaces representados como líneas con marcas entrecortadas de color rojo en la figura 9.3). Nótese que ⟨Huir-Depredador⟩ incluye en su lista de efectos "no deseados" los estados ⟨Somnoliento⟩ y ⟨Con-Hambre⟩. Esto quiere decir que el *Comportamiento* ⟨Huir-Depredador⟩ no tiene como sub-meta que el agente caiga

en un estado de somnolencia, lo cual evita la activación del *Comportamiento* ⟨Dormir⟩ y, por otro, tiende a ignorar si se encuentra con hambre, lo que inhibe la activación del *Comportamiento* ⟨Comer⟩. De esta manera, se garantiza que ⟨Huir-Depredador⟩ competirá, a través de sus enlaces de conflicto (representados como una lista de efectos no deseados), contra otros *Comportamientos* que entorpecerían su ejecución.

Una vez el ratón ha huido exitosamente del gato, el GFA activaría nuevamente el *Comportamiento* ⟨Explorar-Entorno⟩ puesto que durante la huida, ⟨Huir-Depredador⟩ mantuvo la dispersión de energía de activación hacia ⟨Explorar-Entorno⟩ a través de su enlace sucesor. Imagínese ahora que el sistema de anticipación del ratón predice que tras bajar cierta cuesta habrá una gran roca, lo que genera la anticipación ⟨Con-Obstáculos-Cerca⟩. Debido a que es conveniente para ⟨Explorar-Entorno⟩ que el ratón se encuentre libre de obstáculos, éste dispersa energía de activación hacia adelante, a través del enlace sucesor, hacia ⟨Evadir-Obstáculos⟩, el cual se hace con el control después de recibir activación directa de la meta ⟨Mantenerse--A-Salvo⟩ y de la anticipación ⟨Con-Obstáculos-Cerca⟩. El *Comportamiento* ⟨Explorar-Entorno⟩ queda latente, en un estado subconsciente, a la espera de que ⟨Evadir-Obstáculos⟩ cumpla con su función y le regrese el control sobre la actuación del ratón. Como se observa en la figura 9.3, se genera un bucle de "cooperación" mutua entre tales *Comportamientos* (⟨Explorar-Entorno⟩ y ⟨Evadir--Obstáculos⟩), los cuales estarán cediendo continuamente el control el uno al otro, hasta que un nuevo *Comportamiento* entre en escena y se inicie nuevamente el proceso de "competición".

Considérese ahora un escenario alternativo al anteriormente descrito donde se incluye un nuevo *Comportamiento*: ⟨Descansar⟩. Inicialmente, el ratón decide deambular alrededor, para lo cual activa el *Comportamiento* ⟨Explorar-Entorno⟩. Después de cierto tiempo, el gasto de energía producido por el movimiento hace que el motivador ⟨Hambre⟩ genere cierta activación de entrada hacia el *Comportamiento* ⟨Buscar--Comida⟩ y que el motivador ⟨Cansancio⟩[5] induzca energía de activación a ⟨Descan-sar⟩. Supóngase que ⟨Buscar-Comida⟩ acumula mayor cantidad de energía de activación y toma el control consciente del ratón.

Imagínese ahora que el ratón encuentra un trozo de queso y a un gato custodiándolo. Los *Comportamientos* ⟨Comer⟩, ⟨Huir-Depredadores⟩, y ⟨Descansar⟩ (este último debido al desgaste físico del ratón) inician una reñida competencia entre ellos. Cada *Comportamiento* empieza a acumular energía de activación proveniente de la mayor cantidad de fuentes disponibles: de las metas, del estado actual (motivadores, perceptos, anticipaciones, etc.), y de las alianzas o coaliciones con otros *Comportamientos* que puedan generar una dispersión de activación hacia adelante o hacia atrás (a través de los enlaces sucesores y predecesores). Estas coaliciones intentan pues, reclutar la mayor cantidad de "apoyo" posible bajo la consigna de un beneficio mutuo. Por ejemplo, el *Comportamiento* ⟨Explorar-Entorno⟩ dispersaría energía de activación hacia ⟨Huir--Depredadores⟩ puesto que este último promete un estado "sin depredadores" necesario para que ⟨Explorar-Entorno⟩ pueda ejecutarse. Lo mismo sucede con la

[5] Aunque "Cansancio" pertenece más a un Estado Físico del agente, éste puede ser modelado como un motivador puesto que cumple con las premisas básicas relativas a la regiones de intensidad y puntos de activación de un motivador genérico.

coalición entre ⟨Descansar⟩ y ⟨Comer⟩, pues ⟨Comer⟩ promete satisfacer una de las precondiciones de ⟨Descansar⟩: el estado ⟨Saciado⟩. Finalmente, luego de varios ciclos de continua competición y cooperación, el *Comportamiento* que más ha acumulado energía de activación, ha sobrepasado el umbral de activación y cumple con todas sus precondiciones, es ⟨Huir-Depredadores⟩.

Una vez el ratón está libre de peligro, el *Comportamiento* ⟨Comer⟩, el cual se ha mantenido latente en un plano subconsciente mientras ⟨Huir-Depredador⟩ se ejecutaba, entra de nuevo a la competición por el foco de atención. Sin embargo, debido a que en la nueva posición del ratón no hay fuentes de alimento cercanas, el *Comportamiento* ⟨Comer⟩ forma una nueva coalición con ⟨Buscar-Comida⟩, y es éste el que se hace con el control. Después de encontrar la comida, ⟨Comer⟩, el cual ha acumulado suficiente energía de activación y cuyas precondiciones son ahora todas verdaderas, se activa. Una vez el ratón está lleno, el *Comportamiento* ⟨Comer⟩ dispersa energía hacia ⟨Descansar⟩ a través de sus enlaces sucesores...

Como se puede observar, cada RC orquesta la ejecución de los distintos *Comportamientos* a través de la modulación del flujo dinámico de energía de activación que entra y se dispersa en la red. De esta manera, cada RC responde ***adaptativamente*** a las demandas del entorno y NO se ciñe a la ejecución de una secuencia (pre-definida) de pasos que lleva a la consecución de un objetivo o meta (como suele suceder con los planificadores clásicos).

Es importante hacer énfasis además, en la interacción recíproca entre los procesos de competición y cooperación: los *Comportamientos* continuamente cooperan entre sí buscando un beneficio mutuo, pero esta cooperación se traduce en un incremento del nivel de activación de los *Comportamientos* cooperantes, lo cual implica que estos aumentarán su nivel competitivo frente a los demás *Comportamientos*, y como consecuencia, cuanto más alto sea el nivel de activación de los competidores, mayor será la influencia que estos pueden ejercer durante el apoyo cooperativo...

En este caso de estudio se han presentado las dinámicas de dispersión de activación desde una perspectiva meramente descriptiva y cualitativa, la cual resulta útil para ejemplificar el funcionamiento global y las ventajas de emplear este modelo computacional. No obstante, las dinámicas que modula el GFA en cada Red de Comportamientos son mucho más complejas, requieren de continuas coaliciones e inhibiciones, y de múltiples ciclos de acumulación incremental de energía de activación, hasta que finalmente se sobrepasa el umbral de activación requerido para seleccionar el *Comportamiento* que se hará con el foco de atención.

9.2.5. Discusión: Fortalezas y Debilidades del Modelo de Dispersión de Activación

En esta sección se identifican las fortalezas principales del modelo de dispersión de activación (originalmente propuesto por Maes (Maes, 1989), extendido en (Maes, 1992; Dorer, 2005; Nebel y Babovich-Lierler, 2004), y adaptado en esta tesis según las consideraciones mencionadas en la sección 9.2.3). Así mismo, son identificadas sus debilidades

las cuales sirven como punto de partida para la extensión del modelo que se propone en la siguiente sección.

Fortalezas:

- ***Orientabilidad hacia los Comportamientos:*** en el modelo original de Maes (Maes, 1989) los nodos que conforman las redes de dispersión corresponden simplemente a "acciones discretas", donde la listas de efectos deseados y no deseados de cada nodo son consideradas consecuencias absolutas que derivan de la ejecución de su correspondiente acción. En contraste, en la adaptación propuesta, cada nodo de la red corresponde a un *Comportamiento* de representación híbrida como los descritos en el capítulo 7. La hipótesis que se teje sobre este supuesto es que, en entornos dinámicos y realistas, el mecanismo de transición que lleva al sistema desde el estado A hasta el estado B (donde B es un estado deseado o sub-meta), requiere de la ejecución (secuencial y en paralelo) de múltiples acciones, probablemente a través de múltiples ciclos cognitivos que involucren procesos de toma de decisiones, de razonamiento y de aprendizaje.

  Por tanto, los *Comportamientos* son una buena metáfora para describir estas transiciones entre estados. Las listas de efectos deseados y no deseados de cada *Comportamiento* no corresponden a estados absolutos que se se garantizan con la ejecución del *Comportamiento*, sino que modelan un conjunto de expectativas (excitativas e inhibitorias) que pueden llegar a cumplirse o no. De esta manera, mientras el *Comportamiento* no haya satisfecho todas sus expectativas y reciba activación desde los elementos de la red (i.e., desde el estado, desde las metas, y desde otros *Comportamientos*), permanecerá en continua competencia por obtener el foco de atención, aunque esto no garantice que siempre lo obtenga.

- ***Orientabilidad hacia las Metas:*** las metas globales inyectan cierta cantidad γ de energía de activación a un conjunto (x) de *Comportamientos* los cuales prometen alcanzar dichas metas. Estos a su vez incrementan el nivel de activación de otro conjunto (y) de *Comportamientos* que cumplen con las sub-metas de x (es decir, aquellos y que hacen verdaderas las precondiciones de x), y así sucesivamente. Por tanto, esta dispersión de activación hacia atrás garantiza que los *Comportamientos* que contribuyen con determinada meta reciban mayor activación que aquellos que no. Además, los *Comportamientos* que contribuyen con varias metas o sub-metas, recibirán mayor cantidad de energía de activación.

  Si la MAT tiene asociada más de una meta global, los *Comportamientos* que contribuyen a la meta más cercana son favorecidos. Por "meta más cercana" se entiende la secuencia de activación de *Comportamientos* más corta entre la situación actual y la meta. Así mismo, el algoritmo favorece los *Comportamientos* que generan poca competición. Por ejemplo, si la MAT tiene dos metas m_1 y m_2, y hay un *Comportamiento* que promete alcanzar m_1 y dos *Comportamientos* que prometen alcanzar m_2, entonces el algoritmo favorecerá al *Comportamiento* que promete alcanzar m_1, y por consiguiente la probabilidad que tiene m_1 de ser realizada primero que las demás es más alta. Lo mismo aplica para las sub-metas.

El grado de orientabilidad hacia las metas puede ser variable, de acuerdo con la proporción de γ (la cantidad de energía de activación que inyecta una meta a la red) con respecto a ϕ (la cantidad de energía de activación que inyecta el estado por cada proposición verdadera). Si se define $\phi = 0$ la red ejecuta el tradicional encadenamiento hacia atrás (backward chaining), y como consecuencia, la activación de los *Comportamientos* es totalmente orientada hacia las metas, causando así que el agente sea menos "reactivo" ante los cambios. Además, el sistema se ralentiza debido a que el estado actual del entorno no dirige la activación de los *Comportamientos*. Idealmente, se desea que la actuación del agente sea principalmente orientada hacia las metas pero que tome en consideración las oportunidades interesantes que se presentan en el estado percibido. Esto se logra definiendo $\gamma > \phi$, **aunque la proporción óptima es dependiente del contexto**[6].

- ***Relevancia Situacional:*** el algoritmo inyecta más cantidad de energía de activación a los *Comportamientos* que son relevantes para la situación en contraste con aquellos que no. El proceso responsable de esto es la entrada de energía de activación que produce el estado observado (considerando la activación adicional generada por la detección de estímulos novedosos, cambios físicos en los objetos, y estados anticipados[7]) y por la dispersión de activación hacia adelante que producen los *Comportamientos* ejecutables a través de sus enlaces sucesores. Las ventajas de esta característica son: (1) el agente orienta su búsqueda de objetivos y por tanto acelera la activación de *Comportamientos*, y (2) el agente es capaz de explotar las oportunidades que se le presenten, permitiendo que el foco de atención se re-dirija más hacia aquello que está sucediendo en el entorno e incrementando su capacidad de reactividad.

  El mecanismo de dispersión hacia adelante se encarga de hacer que un *Comportamiento* reciba cierta cantidad de energía de activación desde el estado percibido en proporción a su tendencia a convertirse en un *Comportamiento* ejecutable. Esto implica, por ejemplo, que un *Comportamiento* que tiene dos precondiciones p_1 y p_2, de las cuales supóngase que p_1 no puede hacerse verdadera dado el estado actual, reciba relativamente menos activación desde el estado y, por consiguiente, tenga menos probabilidad de participar en la actuación consciente del agente o formar parte de un plan[8].

  Definir $\phi \geq \gamma$, puede provocar que el agente sea más reactivo y que la activación de *Comportamientos* sea más rápida (i.e, que se requieran menos ciclos de activación para hacer ejecutable un *Comportamiento*). Por tanto, el foco de atención se puede dirigir hacia la activación de un *Comportamiento* menos óptimo (debido a que el mecanismo de activación de los *Comportamientos* en la RC es ahora más orientado hacia los estímulos que hacia las metas).

[6] El que esta proporción sea dependiente del contexto, supondría la necesidad de definir este valor de forma empírica para cada concreción contextual de la arquitectura. La solución a este problema se propone en la siguiente sección.

[7] Esta activación adicional que recibe el estado, como se mencionó en la sección 9.2.3, es una adaptación que se propone al modelo original de Maes, con el objetivo de simular los cambios en el foco de atención debidos a la Oscilación.

[8] Sin embargo, éste puede recibir activación desde las metas y emplear dicha activación para persuadir a sus *Comportamientos* predecesores, para que hagan verdaderas sus precondiciones.

- ***Adaptabilidad:*** El mecanismo de activación de *Comportamientos* llevado a cabo por la RC es completamente "dinámico". Tanto el entorno como las metas pueden cambiar en tiempo de ejecución. Como resultado, las entradas/salidas externas de energía de activación, así como los patrones internos de excitación/inhibición, cambiarán de modo que se refleje la situación modificada. Incluso, la influencia externa durante la "planificación" o dispersión de activación es tan importante, que los planes son formados en la medida que esté presente la influencia o perturbación del entorno y de las metas.

  Nótese además que no se da el caso donde el sistema tiene que re-planificar en cada ciclo cognitivo. La "historia" de la activación dispersa juega también un rol importante en la orientación del foco de atención puesto que los niveles de activación no son re-inicializados en cada ciclo. Por tanto, de la misma manera que existe una relación entre la "orientabilidad hacia las metas" y la "relevancia situacional", existe una relación entre la adaptabilidad y la predisposición a llevar a cabo los planes en curso (lo cual se discute en el siguiente punto). Es posible mediar fácilmente entre estos dos extremos seleccionando una proporción específica entre los parámetros γ vs. ϕ vs. π (el nivel de activación promedio). De nuevo, la solución apropiada recae en algún valor intermedio. Los parámetros deberían ser escogidos de manera tal que el sistema no esté continuamente "saltando" entre una meta y otra, pero que a su vez pueda explotar las oportunidades y adaptarse a las situaciones cambiantes.

  Por último, nótese que el algoritmo exhibe otro tipo de adaptabilidad llamado "tolerancia a fallos". Este tipo de adaptabilidad es una consecuencia de la naturaleza distribuida del algoritmo. Puesto que ninguno de los *Comportamientos* es más importante que los otros, las redes son capaces de ejecutarse y dirigir el foco de atención aún bajo precondiciones degradadas. Es posible eliminar *Comportamientos* de la red y que ésta aún encuentre una solución satisfactoria. Por ejemplo, en el caso de estudio, si el *Comportamiento* $\langle$`Buscar-Comida`$\rangle$ es eliminado o inactivado, la red propone una solución que no involucra este *Comportamiento* (e.g., es posible que el agente permanezca deambulando – a través del *Comportamiento* $\langle$`Explorar-Entorno`$\rangle$ – hasta que eventualmente encuentre alimento).

- ***Predisposición hacia los Planes en Curso:*** el algoritmo demuestra un mecanismo implícito de predisposición, favoreciendo los *Comportamientos* que contribuyen a las metas y sub-metas en curso, excepto cuando apremia la necesidad de re-orientar el foco de atención hacia algo diferente. La principal razón por la cual se exhibe la predisposición, es que los niveles de activación no son re-inicializados cada vez que un *Comportamiento* es activado. Como consecuencia, la historia de los efectos pasados producidos por la dispersión de activación juega un rol importante en la re-orientación del foco de atención, en particular cuando la activación desde el estado y desde las metas es relativamente pequeña en comparación con el nivel de activación promedio π. Pero incluso si éste no es el caso, el algoritmo exhibe cierta predisposición hacia la ejecución de los planes en curso. Más específicamente, éste demuestra dos tipos de predisposición: horizontal y vertical.

  Predisposición Horizontal: este tipo de predisposición se evidencia cuando el algoritmo favorece los *Comportamientos* que contribuyen a la meta actual (o la meta

en la que el agente se ha estado enfocando anteriormente). Por ejemplo, en el caso de estudio, el algoritmo seleccionaba uno a uno los *Comportamientos* hasta que las sub-metas fuesen logradas, sin que el foco de atención saltase de una meta a otra (e.g., primero se activó el *Comportamiento* ⟨`Buscar-Comida`⟩; cuando la comida se encontró, se activó el *Comportamiento* ⟨`Comer`⟩; y cuando el estado ⟨`Saciado`⟩ se disparó, el ratón dejó de comer para activar luego el *Comportamiento* ⟨`Dormir`⟩). Para lograr un efecto de predisposición horizontal, el umbral de activación de la red (θ) debe ser incrementado, de manera tal que las dinámicas de competición y cooperación (que tienen lugar durante la ejecución de un ciclo cognitivo) puedan inyectar mayor cantidad de activación sobre el *Comportamiento* que promete alcanzar la meta actual, y a su vez sea ignorada, por ejemplo, la influencia de nuevas metas que intentan desviar el curso de ejecución de la meta actual.

Predisposición Vertical: este tipo de predisposición favorece a los *Comportamientos* que contribuyen al logro de una meta "hermana" (una sub-meta de la misma meta global). De nuevo en este caso, si el umbral es lo suficientemente alto, el algoritmo primero ejecuta todos los *Comportamientos* que contribuyen a una de las sub-metas, luego, cambia el foco de atención hacia otra sub-meta y dispersa la energía de activación hacia los *Comportamientos* asociados; y así sucesivamente.

Las dinámicas de orientación del foco de atención coordinadas por cada MAT pueden estar sujetas a cierto nivel de "inercia" (que se modifica a través de los parámetros globales), el cual evita los efectos producidos por las metas y el estado del entorno continuamente cambiantes. Especialmente, en entornos muy dinámicos, resulta necesario hacer que la MAT se adapte lentamente, o de lo contrario es posible que no logre concluir ninguna de sus metas y sub-metas. Por lo tanto, el mecanismo de activación de *Comportamientos* que dirige cada MAT puede hacerse menos adaptativo y más orientado hacia los planes en curso si se escogen valores para γ y ϕ relativamente pequeños en comparación con π. Por otro lado, una MAT puede hacerse menos predispuesta hacia la consecución de las metas en curso si se escogen valores para γ y ϕ relativamente altos en comparación con π.

- ***Prevención de Metas en Conflicto:*** una secuencia de activación de *Comportamientos* que no sea óptima puede incrementar dramáticamente el número de pasos (ciclos cognitivos) necesarios para alcanzar una meta. Por tanto, cada MAT está capacitada para arbitrar, a través de las reglas de inhibición, los conflictos entre el conjunto de expectativas (efectos deseados y no deseados) que define cada *Comportamiento* de la RC respectiva. El nivel de activación de los *Comportamientos* que intentan "deshacer" una meta protegida (i.e., una meta alcanzada previamente) es disminuido en una fracción de δ. Si δ es lo suficientemente alta (en relación a γ y ϕ), el proceso de activación resulta en un *Comportamiento* que promete seguir conservando dicha meta protegida. Lo mismo aplica para todas las sub-metas.

  Cada *Comportamiento* disminuye el nivel de activación de otros *Comportamientos* que intentan hacer falsas sus precondiciones. De nuevo, esto resulta en la activación de un *Comportamiento* cuyas sub-metas son protegidas y como consecuencia, son evitados los conflictos por metas.

- ***Orientabilidad hacia el Procesamiento Deliberativo:*** una RC sólo puede prever un alcance local (en el tiempo) de sus consecuencias, el cual es determinado por el umbral θ. Por tanto, el comportamiento global de la RC puede hacerse más o menos deliberativo incrementando dicho umbral. El incremento de θ ocasiona que el proceso de dispersión de activación se prolongue por más tiempo (es decir, por un mayor número de ciclos), evitando que el algoritmo caiga en máximos locales.

  Idealmente, se desea configurar un umbral con un valor muy alto (e.g., igual a la activación total de la red). Esto garantizaría que el proceso de dispersión de activación se prolongue lo suficiente de manera que pueda ser seleccionado en cada paso el *Comportamiento* óptimo. Los inconvenientes con establecer un umbral demasiado alto son: (1), que el proceso de activación de *Comportamientos* requeriría mucho tiempo (especialmente para agentes que operan en entornos altamente cambiantes); y (2), que el resultado sería que el agente se estancaría "mentalmente" tratando de considerar los diferentes efectos a largo plazo de las decisiones que toma, resultando, probablemente, en una perdida de esfuerzo proporcional al grado de impredecibilidad del ambiente.

- ***Velocidad:*** la contraparte de la orientabilidad hacia el procesamiento deliberativo es la velocidad. La activación de *Comportamientos* puede hacerse más rápida variando el umbral θ (como se explicó anteriormente). El proceso de activación de *Comportamientos* resultante es, sin embargo, menos "deliberativo", lo cual significa que es menos orientado hacia las metas, menos orientado hacia las situaciones, menos perceptivo hacia las metas en conflicto, y menos orientado hacia los planes en curso. No obstante, en algunas situaciones resulta necesario y útil que el agente pueda reaccionar rápidamente sin desperdiciar demasiado tiempo construyendo planes deliberativos.

  El algoritmo no se ve afectado por los problemas comunes identificados en los planificadores clásicos. Esto es debido a que cada RC no construye un árbol de búsqueda ni mantiene un estado actual hipotético (lo que comúnmente se conoce como el problema de explosión combinatorial). Adicionalmente, el algoritmo evalúa diferentes caminos en paralelo, de tal forma que no tiene que partir de cero cuando un camino no produce una solución, pero lo que sí puede hacer es moverse fácilmente de un plan a otro. La RC no "re-planifica" en cada ciclo cognitivo, debido a que el algoritmo no reinicializa los niveles de activación a cero cuando se ha activado un *Comportamiento*. Como resultado de estas características, el algoritmo es computacionalmente menos costoso.

Debilidades:

- ***Estructura Rígida:*** la estructura de una RC, esto es, el conjunto de *Comportamientos* y conexiones entre los mismos (a través de los enlaces sucesores, los enlaces predecesores, y los enlaces de conflicto), requiere una definición rígida y predeterminada, que no permite que la red se adapte estructuralmente a las necesidades internas (del propio agente) ni externas (ambientales) en tiempo de ejecución. La consecuencia principal de esta debilidad es que la red no puede auto-repararse en

aquellos casos donde el diseño estructural requiere de cambios. La necesidad de realizar estos cambios en la estructura se evidencia a través de:

- *La formación de ciclos de auto-referencia (callejones sin salida):* es probable que las dinámicas de dispersión de activación de la RC formen caminos (secuencias de *Comportamientos*) que no conduzcan a la consecución de ninguna meta global y que por el contrario, formen ciclos de activación que se repiten una y otra vez sin ser advertidos por el agente. Esta anomalía es ocasionada por: (1) la definición circular de los enlaces sucesores y predecesores (e.g., $a \rightarrow b$, $b \rightarrow c$, $c \rightarrow d$, y $d \rightarrow a$); o (2) la ausencia de enlaces de conflicto (inhibitorios) que disminuyan la acumulación de energía de activación en aquellos *Comportamientos* que ya han satisfecho cierta sub-meta.
- *Comportamientos Inalcanzables:* es probable que en el diseño de la RC, en especial cuando ésta se compone de múltiples nodos, existan *Comportamientos* que se encuentran desconectados de otros (i.e., que no poseen enlaces sucesores, predecesores, o de conflicto con otros *Comportamientos*). En este caso, la activación de dichos *Comportamientos* dependerá únicamente de la energía de activación recibida desde una fuente externa (i.e., desde el estado o desde alguna meta, si es que está relacionado con alguna), lo cual hará que su capacidad de "competir" contra otros *Comportamientos* sea muy débil, y por tanto, tendrá una probabilidad reducida o nula de activarse.
- *Auto-inhibición:* cada *Comportamiento* define ciertas proposiciones (en su lista de efectos no deseados) que inhiben las precondiciones de otros *Comportamientos*, e incluso, en algunos casos, producen una auto-inhibición. Por ejemplo, el *Comportamiento* ⟨`Evadir-Obstáculos`⟩ tiene como precondición la proposición ⟨`Con-Obstáculos-Cerca`⟩, la cual debe ser verdadera para su activación, y además define la misma proposición en su lista de efectos no deseados, la cual hará falsa después de ejecutarse. Si el hacer falsa dicha proposición representa para el *Comportamiento* disminuir el nivel de activación de otros competidores, tendría sentido conservarla en su lista de efectos no deseados, de lo contrario, sólo haría al *Comportamiento* menos apto para la competición (puesto que se inhibiría a sí mismo).
- *Omisión de Conexiones Excitativas:* es posible que en el diseño de la RC se omitan ciertas conexiones entre *Comportamientos*, entre estos y el estado, y entre estos y las metas. Probablemente las repercusiones de esto no sean tan dramáticas, sin embargo, la ausencia de estas conexiones provocaría que la dispersión de energía de activación que fluye a través de los enlaces sucesores y predecesores tuviese menos relevancia y, como consecuencia, el impacto de la "cooperación" entre *Comportamientos* sería reducida.

Identificadas estas debilidades sobre el diseño estructural de la RC, se puede decir entonces que la red no posee una estrategia que optimice su diseño a través de un proceso adaptativo. Como consecuencia, se reconoce la necesidad de incorporar un mecanismo que dirija la optimización del diseño estructural de la red.

- ***Parámetros Globales Predefinidos:*** como se evidenció en las fortalezas antes mencionadas, los parámetros globales del algoritmo determinan, en gran medida,

la efectividad y las características del mecanismo de activación de *Comportamientos*. La configuración de tales parámetros globales (θ, δ, γ, ϕ, y π) es un aspecto que depende del contexto del problema, no sólo debido a que cada RC requiere diferentes grados de orientabilidad hacia las metas, orientabilidad hacia las situaciones, velocidad, adaptabilidad, etc., sino también porque el tamaño y la estructura de cada RC determinan dichas características. Por ejemplo, en una RC con una gran cantidad de *Comportamientos*, el umbral de activación θ debe ser definido lo suficientemente alto de manera que en cada ciclo cognitivo se garantice que los *Comportamientos* más relevantes acumulen mayor cantidad de energía de activación en comparación con los demás.

Debido a su dependencia del contexto, los parámetros globales son entonces definidos a priori y de forma empírica por un diseñador (humano). Teniendo en cuenta que cada RC (asociada a una MAT específica) puede definir una configuración de parámetros totalmente diferente a las demás RCs (dependiendo del tipo de influencia que deban tener sus dinámicas de dispersión: orientadas hacia las metas vs. orientadas hacia el estado percibido, adaptabilidad vs. predisposición a los planes en curso, orientadas hacia el procesamiento deliberativo vs. reactividad o velocidad, etc.), se requiere definir entonces, de manera adecuada, los valores respectivos para cada parámetro global, de manera que cada RC refleje la intencionalidad para la cual ha sido diseñada.

Adicionalmente, definir el valor de cada parámetro global es una tarea que incrementa en complejidad debido a dos factores: (1) al aumento del número de RCs requeridas (en función del número de MATs definidas para cada agente); y (2) a la relatividad con que pueden relacionarse entre sí tales parámetros (e.g., se mencionaba que el agente es más orientado hacia la ejecución de los planes y menos adaptativo cuando se definen valores para γ y ϕ "relativamente" pequeños en comparación con π, evidenciando que la proporción entre tales parámetros no es una medida que se pueda definir con precisión y de forma absoluta para todas las RCs, sino que es una medida relativa la cual debería ser ajustada en tiempo de ejecución por el propio agente).

- En síntesis se puede concluir que, en los modelos de dispersión de activación revisados (Maes, 1989; Maes, 1992; Dorer, 2005; Nebel y Babovich-Lierler, 2004), la eficiencia global del algoritmo está sujeta al criterio y a la pericia del diseñador de la concreción contextual de la arquitectura quien define tanto los valores de los parámetros globales como la estructura misma de cada RC. Como consecuencia, el sistema no está dotado con la capacidad de realizar análisis introspectivos acerca de la formación de sus propias estructuras cognitivas, así como tampoco está capacitado para ejercer ningún tipo de "adaptación" (evolución) de tales estructuras ante las condiciones cambiantes que tienen lugar internamente (necesidades del agente) y externamente (cambios ambientales). Con respecto a esto último, dadas ciertas condiciones ambientales, el agente puede requerir definir, para cada una de sus MATs, diferentes parámetros globales con el objetivo de "personalizar" o adaptar cada una de estás a dichas condiciones. Por ejemplo, una MAT cuya meta sea no tropezar con ningún objeto mientras se camina, debe ser una MAT con parámetros que orienten su funcionamiento a un comportamiento global más adaptativo

y reactivo, en comparación con una MAT cuya meta sea planificar el apilamiento (en un orden específico) de un conjunto de objetos, y cuyo comportamiento global debe estar orientado al procesamiento deliberativo, la consecución de las metas y el seguimiento de los planes en curso.

9.3. Ontogénesis del Modelo de Consciencia Sintética

En esta sección se describe un modelo evolutivo que atiende dos aspectos: (1) una propuesta de solución a las debilidades del modelo de dispersión de activación identificadas en la sección anterior, y (2) la definición de un mecanismo que coordina el proceso de orquestación de la consciencia a través de procesos darwinianos.

En este punto, resulta conveniente retomar la idea de las "estructuras cognitivas innatas" vs. las "estructuras cognitivas adquiridas". En el módulo de Orquestación de la Consciencia de METÁFORA, las estructuras cognitivas innatas corresponden a aquellas Redes de Comportamientos las cuales son definidas, bien a través del proceso filogenético del agente (i.e, determinadas en cada fase evolutiva – lo cual será descrito en la sección 10.4), o mediante el diseño a priori, preprogramado, de dichas estructuras por parte de un diseñador humano). Por otro lado, las estructuras cognitivas adquiridas hacen referencia a las Redes de Comportamientos que se "desarrollan" durante el ciclo de vida del agente a través de un proceso ontogenético.

La razón por la cual se requiere distinguir estos dos tipos de RCs se debe a que son mutuamente cooperantes: las RCs innatas permiten al agente interactuar con un entorno sin tener mucho conocimiento acerca de éste; en la medida que el agente interactúa con este entorno, adapta y ajusta la anatomía de sus RCs a través de un proceso darwiniano; luego, estas nuevas RCs adaptadas y mejoradas servirán como base para la formación de nuevas estructuras cognitivas innatas en la siguiente generación; y así sucesivamente.

Considérese la siguiente idea que esboza el neurofisiólogo Antonio Damasio (Damasio, 1994): *Ni nuestro cerebro ni nuestra mente son "tabulae rasae"*[9] *cuando nacemos. Pero tampoco están completamente determinados desde el punto de vista genético. Los genes se encargan de un componente cerebral con estructura pre-establecida (innata), y de otro componente en el que la estructura pre-establecida todavía ha de determinarse bajo la influencia de tres elementos: (1) la estructura pre-establecida; (2) la actividad y las circunstancias individuales (experiencia); y (3) las presiones de auto-organización que surgen de la misma complejidad del sistema.*

Adaptando la idea de Damasio, la experiencia que adquiere el agente como consecuencia de su interacción con el entorno, modela el diseño de las estructuras cognitivas (RCs): la propia estructura y las dinámicas entre sus componentes puede cambiar a lo largo de todo el ciclo de vida para reflejar diferentes experiencias del agente y, como resultado, el diseño de tales estructuras cognitivas sigue cambiando. Las RCs no sólo son

[9] En filosofía, tabulae rasae o tabla rasa hace referencia a la tesis epistemológica de que cada individuo nace con la mente "vacía", es decir, sin cualidades innatas, de modo que todos los conocimientos y habilidades de cada ser humano son exclusivamente fruto del aprendizaje a través de sus experiencias y sus percepciones sensoriales.

receptivas a los resultados de la primera experiencia, sino que son repetidamente dúctiles y modificables por las experiencias continuadas.

Esta *plasticidad* en la formación de las estructuras cognitivas (RCs) que sirven como substrato al proceso de orquestación de la consciencia sintética, están sujetas a las presiones auto-organizativas que articulan los procesos darwinianos (proceso que ha sido introducido ya en las secciones 4.4.3.2 y 6.3.2.2). De nuevo, la metáfora de los procesos darwinianos resulta útil en este nivel organizativo de la arquitectura (ontogenético), permitiendo que las RCs "compitan" entre sí dentro de un área de trabajo, auto-replicándose y recombinándose, hasta crear la "resonancia" suficiente que permite que una determinada RC pase a un primer plano de la consciencia.

Téngase en consideración que cada RC es la estructura cognitiva que se desarrolla a través de procesos continuos de interacción (con el ambiente) y de evolución (mediante los procesos darwinianos); y que la MAT es el componente que contiene la RC asociada junto con el GFA respectivo. Con el objetivo de no sobrecargar el proceso de orquestación de la consciencia, el desarrollo ontogenético es realizado a lo largo de múltiples ciclos cognitivos y no en cada uno de estos, de lo contrario el agente no podría actuar reactivamente ante los cambios del ambiente. Los resultados parciales de este enfoque propuesto han sido publicados en (Romero, 2010).

9.3.1. Máquina Darwiniana para el Proceso Ontogenético

En esta sección se describe la Máquina Darwiniana (MD) encargada de coordinar el proceso de desarrollo ontogenético de las Redes de Comportamientos. Para esto, se propone emplear una solución basada en Programación Genética, más concretamente, empleando la Programación de Expresiones Genéticas (PEG) (Ferreira, 2001). Los principios básicos sobre la programación genética fueron expuestos en la sección 2.2.4.2.

Usualmente, en la PEG, se evoluciona una población de individuos (donde cada uno de ellos representa un programa informático o una porción de éste), con el objetivo de optimizar la solución a un problema. Sin embargo, en la siguiente sección se propone el empleo de este modelo computacional (la PEG) no para evolucionar programas informáticos, sino para optimizar el diseño de las Redes de Comportamientos descritas en la sección anterior.

> *"Los problemas no se resuelven con nueva información, sino organizando lo que sabemos desde hace mucho tiempo".*
> — Ludwig Wittgenstein, Philosophical Investigations, 1953.

Tomando como referencia la cita anterior, se puede decir que las soluciones (RCs evolucionadas y optimizadas) generadas por la máquina darwiniana se logran merced a las combinaciones y variaciones de fragmentos del diseño de diferentes RCs, los cuales (probablemente) nunca se habían combinado antes de cierto modo específico. Vale la pena evocar el caso, por ejemplo, cuando los expertos en aeronáutica han señalado que todas las partes de los aviones ya se tenían antes de que estos aparecieran (e.g., el motor, el diseño de los alerones, la forma de las alas, etc.), pero hubo de llegar alguien a quien

se le ocurriera combinarlas de una forma determinada para producir la primera aeronave que funcionara con éxito.

Entiéndase entonces la función de la Máquina Darwiniana (MD) que opera en este nivel de organización de la arquitectura, como un componente de *introspección* que rediseña, en tiempo de ejecución, las estructuras cognitivas subyacentes al proceso de consciencia sintética del agente. Es por este motivo que el *Mecanismo de Gestión de los procesos EOF (Epigenéticos, Ontogenéticos y Filogenéticos)* del módulo Meta-Cognitivo de METÁFORA, es el encargado de dirigir dicho proceso de optimización de los diseños.

9.3.1.1. Representación Fenotípica: Árboles de Expresión

Considérese la siguiente Red de Comportamientos (simplificada):

Comportamiento: C1:	
Precondiciones:	⟨proposición, P1⟩ ⟨proposición, P5⟩
Efectos Deseados:	⟨Meta, M1⟩
Efectos No Deseados:	⟨proposición, P2⟩

Comportamiento: C2:	
Precondiciones:	⟨proposición, P2⟩ ⟨proposición, P3⟩
Efectos Deseados:	⟨proposición, P4⟩
Efectos No Deseados:	null

Comportamiento: C3:	
Precondiciones:	⟨proposición, P4⟩
Efectos Deseados:	⟨proposición, P5⟩
Efectos No Deseados:	⟨proposición, P3⟩

En la figura 9.4 se representa esta Red de Comportamientos. En la parte a. de la figura, se presenta el modelo conceptual de la RC (es decir, todos los *Comportamientos* que la conforman y los diferentes enlaces de excitación e inhibición entre los mismos); en la parte b. de la figura, se presenta un árbol de expresiones que representa el modelo fenotípico de la RC; y en la parte c. de la figura se presenta el modelo genotípico de la RC (empleando un cromosoma de longitud determinada compuesto por tres genes y una función de enlace).

A lo largo de esta sección y de la siguiente, se explicarán al detalle los mecanismos de conversión y traducción desde una representación hacia otra (fenotipo $\rightarrow$ genotipo, y

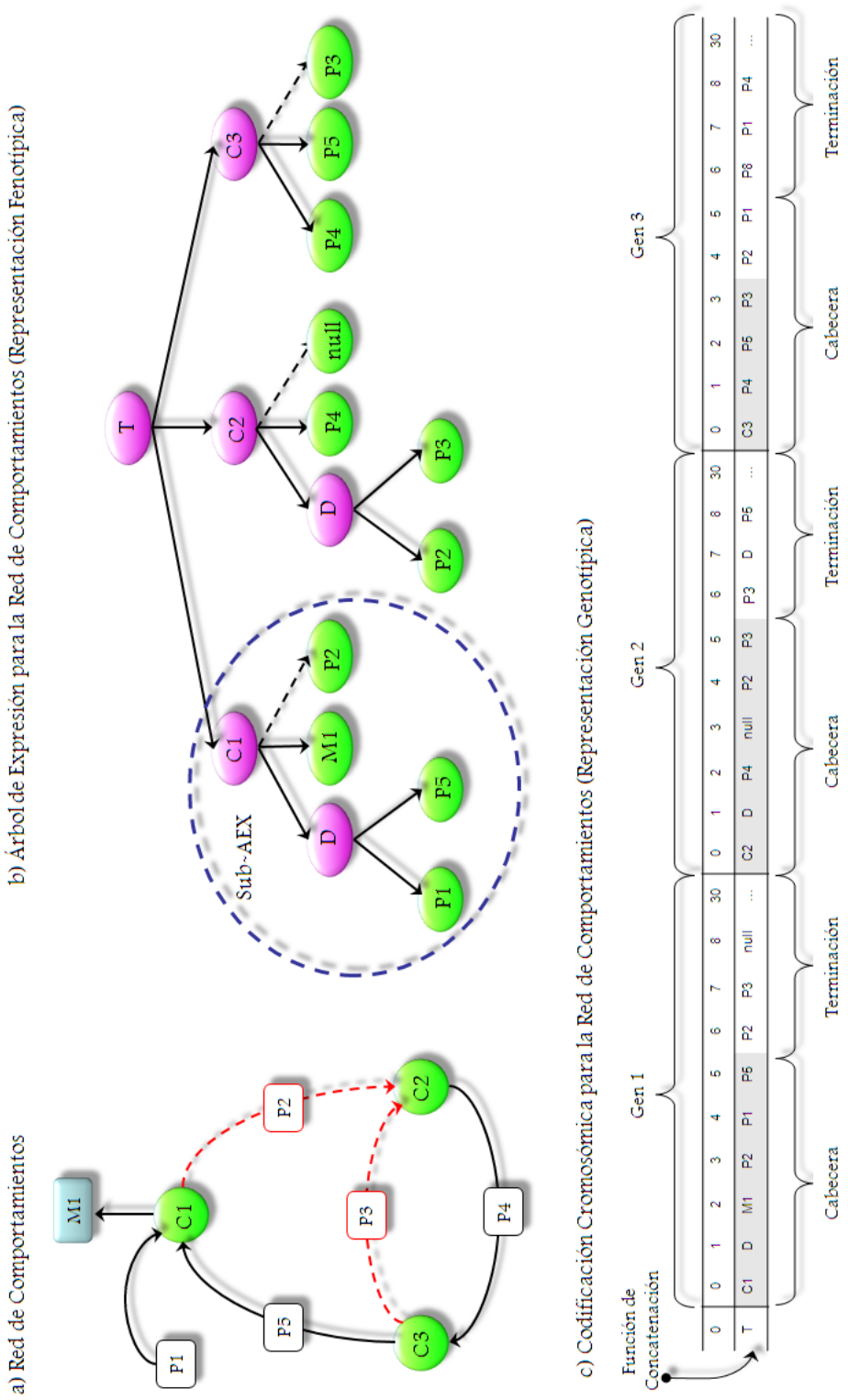


Figura 9.4: Representación Fenotípica y Genotípica de una RC.

viceversa). Sin embargo, antes de entrar en materia, conviene tener claro cómo se estructuran las representaciones fenotípicas de una RC a través de los Árboles de Expresión (AEX). En un AEX se definen dos tipos de nodos: los *nodos función* y los *nodos terminal*. El conjunto de nodos terminales está compuesto por las posibles proposiciones que sirven para la activación e inhibición de los *Comportamientos*. Por otro lado, el conjunto de nodos función está compuesto por los operadores constructos que constituyen una RC.

Concretamente, los nodos terminales poseen una aridad de 0 (i.e, son los nodos "hoja" del árbol), y corresponden a cuatro tipos de proposiciones asociadas a cada *Comportamiento*: las precondiciones, los efectos deseados, los efectos no deseados, y las metas. En la parte b. de la figura 9.4, estas proposiciones están representadas por todos los nodos $\mathbf{P}_1$, $\mathbf{P}_2$,... $\mathbf{P}_n$, por todos los nodos $\mathbf{M}_1$, $\mathbf{M}_2$,... $\mathbf{M}_n$, y por las referencias nulas **null** (las referencias nulas resultan útiles para indicar que alguna de las listas de efectos del *Comportamiento* está vacía).

En cuanto a los nodos función, estos pueden ser:

- **Funciones de Conectividad**: estas funciones agrupan un número determinado de nodos, los cuales pueden ser tanto funciones como terminales. Se definen 2 tipos de funciones de conectividad: **D**, es una función que agrupa dos nodos; y **T**, es una función que agrupa tres nodos. Si se requiere agrupar más de tres nodos, se utiliza una combinación de las funciones anteriores. Por ejemplo, para agrupar 8 nodos, se podría emplear un nodo **T** cuyas ramas contendrían a su vez tres nodos de conectividad: dos nodos **T** y un nodo **D** (3 + 3 + 2 = 8). Se definen sólo dos funciones de conectividad puesto que no es conveniente emplear un gran número de funciones, esto debido a que el tamaño del espacio de búsqueda aumentaría (principio de parsimonia, (Poli et al., 2008)).
- **Funciones de Comportamiento**: estas funciones representan los *Comportamientos* contextuales que constituyen la RC (representados por **C1**, **C2**,... $\mathbf{C}_n$). Cada nodo de este tipo tiene una aridad de 3, donde la primera rama representa las precondiciones, la segunda rama los efectos deseados, y la tercera rama los efectos no deseados del *Comportamiento*. Una función de comportamiento se conecta únicamente con nodos terminales o con funciones de conectividad. Por ejemplo, obsérvese la función comportamiento $\mathbf{C}_1$ de la figura 9.4.b. Esta función posee tres ramas, la primera rama a la izquierda, contiene dos nodos terminales $\mathbf{P}_1$ y $\mathbf{P}_5$, agrupados a través de una función de conectivad **D**, los cuales representan las precondiciones del *Comportamiento*; en la segunda rama, se encuentra el nodo terminal $\mathbf{M}_1$ (meta) el cual representa un efecto deseado del *Comportamiento*; y finalmente, en la tercera rama, se encuentra el nodo terminal $\mathbf{P}_2$ que representa el efecto no deseado de $\mathbf{C}_1$. Nótese que para efectos explicativos, las ramas inhibitorias en el AEX se encuentran punteadas, igual que sus correspondientes análogos en el modelo estructural de la RC en la parte a. de la figura 9.4. Nótese además que el nodo terminal $\mathbf{P}_1$ en la primera rama de $\mathbf{C}_1$ representa una proposición que es activada por un estímulo externo (ambiental), esto es, que no corresponde a ningún efecto excitativo o inhibitorio producido por algún otro *Comportamiento* (lo cual se comprueba al verificar que este nodo terminal no forma parte de los efectos deseados de ningún otro *Comportamiento* en el AEX).

Como se puede observar en la figura 9.4, la traducción *fenotipo* → *genotipo* de una RC, o mejor, la conversión de un AEX en un cromosoma lineal, implica por un lado, la subdivisión del AEX en varios sub-árboles de expresión (Sub-AEX) y, por otro, la subdivisión del cromosoma en segmentos de longitud preestablecida, los *genes*, los cuales se corresponden uno a uno con cada Sub-AEX. Cada gen define, por sí mismo, un sólo *Comportamiento* de la RC (i.e., sus precondiciones, sus efectos deseados, y sus efectos no deseados). Así pues, la concatenación de múltiples genes conforma un cromosoma el cual representa toda una Red de Comportamientos. Por ahora, se centrará la atención en el diseño de cada gen, y en la sección 9.3.1.3 se discutirá como estos se organizan para conformar un cromosoma.

9.3.1.2. Marcos de Lectura Abiertos y Genes

La estructura organizacional de los genes que emplea la PEG es concebida en términos de marcos de lectura abiertos (MLA). En biología, un MLA, o la secuencia de codificación de un gen, empieza con un *codón*[10] de inicio, continua con los *codones* de aminoácidos, y termina con un *codón* de finalización.

Como se mencionó anteriormente, cada gen codifica un sólo *Comportamiento* de la RC. Por ejemplo, considérese el siguiente *Comportamiento*:

Comportamiento: C1:	
Precondiciones:	⟨proposición, P1⟩ ⟨proposición, P2⟩
Efectos Deseados:	⟨meta, M1⟩ ⟨proposición, P3⟩ ⟨proposición, P4⟩
Efectos No Deseados:	⟨proposición, P5⟩

El AEX correspondiente se presenta en la figura 9.5 y su representación genética se codifica en la expresión 9.1.

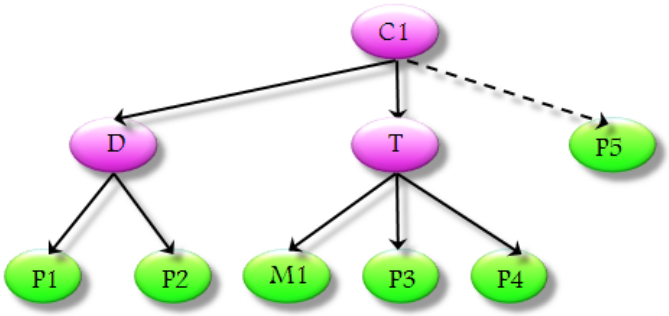


Figura 9.5: Representación Fenotípica del Comportamiento C1.

[10]Un codón es una sub-unidad genética de tres nucleótidos, en un ARN mensajero, que determina qué aminoácido se encuentra en un punto en una cadena proteínica.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	30
C_1	D	T	P_5	P_1	P_2	M_1	P_3	P_4	P_1	P_2	P_2	P_3	null	P_6	...

Expresión Genética 9.1

Nótese que un MLA se representa empleando la notación Karva (Ferreira, 2001), la cual consiste en inferir el genotipo de la RC a partir de su representación fenotípica. Esto es, se representa cada nodo del AEX haciendo un recorrido de arriba hacia abajo y de izquierda a derecha del árbol. Por ejemplo, el nodo raíz del AEX ($\mathbf{C}_1$), se sitúa en la posición 0 del MLA; a continuación se recorre horizontalmente el siguiente nivel del árbol donde están los nodos **D**, **T** y $\mathbf{P}_5$, los cuales se sitúan respectivamente en las posiciones 1, 2, y 3 del MLA; y así sucesivamente. Finalmente se obtiene un MLA de 9 elementos (posiciones 0 - 8 del gen que se muestran sombreadas en la expresión 9.1) para el *Comportamiento* propuesto. La utilidad de la región no sombreada del gen se explicará más adelante.

La traducción en el sentido contrario, es decir, la construcción de un AEX a partir de un MLA de un gen, sigue las siguientes pautas:

1. El primer elemento del MLA corresponde al nodo raíz del AEX, esto es, una función comportamiento.
2. Dependiendo del número de argumentos de cada elemento del MLA (recuérdese que las funciones pueden tener distintas cantidades de argumentos mientras que las terminales tienen una aridad de 0), se construye un nivel del árbol formado por tantos nodos como el número de argumentos tengan las funciones del nivel anterior.
3. De izquierda a derecha, son rellenados los nodos, en el mismo orden que aparecen en el vector de elementos del MLA.
4. El proceso se repite desde el segundo paso, hasta que se construya un nivel del árbol conformado sólo por terminales (nodos hoja). Nótese que el hecho de que se garantice que cada AEX tenga sólo nodos terminales en su nivel más bajo es equivalente a decir que el AEX es *sintácticamente correcto*.

Por ejemplo, considérese ahora el MLA para el *Comportamiento* $\mathbf{C}_2$ que se presenta en la expresión 9.2:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	30
C_2	P_1	T	D	M_2	D	P_2	P_1	P_6	P_3	P_5	P_2	P_3	null	P_6	...

Expresión Genética 9.2

Aplicando las pautas antes descritas a la expresión genética 9.2 se obtiene que: el nodo raíz del árbol (primer nivel) corresponde a la función comportamiento $\mathbf{C}_2$. Debido

a que ésta función tiene una aridad igual a tres, entonces en el segundo nivel del árbol deben construirse tres nodos, los cuales corresponden a los elementos $\mathbf{P}_1$, **T** y **D**; en el tercer nivel del árbol, se construyen cinco nodos debido a que las dos funciones de conectividad en el nivel anterior tienen una aridad de 3 y 2 argumentos respectivamente, con lo cual se agregan los siguientes cinco elementos del MLA ($\mathbf{M}_2$, **D**, $\mathbf{P}_2$, $\mathbf{P}_1$, y $\mathbf{P}_6$); por último, en el cuarto nivel del árbol se crean 2 nodos (debido a la aridad de la función **D** del nivel anterior), y se insertan los siguientes dos elementos del MLA ($\mathbf{P}_3$ y $\mathbf{P}_5$). En la figura 9.6 se aprecia el AEX resultante.

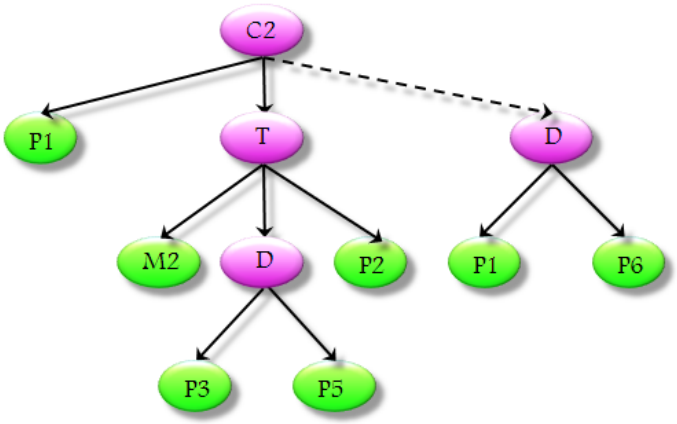


Figura 9.6: Representación Fenotípica de C2 a partir del MLA 9.2.

Y el *Comportamiento* resultante sería:

Comportamiento: C2:	
Precondiciones:	⟨proposición, P1⟩
Efectos Deseados:	⟨meta, M2⟩ ⟨proposición, P2⟩ ⟨proposición, P3⟩ ⟨proposición, P5⟩
Efectos No Deseados:	⟨proposición, P1⟩ ⟨proposición, P6⟩

En el contexto de la PEG, los cromosomas (i.e, las representaciones lineales de las RCs) poseen una longitud invariable, y estos a su vez pueden estar conformados por uno o más genes de igual longitud. Por tanto, la longitud de un gen también es invariable. Entonces, nótese que aquello que varía en la PEG no es la longitud de los genes sino la longitud de los MLAs. Ahora, la longitud de un MLA puede ser igual o menor que la longitud de un gen. En el primer caso, el codón de finalización coincide con la finalización del gen, y en el segundo caso, el codón de finalización se encuentra en algún punto del gen antes de su terminación, quedando una región del gen sin codificar. Estas regiones sin codificar en un gen son realmente la esencia de la PEG puesto que permiten modificar el genoma de cada RC a través de la aplicación, sin restricción, de los operadores

genéticos, generando siempre estructuras cognitivas sintácticamente correctas. Así pues, el hecho de que cada MLA pueda tener una longitud diferente concede gran flexibilidad al diseño evolutivo de RCs de tamaño y grado de complejidad variables.

Los genes están compuestos por dos segmentos: una *cabecera* y una *terminación*. La *cabecera* contiene elementos que representan tanto funciones como terminales, mientras la *terminación* contiene sólo terminales. En cada concreción contextual de la arquitectura, debe ser definida la longitud de la cabecera (c), mientras que la longitud de la terminación (t) se calcula en función de c y de una constante n que representa la cantidad de argumentos de la función con mayor aridad. Así pues, la longitud de t se obtiene de la siguiente manera:

$$t = c \times (n - 1) + 1 \tag{9.16}$$

Por ejemplo, considérese nuevamente la expresión genética 9.2 antes presentada. Suponiendo que se define un valor para la cabecera $c = 10$[11]. Teniendo en cuenta que las funciones con mayor cantidad de argumentos son la función de comportamiento y la función de conectividad **T** (cuya aridad es 3), entonces se tiene que $n = 3$. Luego $t = 21$ y la longitud total del gen es de 31 elementos ($c + t$). En este caso, obsérvese que el MLA finaliza en la posición 10 donde se encuentra el nodo terminal $\mathbf{P}_5$ (véase la figura 9.6), la cabecera finaliza en la posición 9 (debido a que $c = 10$), y el gen finaliza en la posición 30, como se observa en la expresión 9.3.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
C_2	P_1	T	D	M_2	D	P_2	P_1	P_6	P_3	P_5	P_2	P_3	null	P_6	P_2

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
P_1	P_4	P_5	P_2	P_1	P_1	P_3	P_4	P_2	P_5	P_5	P_5	P_4	P_3	P_2	

Expresión Genética 9.3

Supóngase ahora que el gen de la expresión 9.3 sufre una mutación en la posición 2, variando la función **T** por la terminal $\mathbf{P}_5$, de lo cual se obtiene el gen de la expresión 9.4, y cuya representación fenotípica se presenta en la figura 9.7.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	30
C_2	P_1	P_5	D	M_2	D	P_2	P_1	P_6	P_3	P_5	P_2	P_3	null	P_6	...

Expresión Genética 9.4

[11] Se escoge un valor bastante pequeño en este ejemplo con el fin de simplificar su representación, sin embargo, en las concreciones contextuales, este parámetro debe ser estimado apropiadamente y su valor debe reflejar un tamaño máximo deseable para la estructura de cada *Comportamiento* de las RCs.

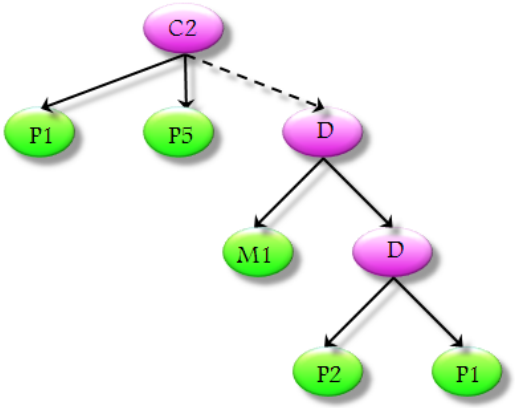


Figura 9.7: Representación Fenotípica del Comportamiento C2 a partir del MLA 9.4.

Obsérvese que debido a que la función **T** es reemplazada por la terminal $\mathbf{P}_5$, el AEX reduce su profundidad en la rama de los efectos deseados, mientras que aumenta su profundidad en la rama de los efectos no deseados. De esta manera, el MLA se contrae desde la posición 10 hasta la posición 7 del gen, creando un *Comportamiento* estructuralmente distinto al de la figura 9.6, lo cual afecta las dinámicas de dispersión de activación de la RC debido a los nuevos enlaces de excitación e inhibición incorporados[12]. De manera similar, las variaciones genéticas (mutaciones o recombinaciones) pueden generar también cambios en el MLA haciendo que éste requiera más elementos (proposiciones) para construir el AEX, y por lo tanto causando que la longitud del MLA aumente.

De esta manera, a pesar de que su longitud es fija, cada gen tiene el potencial de codificar AEXs de diferentes tamaños y formas, siendo el más simple, un AEX compuesto por sólo un nodo (e.g., cuando el primer elemento del gen es un nodo terminal); y el más extenso, un AEX compuesto por tantos nodos como los permitidos por la longitud del gen (e.g., cuando todos los elementos de la cabecera son funciones con el máximo número de argumentos, es decir, funciones de conectividad T).

En resumen, para asegurar que las modificaciones en el genoma produzcan AEXs sintácticamente correctos, se debe garantizar que:

- El primer elemento de cada gen (que corresponde al nodo raíz de su respectivo AEX), debe ser siempre una función comportamiento, puesto que cada gen codifica la estructura de un *Comportamiento* específico de la RC. Como se verá más adelante, las variaciones causadas por los operadores genéticos siempre respetan esta regla.
- Siempre se deben aplicar las validaciones antes descritas para la construcción de nodos función y nodos terminal (e.g., que la función comportamiento sólo se conecta con funciones de conectividad o con terminales, etc.).

[12]Esta capacidad que tiene el algoritmo de convertir una estructura cognitiva en otra totalmente diferente a partir de las variaciones en el genoma, resulta similar a lo que en la "biología del desarrollo" se conoce como *homeosis*, que es la transformación de una parte del cuerpo en otra como resultado de la mutación o de la alteración de la expresión de genes críticos.

- Por último, que los nodos función sólo pueden formar parte de la cabecera del gen, puesto que en la terminación del mismo sólo habrán nodos terminales.

9.3.1.3. Cromosomas Multigénicos

En la PEG es posible definir cada cromosoma como la composición de uno (cromosoma unigénico) o varios genes (cromosoma multigénico). En la presente propuesta, se define una estructura multigénica para cada cromosoma teniendo en cuenta las siguientes justificaciones:

- Los cromosomas multigénicos permiten la articulación de estructuras más complejas y jerárquicas a partir de bloques de construcción más simples, los genes. Se propone que cada cromosoma contenga múltiples genes, donde cada gen representa, a través de un sub-árbol de expresión único (funcional y estructuralmente), la estructura de cada *Comportamiento* que conforma la RC. De esta manera, cada Sub-AEX es una entidad estructuralmente independiente, y a su vez forma parte de una estructura jerárquica más compleja (i.e., la RC expresada como la conexión entre los diferentes sub-AEXs), evidenciándose, como es característico de los sistemas complejos, que "el todo" es más que la suma de sus partes.
- La articulación de múltiples genes que cooperan en la construcción de una estructura más compleja (la RC) es una manera de "co-evolucionar". El esfuerzo individual que hace cada gen por refinar la estructura de cada *Comportamiento* resulta en beneficios recíprocos para todo el cromosoma. Esto es pues, una micro co-evolución de las estructuras cognitivas dirigida por el proceso ontogenético (como se discutió en la sección 4.4.3.3).
- Debido a que se requiere que el proceso de desarrollo evolutivo de las RCs no consuma cantidades significativas de tiempo, se propone aplicar la metáfora de las "islas" para introducir cierta aceleración en dicho proceso evolutivo. Cada gen del cromosoma, evolucionará de forma aislada y en paralelo a los demás, de tal manera que su búsqueda de un óptimo local desencadenará en la emergencia de un óptimo global que se obtiene al evaluar la estructura de toda la RC compuesta por los diferentes sub-AEXs.

Como se puede apreciar en el ejemplo de la figura 9.8, la descripción fenotípica y genotípica de la RC requiere de la interacción de tres Sub-AEXs de diversos tamaños y formas, donde cada uno expresa los diferentes *Comportamientos* que componen la red. El AEX resultante integra los diferentes Sub-AEXs a través de una *función de concatenación*, la cual corresponde, en el ejemplo, a una función de conectividad **T**. Las funciones de concatenación no son generadas por el proceso mismo de evolución dirigido por la PEG, sino a través de un mecanismo de post-traducción[13] que determina el número de Sub-AEXs que deben ser concatenados, y a partir de éste se genera la correspondiente función de conectividad (**D** o **T**) o una combinación de éstas (en el caso de que el número

[13] Se emplea el término de post-traducción para hacer hincapié en que las funciones de concatenación son agregadas a la solución "después" de decodificar el cromosoma y sus diferentes genes en árboles de expresión (AEXs).

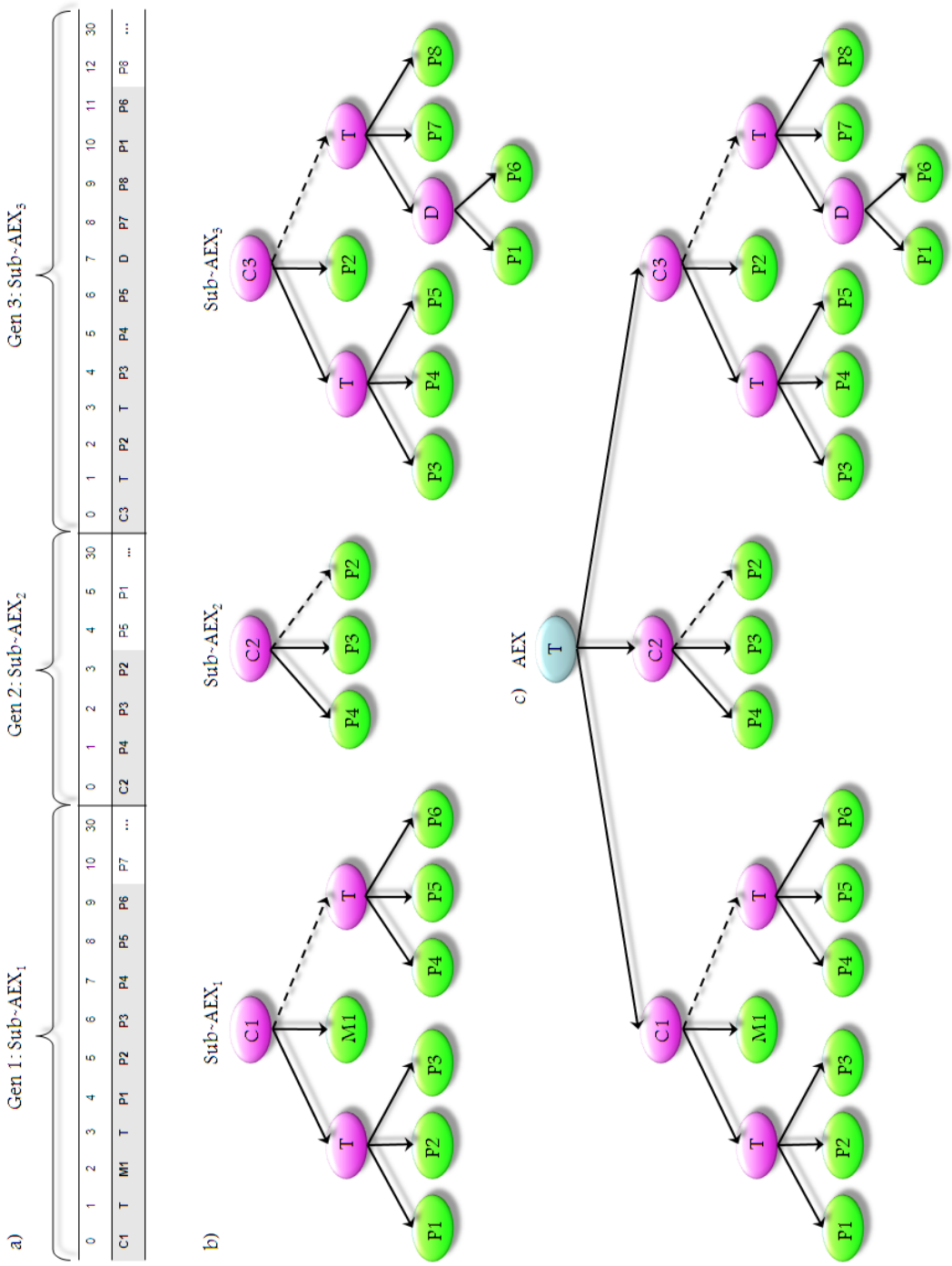


Figura 9.8: Cromosoma Multigénico. a) cromosoma formado por tres genes. En aras de la claridad los elementos del MLA de cada gen se muestran sombreados y los segmentos "terminación" de cada gen no son presentados. b) Sub-árboles de expresión (Sub-AEXs) codificados por cada gen. c) AEX resultante que emplea la función de conectividad T como función de concatenación. Este AEX representa la estructura total de la RC.

de Sub-AEXs sea superior a 3). Estableciendo una analogía con la biología, este proceso resulta similar al ensamblaje de diferentes sub-unidades proteicas que genera como

resultado, proteínas multiméricas (complejos proteicos) como las de la polimerasa del ADN.

Por último, resulta evidente que cada cromosoma multigénico definirá tantos genes como *Comportamientos* tenga el agente. Si una RC determinada no emplea alguno de los *Comportamientos* del agente, simplemente el gen correspondiente determinará la ausencia de conexiones de dicho *Comportamiento* con los demás. Esto es posible asignando la terminal "null" en las ramas del Sub-AEX correspondientes a los efectos deseados y no deseados del *Comportamiento* (segundo y tercer argumento de la función comportamiento). De esta manera, el *Comportamiento* quedaría fuera de las dinámicas de dispersión de la RC. Por ejemplo, el *Comportamiento* $\mathbf{C}_4$ que describe el gen de la expresión 9.5, no se conectará con ningún otro *Comportamiento* de la red (a menos de que sufra alguna variación en su genoma, claro está).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	30
C_4	P_6	null	null	P_2	D	P_2	P_1	P_6	P_3	P_5	P_2	P_3	P_6	P_7	...

Expresión Genética 9.5

9.3.1.4. Dominio Genético para los Parámetros Globales

Con el objetivo de evolucionar no sólo la estructura topológica de la RC sino la eficiencia de la misma (lo cual se determina a través de los "parámetros globales" que se encargan de modular las dinámicas de la RC), se define entonces un gen adicional al cromosoma multigénico compuesto por cinco elementos, uno por cada parámetro global (π, α, δ, γ, y ϕ). De este modo, cada cromosoma queda conformado por un determinado número de genes que representan los diferentes *Comportamientos* de la red (que se denominarán de ahora en adelante *genes estructurales*), y por un gen que representa los parámetros globales de la RC (que se denominará *gen funcional*).

Cromosoma:

Gen Estructural 1						Gen Estructural 2						Gen Funcional				
0	1	2	3	4	30	0	1	2	3	4	30	0	1	2	3	4
C_1	D	M_2	P_1	P_2	...	C_2	P_3	P_1	D	P_4	...	3	0	7	5	9
												π	α	δ	γ	ϕ

D_{pg}:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	...	20
1	5	10	15	20	25	30	35	40	45	50	55	60	65	...	95

Expresión Genética 9.6

En la expresión genética 9.6, se presenta un cromosoma compuesto por 3 genes: dos genes estructurales que codifican respectivamente dos Sub-AEXs diferentes, y un gen funcional (sombreado) que representa los parámetros globales de la RC. Adicionalmente, se presenta un vector que corresponde al dominio de los parámetros globales (D_{pg}). La conexión entre el gen funcional y el vector D_{pg} es la siguiente:

- Cada posición del gen funcional representa un parámetro global de la RC estrictamente en el siguiente orden: la posición 0 representa π, la posición 1 representa α, la posición 2 representa δ, la posición 3 representa γ, y la posición 4 representa ϕ.
- El valor de cada elemento del gen funcional indica el índice del elemento del vector D_{pg} asociado. Por ejemplo, en el gen funcional π tiene un valor de 3, lo cual significa que su valor real corresponde al elemento en la posición 3 del vector D_{pg}, es decir 15. Consecuentemente, según el ejemplo, los valores para los parámetros globales sería: $\pi = 15$, $\alpha = 0$, $\delta = 35$, $\gamma = 25$, y $\phi = 45$.

Como resulta evidente, los valores de cada parámetro global pueden variar a través del proceso evolutivo, mediante el cambio de referencia de los elementos del vector D_{pg}. Adviértase que la longitud del vector D_{pg} es predefinida y debe contener tantos valores como los requeridos para ajustar las dinámicas de la RC (idealmente se empleará un vector de 100 posiciones, con valores del 1 al 100, sin embargo esto puede variar en cada concreción contextual de la arquitectura).

9.3.1.5. Operadores Genéticos

En esta sección se describen los operadores genéticos empleados durante el proceso ontogenético, los cuales proveen el grado de diversificación genética necesario que da paso a la evolución de las RCs. A continuación se explica cómo la potente interacción entre los operadores genéticos de selección, replicación, recombinación y mutación, no sólo posibilita la evolución efectiva a través de los procesos de diversificación y refinamiento continuo, sino que además permite la duplicación con variación de los bloques de construcción, su circulación dentro del acervo genético, y la emergencia de nuevas estructuras cognitivas más complejas a partir de dichos bloques.

Los operadores genéticos aquí expuestos resultan similares a los empleados por los Algoritmos Genéticos (descritos en las secciones 7.4.3.1 y 8.6.3.3) aunque con ciertas variaciones que conservan la coherencia representacional tanto de los cromosomas como de los AEXs. Adicionalmente, cabe notar que los operadores genéticos actúan de forma diferente sobre cada dominio específico, esto debido a las diferencias entre los alfabetos empleados por cada tipo de gen y por cada segmento del gen. Esto es, por ejemplo, el segmento cabecera de un gen estructural emplea el alfabeto $A_c = \{funciones\} \cup \{terminales\}$, el segmento terminación de un gen estructural emplea el alfabeto $A_t = \{terminales\}$, y los genes funcionales emplean un alfabeto conformado por un conjunto de constantes tal que $A_f = D_{pg}$.

Operadores de Selección y Replicación: todos los algoritmos basados en computación evolutiva definen un esquema para seleccionar los cromosomas que pasarán a la siguiente generación, de acuerdo con una función de aptitud. El operador de selección se basa en el esquema probabilístico del método de la ruleta (Goldberg, 1989) e incorpora la clonación del mejor cromosoma (selección elitista) como una imitación aproximada al proceso biológico de selección natural, lo cual refleja muy buenos resultados (Ferreira, 2001).

De acuerdo con el método de la ruleta, los cromosomas son seleccionados para ser replicados empleando un método estocástico que toma en cuenta su nivel de aptitud (como se explicó en la sección 7.7.1). Aunque el operador de replicación resulta vital, es el operador menos interesante. Durante la replicación, los cromosomas son copiados fielmente dentro de la siguiente generación. cuanto más apto sea el cromosoma (es decir, cuanto más alta sea su aptitud) mayor será su probabilidad de reproducirse. Por consiguiente, durante la replicación, el genoma de los cromosomas seleccionados es copiado tantas veces como lo favorezca la ruleta, y a su vez, la ruleta será girada tantas veces como cromosomas se requieran en la población, manteniendo siempre un tamaño constante de ésta.

Operador de Mutación: el operador de mutación puede actuar en cualquier punto del cromosoma. Lo único que se debe tener en cuenta es el alfabeto que emplea cada segmento del gen. Respecto a este operador son tomadas en cuenta las siguientes consideraciones:

- Las mutaciones que tienen lugar en el segmento cabecera de un gen estructural pueden modificar el elemento mutado por cualquier función de conectividad (**T** o **D**) o bien por algún elemento del conjunto de terminales (proposiciones y metas).
- Nótese que a fin de preservar la consistencia fenotípica, la mutación puede efectuarse en cualquier punto del cromosoma excepto en el primer elemento de cada gen, esto es, donde se sitúa la función comportamiento **C** que inicia la secuencia génica relativa a la estructura de cada *Comportamiento*.
- Así mismo, se restringe el uso de la función comportamiento **C** en cualquier otro punto del cromosoma, con lo cual las mutaciones nunca consideran la inserción de este elemento.
- Por otro lado, las mutaciones que ocurren en el segmento terminación de un gen estructural sólo pueden modificar el elemento mutado por un elemento correspondiente al conjunto de terminales.
- Por último, una mutación de un elemento del gen funcional del cromosoma sólo puede cambiarse por otro elemento del conjunto de constantes que se definen para los parámetros globales, esto es, la mutación cambia el elemento al que refiere el parámetro global por cualquier otro elemento del vector D_{pg}.

Por defecto se define una probabilidad de mutación (p_m) equivalente a 0.5, que se aplica dos veces: una sobre los genes estructurales y la otra sobre el gen funcional. Por ejemplo, considérese el cromosoma de la expresión 9.7:

Gen Estructural 1							Gen Estructural 2							Gen Funcional				
0	1	2	3	4	5	30	0	1	2	3	4	5	30	0	1	2	3	4
C_1	P_1	P_2	P_3	P_4	P_5	...	C_2	P_3	P_1	P_4	D	P_6	...	3	0	7	5	9

Expresión Genética 9.7

Este cromosoma es mutado en dos posiciones: en la posición 1 del primer gen estructural, y en la posición 3 del gen funcional, produciendo como resultado el gen de la expresión 9.8.

Gen Estructural 1							Gen Estructural 2							Gen Funcional				
0	1	2	3	4	5	30	0	1	2	3	4	5	30	0	1	2	3	4
C_1	D	P_2	P_3	P_4	P_5	...	C_2	P_3	P_1	P_4	D	P_6	...	3	0	7	12	9

Expresión Genética 9.8

Como se había comentado anteriormente, los cambios producidos por las mutaciones pueden afectar drásticamente en la topología y funcionalidad de la RC. En el caso del cromosoma mutado de la expresión 9.8, el MLA codificado por el primer gen estructural para el *Comportamiento* $\mathbf{C}_1$ se extiende en tamaño como consecuencia de modificar la proposición $\mathbf{P}_1$ por la función **D**, la cual incorpora dos nuevos elementos al MLA (por tanto, el MLA se extiende ahora desde la posición 0 hasta la 5). Por otro lado, la mutación en la posición 3 del gen funcional ha cambiado la referencia del parámetro global γ por un valor de 12, lo cual se corresponde con un valor de "60" obtenido del vector D_{pg} (véase la expresión 9.6). Esta mutación en el parámetro γ (que ha aumentado el valor original de 25 a un valor de 60) influye en la eficiencia y en las dinámicas de dispersión de energía de la RC, haciéndola más adaptativa y menos orientada hacia los planes en curso. Como es de esperarse, las mutaciones en las regiones no codificadas de cada gen no desencadenan ningún efecto fenotípico en la estructura de la RC.

Operador de Transposición de Secuencias de Inserción: los elementos transponibles que define la PEG son fragmentos del genoma que pueden ser activados y trasladados a cualquier otro punto del cromosoma. Concretamente, una secuencia de inserción (SI) es un transposón[14] que define una secuencia de elementos dentro del cromosoma, elegida de forma aleatoria, la cual es copiada e insertada en cualquier punto del segmento cabecera de otro gen, excepto en la primera posición (donde siempre debe estar una función comportamiento). Por defecto, la probabilidad de transposición (p_{si}) es definida en 0.1. El operador de transposición elige aleatoriamente el cromosoma, el punto de inicio de la SI en el cromosoma, la longitud de dicha SI, y un punto destino dentro del mismo cromosoma. Este operador de transposición actúa sólo en los genes estructurales, debido a

[14]Un transposón es una secuencia de ADN que puede moverse autosuficientemente a diferentes partes del genoma de una célula, un fenómeno conocido como transposición.

la incompatibilidad entre los alfabetos de los genes estructurales y el gen funcional. Por ejemplo, considérese el cromosoma de la expresión 9.9 que contiene dos genes estructurales:

Gen 1									Gen 2								
0	1	2	3	4	5	6	7	30	0	1	2	3	4	5	6	7	30
C_1	D	P_1	P_2	P_5	P_3	P_4	P_9	...	C_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	...

Expresión Genética 9.9

Supóngase que el transposón "D P_1 P_2" (posiciones sombreadas 1-3 del Gen 1) ha sido escogido aleatoriamente por el algoritmo para ser transpuesto en las posiciones 2-4 del Gen 2, de lo cual se obtiene el cromosoma de la expresión 9.10.

Gen 1									Gen 2								
0	1	2	3	4	5	6	7	30	0	1	2	3	4	5	6	7	30
C_1	D	P_1	P_2	P_5	P_3	P_4	P_9	...	C_2	P_3	D	P_1	P_2	P_4	P_5	P_6	...

Expresión Genética 9.10

Nótese que, por un lado, la secuencia del transposón queda duplicada en el mismo cromosoma aunque, por otro lado, se ha modificado la estructura del *Comportamiento* que definía el Gen 2, lo cual modifica a su vez la topología de la RC. El transposón insertado "desplaza" los elementos del Gen 2 en el punto de inserción, conservándose siempre la regla de mantener los elementos función sólo en la cabecera del gen, esto es, que si al efectuar el desplazamiento algunos elementos función sobrepasan la longitud máxima de la cabecera, estos serán eliminados. A pesar de la duplicación del transposón en los dos genes, los efectos fenotípicos en la RC son totalmente diferentes: el *Comportamiento* $\mathbf{C}_1$ excita la proposición $\mathbf{P}_1$ e inhibe la proposición $\mathbf{P}_2$; mientras que el *Comportamiento* $\mathbf{C}_2$ excita las proposiciones $\mathbf{P}_2$ y $\mathbf{P}_7$, e inhibe la proposición $\mathbf{P}_1$. A pesar de la inserción, obsérvese que la organización estructural del cromosoma se mantiene, por consiguiente todos los cromosomas creados por el operador de transposición produce RCs sintácticamente correctas.

Operador de Recombinación en un punto: este operador genético se encarga por un lado, de escoger aleatoriamente dos cromosomas (teniendo mayor probabilidad de ser escogidos los cromosomas con mayor aptitud); y, por otro, de intercambiar su material genético en un punto determinado (escogido también de forma aleatoria, exceptuando la primera posición del gen), creando así dos nuevos cromosomas descendientes. Supóngase por ejemplo, que el algoritmo ha escogido los dos cromosomas padres de la expresión 9.11.

Considérese ahora, que se ha escogido aleatoriamente la posición 3 del Gen 1 para realizar la recombinación entre los dos cromosomas. Intercambiando el material genético

Cromosoma 1:

Gen 1										Gen 2									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
C_1	D	P_1	D	P_2	P_3	P_4	P_9	P_7	P_6	C_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_8	P_2

Cromosoma 2:

Gen 1										Gen 2									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
C_1	D	D	P_3	D	P_1	P_2	P_7	P_8	P_2	C_2	P_5	D	P_3	D	P_1	P_2	P_4	P_6	P_7

Expresión Genética 9.11

de los dos cromosomas padres, se obtienen dos nuevos cromosomas hijos, los cuales se presentan en la expresión 9.12.

Cromosoma 3:

Gen 1										Gen 2									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
C_1	D	P_1	P_3	D	P_1	P_2	P_7	P_8	P_2	C_2	P_5	D	P_3	D	P_1	P_2	P_4	P_6	P_7

Cromosoma 4:

Gen 1										Gen 2									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
C_1	D	D	D	P_2	P_3	P_4	P_9	P_7	P_6	C_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_8	P_2

Expresión Genética 9.12

Es importante destacar que con esta clase de recombinación, la mayoría de las veces, la descendencia creada exhibe diferentes características (topológicas y funcionales) que las de sus cromosomas padres. El operador de recombinación es una fuente de variación genética muy importante, siendo, después del operador de mutación, uno de los operadores más empleados en la programación genética.

Intuitivamente, si el punto de recombinación cae sobre la región de un gen funcional, el intercambio de información genética producirá descendientes igualmente válidos debido a que, como ya se mencionó anteriormente, los cromosomas y los genes tienen todos longitudes invariables, impidiendo que se mezcle, por ejemplo, información del gen funcional de un cromosoma con la información de los genes estructurales de otro cromosoma. Por defecto, la probabilidad de que un par de cromosomas padres se recombinen en un punto es $p_{rup} = 0.3$. Una vez producida la recombinación, el nivel de aptitud para cada cromosoma hijo se calcula como un promedio de las aptitudes de los cromosomas padres proporcional al punto de cruce. Por ejemplo, supóngase 2 cromosomas (a y b), cada uno con un nivel de aptitud de 215 y 78 respectivamente. Si la longitud total de

ambos cromosomas es de 30 elementos y el punto de cruce seleccionado aleatoriamente es 17, las aptitudes para los cromosomas hijos (c y d) serían:

$$c.aptitud = (a.aptitud \times (17/30)) + (b.aptitud \times (13/30)) = 155.6333 \quad (9.17)$$
$$d.aptitud = (a.aptitud \times (13/30)) + (b.aptitud \times (17/30)) = 137,3666 \quad (9.18)$$

Este cálculo proporcional de los niveles de aptitud permite que los cromosomas hijos hereden no sólo la información genética de sus padres, sino también la eficiencia relativa que han tenido estos durante los períodos de aprendizaje previos a la reproducción.

Operador de Recombinación en dos puntos: este operador actúa de manera similar al anteriormente descrito, con la diferencia de que son definidos aleatoriamente dos puntos de intercambio de información entre los cromosomas padres. La probabilidad de que dos cromosomas padres se recombinen en dos puntos se define por defecto como $p_{rdp} = 0.3$, y el nivel de aptitud para cada cromosoma hijo generado corresponde a un promedio de las aptitudes de los cromosomas padres. A fin de ilustrar el funcionamiento de este operador, considérese los cromosomas de la expresión 9.13. La traducción fenotípica de estos dos cromosomas produce los Sub-AEXs y las RCs respectivas que se presentan en la figura 9.9. Obsérvese que en este caso, los cromosomas padres generan dos *Comportamientos* que no se conectan entre sí, evitando que se disperse energía (excitativa o inhibitoria) a través de la red. Topológicamente, los dos cromosomas producen RCs con estructuras "no consistentes".

Cromosoma 1 (Padre):

Gen 1											Gen 2										
0	1	2	3	4	5	6	7	8	9	10	0	1	2	3	4	5	6	7	8	9	10
C_1	D	T	P_5	P_1	D	P_4	P_2	P_3	P_2	P_3	C_2	D	D	P_6	P_7	P_9	M_1	P_9	P_9	P_7	P_5

Cromosoma 2 (Padre):

Gen 1											Gen 2										
0	1	2	3	4	5	6	7	8	9	10	0	1	2	3	4	5	6	7	8	9	10
C_1	P_2	P_8	P_4	P_6	D	P_2	P_3	P_4	P_2	P_1	C_2	D	D	P_6	P_3	D	P_1	P_3	P_5	P_9	P_2

Expresión Genética 9.13

Supóngase ahora que el operador genético ha generado aleatoriamente dos puntos de recombinación: la posición 4 del primer gen y la posición 5 del segundo gen. Los cromosomas descendientes de esta recombinación genética se presentan en la expresión 9.14.

La figura 9.10 presenta la traducción fenotípica para los cromosomas descendientes. Nótese que de los dos cromosomas producidos por el operador de recombinación, uno de ellos (el cromosoma 3) ha generado una estructura consistente para una RC, donde es posible inferir que el *Comportamiento* $\mathbf{C}_1$ dispersa energía hacia el *Comportamiento* $\mathbf{C}_2$

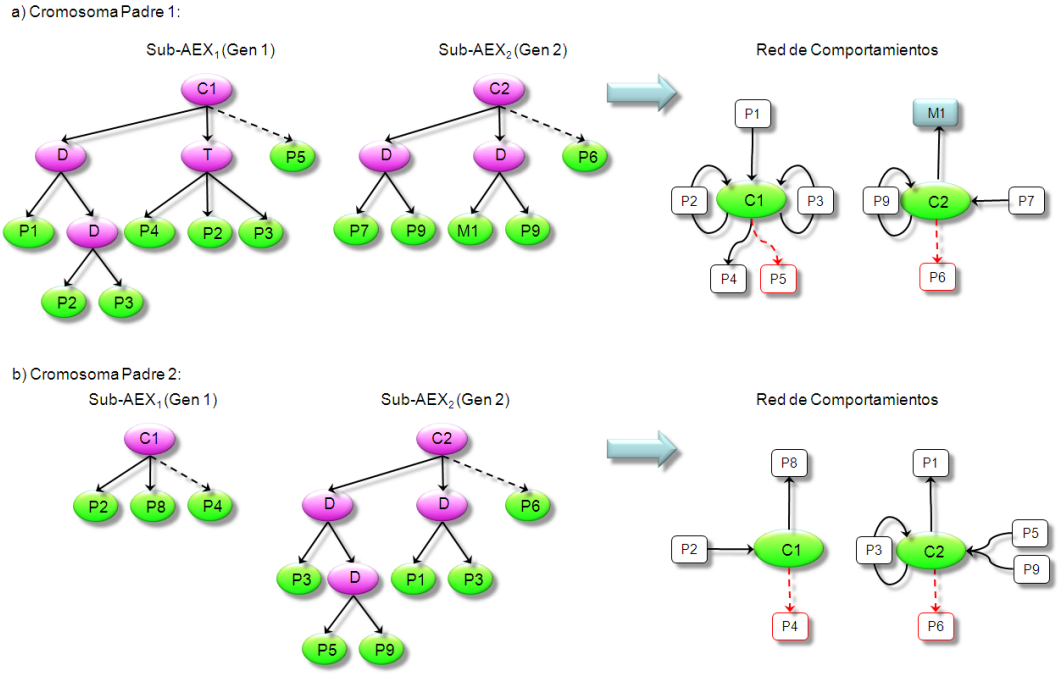


Figura 9.9: Representación Fenotípica de la Expresión 9.13.

Cromosoma 3 (Hijo):

Gen 1											Gen 2										
0	1	2	3	4	5	6	7	8	9	10	0	1	2	3	4	5	6	7	8	9	10
C_1	D	T	P_5	P_6	D	P_2	P_3	P_4	P_2	P_1	C_2	D	D	P_6	P_3	D	M_1	P_9	P_9	P_7	P_5

Cromosoma 4 (Hijo):

Gen 1											Gen 2										
0	1	2	3	4	5	6	7	8	9	10	0	1	2	3	4	5	6	7	8	9	10
C_1	P_2	P_8	P_4	P_1	D	P_4	P_2	P_3	P_2	P_3	C_2	D	D	P_6	P_7	P_9	P_1	P_3	P_5	P_9	P_2

Expresión Genética 9.14

(a través de $\mathbf{P}_3$), el cual, una vez ha obtenido el foco de control, inhibe el *Comportamiento* $\mathbf{C}_1$ (a través de $\mathbf{P}_6$) y promete alcanzar la meta $\mathbf{M}_1$ como resultado de su actuación.

Como se puede observar, el operador de recombinación ha propiciado la *"emergencia"* de una nueva estructura cognitiva más compleja y consistente a partir de la combinación de sub-estructuras básicas y arbitrarias. No obstante, la aplicación de este operador (así como la de los demás operadores genéticos) no siempre genera estructuras consistentes, complejas y robustas, como es el caso del cromosoma 4. Se requiere entonces por un lado, de la interacción mutua de los diferentes operadores genéticos, con el fin de equilibrar los procesos de explotación vs. exploración del espacio de búsqueda; y, por otro, de la orientación adaptativa que ejerce el mecanismo de selección en cada ciclo evolutivo,

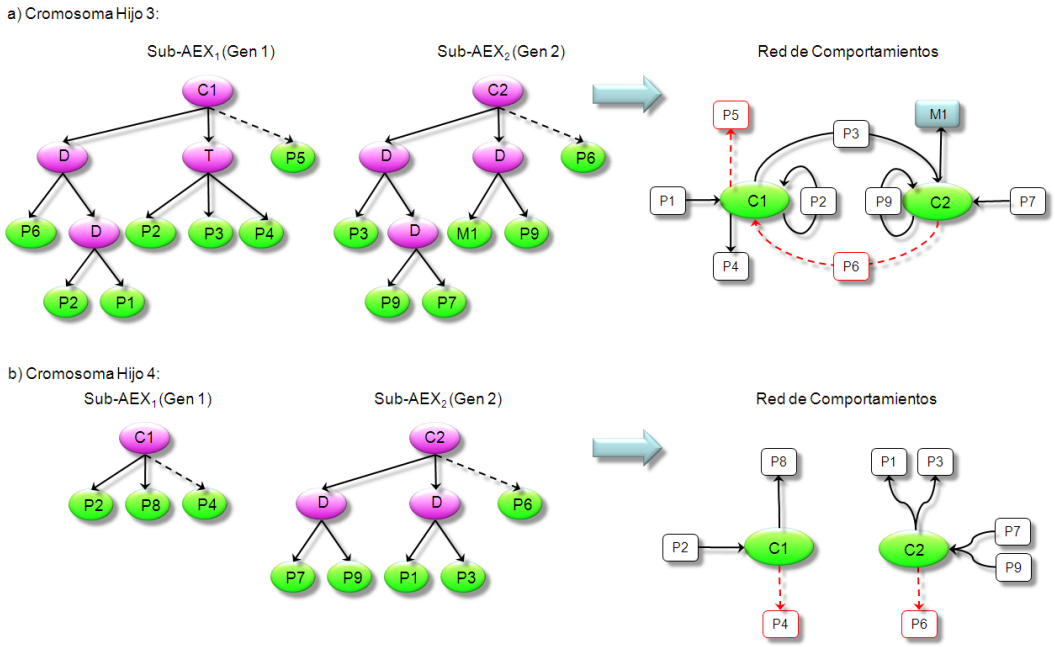


Figura 9.10: Representación Fenotípica de la Expresión 9.14.

a través de la evaluación de la aptitud de cada cromosoma de la población (tema que se discutirá en el apartado 9.3.2).

Por último, cabe notar que los operadores genéticos de recombinación materializan la metáfora sobre el "*sexo*" (discutida en la sección 4.4.3.2) como acelerador de los procesos darwinianos. La combinación del material genético de aquellos cromosomas más aptos acelera el proceso evolutivo, provocando que el algoritmo converja con más prontitud en óptimos globales y evitando, por supuesto, que caiga en óptimos locales. En estas líneas, un óptimo local hace referencia a la estructura consistente que expresa un gen para un *Comportamiento* específico, mientras que un óptimo global se refiere a la estructura completa de una RC definida por su correspondiente cromosoma, donde la interacción de los múltiples genes refleja la eficiencia de la red.

Algoritmo para la Máquina Darwiniana (RCs): en el cuadro 9.2 se presenta el algoritmo empleado por la Máquina Darwiniana basada en PEG. Con respecto a la función $generarPoblacionInicialCromosomas$ en el paso 6 del algoritmo 9.2, se deben tomar en cuenta las siguientes consideraciones:

- La generación de una población inicial de cromosomas responde a la necesidad de crear un repertorio pseudo-aleatorio de representaciones genéticas de diversas RCs sobre las cuales se inicia el proceso evolutivo. Con pseudo-aleatorio se quiere aludir a la idea de que el repertorio inicial de cromosomas se genera mediante dos procesos: (1) el diseñador de la concreción contextual de la arquitectura define por cada meta global, al menos un cromosoma que representa una RC simplificada, esto es, una RC que define las precondiciones para cada *Comportamiento* de la RC

Método: EjecutarPEG

```
1.  Variables
2.     pob            ↓↑ población de RCs.
3.     subpob1        ↓↑ sub-población temporal de RCs.
4.     subpob2        ↓↑ sub-población temporal de RCs.
5.  Inicio Procedimiento
6.     pob ← generarPoblacionInicialCromosomas()
7.     Mientras que se cumpla la condición de activación de la
       PEG hacer:
8.        Para cada cromosoma creado hacer:
9.           aex ← generarAEX(pob.cromosoma)
10.          rc ← generarRC(aex)
11.          pob.cromosoma.aptitud ← evaluarRC(rc)
12.       Fin Para
13.       subpob1 ← seleccionarMejoresCromosomas(pob)
14.       pob ← eliminarPeoresCromosomas(pob)
15.       Mientras que no se supere el tamaño máximo de la
          población hacer:
16.          subpob2 ← replicarCromosomas(subpob1)
17.          subpob2 ← recombinarUnPunto(subpob2.crom, subpob2.crom)
18.          subpob2 ← recombinarDosPuntos(subpob2.crom,
             subpob2.crom)
19.          subpob2 ← transponerSecuenciasInsercion(subpob2.crom)
20.          subpob2 ← mutarCromosoma(subpob2.crom)
21.       Fin Mientras que
22.       pob ← agregarNuevosCromosomas(subpob2)
23.    Fin Mientras que
24. Fin Procedimiento
```

Algoritmo 9.2: Algoritmo para la MD basada en PEG.

aunque no necesariamente los efectos deseados y no deseados del mismo; y (2) un proceso que genera aleatoriamente otro conjunto de cromosomas, respetando las restricciones relativas a cada dominio de cada gen (e.g., en un gen estructural, la cabecera puede incluir funciones de conectividad y terminales, mientras que en la terminación sólo pueden generarse terminales; y en los genes funcionales sólo pueden incluirse valores del vector D_{pg}).

Las razones por las cuales se sugiere que el mecanismo de generación inicial de cromosomas sea pseudo-aleatorio y no totalmente aleatorio, se debe por un lado, a que si el algoritmo parte de una base predefinida por el diseñador (i.e., estructuras cognitivas innatas), el proceso evolutivo convergerá más prontamente; y, por otro, a que una generación de cromosomas totalmente aleatoria requeriría una función de aptitud lo suficientemente sofisticada [15] como para evaluar la creación de nuevos

[15] En la siguiente sección, se propone una función de aptitud que dirige el proceso evolutivo, evaluando la creación estructural de cada *Comportamiento* de la red. Sin embargo, esta función de aptitud no está diseñada para soportar la creación de nuevos *Comportamientos* que no se hayan considerado previamente como parte del repertorio de estructuras cognitivas que posee el agente. Este punto, aunque resulte muy tentador de tratar en esta tesis, se escapa del alcance de la misma debido a que requiere de un análisis complejo adicional, no obstante, sí que se considera como parte del trabajo futuro.

Comportamientos en la red "partiendo desde cero". Esto sin embargo no implica que deba ser predefinido siempre un conjunto inicial de cromosomas, simplemente se resalta la conveniencia de acelerar el proceso evolutivo. Por ejemplo, según lo propuesto anteriormente, sería posible definir por cada meta global del agente, un conjunto de *Comportamientos* cuya interacción podría alcanzar dicha meta. Entonces, el diseñador podría, simplemente, establecer un número determinado de genes, uno por cada *Comportamiento*, y definir el conjunto específico de precondiciones que deben ser ciertas para producir la activación de cada uno de estos, aunque por otro lado, no se definan los efectos deseados y no deseados producidos por los mismos, dejando en manos del proceso darwiniano el descubrimiento de dichas conexiones (excitativas e inhibitorias) que describirían las dinámicas de la RC.

En resumen, el proceso ontogenético tiene como objetivo evolucionar el diseño estructural y funcional de la RC, refinando por un lado, las conexiones entre *Comportamientos* las cuales determinan las dinámicas de cooperación y competencia entre los mismos; y, por otro, los parámetros globales de la red que determinan el funcionamiento del mecanismo de dispersión de energía de activación, y como consecuencia, la eficiencia y orientabilidad de la RC.

- Lo que se ha descrito anteriormente aplica cuando el sistema (multi-agente) no dispone de una base filogenética, es decir, aplica para la primera generación de agentes. Como se verá más adelante (sección 10.4), el sistema produce múltiples generaciones de agentes, donde cada generación establece una base filogenética de estructuras cognitivas, esto es, una base de conocimiento innata sobre la cual se desarrollan procesos epigenéticos y ontogenéticos subsecuentes. En este caso, el proceso ontogenético toma como base las estructuras cognitivas (RCs) generadas por el proceso filogenético en la generación anterior, y las refina en la actual generación. Debido a que el proceso de generación de la población de cromosomas sigue siendo pseudo-aleatorio, adicionalmente a las RCs generadas por el proceso filogenético (estructuras innatas), se generan aleatoriamente un conjunto de cromosomas que también participaran en el proceso evolutivo, esto con el fin de agregar diversidad en la población de cromosomas y que, como consecuencia, se exploren nuevos espacios de búsqueda que permitan refinar (a través de la replicación, recombinación y variación) las estructuras innatas.

Con respecto a la función $eliminarPeoresCromosomas$ del paso 14 del algoritmo, ésta se encarga de reducir la población de cromosomas a un porcentaje determinado (por defecto se establece un 40 %) de la capacidad máxima definida para la población, eliminando aquellos cromosomas restantes que tienen la aptitud más baja. La idea principal de este método de recorte consiste por un lado, en mantener una población de cromosomas de tamaño constante que evite sobredimensionamientos y sobrecargas en el proceso; y, por otro, en proporcionar el espacio suficiente (el 60 % restante del tamaño de la población) donde se crearán, en la siguiente generación, los nuevos cromosomas fruto de la ejecución de los operadores genéticos. Intuitivamente, la región de la población que se mantiene en el 40 % son los cromosomas que han demostrado ser los más aptos a lo largo de las múltiples generaciones producidas por el proceso ontogenético.

9.3.2. Selección de la Máquina de Atención Apropiada

En esta sección se expone el mecanismo por el cual el módulo de Orquestación de la Consciencia elige la MAT que se hará con el control consciente del agente y selecciona aquellas MATs que actuarán de forma subconsciente (automatismos). Este mecanismo se basa pues en dos subprocesos los cuales se esbozan a continuación: la evaluación de la función de aptitud y la selección por resonancia de la mejor aptitud.

9.3.2.1. Evaluación de la Función de Aptitud

Al igual que los algoritmos genéticos descritos en las secciones 7.4.3.1 y 8.6.3.3, en la PEG, cada cromosoma generado por el proceso ontogenético debe ser evaluado en cada generación[16] con el fin de determinar su "aptitud". Por tanto, la aptitud refleja qué tan apta es cada RC para atender la situación actual percibida por el agente. De esta manera, la función que evalúa la aptitud de cada cromosoma ejerce cierta presión que dirige el proceso evolutivo hacia la construcción de RCs estructural y funcionalmente eficientes. Así pues, la función de aptitud es un tipo de función objetivo que evalúa la optimalidad de cada solución (RC) codificada por un cromosoma. En este orden de ideas, la función de aptitud evalúa dos aspectos para cada cromosoma: el *Aspecto Estructural* y el *Aspecto Funcional*.

Aspecto Estructural: con el fin de promover la evolución de diseños estructurales correctos y eficientes, la función de aptitud evalúa, por cada cromosoma, los siguientes aspectos:

A: este término de la función determina si al menos un *Comportamiento* de toda la red promete alcanzar una meta global. El término A genera una retroalimentación positiva si se cumple la condición, o genera una retroalimentación negativa de lo contrario. Se tiene entonces que:

$$A = \begin{cases} a1, & \text{si } \exists\ comp\ \in RC \mid ED_{comp}\ \cap\ G(t)\ \neq\ \emptyset \\ a2, & \text{de lo contrario} \end{cases} \tag{9.19}$$

donde: $comp$ es algún *Comportamiento* codificado por un gen del cromosoma. ED_{comp} es la lista de efectos deseados del *Comportamiento*. $G(t)$ es el conjunto de metas globales del agente. $a1$ es una constante de retroalimentación positiva (por defecto +100). $a2$ es una constante de retroalimentación negativa (por defecto -100).

B: este término valida la existencia de enlaces entre todos los *Comportamientos* de la red, es decir, verifica que no existan *Comportamientos* "desconectados". Por cada

[16] Considérese una generación como un ciclo evolutivo completo, es decir, un paso del algoritmo de la MD basada en PEG (véase el algoritmo 9.2).

Comportamiento que defina un enlace sucesor o predecesor hacia otro *Comportamiento* diferente de sí mismo, el término B genera una retroalimentación positiva, mientras que por cada *Comportamiento* que no se encuentre conectado a ninguno, el término B genera una retroalimentación negativa. Se tiene entonces que:

$$B = n_c \times b1 + n_d \times b2 \tag{9.20}$$

donde: n_c es el número de *Comportamientos* conectados con otros diferentes de sí mismo. $b1$ es una constante de retroalimentación positiva (por defecto +10). n_d es el número de *Comportamientos* desconectados. $b2$ es una constante de retroalimentación negativa (por defecto -20).

C: este término valida la no existencia de ciclos de auto-referencia o callejones sin salida. Se verifica entonces que exista al menos un camino posible entre cada uno de los *Comportamientos* de la red y la meta global. Esto es, por cada *Comportamiento* de la red se recorren los enlaces sucesores hasta determinar si se llega a la meta final, caso en el que se genera una retroalimentación positiva, o de lo contrario se genera una retroalimentación negativa. Nótese que este término depende del término A, con lo cual, si no hay una meta global asociada a la RC, se genera una retroalimentación negativa total. Se tiene entonces que:

$$C = \begin{cases} n_{cc} \times c1 + n_{sc} \times c2, & \text{si la RC tiene asociada una meta global} \\ c3, & \text{de lo contrario} \end{cases} \tag{9.21}$$

donde: n_{cc} es el número de *Comportamientos* que definen al menos un camino hasta la consecución de la meta. $c1$ es una constante de retroalimentación positiva (por defecto +10). n_{sc} es el número de *Comportamientos* que no definen un camino hasta la meta global. $c2$ es una constante de retroalimentación negativa (por defecto -20). $c3$ es una constante de retroalimentación negativa total (por defecto -50).

D: este término valida, por cada gen, que las proposiciones del *Comportamiento* (i.e., precondiciones, efectos deseados y efectos no deseados) "no" sean ambiguas, esto es, que no exista más de un valor asociado a la proposición observada. Por ejemplo, las proposiciones $\langle$`Con-Hambre`$\rangle$ y $\langle$`Sin-Hambre`$\rangle$ son valores opuestos para el motivador $\langle$`Hambre`$\rangle$, y no pueden co-existir como precondiciones simultáneas en un *Comportamiento*. Nótese que la ambigüedad se determina sólo dentro de cada subconjunto de proposiciones del *Comportamiento*, por ejemplo, la proposición $\langle$`Con-Hambre`$\rangle$ puede formar parte de las precondiciones y la proposición $\langle$`Sin-Hambre`$\rangle$ puede formar parte de los efectos deseados, pero debido a que no co-existen dentro del mismo subconjunto de proposiciones no se considera ambigüedad. Se tiene entonces que:

$$D = \sum_{i=0}^{k} n_{pn} \times d1 + n_{pa} \times d2 \tag{9.22}$$

donde: k es el número total de *Comportamientos* (genes) de la RC. n_{pn} es el número de proposiciones (i.e., precondiciones, efectos deseados, y efectos no deseados) que no tienen valores ambiguos. $d1$ es una constante de retroalimentación positiva (por defecto +10). n_{pa} es el número de proposiciones (i.e., precondiciones, efectos deseados, y efectos no deseados) con valores ambiguos. $d2$ es una constante de retroalimentación negativa (por defecto -20).

E: este término valida que cada proposición que forma parte de los "efectos deseados" de cada *Comportamiento* de la red sea estructuralmente válida. Esto es, el término E genera una retroalimentación positiva por cada proposición que forma parte de los efectos deseados de un *Comportamiento*, si se cumple que: (1) la proposición no se encuentra a su vez en la lista de efectos no deseados del mismo *Comportamiento*, con lo cual se garantiza la congruencia: una proposición no puede ser efecto deseado y efecto no deseado a la vez; y (2) el efecto deseado se corresponde al menos con una precondición de otro *Comportamiento* diferente en la red, con lo cual se asegura que haya un flujo de dispersión de energía hacia adelante. De lo contrario, el término E genera una retroalimentación negativa.

$$E = \sum_{i=0}^{k} n_{edv} \times e1 + n_{ednv} \times e2 \tag{9.23}$$

donde: k es el número total de *Comportamientos* (genes) de la RC. n_{edv} es el número de efectos deseados válidos (es decir, que cumplen con las dos condiciones anteriores). $e1$ es una constante de retroalimentación positiva (por defecto +10). n_{ednv} es el número de efectos deseados que no cumplen con alguna de las dos condiciones para ser válido. $e2$ es una constante de retroalimentación negativa (por defecto -20).

F: este término valida que las proposiciones que forman parte de los "efectos no deseados" de cada *Comportamiento* de la red sean estructuralmente válidas. Esto es, el término F genera una retroalimentación positiva por cada proposición que forma parte de los efectos no deseados de un *Comportamiento*, si se cumple que: (1) la proposición no se encuentra a su vez en la lista de precondiciones del mismo *Comportamiento*, a menos que otro *Comportamiento* también tenga la proposición como una de sus precondiciones, con lo cual se evita que el efecto no deseado produzca una auto-inhibición del *Comportamiento*[17]; (2) la proposición no se encuentra a su vez en la lista de efectos deseados del mismo *Comportamiento*, con lo cual se garantiza la congruencia: una proposición no puede ser efecto no deseado y efecto deseado a la vez; y (3) el efecto no deseado se corresponde al menos con una precondición de otro *Comportamiento* diferente en la red, con lo cual se asegura que haya un flujo de dispersión de energía inhibitoria que

[17]Como consecuencia, la auto-inhibición sólo es posible si simultáneamente se produce una inhibición de otro *Comportamiento*, de lo contrario no se justifica que una proposición forme parte de la lista de precondiciones y de la lista de efectos no deseados del mismo *Comportamiento*.

evita los conflictos entre *Comportamientos*. De lo contrario, el término F genera una retroalimentación negativa.

$$F = \sum_{i=0}^{k} n_{env} \times f1 + n_{ennv} \times f2 \tag{9.24}$$

donde: k es el número total de *Comportamientos* (genes) de la RC. n_{env} es el número de efectos no deseados válidos (es decir, que cumplen con las tres condiciones anteriores). $f1$ es una constante de retroalimentación positiva (por defecto +10). n_{ennv} es el número de efectos no deseados que no cumplen con alguna de las tres condiciones para ser válido. $f2$ es una constante de retroalimentación negativa (por defecto -20).

Finalmente, la función de aptitud que evalúa el aspecto estructural de cada RC se presenta a continuación:

$$FAAE_i = (k_1 \times A_i) + (k_2 \times B_i) + (k_3 \times C_i) + (k_4 \times D_i) + (k_5 \times E_i) + (k_6 \times F_i) \tag{9.25}$$

donde: i es una RC específica. las constantes $k_1, k_2, \ldots k_6$ son pesos asignados a cada término de la función, con lo cual es posible ejercer mayor presión sobre alguno de estos. Por defecto todos los pesos son iguales a 1.

Aspecto Funcional: con el fin de dirigir la evolución de cada RC hacia un diseño funcional refinado, la función de aptitud evalúa, por cada cromosoma, la eficiencia de la red en términos de la configuración de sus parámetros globales (codificados en el gen funcional).

Según el genetista Richard Dawkins (Dawkins, 1992) el proceso adaptativo de los organismos es moldeado por distintas *"presiones selectivas"*, determinadas por el ambiente, las cuales buscan establecer un punto de equilibrio o de adaptación, a partir del cual continúa el proceso evolutivo. De manera similar, la eficiencia funcional de una RC está determinada por diversas presiones selectivas ejercidas por el propio ambiente en el cual se encuentra inmerso el agente. Por ejemplo, dado un ambiente altamente cambiante y dinámico, existe cierta presión selectiva sobre el agente que lo induce a re-configurar sus RCs con el fin de hacerlas más "reactivas" y "adaptativas" ante los cambios del ambiente. Concretamente, existe una presión selectiva por cada uno de los parámetros globales de una RC. Cada presión selectiva ejerce una fuerza hacia sí misma, y el resultado de esas múltiples tensiones es lo que sitúa a cada cromosoma en un punto del espacio de búsqueda. Idealmente, el objetivo es situar la mayor cantidad de cromosomas en un área cercana al óptimo global.

Para determinar el nivel de adaptación gradual que adquiere cada solución (cromosoma) a lo largo del tiempo, se deben establecer dos puntos de comparación por cada parámetro global analizado: (1) la configuración paramétrica de la RC para un estado del sistema anterior (e_{ant}), y (2) la configuración paramétrica de la RC para un estado del sistema actual (e_{act}) donde se espera que la eficiencia de la red haya incrementado[18].

[18] Adviértase que entre los estados e_{ant} y e_{act} del sistema pueden transcurrir uno o varios ciclos cognitivos, dependiendo en gran medida de la frecuencia con la que será activado el proceso ontogenético en una concreción contextual determinada de la arquitectura.

Concretamente, la eficiencia de la red se mide en términos de las demandas circunstanciales del ambiente y los intereses propios del agente vs. el ajuste de los parámetros de la RC frente a tales demandas e intereses. Igual que para el aspecto estructural, se definen múltiples términos que determinan la función de aptitud del aspecto funcional de una RC:

G: este termino define el ajuste del parámetro global γ (la cantidad de energía de activación que una meta inyecta a la red). Para efectuar el ajuste apropiado de este parámetro (a través de la presión selectiva) se definen entonces dos variaciones absolutas: Δ_{frecM} y Δ_{γ}, las cuales determinan respectivamente, si el ambiente requiere que el agente oriente su actuación hacia la consecución de metas y planes, y si éste realmente lo hace o no. Δ_{frecM} es la variación absoluta de la frecuencia de activación (o re-activación) de las metas y sub-metas del agente. Por ejemplo, supóngase que en el estado e_{ant} del sistema se registraron 3 activaciones o re-activaciones de metas ($frecM_{e_{ant}}$) efectuadas por diferentes motivadores durante un número determinado de ciclos cognitivos (e.g., 10 ciclos). Ahora, en un estado e_{act} del sistema se registran 5 activaciones de metas ($frecM_{e_{act}}$), con lo cual la variación absoluta de la frecuencia es: $\Delta_{frecM} = frecM_{e_{act}} - frecM_{e_{ant}} = 5 - 3 = 2$. Supóngase ahora que el parámetro γ tenía un valor de $\gamma = 17$ en el estado e_{ant}, y que en el estado e_{act} tiene un valor de $\gamma = 45$, con lo cual la variación absoluta de γ es: $\Delta_{\gamma} = \gamma_{e_{act}} - \gamma_{e_{ant}} = 45 - 17 = 28$.

Con el fin de medir las variaciones de modo más preciso, eliminando las diferencias de escala para que puedan ser equiparables entre sí, es necesario expresar las variaciones en términos relativos. Por tanto, la variación relativa de cada una de estas variaciones absolutas se mide a través de una tasa de variación respectiva:

$$T_{frecM} = \frac{\Delta_{frecM}}{frecM_{e_{ant}}} = \frac{5-3}{3} = 0.6666 \tag{9.26}$$

$$T_{\gamma} = \frac{\Delta_{\gamma}}{\gamma_{e_{ant}}} = \frac{45-17}{17} = 1.64705 \tag{9.27}$$

Empleando estas tasas de variación como punto de comparación entre las dos variables, se tiene que la retroalimentación producida por el término G corresponde a:

$$G = \begin{cases} 100 - g1, & \text{si } (T_{frecM} > 0 \wedge T_{\gamma} > 0) \vee (T_{frecM} < 0 \wedge T_{\gamma} < 0) \\ g2, & \text{de lo contrario} \end{cases} \tag{9.28}$$

donde: $g1$ corresponde a la diferencia absoluta entre las tasas de variación ($g1 = \mid T_{frecM} - T_{\gamma} \mid$). $g2$ corresponde a una constante de retroalimentación negativa (por defecto -100).

Obsérvese que en el caso del ejemplo, debido a que la frecuencia de activación de metas incrementó del estado e_{ant} al estado e_{act}, y debido a que la RC ha aumentado como

consecuencia el valor del parámetro γ, cumpliéndose la condición $T_{frecM} > 0 \wedge T_\gamma > 0$; el término G produce entonces una retroalimentación positiva: $G = 100 - \mid T_{frecM} - T_\gamma \mid = 100 - (|0.66666 - 1.64705|) = 100 - 0.9804 \cong 99.02$.

Nótese además, que en el peor de los casos, cuando $g1$ tiene un valor muy alto, la retroalimentación que produce el término G sigue siendo positiva (aunque muy pequeña), mientras se cumplan las dos condiciones de la ecuación 9.28. Por ejemplo, supóngase que el parámetro γ cambia drásticamente desde un valor de 1 en e_{ant} hasta un valor de 100 en e_{act} (valor máximo que puede alcanzar un parámetro global); se tiene entonces que su tasa de variación es igual a $T_\gamma = (100-1)/1 = 99$, y que la retroalimentación es igual a $G = 100-(|0.6666-99|) \cong 1.6666$. En el caso ideal, las tasas de variación deberían ser iguales ($T_{frecM} = T_\gamma$), con lo que la retroalimentación del término G debería ser igual a 100. Es importante resaltar cómo el término $g1$ determina una retroalimentación positiva que varía en función de la distancia entre las dos tasas de variación (T_{frecM} y T_γ).

Intuitivamente, la presión selectiva que ejerce el término G sobre la función de aptitud promueve el aumento del valor del parámetro γ si la frecuencia de activación de metas incrementa, o promueve su disminución si dicha frecuencia disminuye proporcionalmente.

H: este término determina el ajuste del parámetro global ϕ (la cantidad de energía de activación que una proposición verdadera inyecta a la red). De manera similar al término anterior, H se define en función de dos tasas de variación: T_{frecC} y T_ϕ. Por su parte, T_{frecC} es una tasa de variación que determina el cambio de frecuencia con la que son percibidos los cambios ambientales (es decir, los cambios físicos en los objetos observados, los estímulos novedosos, y los elementos relevantes)[19]. Por otro lado, T_ϕ determina una tasa de variación en el valor del parámetro global ϕ entre dos estados del sistema. La presión selectiva que ejerce el término H estaría determinada entonces por:

$$H = \begin{cases} 100 - h1, & \text{si } (T_{frecC} > 0 \wedge T_\phi > 0) \vee (T_{frecC} < 0 \wedge T_\phi < 0) \\ h2, & \text{de lo contrario} \end{cases} \tag{9.29}$$

donde: $h1$ corresponde a la diferencia absoluta entre las tasas de variación ($h1 = \mid T_{frecC} - T_\phi \mid$). $h2$ corresponde a una constante de retroalimentación negativa (por defecto -100). $T_{frecC} = \frac{\Delta_{frecC}}{frecC_{e_{ant}}}$. $T_\phi = \frac{\Delta_\phi}{\phi_{e_{ant}}}$. Nótese que las tasas de variación del término H se tratan de modo similar al término G.

Por ejemplo, supóngase que para el estado e_{ant} se determina una frecuencia de 6 cambios ambientales detectados durante un intervalo específico de ciclos cognitivos (e.g., durante 10 ciclos); ahora, en el estado e_{act} se determina una frecuencia de 3 cambios ambientales durante el siguiente intervalo de ciclos cognitivos; luego $T_{frecC} = (3-6)/6 = -0.5$. Supóngase ahora que en el estado e_{ant} el parámetro global tiene un valor de $\phi = 25$, mientras que en el estado e_{act} el mismo parámetro sufre una

[19]Recuérdese que estos son los tres tipos de factores descritos en la sección 9.2.1 que determinan los períodos de oscilación de la atención por cambios en la situación percibida por el agente.

mutación y cambia al valor $\phi = 63$. En este caso, la tasa de variación respectiva es: $T_\phi = (63 - 25)/25 = 1.52$. Obsérvese que en este ejemplo, la variación del parámetro ϕ no se ha ajustado correctamente a la disminución de la frecuencia de cambios ambientales detectados ($T_{frecC} < 0$ mientras que $T_\phi > 0$), produciendo una retroalimentación negativa. Intuitivamente, la presión selectiva que ejerce este término sobre la función de aptitud favorece el aumento del valor del parámetro ϕ cuando la frecuencia de cambios ambientales percibidos por el agente aumenta. Así mismo, favorece la disminución del valor del parámetro ϕ cuando la frecuencia asociada ($frecC$) también disminuye.

I: este término determina el ajuste del parámetro global π (la cantidad de energía de activación promedio que circula dentro de la red). Debido a que este parámetro global afecta directamente las características de *adaptabilidad vs. predisposición hacia los planes en curso* de la RC, se tiene en cuenta la frecuencia de activación del conjunto de sub-metas asociadas a una meta más general. Esto es, dada una meta activa en la estructura de metas, se determina el número de sub-metas asociadas que son activadas durante un intervalo específico de ciclo cognitivos. De manera similar a los términos anteriores, se especifica una tasa de variación que determina el cambio de frecuencia de activación de sub-metas T_{frecSM} y una tasa de variación que indica el cambio del parámetro global π (T_π) entre dos estados del sistema (e_{ant} y e_{act}).

$$I = \begin{cases} 100 - i1, & \text{si } (T_{frecSM} > 0 \wedge T_\pi > 0) \vee (T_{frecSM} < 0 \wedge T_\pi < 0) \\ i2, & \text{de lo contrario} \end{cases} \tag{9.30}$$

donde: $i1$ corresponde a la diferencia absoluta entre las tasas de variación ($i1 = \mid T_{frecSM} - T_\pi \mid$). $i2$ corresponde a una constante de retroalimentación negativa (por defecto -100). $T_{frecSM} = \frac{\Delta_{frecSM}}{frecSM_{e_{ant}}}$. $T_\pi = \frac{\Delta_\pi}{\pi_{e_{ant}}}$.

Intuitivamente, si el estado del ambiente influye sobre el agente provocando que su actuación esté más orientada hacia la consecución de un conjunto de metas jerárquicas (es decir, un conjunto de sub-metas especificas que llevan al logro de una meta más general) entonces el valor del parámetro π debe aumentar. Contrariamente, si se requiere que la actuación del agente sea más adaptativa, entonces el valor del parámetro π debe disminuir.

J: este término determina el ajuste del parámetro global δ (la cantidad de energía de activación que una meta protegida libera de la red). La variable que interesa controlar con este término es el cambio de δ con respecto a la necesidad de proteger las metas alcanzadas vs. los ciclos de auto-referencia. Por ejemplo, supóngase que el agente debe lograr las metas m_1, m_2, m_3. En un instante determinado, el agente logra la meta m_3 impidiendo alcanzar la meta m_2 desde el estado actual, por lo cual se requiere primero "deshacer" la meta alcanzada m_3 para luego si alcanzar m_1, m_2, m_3 en ese orden. En este caso, el valor del parámetro δ debe ser lo suficientemente pequeño como para permitir deshacer la meta alcanzada. En un caso contrario, donde el orden de consecución de las metas no interfiera en el proceso, un valor de δ alto puede ser el más apropiado. Por

tanto, el indicador que ejerce la presión selectiva adecuada es el número de ciclos de auto-referencia[20] que ocurren durante la actuación del agente. Formalmente se tiene que:

$$J = \begin{cases} 100 - j1, & \text{si } (T_{auto} > 0 \wedge T_\delta < 0) \vee (T_{auto} < 0 \wedge T_\delta > 0) \\ j2, & \text{de lo contrario} \end{cases} \tag{9.31}$$

donde: $j1$ corresponde a la diferencia absoluta entre las tasas de variación ($j1 = \mid T_{auto} - T_\delta \mid$). $j2$ corresponde a una constante de retroalimentación negativa (por defecto -100). $T_{auto} = \frac{\Delta_{auto}}{auto_{e_{ant}}}$. $T_\delta = \frac{\Delta_\delta}{\delta_{e_{ant}}}$. Δ_{auto} es la diferencia entre el número de ciclos de auto-referencia detectados en los estados e_{ant} y e_{act}. Δ_δ indica el cambio de valor para el parámetro δ durante los estados e_{ant} y e_{act}.

Intuitivamente, si la RC cae recurrentemente en ciclos de auto-referencia sin poder alcanzar otras metas, entonces el valor del parámetro δ debe ser disminuido; por el contrario, si no son detectados ciclos de auto-referencia, la RC debe garantizar no "deshacer" las metas alcanzadas protegiéndolas contra otras metas en conflicto, caso en el cual δ debe aumentar.

L: este término determina el ajuste del parámetro global θ (el umbral de activación mínimo requerido para activar un *Comportamiento*). Debido a que este parámetro global dirige la velocidad de respuesta de la RC (*reactividad vs. deliberación*), L debe controlar entonces el grado de reactividad/deliberación del agente en cada situación. Similar a los términos anteriores, L evalúa el comportamiento de las tasas de variación: T_{frecEC} y T_θ.

Por su parte, T_{frecEC} corresponde a la variación relativa entre las frecuencias con que son percibidos los "elementos cambiantes" del ambiente, en dos estados diferentes del sistema (e_{ant} y e_{act})[21]. La frecuencia de elementos cambiantes indica el número de elementos (metas y perceptos) cuyo cambio de estado ocurre durante el proceso de activación de un *Comportamiento* ejecutable (proceso que puede tardar varios bucles de activación).

Por ejemplo, supóngase que en el estado e_{ant} la RC requiere de 7 bucles de activación para escoger el *Comportamiento* ejecutable. Durante este proceso el agente percibe

[20] Recuérdese que un ciclo de auto-referencia se da cuando se determina una referencia circular en la activación de *Comportamientos*: $a \rightarrow b, b \rightarrow c, c \rightarrow a$. Una de las causas principales de los ciclos de auto-referencia se relaciona con el hecho de que el sistema no puede alcanzar una meta por evitar deshacer una meta previamente satisfecha.

[21] Recuérdese que una RC puede efectuar entre uno y varios bucles de activación (durante los cuales se ejecutan las diferentes dinámicas de dispersión de energía en la red) antes de seleccionar un *Comportamiento* "ejecutable". El parámetro θ define el umbral de activación que debe sobrepasar un *Comportamiento* para convertirse en ejecutable. Si en un determinado bucle de activación ningún *Comportamiento* sobrepasa θ, entonces éste es disminuido por una fracción. Un valor alto para θ supone una RC más deliberativa, incrementando el número de bucles de activación durante los cuales las dinámicas de competencia y cooperación tienen mayor oportunidad de escoger el *Comportamiento* más apropiado para la situación. Contrariamente, un valor bajo para θ supone una RC que actúa de manera reactiva (es decir, que requiere de menos bucles de activación) ante las situaciones cambiantes del ambiente.

el cambio de estado de 3 perceptos (e.g., asúmase tres perceptos nuevos que reflejan el cambio de alguno de los atributos físicos de 3 objetos del ambiente observado), luego la frecuencia en dicho estado sería $frecEC_{ant} = 3/7 = 0.4285$. Ahora supóngase que en el estado e_{act}, la RC ejecuta 4 bucles durante los cuales son percibidos 2 cambios ambientales, luego la frecuencia para este estado corresponde a $frecEC_{act} = 4/2 = 0.5$. La tasa de variación de las frecuencias sería entonces: $T_{frecEC} = (0.5 - 0.4285)/0.4285 = 0.1668$, lo cual al ser un valor positivo indica que ha "aumentado" el número de elementos cambiantes percibidos entre un estado y otro del sistema, y como consecuencia, la RC requiere hacerse más reactiva con el objetivo de poder atender tales cambios ambientales con mayor velocidad. Ahora, supóngase que la tasa de variación del parámetro θ corresponde a: $T_\theta = (52 - 80)/80 = -0.35$. La retroalimentación para este término se expresa formalmente como sigue:

$$L = \begin{cases} 100 - l1, & \text{si } (T_{frecEC} > 0 \wedge T_\theta < 0) \vee (T_{frecEC} < 0 \wedge T_\theta > 0) \\ l2, & \text{de lo contrario} \end{cases} \tag{9.32}$$

donde: $l1$ corresponde a la diferencia absoluta entre las tasas de variación ($l1 = \mid T_{frecEC} - T_\theta \mid$). $l2$ corresponde a una constante de retroalimentación negativa (por defecto -100). $T_{frecEC} = \frac{\Delta_{frecEC}}{frecEC_{e_{ant}}}$. $T_\theta = \frac{\Delta_\theta}{\theta_{e_{ant}}}$.

Debido a que $T_{freEC} > 0$ y $T_\theta < 0$, la retroalimentación generada para este término es positiva. Intuitivamente, se desea que en la medida que la frecuencia $freEC$ aumenta (i.e, que los elementos del ambiente, como los perceptos y las metas, sean altamente cambiantes y dinámicos), entonces el parámetro global θ debe disminuir (es decir, la RC debe hacerse más reactiva y por tanto debe ejecutar menos bucles de activación); mientras que si la frecuencia $freEC$ disminuye (i.e, los elementos del ambiente se vuelven más estáticos), entonces el valor del parámetro θ debe aumentar (y hacer más deliberativa la RC); con lo cual estas dos variables resultan ser inversamente proporcionales.

M: este término valida los "efectos deseados" configurados para cada *Comportamiento*. El término M genera una retroalimentación positiva si en el estado actual percibido por el agente (e_{act}) se encuentra una proposición (i.e., un percepto, una meta, una anticipación, etc.) que se corresponda con alguna proposición de la lista de "efectos deseados" del *Comportamiento* que posee el control consciente del agente en ese instante. Por tanto, este término verifica si los "efectos deseados" que promete alcanzar cada *Comportamiento* son factibles. Formalmente se tiene que:

$$M = m1 \times \sum_{i=0}^{k} edv \tag{9.33}$$

donde: $m1$ es una constante de retroalimentación positiva (por defecto +100). k es la cantidad de proposiciones definidas en la lista de efectos deseados del *Comportamiento* activo (ED_{comp}). edv es una función que determina si el efecto deseado ha sido "verificado" en el estado actual ($S(t)$) formado por la memoria de trabajo, es decir, si se cumple que: $\exists\, p \in S(t) \mid p \cap ED_{comp} \neq \emptyset$.

N: este término valida los "efectos no deseados" configurados para cada *Comportamiento*. El término N genera una retroalimentación negativa si en el estado actual percibido por el agente (e_{act}) se encuentra una proposición que se corresponda con alguna de la lista de "efectos no deseados" del *Comportamiento* que posee el control consciente del agente en ese instante. Por tanto, este término verifica si los "efectos no deseados" que promete evitar cada *Comportamiento* son realmente eludidos. Formalmente se tiene que:

$$N = n1 \times \sum_{i=0}^{k} endv \tag{9.34}$$

donde: $n1$ es una constante de retroalimentación negativa (por defecto -200). k es la cantidad de proposiciones definidas en la lista de efectos no deseados del *Comportamiento* activo (END_{comp}). $endv$ es una función que determina si el efecto no deseado ha sido identificado en el estado actual ($S(t)$) formado por la memoria de trabajo, es decir, si se cumple que: $\exists\, p \in S(t) \mid p \cap END_{comp} = \emptyset$.

Finalmente, la función de aptitud que evalúa el aspecto funcional de cada RC se presenta a continuación:

$$FAAF_i = (k_7 \times G_i) + (k_8 \times H_i) + (k_9 \times I_i) + (k_{10} \times J_i) + (k_{11} \times L_i) + (k_{12} \times M_i) + (k_{13} \times N_i) \tag{9.35}$$

donde: i es una RC específica. las constantes $k_7, k_8, \ldots k_{13}$ son pesos asignados a cada término de la función, con lo cual es posible ejercer mayor presión sobre alguno de estos. Por defecto todos los pesos son iguales a 1.

Es posible variar, a través de la modificación de los pesos respectivos, el impacto que tiene la evaluación del diseño funcional en contraste con la evaluación del diseño estructural y viceversa. Por ejemplo, si se ajustan los pesos $k_7, k_8, \ldots k_{13}$ de la función FAAF a un valor de 2, el proceso adaptativo se orientará preferiblemente hacia la búsqueda de una RC cuyo diseño funcional sea eficiente, aunque la estructura de dicha RC no esté del todo refinada. Idealmente, se desea que el proceso adaptativo genere RCs con diseños eficientes tanto estructural como funcionalmente. En todo caso, queda a criterio del diseñador de cada concreción contextual de la arquitectura el modificar estos pesos.

Como se puede observar, $FAAF$ es una función multiobjetivo que representa las diferentes presiones selectivas ejercidas por el ajuste de cada parámetro. Las diferentes características de cada RC (orientabilidad hacia las metas vs. orientabilidad situacional, adaptabilidad vs. predisposición hacia los planes en curso vs. prevención de metas en conflicto, procesamiento deliberativo vs. velocidad (reactividad), etc.) son configuradas como consecuencia del proceso que adapta las interdependencias entre las distintas presiones selectivas. Por ejemplo, puede decirse que una RC es una red adaptativa y poco orientada hacia los planes en curso, si ésta define valores para γ y ϕ relativamente altos en comparación con π, con lo cual las presiones ejercidas por estos tres parámetros globales generan una interdependencia específica que caracteriza la red. Así pues, la interdependencia entre estos parámetros globales consiste en que el proceso adaptativo debe averiguar el valor adecuado para cada uno los parámetros de manera tal que se maximice la función $FAAF$ y como consecuencia la RC se haga más eficiente.

Obsérvese que no se establece una correlación directa entre los parámetros globales que emplea cada término de la función de aptitud $FAAF$ (e.g., no existe más de un parámetro global en cada término), debido a que ningún parámetro global es una variable dependiente de otro, esto es, que el cambio de valor en un parámetro global no influye necesariamente en el cambio de los demás parámetros globales. El cambio de un parámetro global (variable independiente), producido por las presiones que ejerce el ambiente, influye entonces sobre la variable dependiente asociada (e.g., la variación absoluta de la frecuencia de activación de una meta, etc.), y por tanto el proceso de adaptación consiste en ajustar el parámetro global a tales cambios ambientales.

A pesar de que la evaluación de los diferentes términos de la FAAF dé la impresión de que el ambiente es el encargado de determinar totalmente la actuación del agente, realmente existe una gran influencia de los intereses y objetivos propios del mismo, que aunque tácitos en las ecuaciones, ejercen también una gran presión adaptativa. Por ejemplo, en el término G se mide la frecuencia con la que son activadas o re-activadas las metas del agente en términos de las propias demandas del ambiente que ocasionan tales activaciones de las metas, sin embargo, existen además objetivos propios del agente, como por ejemplo la auto-regulación homeostática de sus motivadores, que conducen también a la activación de metas y sub-metas; y de la misma forma sucede con los demás términos de la FAAF. En otras palabras, esto quiere decir que el proceso por el cual se modela progresivamente el diseño funcional de cada RC, depende tanto de las presiones ambientales como de las necesidades internas del agente, sin embargo, estas últimas están altamente condicionadas por las primeras, y a su vez las primeras cambian como consecuencia de la satisfacción de las segundas por parte del agente.

Finalmente, nótese cómo las diferentes debilidades del modelo de dispersión identificadas en la sección 9.2.5 se incluyen como variables a optimizar dentro de las funciones de aptitud: (1) comportamientos inalcanzables (término B, ecuación 9.20); (2) formación de ciclos de auto-referencia (término C, ecuación 9.21); (3) omisión de conexiones excitativas (término E, ecuación 9.23); (4) efectos de auto-inhibición (término F, ecuación 9.24); (5) estructura rígida de la RC (todos los términos de la función $FAAE$: 9.25); y (6) parámetros globales predefinidos (todos los términos de la función $FAAF$: 9.35).

9.3.2.2. Selección por Resonancia y Mejor Aptitud

En esta sección se describe el proceso por el cual se selecciona la MAT que controlará el foco de atención y la actuación consciente del agente. El proceso consiste de varias fases, como se puede observar en la figura 9.11, las cuales se detallan a continuación.

Considérese que se encuentra activa la meta global n la cual tiene asociado un conjunto de MATs (Repertorio de MATs[22]), donde cada MAT define una RC con una estructura distinta a las demás, y donde al menos un *Comportamiento* de cada RC/MAT promete alcanzar la meta n. Las MATs que se encuentran en el repertorio *"compiten"* entre sí por dos motivos: (1) ser la MAT que controle el foco de atención y la actuación consciente del

[22] Se ha denominado el término Repertorio de MATs a la población total de cromosomas que codifican diferentes diseños estructurales y funcionales de las RCs. Así pues, en el Repertorio de MATs es donde tienen lugar los operadores genéticos de la Máquina Darwiniana que replican, recombinan y varían los cromosomas con mejor aptitud.

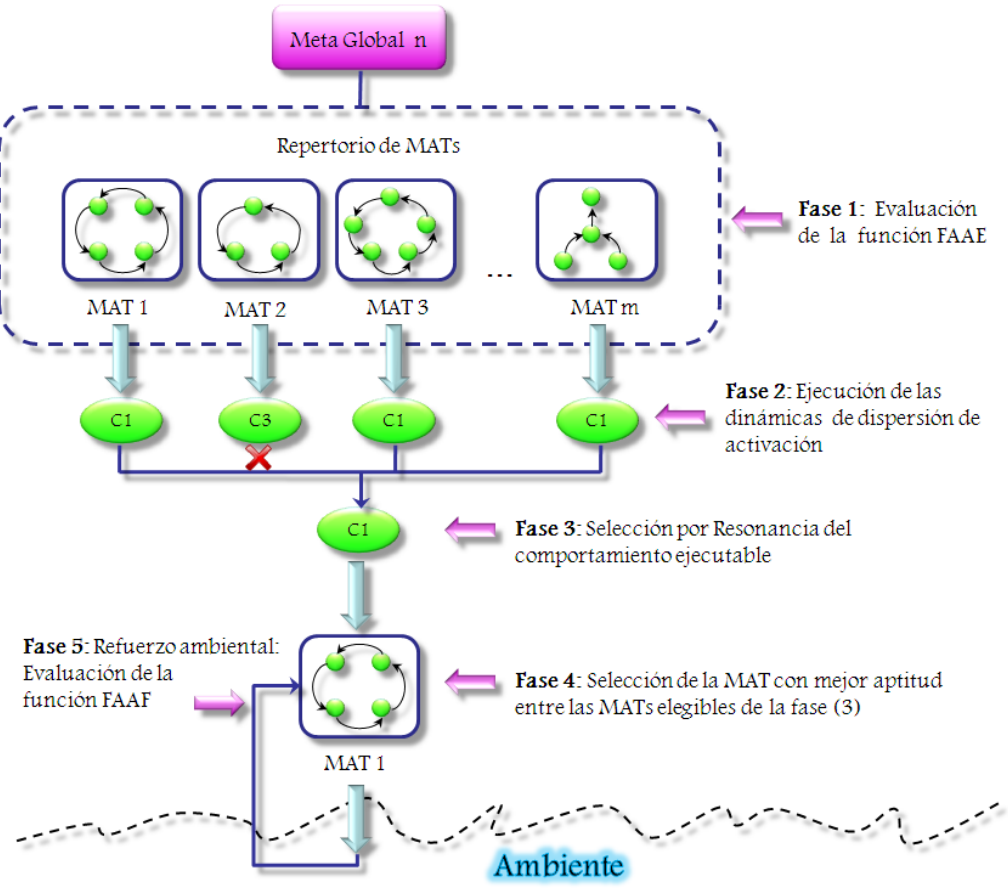


Figura 9.11: Fases de Selección de una MAT.

agente durante un intervalo de tiempo específico, y (2) ser la MAT que acumule mayor cantidad de retroalimentación positiva a fin de aumentar progresivamente el nivel aptitud, y como consecuencia, aumentar su probabilidad de replicarse en las siguientes generaciones donde se activará la Máquina Darwiniana. En este orden de ideas, el proceso de selección de una MAT sigue las siguientes fases:

Fase 1: Evaluación de la función FAAE. Por cada MAT que se encuentre en el repertorio se evalúa la función de aptitud para el aspecto estructural de la RC asociada (FAAE, ecuación 9.25).

Fase 2: Ejecución de las dinámicas de dispersión de activación. Por cada MAT del repertorio que haya obtenido una retroalimentación positiva al evaluarse su FAAE en la fase anterior, se procede a ejecutar las dinámicas de dispersión de activación de su RC correspondiente. Esta fase puede requerir de múltiples bucles de acumulación de activación hasta que la RC asociada a cada MAT produzca un *Comportamiento* ejecutable (como se describió en el algoritmo 9.1). Según se establezca para cada concreción contextual de la arquitectura, es posible determinar que cada MAT no sólo proponga el *Comportamiento* que pretende hacerse con el control consciente del agente, sino que además postule un segundo *Comportamiento*, el siguiente en orden de prioridad de la RC (i.e, el segundo

Comportamiento que ha acumulado mayor cantidad de activación), el cual será ejecutado como un proceso subconsciente (automatismo).

Fase 3: Selección por resonancia del comportamiento ejecutable. En esta fase, las MATs *"cooperan"* y *"compiten"* entre sí formando *"resonancias"* cada vez más fuertes. La resonancia consiste, como se mencionó en la metáfora del *coro aleluya* (véase la sección 4.6), en que las diferentes pautas cognitivas (en este caso, las RCs) se sincronizan gradualmente hasta que todas (o por lo menos la mayoría) recomiendan la activación del mismo *Comportamiento*. Obviamente no se pretende obtener una resonancia absoluta, por el contrario, pueden existir RCs que proponen *Comportamientos* distintos debido al mismo diseño estructural y funcional de la red, pero lo que sí se pretende lograr es un nivel de sincronización donde la mayoría de MATs (al menos la mitad más uno) se "ponga de acuerdo" en qué *Comportamiento* debe ser activado.

El proceso empleado para determinar la resonancia, es similar al mecanismo de *agregación por votación* que lleva a cabo el módulo Efector (descrito en la sección 7.7.3). Evidentemente, aquellas MATs que se han replicado con mayor frecuencia y fidelidad, generarán mayor resonancia a través de la sincronización de sus votos con las MATs parientes. En el ejemplo de la figura 9.11, la mayoría de votos los ha obtenido el *Comportamiento* $C1$.

Recuérdese que el mecanismo de agregación por votación es un método posicional que determina un orden de preferencia. De esta manera, por ejemplo, podría generarse un listado de *Comportamientos* (por orden de votos obtenidos) como el siguiente: $C1$-$C3$-$C5\ldots$ donde se deduce que el *Comportamiento* $C1$ ha tenido LA mayor resonancia, siguiéndole el *Comportamiento* $C3$, y así sucesivamente. En el caso de que esté activado el modo de "ejecución de automatismos" en la concreción contextual de la arquitectura, $C1$ se haría con el control consciente mientras que $C3$ y $C5$ se ejecutarían de forma subconsciente siempre y cuando (como se estableció en la sección 8.3) no existan recursos compartidos, es decir, que tanto $C1$, $C3$, como $C5$ no operen sobre ningún actuador en común; de lo contrario, sólo se activaría $C1$, no habría procesamiento subconsciente de trasfondo, y los *Comportamientos* no elegidos continuarían creando copias de sí mismo en el área de trabajo a fin de hacerse más competitivos en la siguiente activación del proceso de selección de MATs.

Es de resaltar que la velocidad y precisión de la sincronización depende en gran medida de la cantidad de MATs en cada Repertorio de MATs. Para reducir a la mitad las irregularidades de sincronización se necesitan cuatro veces más MATs. Esto obedece a la **ley probabilística de los grandes números** (Pollard, 2003) la cual enuncia que *"el promedio de una sucesión de variables aleatorias tiende a converger al valor promedio de la esperanza individual, es decir, el promedio de una muestra al azar de una población de gran tamaño tenderá a estar cerca de la media de la población completa"*. Lo anterior implica que, para una sincronización más rápida y precisa, se requiere un gran número de MATs (en el orden de 1000) por cada repertorio, lo cual afectaría dramáticamente el desempeño y los tiempos de respuesta del sistema.

Como solución alternativa al incremento excesivo de MATs en cada repertorio, es posible determinar un mecanismo de asignación de pesos que influya sobre el proceso de votación, de manera tal que la competencia se haga más reñida. Debido a que cada

MAT ha tenido una evaluación de su aptitud (en la fase 1) completamente diferente a las de demás, cada una de éstas puede hacer más énfasis sobre su voto multiplicándolo por un factor proporcional a su aptitud. Recuérdese que en el método de agregación por votos cada opción recibe un puntaje de acuerdo con su prioridad, esto es, por ejemplo, suponiendo que la MAT_1 propone la secuencia de *Comportamientos* $C1$-$C3$, entonces $C1$ recibirá 10 puntos por ser el primero de la secuencia (el *Comportamiento* con mayor acumulación de energía), y $C3$ recibirá 5 puntos por ser el segundo. Ahora, adicionalmente se pueden multiplicar estos puntos por una fracción de la aptitud de la RC respectiva, por ejemplo, los votos para $C1$ podrían ser: $votos_{C1} = 10 \times (0.1 \times aptitud)$, es decir, el voto para $C1$ estaría soportado por el 10 % del valor de la aptitud de la RC respectiva. Queda a criterio del diseñador de la concreción contextual el incorporar este factor de énfasis basado en la aptitud de la RC, o el definir en su caso, un tamaño de los repertorios lo suficientemente grande que soporte la sincronización efectiva.

Fase 4: Selección de la MAT con mejor aptitud. En esta fase se selecciona, únicamente entre las MATs que votaron por el *Comportamiento* que resultó elegido en la fase 3, aquella que tenga la mayor aptitud. La MAT elegida se encargará de gestionar el foco de atención por lo menos hasta que nuevamente se active el proceso de selección de una MAT (más adelante se discutirán los criterios que determinan la frecuencia con la que se activa dicho proceso de selección, en términos del tiempo que puede mantener la concentración el agente sobre un conjunto de estímulos en particular). Formalmente se tiene que:

$$MAT_{activa} = \max_{i \in N(t,comp)} (Aptitud(i,t)) \tag{9.36}$$

donde: $N(t, comp)$ es el conjunto de MATs que votaron por el *Comportamiento* ganador $comp$ en el instante t de la fase 3. $Aptitud$ es la aptitud acumulada por cada MAT en el transcurso del tiempo ($\sum FAAE + \sum FAAF$). MAT_{activa} es la MAT que finalmente se hace con el control del foco de atención.

Fase 5: Evaluación de la función FAAF. La MAT elegida en la fase 4 ha estado gestionando el foco de atención del agente durante uno o varios ciclos cognitivos. Cada cierto número de ciclos cognitivos (icc_{FAAF})[23] la MAT que se encuentra activa recibe un refuerzo ambiental que corresponde a la evaluación de la función de aptitud del aspecto funcional de su RC asociada (función FAAF, ecuación 9.35). Esto es, se evalúa la eficiencia de la RC que posee el control de la atención durante el proceso mismo de interacción del agente con su ambiente.

Por motivos de simplicidad, en la figura 9.11 no se muestran otros módulos de la arquitectura que también intervienen en el proceso y que se sobrentiende están presentes. Concretamente, intervienen los módulos de Percepción y Efector, y la Memoria de Trabajo (MT). La MAT que se encuentra activa recaba la información vigente que se encuentra en la MT, y envía al módulo Efector las acciones que propone el *Comportamiento* ejecutable. La MT es actualizada constantemente por el módulo de Percepción. Adicionalmente, cabe mencionar que la MT es el área de trabajo donde tienen lugar las múltiples copias de RCs y las variaciones de las mismas, es decir, que es en la MT donde

[23] Como se mencionó en la sección 9.3.2.1, se define por defecto un intervalo de 10 ciclos cognitivos después de los cuales se compara el estado actual del sistema e_{act} con el estado anterior e_{ant}.

se encuentran los distintos repertorios donde se producen los procesos de competición y cooperación entre las MATs por obtener el control sobre el foco de atención.

Por otro lado, como se puede observar, la aptitud relativa al aspecto estructural (FAAE) es evaluada para cada RC en un modo *fuera de línea* (off-line), es decir, se genera una retroalimentación (positiva o negativa) por cada RC, sin la necesidad de probar dicha RC en el estado actual del ambiente. Por otro lado, la aptitud relativa al aspecto funcional (FAAF) es evaluada *en línea* (on-line) o en tiempo de ejecución, sólo para aquella RC que se encuentra interactuando con el ambiente en el estado actual.

Respecto al proceso general de selección de MATs cabe mencionar algunas consideraciones:

- El proceso de selección de una MAT, antes descrito, puede iniciarse cuando se satisface alguna de las siguientes condiciones: (1) el *Comportamiento* ejecutable de la MAT que se encuentra activo ya no es válido, esto es, que no se cumplen todas sus precondiciones de activación; (2) se cumple cierto intervalo de ciclos cognitivos; o (3) la meta global del agente ha cambiado.

  En el caso (1), el proceso de selección se reinicia en la fase 2 con el fin de obtener un *Comportamiento* que atienda la nueva situación. No se efectúa la fase 1 puesto que la estructura de las RCs no ha cambiado y por tanto no se requiere de una nueva evaluación de la FAAE.

  En el supuesto de que las condiciones ambientales cambien con demasiada frecuencia requiriéndose la re-orientación continua del foco de atención y produciéndose saltos reiterados de activación entre un *Comportamiento* y otro, como en el caso (1) antes descrito, es posible que como consecuencia se ralenticen significativamente los tiempos de respuesta del agente. Una solución conveniente, que corresponde al caso (2), es determinar un criterio para que el proceso de selección no se reinicie cada vez que se requiera activar un *Comportamiento* diferente, sino que en su lugar se reinicie cada cierto intervalo de ciclos cognitivos ($icc_{controlMAT}$), durante los cuales, la RC que se encuentra activa mantenga el control autónomo del foco de atención.

  Finalmente, en el caso (3), cuando una meta global ha cambiado, el proceso de selección de MATs cambia de dominio y se reinicia ahora sobre el Repertorio de MATs asociado a la nueva meta global del agente. Recuérdese que cada meta global del agente tiene asociado un conjunto de soluciones (Repertorio de MATs) que evolucionan independientemente y en diferentes repertorios (es decir, coevolucionan).
- El proceso ontogenético encargado de replicar las RCs con mejor aptitud y de inyectar diversidad en cada Repertorio de MATs, es activado cada cierto intervalo de ciclos cognitivos (icc_{onto}), intervalo que debe ser determinado de manera conveniente de manera que, por un lado, permita reclutar la suficiente masa crítica de clones de RCs que produzca la resonancia adecuada para seleccionar del *Comportamiento* más apropiado ante la situación actual; y, por otro, evite la ralentización de los tiempos de respuesta del agente debido a la sobrecarga del procesamiento

requerido por cada fase de ejecución del proceso ontogenético. Formalmente, se debe tener en cuenta la siguiente restricción:

$$icc_{onto} > icc_{controlMAT} > icc_{FAAF} \tag{9.37}$$

9.4. Modelo Dinámico para el Mecanismo de Orquestación de la Consciencia

En esta sección se especifica la interacción entre los diferentes módulos, componentes y mecanismos de la arquitectura que intervienen durante el proceso de orquestación de la consciencia del agente.

Dentro de lo que cabe destacar del diagrama de actividades de la figura 9.12, son aquellos procesos que se ejecutan en paralelo. Con el fin de que la actuación del agente no se vea interrumpida por la activación del *proceso ontogenético* ni por la activación del *proceso de selección de MATs*, se implementa un hilo de procesamiento concurrente donde el último *Comportamiento* consciente o sub-consciente (automatismo), activado por la MAT que controlaba el foco de atención en un estado anterior del sistema, se encarga de ejecutar uno o varios ciclos cognitivos (los necesarios hasta que dichos procesos concomitantes finalicen su ejecución), en cada uno de los cuales el *Comportamiento* procesa la información almacenada en la MT y genera una acción apropiada.

El propósito de este diseño concurrente es que el agente no detenga repentinamente su actuación debido a la necesidad de ejecutar alguno de los dos procesos antes mencionados (cuyo procesamiento en todo caso no debe superar el orden de tiempo de los milisegundos) sino que contrariamente, el agente deberá mantener una actuación coherente y continua mientras se ejecutan tales procesos. No obstante, es posible que en cierto momento la actuación del agente se vea interrumpida voluntariamente, esto debido a que el propio *Comportamiento* puede definir, por ejemplo, una acción del tipo ⟨`No-Hacer--Nada`⟩, pero no debido a una situación de cuello de botella donde la ejecución del *Comportamiento* tiene que esperar la finalización del proceso ontogenético o del proceso de selección de MATs.

En aquellos ciclos cognitivos donde no se ejecuta el proceso ontogenético ni el proceso de selección de MATs, entonces se procede directamente a ejecutar las dinámicas de dispersión de energía en la MAT que tiene el control del foco de atención, y como consecuencia se activa: (1) un *Comportamiento* ejecutable que se hará con el control consciente del agente, y (2) un *Comportamiento* que se ejecutará como un proceso subconsciente siempre y cuando no existan conflictos por recursos con (1). Nótese entonces que la actividad ⟨`Ejecutar Comportamiento`⟩ (en el diagrama) implica la ejecución de un proceso consciente, y la posible ejecución de un proceso subconsciente.

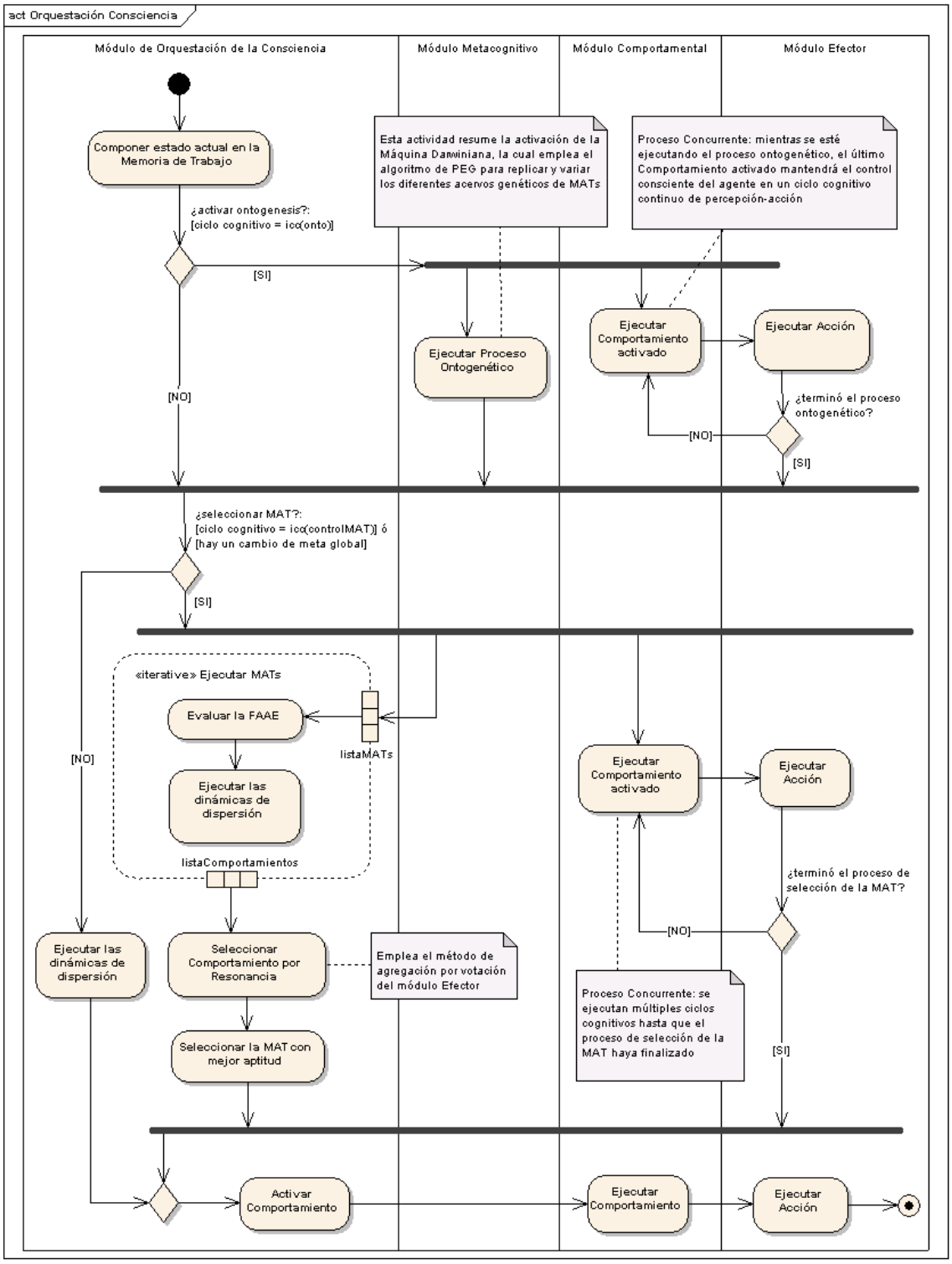


Figura 9.12: Diagrama de Actividades para el Proceso de Consciencia Sintética.

9.5. Discusión

En esta sección se resumen los aportes más significativos propuestos para el modelo de Orquestación de la Consciencia Sintética, así como también se discuten algunos aspectos que podrían ser refinados y mejorados en futuras extensiones de la arquitectura.

Cabe resaltar que el modelo de consciencia sintética propuesto en este capítulo es una aproximación reduccionista a la simulación fenomenológica de una consciencia primaria, descrita desde una perspectiva mecanicista. Sin embargo, este es, incluso, un nivel de especificación complejo de lograr, el cual se enmarca como un requisito suficiente dentro del alcance de la presente tesis. En cuanto al modelo general propuesto es posible resaltar los siguientes aspectos:

- Se ha planteado un modelo de consciencia sintética con diferentes niveles de adaptabilidad: (1) el nivel de *"adaptabilidad"* proporcionado por el modelo de dispersión que define cada estructura cognitiva (RC), el cual modula, adaptativamente, la acumulación de energía de activación sobre aquellos *Comportamientos* más relevantes para la situación del agente (i.e., el estado percibido y las metas); y (2) el nivel de *"plasticidad"* que ofrece el proceso ontogenético en el desarrollo evolutivo tanto de la estructura como de la funcionalidad de las RCs definidas en (1).
- La fenomenología de la consciencia sintética es descrita en términos de múltiples bucles recurrentes, conformados por la interacción continua de procesos de *competencia* y *cooperación* entre los elementos constitutivos, y moldeados por la acción de señales de retroalimentación positiva y negativa. Las dinámicas de *competencia* y *cooperación* actúan en diferentes niveles de la consciencia sintética del agente. Así, a un micro-nivel, la *competencia* se evidencia a través de las conexiones inhibitorias (enlaces de conflicto) entre los *Comportamientos* de una RC, mientras que la *cooperación* se evidencia mediante las conexiones excitativas (enlaces sucesores y predecesores) entre los mismos. Por otro lado, a un macro-nivel, la *competencia* se refleja a través del esfuerzo individual que realiza cada RC del repertorio por alcanzar el valor máximo para su "aptitud". En la medida que el valor de la aptitud de una RC aumenta, las probabilidades de hacerse con el foco de atención incrementan proporcionalmente. Finalmente, la *cooperación*, en un macro-nivel, se refleja a través de la "resonancia" producida por la sincronización de las diferentes RCs en cada repertorio, lo cual se logra mediante la replicación gradual de las RCs más exitosas. Se observa, así como en otros módulos de la arquitectura, que la cooperación en un nivel inferior de interacción (i.e., el bucle interior) representa una competencia en el nivel inmediatamente superior (i.e., el bucle exterior), la cual a su vez se convierte en cooperación en el nivel siguiente, y así sucesivamente.
- Se ha planteado la consciencia sintética del agente, como un *"fenómeno persistente"* y distribuido que reconstruye continuamente representaciones acerca de los estados cambiantes externos (ambientales) e internos (metas, motivaciones, recuerdos, etc.). Este planteamiento contrasta totalmente con la idea de un procesamiento ejecutivo central, o con la teoría del homúnculo de la consciencia[24].

[24]La teoría del homúnculo explica un fenómeno en términos del propio fenómeno que se supone que explica. Esta teoría describe un espectador central, un hombrecillo, que reflexiona internamente sobre los

- En el modelo propuesto se observa una vez más la aplicación del *principio de funciones comunes* enunciado en la sección 4.2: a partir de un conjunto "finito y único" de *Comportamientos* es posible construir una cantidad ilimitada de estructuras cognitivas de orden superior (las RCs), las cuales difieren en estructura, propósito, y funcionalidad una de otras.
- El modelo propuesto permite articular los cambios de atención selectiva del agente a través de la integración de señales como la *"intensidad"*, la *"frecuencia"*, y el grado de *"novedad"* con la que son percibidos los estímulos sensoriales. Desde este punto de vista, cada *Comportamiento* de una RC tiene la capacidad de combinar estas señales con el transcurso del tiempo, de manera que, a mayor cantidad de activación acumulada de estas señales, mayor probabilidad tendrá el *Comportamiento* de controlar el foco de atención. Por tanto, cada *Comportamiento* de la red funciona como un *integrador* (recuérdese el bloque de construcción cibernético descrito en la sección 4.5) el cual puede incrementar el valor de energía de activación interna en proporción directa a la potencia de sus señales de entrada, mientras que simultáneamente libera exponencialmente dicha energía de activación hasta alcanzar un estado de reposo. Cada *Comportamiento* envía señales excitativas hacia *Comportamientos* cooperantes, mientras que intenta inhibir aquellos otros que son competidores. Esto es, por supuesto, retroalimentación positiva en acción!

En comparación con las arquitecturas cognitivas más reconocidas que se basan también en el modelo de Redes de Comportamientos (LIDA (Negatu et al., 2007), CMattie (Anwar y Franklin, 2003), GW (Franklin et al., 2005), ALIVE (Maes, 1994), etc.), y en comparación con las extensiones más relevantes propuestas para el mismo modelo (Maes, 1992; Dorer, 2005; Nebel y Babovich-Lierler, 2004), se puede argumentar que:

- El modelo propuesto posee la capacidad de *"aprender"* de la experiencia. El diseño de cada RC no es definido a priori sino que es el resultado de un proceso de refinamiento incremental (ontogénesis) influido por las restricciones del ambiente circundante y por la necesidad del propio agente de satisfacer sus metas y motivaciones.
- A pesar de que METÁFORA define una estructura rígida en el sentido que describe un conjunto finito de módulos y componentes cuya funcionalidad es específica, exhibe, por otro lado, una arquitectura flexible debido a su capacidad para autoorganizar y re-diseñar las estructuras cognitivas que emplea para dirigir los procesos que controlan la atención y la consciencia del agente.
- El proceso ontogenético brinda la capacidad de re-definir, a través de un proceso darwiniano, la funcionalidad de cada *Comportamiento* (i.e., sus precondiciones, efectos deseados y no deseados), así como las dinámicas que definen la orientabilidad de la Red de Comportamientos.

actos inteligentes del agente. El problema aparece cuando se intenta explicar cómo este hombrecillo lleva a cabo esta tarea, lo cual llevaría a proponer otro proceso interno al hombrecillo, y así sucesivamente, produciéndose una regresión infinita de "reflexiones sobre cómo reflexionar, sobre cómo reflexionar... sobre cómo actuar. Esto también es conocido como el efecto del "juego de las muñecas rusas".

- La aproximación propuesta permite la ejecución concurrente de procesos conscientes y subconscientes, siempre y cuando estos últimos no presenten conflictos por recursos con los primeros.
- El modelo de consciencia sintética de METÁFORA propone un conjunto adicional de entradas de activación a la RC: activación por estímulos novedosos (ecuación 9.4), activación por cambios físicos en los objetos (ecuación 9.5), y activación por elementos relevantes (ecuación 9.6). Estas activaciones adicionales representan la característica de *oscilación* del proceso atencional, característica que no había sido tomada en cuenta en el modelo inicial de Maes (Maes, 1989) ni en ninguna de sus extensiones, y que se estima es un factor que mejora el proceso de re-orientación adaptativa del foco de atención.
- La idea de emplear múltiples Máquinas de Atención (MATs), donde cada una se encarga de cumplir con un objetivo diferente, le permite al agente descomponer, apropiadamente, un problema complejo en varios sub-problemas (o sub-metas) a solucionar. Este modelo, como se verá en la sección 10.3 del siguiente capítulo, forma las bases para la articulación de procesos cognitivos de mayor complejidad: la extracción y ejecución de planes, y la simulación interna de alternativas.
- A diferencia del modelo original propuesto por Maes (Maes, 1989) donde se formulan los nodos de la *red de comportamientos* como componentes pre-programados conformados por instrucciones muy básicas, en la presente tesis se propone cada nodo como un *Comportamiento* complejo, de naturaleza híbrida, con capacidad para: coordinar procesos de aprendizaje explícito e implícito, razonar bajo condiciones de incertidumbre, generalizar y especializar el conocimiento adquirido, procesar información bajo condiciones degradadas (tolerancia a fallos y robustez), y articular un proceso adaptativo y auto-organizativo en la creación y destrucción de las unidades de conocimiento.

Respecto a las características que pueden ser refinadas en futuras extensiones del modelo se tiene:

- La calidad del proceso ontogenético es medida a través de la evaluación de las funciones de aptitud FAAE y FAAF. Estas funciones han sido propuestas en términos de las condiciones cambiantes del ambiente, las metas y motivaciones del agente, y los parámetros globales definidos para cada RC. Así pues estas funciones modulan adecuadamente el proceso de auto-regulación homeostática de los motivadores del agente. No obstante, si el modelo motivacional de METÁFORA fuera extendido a un modelo más complejo que no sólo definiera un conjunto de motivadores sino que además incorporara rasgos sobre la personalidad del agente (i.e., estados de ánimo, estados corporales, actitudes, emociones, etc.), entonces resultaría conveniente proponer términos adicionales para estas funciones que exhibieran la influencia de tales aspectos en el re-diseño de las RCs. En otras palabras, el diseño incremental de las RCs debe estar condicionado tanto por los requisitos ambientales como por las necesidades internas del agente las cuales están determinadas por los módulos Motivacional e Intencional de la arquitectura.
- A pesar de que el proceso ontogenético tiene la capacidad de re-diseñar la funcionalidad de un *Comportamiento* y la conectividad de éste con otros, no está dotado

con la competencia para crear nuevos *Comportamientos* desde cero, a partir de las demandas externas (ambientales) e internas (motivacionales). Se presume que este proceso requeriría, por un lado, de una categorización de los diferentes estímulos (i.e., perceptos, estados anticipados, motivadores, etc.), las posibles acciones que pueden ejecutarse, y las metas y estados que prometería alcanzar dicho *Comportamiento*; y, por otro, necesitaría de una función de aptitud lo suficientemente sofisticada y eficiente como para determinar que dicha categorización creada por el proceso ontogenético define correctamente una nueva habilidad del agente.

- Como se mencionó en la sección 9.4, el módulo de Orquestación de la Consciencia ejecuta, de manera concurrente, tres procesos: el proceso ontogenético, el proceso de selección de MATs, y el proceso de activación de *Comportamientos*. La eficiencia algorítmica de tales procesos depende en gran medida del valor definido para las siguientes variables: (1) la cantidad total de *Comportamientos* especificados para cada agente, (2) la cantidad total de metas globales (que corresponde, consecuentemente, al número total de Repertorios de MATs), y (3) el tamaño máximo de soluciones (población de cromosomas) al que puede expandirse cada Repertorio de MATs. Si estas tres variables se mantienen en proporciones no muy altas[25], la eficiencia algorítmica de tales procesos puede ser considerada aceptable. Sin embargo, si estas variables aumentan (lo cual se convierte en una necesidad cuando el agente se encuentra inmerso en ambientes realistas y altamente dinámicos), la eficiencia algorítmica de cada ciclo cognitivo del agente puede verse afectada.

  Debido a los múltiples bucles anidados que define cada uno de estos procesos y al aumento de las tres variables antes mencionadas, la complejidad algorítmica (temporal y espacial) que puede llegar a tener un ciclo cognitivo es de orden polinomial. Como consecuencia y a diferencia por ejemplo de una complejidad de orden exponencial, el problema de la eficiencia algorítmica resulta tratable. A fin de mantener los tiempos de respuesta dentro de los rangos esperados (100 - 500 ms) para cada ciclo cognitivo, la solución más evidente apunta hacia el diseño de un sistema distribuido para la ejecución de tales procesos. Esto quiere decir, que no sólo basta con la implementación concurrente de dichos procesos a través de una programación multihilo que simule, localmente, una ejecución en paralelo (pseudo-paralelismo); sino que probablemente se requiera de un paralelismo "real" soportado por un diseño distribuido, donde cada proceso cognitivo es ejecutado concurrentemente en distintas máquinas y luego es sincronizado por el hilo principal de proceso que articula el agente, con el objetivo de obtener finalmente tiempos de respuesta adecuados.

  Probablemente deba aplicarse un cambio de enfoque que proponga cada *"agente cognitivo"*, no como el ensamblaje de múltiples procesos concurrentes que son ejecutados por un sólo *"agente software"*, sino que, por el contrario, como la orquestación de múltiples *agentes software*, donde cada uno gestiona un proceso cognitivo individual y especializado. En otras palabras, a diferencia de lo que se

[25] Como se verá en el capítulo 12, en los diferentes experimentos realizados se empleó, por cada agente, no más de 10 *Comportamientos*, no más de 5 metas globales (repertorios), y no más de 20 cromosomas por repertorio; cantidades que no perjudicaron los tiempos de respuesta del agente (entre 100 y 500 milisegundos, según lo especificado en la sección 6.5).

ha propuesto en la presente tesis donde las funciones de cada agente cognitivo son coordinadas por un sólo agente de software (i.e., en una relación uno-a-uno), sería conveniente proponer un enfoque donde cada agente cognitivo estuviese conformado por múltiples agentes software (i.e., en una relación uno-a-muchos). Esto resulta compatible con la idea de Minsky, uno de los padres de las ciencias de la computación y cofundador del laboratorio de inteligencia artificial del MIT, quien en su libro de la Sociedad de la Mente ("The Society of Mind") (Minsky, 1988) propone que la inteligencia no es el producto de un mecanismo singular, si no que proviene de las interacciones gestionadas de una gran variedad de agentes.

Capítulo 10

Proceso de Meta-Cognición

A lo largo de los capítulos anteriores se han descrito parcialmente algunos de los mecanismos que controla el módulo Meta-cognitivo, como por ejemplo, el proceso epigenético que tiene lugar en el módulo Comportamental y en el SPA, el mecanismo de sincronización (integración vertical) entre las representaciones simbólicas y sub-simbólicas en los módulos Comportamental y Declarativo, y el proceso ontogenético que ocurre en el módulo de Orquestación de la Consciencia. Este capítulo, por su parte, describe los mecanismos restantes, aunque no menos importantes, que controlan y regulan los procesos cognitivos de la arquitectura.

10.1. Motivación

Sobre el término meta-cognición, al igual que sobre otros del ámbito de la psicología, es muy difícil encontrar una definición única que sea comúnmente aceptada por todos los investigadores en este campo del conocimiento, lo que determina obligatoriamente el análisis de sus componentes para poder ofrecer una aproximación adecuada.

El tema de la meta-cognición comenzó a interesar plenamente a partir de la década de los sesenta, destacando los trabajos de Flavell (Flavell, 1976), de quien se puede apreciar una primera definición del término:

"Meta-cognición se refiere al conocimiento de uno mismo respecto de los propios procesos cognitivos y sus productos o a cualquier cosa relacionada con ellos, por ejemplo, las propiedades de la información o los datos relevantes para el aprendizaje... Meta-cognición se refiere, entre otras cosas, al control activo y a la consecuente regulación y orquestación de estos procesos en relación con los objetos de conocimiento a los que se refieren, normalmente al servicio de alguna meta concreta u objetivo".

De esta definición pueden destacarse dos notas distintivas de la meta-cognición, una, su contenido y, otra, su función. En cuanto al contenido, decir que la meta-cognición es un proceso relacionado con el conocimiento que puede alcanzar el agente sobre sus propios procesos mentales. En cuanto a la función, decir que el hecho de poder acceder

a los propios procesos cognitivos permite al agente un mejor control de su actividad. En ese sentido, se puede aceptar que sus dos principales elementos integrantes son: la consciencia del propio conocimiento y sus procesos, y el control o regulación que sobre estos se ejerce.

Relacionando la definición anterior de Flavell con el proceso de meta-cognición de METÁFORA se tiene que:

1. El conocimiento que tiene el agente sobre sus propios procesos mentales, es decir, el contenido de la meta-cognición, se construye como resultado de la interacción entre los módulos Comportamental, Declarativo, y de Orquestación de la Consciencia. Estos módulos proveen las estructuras cognitivas que serán objeto del proceso de regulación.
2. La función de control y regulación que ejerce el módulo Meta-cognitivo sobre el contenido, se evidencia a través de la ejecución de los siguientes mecanismos: el Mecanismo de Gestión de EOF (Epigénesis, Ontogénesis y Filogénesis), el Mecanismo de Transmisión de Conocimiento, el Mecanismo de Retroalimentación, el Mecanismo de Generación de Planes, el Mecanismo de Regulación y el Mecanismo de Sincronización.

Por otro lado, Brown (Brown, 1987) argumenta que la esencia de la actividad meta-cognitiva es entendida como la capacidad de combinar y equilibrar interactivamente las variables *de la persona*, *de la tarea*, y *de la estrategia*.

Las *variables de la persona/agente* permiten diferenciar los propios procesos mentales de los ajenos, pero teniendo en cuenta que existen elementos comunes para ambos. Se construyen a lo largo del desarrollo y permiten valorar en cada momento lo que se conoce o no, el grado de certeza o las limitaciones del conocimiento que posee el sujeto sobre algo o sobre sí mismo (sistema de creencias).

Las *variables de la tarea* aluden a la naturaleza de la información que maneja el sujeto cuando reconoce que dicha información puede afectar a su proceso de conocimiento. El conocimiento de las variables que definen la tarea (i.e., la amplitud, el grado de dificultad, etc.), ayuda a determinar o seleccionar los procedimientos para su resolución. Y, de igual forma que en las variables de la persona, la comprensión de la influencia de estas variables también se adquiere de modo progresivo con el desarrollo evolutivo de los sujetos (ontogénesis).

Las *variables de la estrategia* suponen una reflexión sobre las estrategias cognitivas y meta-cognitivas empleadas por el sujeto. Son procedimientos que permiten ir de una situación a otra, relacionar unas tareas con otras y conseguir así objetivos y metas más complejas; son, en suma, procedimientos de resolución.

Brown (Brown, 1987) considera la existencia de un conocimiento procedimental que se reflejaría en las actividades utilizadas para regular y revisar el conocimiento. Se trata de actividades de planificación que se realizan antes de comprender y solucionar un problema (e.g., la predicción de los resultados, la programación del tiempo, la determinación de los procedimientos y recursos a utilizar en la realización de una tarea, etc.), actividades de supervisión (monitoring) durante el aprendizaje (e.g., la verificación, la revisión,

etc.) y actividades de control sobre los resultados (e.g., la evaluación de los resultados y la retroalimentación). La regulación y control del conocimiento va referida a la participación activa del agente que aprende, en tres momentos: antes de iniciar la actividad de aprendizaje (e.g., predecir, organizar, etc.), durante el proceso de aprendizaje (e.g., ajustar, revisar, etc.) y después del aprendizaje (e.g., evaluar, retroalimentar, etc.).

Aplicando la teoría de Brown al proceso de meta-cognición de METÁFORA se tiene que:

1. Se definen las *variables del agente* (en analogía a las variables de la persona) como el conocimiento que se adquiere, a través de procesos epigenéticos, durante el transcurso del ciclo de vida del agente. Este conocimiento puede ser declarativo (autobiográfico, episódico, semántico, factual, etc.), y procedimental (reglas y chunks).
2. Se definen las *variables de la tarea*, como el conocimiento relacionado con el conjunto de características involucradas en el logro de una tarea u objetivo del agente (e.g., un conjunto específico de metas, perceptos, motivadores, acciones, y *Comportamientos*). La interacción específica entre dichas características se representa mediante cada Red de Comportamientos definida por el agente, cuya estructura y funcionalidad se desarrolla a través de un proceso ontogenético.
3. Se definen las *variables de la estrategia*, como el conocimiento que posee el agente para resolver problemas más complejos que involucran más de una tarea. Concretamente, este tipo de conocimiento es representado mediante planes deliberativos que son construidos como resultado del proceso ontogenético.
4. Se define un conjunto de actividades sobre el proceso global de cognición:

   a) Actividad de Planificación: en esta actividad interviene el Mecanismo de Confección y Validación de Planes que se encarga de construir estructuras complejas para la resolución de problemas, a partir de la combinación de estructuras cognitivas más básicas.

   b) Actividad de Supervisión: en esta actividad interviene, por un lado, el Mecanismo de Sincronización que se encarga, como su nombre lo indica, de sincronizar las estructuras cognitivas del nivel cognitivista con las del nivel emergente, a través de los mecanismos de aprendizaje vertical (ascendentes y descendentes); y, por otro, el Mecanismo de Confección de Planes que se encarga de realizar, en tiempo de ejecución, un seguimiento del plan en curso a fin de determinar si es factible la continuación de dicho plan o si es necesaria una re-planificación.

   c) Actividad de Control: en esta actividad están involucrados los Mecanismos de Retroalimentación y Regulación. El Mecanismo de Retroalimentación modula todos los procesos de aprendizaje y adaptación definidos por la arquitectura a fin de satisfacer las metas e intereses del agente. Por su parte, el Mecanismo de Regulación ajusta todos los parámetros que emplea el sistema con el fin de hacer más eficientes los procesos cognitivos.

   d) Actividad de Construcción y Propagación de Unidades de Conocimiento: en esta actividad intervienen, por un lado, el Mecanismo de Gestión de EOF

el cual, a través del proceso filogenético, evoluciona los distintos tipos de estructuras cognitivas (denominadas "memes") a nivel poblacional, es decir, a nivel de todo el sistema multi-agente; y, por otro, el Mecanismo de Transmisión de Conocimiento que coordina la transferencia y asimilación de las unidades de conocimiento evolucionadas (memes) dentro de la población de agentes.

10.2. Control y Regulación Homeostática Guiada por la Retroalimentación

Como se expuso en la sección 6.4, la arquitectura define múltiples subsistemas jerárquicos de lazo cerrado los cuales controlan, mediante los bucles de retroalimentación asociados, las distintas variables de regulación que son – a la vez – causa y efecto. Concretamente, esta sección describe como el Mecanismo de Retroalimentación modula las diferentes señales de información de retorno (externas e internas) con el fin de dirigir los diferentes procesos de aprendizaje y adaptación del agente. A lo largo de los últimos capítulos se ha descrito parcialmente la manera en que estas señales son procesadas por los distintos módulos de la arquitectura, lo cual es sintetizado a continuación:

- Retroalimentación en los componentes del módulo Comportamental (proceso epigenético): (1) el componente RES utiliza la retroalimentación para determinar la eficiencia de los procesos de generalización y especialización de las reglas subsimbólicas extraídas; (2) el componente SIA emplea la retroalimentación para dirigir el proceso de aprendizaje a través del refuerzo o penalización de los anticuerpos que han sido seleccionados para responder a cada estímulo sensorial; y (3) la red PMR emplea la retroalimentación para guiar los procesos de aprendizaje supervisado (retropropagación) y aprendizaje por refuerzo (algoritmo de *Q-learning*) - (Véase las secciones 7.5.2, 7.4.2, y 7.3.2)
- Retroalimentación en el Sistema de Anticipación y Prospección (proceso epigenético): este sistema emplea dos tipos de señales de retroalimentación. Una señal determina el refuerzo ambiental obtenido al ejecutarse la parte acción del clasificador elegido, y la otra señal determina la precisión con la que el clasificador elegido ha estimado el estado anticipado del sistema (véase la sección 8.6.3).
- Retroalimentación en las las Redes de Comportamientos (proceso ontogenético): el proceso evolutivo que opera sobre los repertorios de RCs es modulado por dos señales de retroalimentación, una de ellas evalúa la estructura topológica de la RC generada y la otra evalúa la funcionalidad y eficiencia de dicha RC (véase la sección 9.3.2.1).
- Retroalimentación en la Planificación (proceso ontogenético): el proceso de generación evolutiva de planes requiere de dos señales de retroalimentación, una que evalúa la eficiencia del plan antes de que sea ejecutado (evaluación fuera de línea), y otra que evalúa el plan en tiempo de ejecución (evaluación en línea). Este mecanismo de retroalimentación será detallado en la sección 10.3.4.

- Retroalimentación de Memes (proceso filogenético): el mecanismo memético por el cual son evolucionadas y transmitidas las estructuras cognitivas o memes, requiere de una señal de retroalimentación que determine la aptitud y desempeño de cada uno de estos memes en la siguiente generación.

Las señales de retroalimentación pueden ser externas, internas, o las dos. La retroalimentación externa es una señal ambiental que determina si los "efectos directos" de la actuación del agente producen algún tipo de beneficio o perjuicio para el mismo (e.g., para un animat, intentar atravesar un muro ocasionaría un refuerzo negativo mientras que encontrar una fuente de energía o alimento produciría un refuerzo positivo). Este tipo de retroalimentación es utilizada principalmente por el proceso epigenético durante la especialización de los *Comportamientos* y de los sistemas clasificadores del SPA, por lo que la señal debe ser definida independientemente para cada uno de estos. En entornos que son relativamente estáticos y cuya complejidad es baja, la definición de las señales de retroalimentación resulta ser una tarea elemental. Sin embargo, si el entorno es altamente dinámico y complejo (como suelen ser los entornos reales en los que interesaría aplicar el sistema propuesto), el definir una señal de retroalimentación adecuada para cada *Comportamiento* se convierte en una tarea sumamente complicada debido a que esta señal tendría que considerar, de manera exhaustiva, todas las variables que podrían involucrarse y afectar cada situación, y adicionalmente, tendría que indicar cuál debería ser la respuesta más apropiada del *Comportamiento* o del SPA frente a tales situaciones.

Otro inconveniente con las señales externas es que éstas no siempre se encuentran disponibles y, como consecuencia, la retroalimentación ambiental no es lo suficientemente efectiva para orientar el comportamiento global del agente (e.g., en situaciones donde el entorno percibido no sufre cambios significativos durante ciertos intervalos de tiempo debido a que el sistema se encuentra en un estado estacionario, puede que la retroalimentación ambiental sea nula o que no se produzca de forma inmediata). En general, el entorno donde se encuentra inmerso el agente no siempre provee con facilidad una señal de retroalimentación simple, inmediata y escalar, como usualmente se asume en la literatura sobre las técnicas de *aprendizaje automático* (Sutton y Barto, 1998; Bertsekas y Tsitsiklis, 1996; Ferreira, 2006). En su lugar, como consecuencia de la actuación del agente, el entorno "cambia" a un "nuevo estado" el cual puede ser observable (o no) por el agente, produciendo una señal de retroalimentación continua, difusa y variable.

Así pues, resulta evidente que la modulación de los múltiples procesos de *"aprendizaje"* y *"adaptación"* de la arquitectura no puede fiarse por completo de una señal de retroalimentación externa. En tales entornos realistas debe ser determinada, de forma apropiada, una señal de retroalimentación interna producida por el propio agente. En METÁFORA, este tipo de señal interna puede ser generada, por un lado, como una señal de error continua que representa la diferencia entre la respuesta deseada o de referencia y la respuesta real de un *Comportamiento* y, por otro, como la medida variable del grado de satisfacción de los *motivadores* y de las *metas* del agente. En cuanto al primer caso, se debe disponer de una base de conocimiento explícito, experto, e innato (almacenado en forma de reglas RP y RES) a partir del cual se genera una señal de retroalimentación que modula la adquisición de nuevo conocimiento implícito[1].

[1]Este caso de retroalimentación fue previamente discutido en la sección 7.8 donde se introdujo el tema de los procesos de aprendizaje vertical y supervisado del módulo Comportamental. Este tipo de retroalimen-

En cuanto al segundo caso, la señal de retroalimentación interna se determina en función de la tendencia que tiene cada motivador de retornan y mantenerse en su región homeostática. Por tanto, esta señal mide el nivel de efectividad con el que se satisface una necesidad del agente a través de la orquestación de los procesos atencionales (dirigidos por las MATs) y la activación de *Comportamientos*. Más concretamente, un motivador que se encuentre inicialmente *"fuera"* de la región homeostática producirá una señal de retroalimentación interna en términos de la diferencia de sus niveles de intensidad en un intervalo constante de tiempo, esto es:

$$RIM_i = \begin{cases} (Im_{(i,t)} - Im_{(i,t-1)}) \times k1, & \text{Si } Im_{(i,t-1)} < umbral_{(sobre)} \\ (Im_{(i,t)} - Im_{(i,t-1)}) \times k2, & \text{Si } Im_{(i,t-1)} > umbral_{(infra)} \\ k1 & \text{De lo contrario} \end{cases} \quad (10.1)$$

donde: RIM_i es la retroalimentación interna producida por el motivador i. $Im_{(i,t)}$ e $Im_{(i,t-1)}$ son las intensidades del motivador i en dos instantes consecutivos de tiempo (e.g., en dos ciclos cognitivos seguidos). $umbral_{(sobre)}$ y $umbral_{(infra)}$ son los umbrales de sobre e infra-estimulación respectivamente. $k1$ es una constante positiva con un valor por defecto igual a 10. $k2$ es una constante negativa con un valor por defecto igual a -10.

Como se puede inferir de la ecuación anterior, la RMI es una señal de retroalimentación continua (producida en cada ciclo cognitivo), la cual recompensa la actuación del agente cuando ésta tiende a satisfacer sus propias necesidades internas. En otras palabras, se produce una señal de refuerzo positiva si: (1) estando en la región de infra-estimulación, la intensidad del motivador tiene a disminuir con el paso del tiempo; (2) estando en la región de sobre-estimulación, la intensidad del motivador tiene a aumentar con el paso del tiempo; o (3) la intensidad del motivador se mantiene en la región homeostática. De lo contrario, se produce una señal de refuerzo negativa.

En cada ciclo cognitivo, el mecanismo de retroalimentación ejecuta el siguiente procedimiento: a partir de la MAT que posee el control consciente del agente, se identifican las *metas* que pretende alcanzar la MAT y que además se encuentran activas en la Memoria de Trabajo, esto es:

$$listaMetas = metas_{MAT} \cap metas_{MT} \quad (10.2)$$

Luego, si se ha encontrado una meta que cumpla esta condición (i.e., si $listaMetas \neq \emptyset$), se procede a identificar los *estados internos* (EI) que previamente han generado cada una de dichas metas y que además se encuentran activos en la MT (recuérdese que cada meta tiene asociado un conjunto de precondiciones, las cuales se corresponden con estímulos sensoriales y/o con estados internos del sistema), esto es:

$$listaEI = EI_{listaMetas} \cap EI_{MT} \quad (10.3)$$

tación interna se evidencia cuando el nivel cognitivista supervisa el aprendizaje de nuevo conocimiento en el nivel emergente (e.g., cuando se utiliza el algoritmo de retropropagación en las redes PMR, o cuando el SIA recibe un refuerzo proveniente del módulo de Producciones).

Seguidamente, si se encuentra algún estado interno que cumpla con dicha condición (i.e., si $listaEI \neq \emptyset$), se identifican los *motivadores* que han disparado previamente dichos estados internos y que se encuentran activos en la MT, esto es:

$$listaMotiv = EI_{listaEI} \cap EI_{motivadores} \tag{10.4}$$

Por último, por cada motivador identificado se evalúa la ecuación 10.1 y se obtiene la señal de retroalimentación RIM. La señal RIM se emplea luego para modular el funcionamiento tanto de la MAT como del *Comportamiento* activo. Con respecto a la MAT, la RIM es una señal complementaria a la señal de retroalimentación producida por la función FAAF que evalúa el aspecto funcional de la RC (véase la ecuación 9.35). En cuanto al *Comportamiento*, la RIM es empleada, por un lado, para reforzar el proceso aprendizaje en los componentes del nivel emergente (SIA y PMR) y, por otro, para reforzar los procesos de generalización y especialización coordinados por el componente RES. El algoritmo se presenta en el cuadro 10.1.

Método: retroalimentarConMotivadores

```
1.  Variables
2.     mat        ↓ MAT que se encuentra activa en el ciclo cognitivo actual.
3.     comp       ↓ Comportamiento que se encuentra activo en el ciclo cognitivo actual.
4.  Inicio Procedimiento
5.     listaMetas ← MT.identificarMetasActivadas(mat)
6.     Si listaMetas != null entonces:
7.        listaEI ← MT.identificarEstadosInternos(listaMetas)
8.        Si listaEI != null entonces:
9.           listaMotiv ← MT.identificarMotivadores(listaEI)
10.          Si listaMotiv != null entonces:
11.             Para cada Motivador de listaMotiv hacer:
12.                r ← calcularRIM(motiv) //ecuación 10.1
13.                retroalimentar(mat, r)
14.                retroalimentar(comp, r)
15.             Fin Para
16.          Fin Si
17.       Fin Si
18.    Fin Si
19. Fin Procedimiento
```

Algoritmo 10.1: Algoritmo para la Retroalimentación Interna producida por los Motivadores.

Para finalizar, es importante resaltar que el mecanismo descrito en esta sección puede producir, en general, dos tipos de retroalimentación: negativa y positiva. Los bucles de *retroalimentación negativa*, como los producidos por la señal RIM, se encargan de llevar gradualmente al sistema hasta un estado estable o de equilibrio a través de procesos que auto-regulan homeostáticamente el comportamiento oscilante de los motivadores. Esta auto-regulación controla la interacción entre los motivadores, las metas, la maquinaria atencional (MATs), y los *Comportamientos*, con el fin de mantener los niveles de intensidad de los motivadores dentro de su región homeostática (véase la figura 10.1.a). Por tanto, este bucle de auto-regulación interna dirige el comportamiento global del agente

hacia la búsqueda del placer (región homeostática de los motivadores) y la evasión del dolor (regiones de infra y sobre-estimulación de los motivadores); proceso que de alguna forma resulta análogo a la función reguladora del *sistema dopaminérgico* en el cerebro de los organismos superiores (Bozarth, 1994; Miñarro y Rodríguez, 1996).

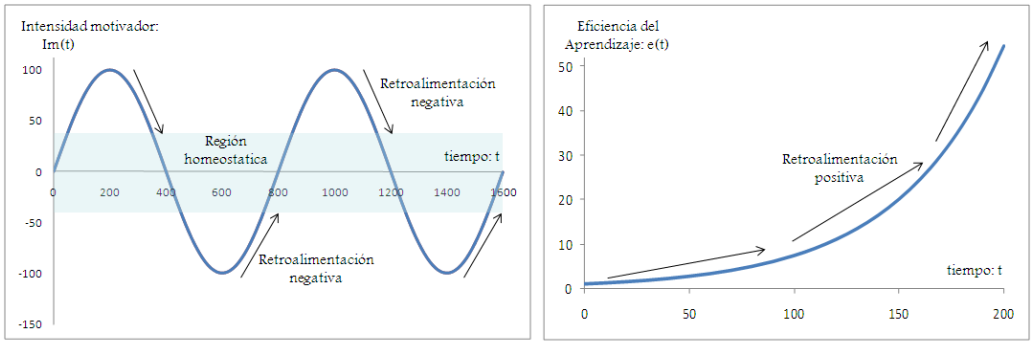


Figura 10.1: Comportamiento de los Bucles de Retroalimentación. a) Regulación homeostática modulada por la retroalimentación negativa. b) Proceso de Aprendizaje modulado por la retroalimentación positiva.

Por otra parte, los bucles de *retroalimentación positiva*, como los generados por las señales de refuerzo interno (en los procesos supervisados) y por las señales de refuerzo externo o ambiental, producen un comportamiento adaptativo a través del cual se intensifica el aprendizaje de situaciones favorables y el olvido de situaciones adversas. Ejemplos de cómo la retroalimentación positiva amplifica la efectividad del aprendizaje/adaptación son: la actualización de las estadísticas de *positividad* y *negatividad* en las Reglas Simbólicas (componentes RES y RP), el incremento gradual del nivel de *aptitud* en los sistemas que emplean algoritmos evolutivos (e.g., en todas las máquinas darwinianas), y el aumento progresivo de los valores-Q en aquellos patrones de activación neuronal que correlacionan apropiadamente una entrada sensorial con una salida motora (e.g., el componente PMR, y las memorias semánticas y episódicas del nivel emergente). Este tipo de retroalimentación hace que la efectividad del aprendizaje/adaptación describa un comportamiento creciente que bien puede ser lineal, exponencial o asintótico (obsérvese la figura 10.1.b).

10.3. Confección y Validación de Planes: Una Aproximación Emergente

"Shakespeare no inventó el vocabulario que utilizaba. Inventó combinaciones de palabras aprendidas, en especial las metáforas *que trasladan relaciones desde un nivel de discurso a otro. De manera similar, muchas conductas inteligentes como la planificación y la creatividad consisten en* ***nuevas combinaciones de cosas viejas****".*

— William Calvin, Cómo piensan los cerebros, 2001.

Hasta el momento se ha propuesto un comportamiento global del agente basado en: (1) respuestas reactivas que se producen en un lazo de "causa y efecto" como respuesta a los cambios ambientales; (2) acciones correlacionadas con un conjunto de entradas sensoriales (perceptos), las cuales reciben un refuerzo del ambiente al ser ejecutadas; (3) un proceso de toma de decisiones basado en un mecanismo anticipatorio que predice los cambios ambientales que acontecerán en un futuro inmediato (en escalas de tiempo de milisegundos), y (4) un mecanismo de planificación a corto y medio plazo (en escalas de tiempo de segundos) articulado por las dinámicas de acumulación y dispersión de energía de activación que cada Red de Comportamientos define a fin de alcanzar una meta concreta.

Complementariamente, esta sección describe una aproximación al proceso de planificación deliberativa a través de la cual se predice una *"secuencia de tareas"* que el agente ejecutará en el medio y largo plazo (en escalas de tiempo de segundos, minutos. o incluso de horas), con el fin de alcanzar un conjunto de metas empleando la menor cantidad de recursos y obteniendo el máximo beneficio. Como se verá más adelante, este proceso involucra la activación de múltiples ciclos cognitivos a través de los cuales se forman, incrementalmente, estructuras cognitivas más complejas: los *"Planes"*.

Conviene advertir que el tema que interesa tratar en esta sección es meramente el de la *planificación automática* (automated planning). La planificación automática es una disciplina de la Inteligencia Artificial que tiene por objeto la producción de estrategias que le permiten a un agente autónomo, a partir de una situación inicial, alcanzar una meta final empleando un número optimizado de recursos y de acciones. A diferencia de los problemas clásicos de control y clasificación, la planificación automática afronta soluciones complejas y con altos niveles de incertidumbre. La dificultad de la planificación depende en gran medida de los supuestos en los que se apoye, por ejemplo: ambientes dinámicos vs. estáticos, restricciones de tiempo, observabilidad completa vs. observabilidad parcial, etc.

Como consecuencia, las distintas aproximaciones en este área cubren un amplio rango que va desde los métodos de planificación determinista (e.g., la progresión y búsqueda hacia adelante, la regresión y búsqueda hacia atrás, la búsqueda heurística, la búsqueda en el espacio de estados, la satisfacibilidad proposicional, etc.), hasta la planificación probabilística (e.g., los procesos de decisión markovianos, los sistemas de transición, la programación dinámica, la optimización combinatorial, los árboles de decisión, etc.), pasando por los métodos de planificación no-determinista (e.g., la regresión no-determinista, la planificación con bucles, la complejidad computacional, etc.). Cabe mencionar además, que incluso los problemas más simples de planificación clásica pueden tener inmensos espacios de búsqueda. Por ejemplo, un problema simple donde se requiera ordenar las piezas de un puzzle en un tablero de 9 casillas (3 x 3), aplicando como regla que sólo se puede mover una pieza, a la vez, hasta una casilla adyacente que esté vacía, produce un espacio de búsqueda de $9! = 362.880$ estados. La explosión combinatorial es evidente.

Debido entonces a la gran complejidad que puede requerir la solución de un problema de planificación, resulta conveniente determinar ciertas restricciones y consideraciones acerca del alcance de dicho proceso de planificación en METÁFORA:

- El mecanismo de generación de planes es una característica que puede ser activada o desactivada, según convenga, dependiendo del dominio en el que se contextualice la actuación del agente. En el caso de desactivarse el mecanismo de generación de planes, el agente empleará sólo la planificación a corto y medio plazo que resulta de ejecutar las dinámicas de la MAT que se encuentra activa en el módulo de Atención.
- A diferencia de los planificadores clásicos, que emplean representaciones vastas del espacio de búsqueda (e.g., los árboles de decisión que crecen incrementalmente en amplitud y profundidad) y/ó que requieren de la definición de un conjunto de métodos, axiomas y operadores que establecen las condiciones para ejecutar secuencialmente un plan (e.g., los planificadores de descomposición de tareas ordenados); se propone un enfoque emergente basado en un *algoritmo evolutivo* el cual, por un lado, permite sesgar adaptativamente la representación de dicho espacio de búsqueda a través de la modulación de los factores de exploración y explotación y, por otro, genera secuencias de tareas (parcialmente optimizadas) que no requieren de la definición de ningún método, axioma u operador para su correcta ejecución (como sí lo requieren en su caso, los planificadores clásicos (Wilkins, 1988; Erol et al., 1994)).
- El mecanismo de generación de planes determina la secuencia óptima de tareas (y metas) que lleva al logro de un estado final, a partir de un estado inicial dado, empleando la mínima cantidad de recursos y obteniendo el máximo beneficio.
- Según el modo de ejecución, la arquitectura define dos tipos de planificación: una *planificación fuera de línea (offline)* y una *planificación en línea (online)*. La planificación fuera de línea consiste en una planificación deliberativa que el agente realiza "antes" de iniciar su actuación en el mundo real y sus efectos en el medio y largo plazo pueden ser estimados a través de un mecanismo de simulación interna. Por otro lado, la planificación en línea se realiza en tiempo de ejecución, y su objetivo es corregir, en el corto plazo, las desviaciones del plan inicial generado por la planificación fuera de línea.
- La planificación fuera de línea implica dos aspectos: (1) la producción de una estructura jerárquica de tareas donde la completitud de cada tarea depende de la completitud de un conjunto de sub-tareas constitutivas, y así sucesivamente hasta que el problema es reducido a tareas atómicas; y (2) la generación de una planificación parcial: se determina una estructura jerárquica de tareas de alto nivel que logran una meta más global, aunque no se determinan las acciones concretas que ejecuta cada una de estas tareas. Las acciones concretas, son definidas a través del mecanismo de simulación interna y son corregidas en tiempo de ejecución por la planificación en línea. Considérese una tarea, como la definición de una meta global que debe ser alcanzada a través de la ejecución de su MAT respectiva.
- La planificación en línea genera planes adaptativos que pueden ser modificados durante la actuación del agente para ajustarse a las necesidades cambiantes (tanto externas como internas).
- La optimalidad de los planes generados está determinada por el mecanismo evolutivo que guía el desarrollo de tales planes en tiempo de ejecución (proceso ontogenético). Este proceso evolutivo emplea un mecanismo de "simulación interna"

para evaluar las consecuencias más probables que acarrearía la ejecución de cada plan en el ambiente "real".

- La planificación deliberativa requiere que el agente posea cierta experiencia previa sobre el mundo. En la medida que el agente disponga de mayor cantidad de *modelos mentales* fiables y eficientes acerca de: (1) las situaciones pasadas (conocimiento declarativo), (2) el estado actual (Memoria de Trabajo), (3) las conductas aprendidas (*Comportamientos* y MATs), y (4) las predicciones sobre los posibles cambios futuros del ambiente (anticipaciones); podrá confeccionar planes y simulaciones internas más ajustadas a la realidad.

Cabe resaltar que el mecanismo de confección de planes no genera estructuras rígidas ni secuencias inmutables de acciones, si no que por el contrario, genera secuencias flexibles de pasos que pueden ser corregidos en tiempo de ejecución. Según (Calvin, 2001), para la mayoría de los movimientos que coordina el cerebro humano, como levantar una taza de café hasta los labios, hay tiempo para la improvisación sobre la marcha. Si la taza es más ligera de lo que se pensaba, se puede corregir la trayectoria antes de que ésta pegue en la nariz. Así pues, en realidad no se necesita de un plan anticipado completo; basta con un objetivo y una corrección periódica de los detalles. Se empieza en la dirección general y se va corrigiendo la trayectoria, como lo hacen los cohetes que viajan a la luna. Es por esto que cada plan define una secuencia general de acciones globales (sub-tareas), mientras que las MATs y los *Comportamientos* precisan los detalles de la actuación, improvisando soluciones en tiempo de ejecución.

10.3.1. Descomposición Jerárquica de Tareas

Tomando como referencia el modelo de planificación de las Redes Jerárquicas de Tareas (HTN, por su acrónimo en inglés), se propone un método representacional similar para los planes generados por el proceso ontogenético. Las HTN son un sistema de planificación jerárquica que descompone las tareas en sub-tareas, hasta conseguir primitivas que pueden ser ejecutadas directamente (Sacerdoti, 1977; Erol et al., 1994). En el caso de las HTN el dominio del planificador se describe básicamente mediante un conjunto de métodos, axiomas y operadores (i.e., las sub-tareas que componen una tarea y las condiciones para ser disparadas). Un problema de planificación debe contener un estado inicial y un conjunto de tareas parcialmente ordenadas a conseguir. El resultado de su aplicación es un conjunto de operadores que permiten, desde el estado inicial, conseguir todas las tareas de la lista.

Un caso específico de las HTN son los planificadores de descomposición jerárquica "ordenada" de tareas. Este tipo de planificadores independientes del dominio, estructuran y representan los planes en el mismo orden en que serán ejecutados. Esto reduce la complejidad del proceso de razonamiento debido a que elimina gran cantidad de incertidumbre relativa al "mundo", lo que facilita la incorporación sustancial de expresividad dentro del algoritmo de planificación. Un ejemplo muy común de este tipo de planificadores es SHOP (Simple Hierarchical Ordered Planner) y sus derivados JSHOP y SHOP2 (Nau et al., 2005).

Se ha mencionado, como punto de referencia, el sistema de planificación basado en las HTN y el planificador SHOP debido a que los planes deliberativos producidos por METÁFORA se estructuran representacionalmente de la misma forma, aunque el proceso de generación de los mismos es completamente emergente, esto es, se generan como resultado del proceso ontogenético y no como el resultado de ejecutar búsquedas hacia adelante y hacia atrás a través del espacio de búsqueda como lo hacen los planificadores de descomposición jerárquica ordenada de tareas. De esta manera, la hipótesis propuesta plantea, como se ha insinuado anteriormente, que los planes generados no requieren de una definición explícita del dominio del problema (i.e., métodos, axiomas, u operadores que sirven como condiciones y determinantes de la activación de cada tarea); sino que, contrariamente, el proceso ontogenético genera planes automáticos definidos como una secuencia de tareas abstractas cuya implementación concreta es determinada por la simulación interna de cada tarea. En otras palabras, el proceso ontogenético primero define un plan como una secuencia de *"tareas de alto nivel"*, cada una representada por una MAT especifica, y luego obtiene una secuencia de *Comportamientos* y acciones concretas por cada tarea a través de la ejecución, en un ambiente simulado, de las dinámicas de dispersión de energía de activación que define la MAT asociada.

Este enfoque emergente de producción de planes hace que el proceso sea aún más independiente del dominio al no requerirse una pre-definición de los mencionados métodos, axiomas y operadores, que sí utilizan los planificadores de descomposición jerárquica ordenada de tareas (SHOP, JSHOP, SHOP2, etc.).

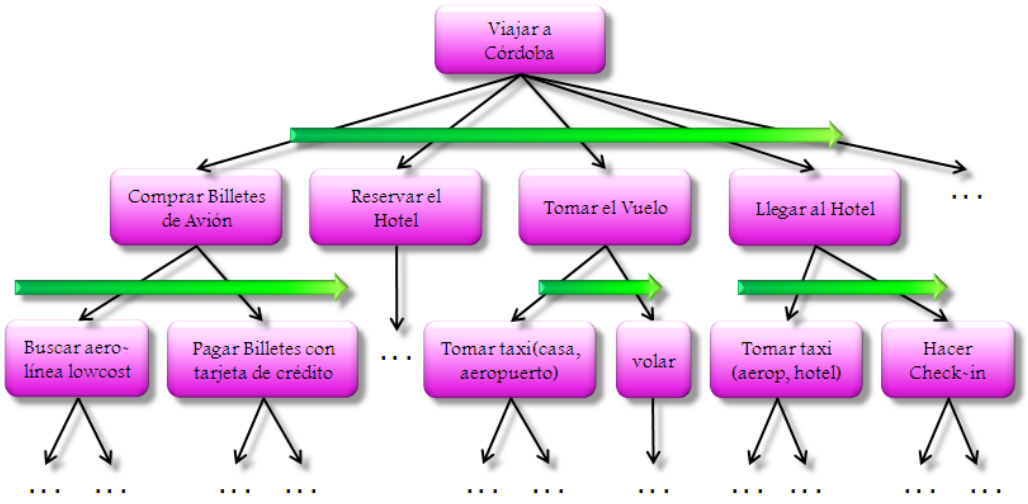


Figura 10.2: Fragmento de la Descomposición Jerárquica de Tareas para un Plan.

Considérese el ejemplo de la figura 10.2, el cual corresponde a una estructura jerárquica que describe, parcialmente, las tareas para ejecutar el plan ⟨`Viajar a Córdoba`⟩. El plan se ha representado como una estructura arbórea de la cual es posible resaltar que:

- El plan se estructura como un conjunto jerárquico de tareas. Por ejemplo, la tarea principal ⟨`Viajar a Córdoba`⟩, requiere de sub-tareas como: ⟨`Comprar Billetes de Avión`⟩, ⟨`Reservar el Hotel`⟩, ⟨`Tomar el Vuelo`⟩,

⟨`Llegar al Hotel`⟩, etc; y cada una de estas tareas se descompone sucesivamente en otras sub-tareas más específicas.

- La ejecución del plan resulta similar a la estrategia de búsqueda que emplea el lenguaje PROLOG (Nau et al., 2005). Esto es, se hace un *recorrido en postorden* del árbol jerárquico y se ejecutan sólo las sub-tareas atómicas (nodos hoja) del árbol. Por ejemplo, supóngase que el plan de la figura 10.2 no puede ser reducido a tareas más atómicas, entonces se ejecutaría la siguiente secuencia de tareas: ⟨`Buscar aerolínea lowcost`⟩ → ⟨`Pagar billetes con tarjeta de crédito`⟩ → ⟨`Reservar el hotel`⟩ → ⟨`Tomar taxi(casa, aeropuerto)`⟩ → ⟨`Volar`⟩ → ⟨`Tomar taxi(aeropuerto, hotel)`⟩ → ⟨`Hacer check-in`⟩.

De manera análoga, los planes deliberativos formados por el proceso ontogenético de METÁFORA tienen una estructura jerárquica de tareas, cuya ejecución es lineal y ordenada.

10.3.2. Proceso Ontogenético y Máquina Darwiniana para la Generación de Planes

A continuación se describe el funcionamiento de la Máquina Darwiniana que emplea el proceso ontogenético para poder manipular y evolucionar los planes automáticos. Primero que todo, para codificar un plan automático (empleando la PEG como herramienta evolutiva) es necesario extender la codificación genética propuesta en el apartado 9.3 para las RCs. Para realizar esta extensión se requiere introducir 3 nuevos conceptos: *Funciones Definidas Automáticamente*, *Génes Homeóticos*, y *Células* , los cuales se detallan a continuación.

Las *Funciones Definidas Automáticamente* (FDA) son un término, acuñado por Koza (Koza, 1992), para referirse a un modo específico de reutilización de código empleado en la Programación Genética (PG). Koza propone el diseño de un "programa principal" que invoca diferentes sub-rutinas, donde cada una de estas sub-rutinas corresponde a una FDA específica. Para esto, se requiere por un lado, de un cromosoma multigénico que define la estructura de una FDA específica, y, por otro, de un gen especial que juegue el rol de programa principal, el cual define el orden y frecuencia con que deben ser invocadas las diferentes FDAs. Este gen especial es llamado *Gen Homeótico*.

En Biología, un *homeobox* es una secuencia de ADN que forma parte de los genes implicados en la regulación del desarrollo (morfogénesis) de los animales. Los genes que tiene un homeobox se llaman *genes homeóticos*. Los genes homeóticos codifican ciertas proteínas que le indican a las células de distintos segmentos del embrión en desarrollo, "qué clase de estructuras tienen que hacer" (e.g., antenas para la cabeza y patas para los tres segmentos torácicos en una mosca). Continuando con la analogía biológica, en la PG, el producto de la expresión de tales genes homeóticos es denominado *Célula* (Ferreira, 2006). Por consiguiente, los genes homeóticos determinan cuáles genes son expresados en determinadas células, y cómo estos interactúan entre sí. En otras palabras, los genes homeóticos determinan cuáles FDAs serán invocadas en un determinado programa principal, y cómo interactúan estas FDAs en dicho programa.

En este orden de ideas, la codificación genética de un Plan y de las estructuras cognitivas subyacentes se acopla a la definición de los conceptos antes descritos de la siguiente forma:

- Una MAT se define como una estructura multigénica (compuesta de varios genes estructurales y uno funcional) la cual describe el diseño de una Red de Comportamientos (RC) específica. Cada estructura genética de este estilo se corresponde con una FDA determinada.
- Un cromosoma se define como la concatenación de múltiples FDAs. Pueden existir tantas FDAs como tipos de MATs diferentes se hayan definido para el agente. Y existen tantos tipos diferentes de MATs como metas globales o "tareas" se definan para el dominio de la concreción contextual. Luego un cromosoma se compone de tantas FDAs como tareas se definan.
- Una célula se define como la concatenación de un cromosoma junto con un gen homeótico. La expresión fenotípica de una célula corresponde entonces a una secuencia de tareas (FDAs) que deben ejecutarse ordenadamente, es decir, un *Plan*.

FDA 0															FDA 1	FDA 2	FDA 3						
Gen Estructural 0					Gen Estructural 1					Gen Funcional					…	…	…	Gen Homeótico					
0	1	2	3	30	0	1	2	3	30	0	1	2	3	4	…	…	…	0	1	2	3	4	5	30
C_1	D	P_2	P_3	…	C_2	P_3	P_1	P_4	…	3	0	7	12	9	…	…	…	T	D	1	2	0	3	…

Expresión Genética 10.1

Considérese la expresión genética 10.1, la cual presenta una célula formada por cuatro FDAs y un gen homeótico. Cada FDA define una MAT/RC específica en términos de un diseño estructural (a través de genes estructurales que determinan los enlaces de activación/inhibición entre los *Comportamientos*) y de un diseño funcional (a través de un gen funcional que especifica los parámetros globales de la RC). Adicionalmente, se presenta un gen homeótico que está expresado en notación Karva, es decir, empleando un codificación genotípica lineal que puede ser traducida a una representación fenotípica en forma de Árboles de Expresión (de manera similar a como se explicó en la sección 9.3.1.2). Una representación gráfica de esta expresión genética se presenta en la figura 10.3.

De forma similar al proceso ontogenético descrito para las MATs/RCs (véase la sección 9.3.1), la Máquina Darwiniana para el gen homeótico está basada en la Programación de Expresiones Genéticas (PEG), aunque con algunas consideraciones especiales las cuales se enuncian a continuación:

- El dominio de las *terminales* empleadas en la codificación del gen homeótico incluye todas las FDAs que componen el cromosoma. Estas terminales son representadas mediante un índice, esto es, la terminal "0" corresponde a la FDA_0, la terminal "1" corresponde a la FDA_1, y así sucesivamente.

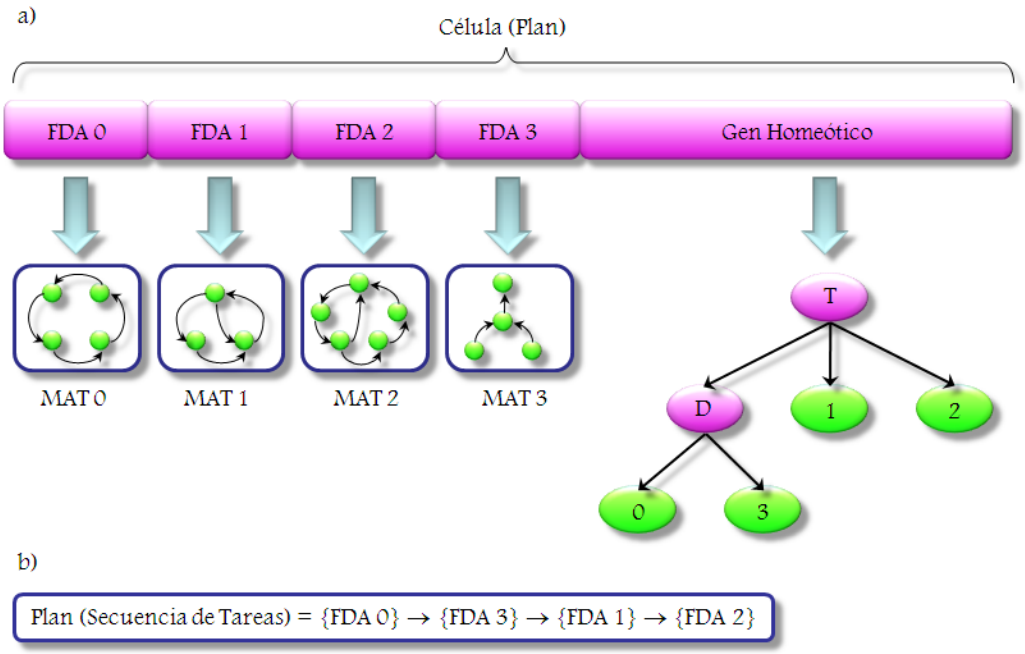


Figura 10.3: Representación de un Plan. a) Representación Genotípica de una Célula. b) Expresión Fenotípica del Plan como una secuencia ordenada de tareas que se extrae de los nodos hoja del sub-árbol representado por el gen homeótico.

- El dominio de las *funciones* empleadas en la codificación genética sólo involucra las funciones de conectividad (**D** y **T**), a través de las cuales se determina la descomposición jerárquica de tareas que conforma el plan. Por ejemplo, en la figura 10.3.a, la tarea principal que propone alcanzar el plan (y la cual se representa de manera implícita), se descompone en dos sub-tareas principales (i.e., el nivel dos del árbol de expresión: 1, 2), de las cuales, la primera sub-tarea se descompone a su vez en dos sub-tareas (i.e., el nivel tres del AEX: 0, 3).
- El gen homeótico, al igual que cada gen estructural de las FDAs, define una parte cabecera y una parte terminación con longitudes específicas, sobre las cuales se establecen los marcos de lectura abiertos (MLA) –véase la sección 9.3.1.2. Gracias a la flexibilidad de los MLA, el gen homeótico puede expresar los planes como estructuras jerárquicas de diferentes tamaños y formas.
- Los operadores genéticos del gen homeótico sólo actúan sobre el dominio de éste, es decir:
  - Operador de Mutación: este operador cambia la información genética del gen homeótico empleando sólo las funciones y terminales antes descritas. La probabilidad de mutación ($p_{(m,h)}$) definida por defecto para este operador es de 0.5.
  - Operador de Transposición de Secuencias de Inserción: al igual que el operador anterior, sólo opera sobre el dominio de las funciones y terminales del gen homeótico. Por defecto se define una probabilidad $p_{(si,h)} = 0.1$.

  - Operadores de Recombinación en un punto y dos puntos: estos operadores actúan sólo en el dominio del gen homeótico de dos células dadas Es decir, sólo se pueden recombinar dos genes homeóticos entre sí, descartándose la posibilidad de recombinación entre dos tipos de genes diferentes (estructurales, funcionales u homeóticos). La probabilidad por defecto para cada operador es de $p_{(rup,h)} = 0.3$.

- El producto de la expresión del gen homeótico, es decir, la célula o *plan*, corresponde a una secuencia de tareas que es extraída de la estructura jerárquica definida por el gen, donde las tareas serán ejecutadas en el mismo orden que define la secuencia. Por ejemplo, en la figura 10.3.b, la secuencia resultante de recorrer el AEX en postorden, extraer los índices o terminales del mismo, y convertir tales índices en sus FDAs respectivas, genera como resultado una secuencia de FDAs (y las MATs correspondientes) que deben ser ejecutadas en un orden específico: primero, debe ser ejecutada la FDA_0 (MAT_0), luego debe ser ejecutada la FDA_3 (MAT_3), a continuación debe ejecutarse la FDA_1 (MAT_1), y por último, deber se ejecutada la FDA_2 (FDA_2). Como se puede observar, la secuencia producida corresponde a un plan "general" cuyas acciones más específicas deben ser extraídas por cada MAT a través de la "simulación" de la ejecución del plan (tema será discutido en la siguiente sección). Así pues, el gen homeótico ha servido como un programa principal que determina cómo deben interactuar las FDAs definidas por el cromosoma con el fin de construir un plan que resuelva una situación que involucra una o varias tareas.
- La estructura jerárquica de tareas que define el gen homeótico, puede también determinar invocaciones recurrentes de tareas. Por ejemplo, resulta factible que para el logro de una determinada tarea, se requiera la siguiente secuencia de sub-tareas: $FDA_3 \rightarrow FDA_0 \rightarrow FDA_3 \rightarrow FDA_2 \ldots$ Obsérvese que la tarea FDA_3 requiere ser ejecutada antes y después de la tarea FDA_0.

Es importante resaltar en este punto, cómo las estructuras cognitivas complejas son el resultado de un *"ensamblaje incremental y adaptativo"* de estructuras cognitivas más simples. Los bloques básicos de conocimiento, esto es, las pautas cognitivas formadas por el conjunto de reglas simbólicas y sub-simbólicas (RP, RES, SIA, y PMR), construyen estructuras cognitivas más complejas que determinan aspectos conductuales del agente: los *Comportamientos*. A su vez, los *Comportamientos* forman coaliciones entre sí creando estructuras cognitivas de orden superior, las Redes de Comportamientos, las cuales se encargan de llevar a cabo tareas específicas cuyo alcance es determinado por una meta global. Por último, los diferentes modos de interacción entre estas Redes de Comportamientos genera diversas estructuras cognitivas y estrategias que operan en la resolución de problemas más complejos: los Planes.

En este orden de ideas, sería posible definir un nivel superior de ensamblaje de estructuras cognitivas, utilizando para ello una estructura genética *"multicelular"*. La idea básica de este postulado se apoya en la generación de una estructura cognitiva que defina múltiples genes homeóticos. Cada gen homeótico describiría una faceta diferente del plan, definiendo condiciones distintas de activación y como consecuencia, distintas secuencias de tareas. Por ejemplo, si se cumplen "x'' condiciones, se activaría el primer gen

homeótico del plan el cual define una secuencia específica de tareas. Por el contrario, si se cumplen "y" condiciones, se activaría el segundo gen homeótico el cual definiría una secuencia de tareas diferente a la del primero, y así sucesivamente. Por otro lado, podría considerarse cada gen homeótico como la representación de un plan parcial o micro-plan de actuación, donde la composición de todos los genes homeóticos formaría un macro-plan o macro-estrategia de resolución de problemas. No obstante, dentro del alcance de esta tesis sólo se abarcará el diseño de soluciones unicelulares, dejando abierta la posibilidad de un diseño multicelular que podría ser retomada en trabajos posteriores.

Finalmente, es importante destacar el aspecto co-evolutivo de este enfoque ontogenético: el "nivel de aptitud" de cada célula, el cual determina el nivel de desempeño de la misma, depende de la evaluación de una función multi-objetivo. Esto quiere decir que, la aptitud de la célula no sólo depende de cuán bueno ha sido el *plan* generado por el gen homeótico, sino que además depende de cuán bien diseñadas (estructural y funcionalmente) están cada una de las FDAs (MATs) que intervienen en la construcción del *plan*. Así pues, la evolución de cada uno de los genes implica la evolución co-determinada de los demás, donde el valor final de la función de aptitud corresponde a la suma de las aptitudes individuales de cada gen, las cuales tienden a ser maximizadas por el mismo proceso ontogenético.

10.3.3. Mecanismo de Simulación Interna

> *"El sistema nervioso es... una máquina calculadora capaz de hacer modelos o paralelismos de los acontecimientos externos... Si el organismo lleva en la cabeza un* ***"modelo a pequeña escala"*** *de la realidad exterior y de sus propias acciones posibles, será capaz de probar varias alternativas, decidir cuál es la mejor, reaccionar a situaciones futuras antes de que surjan, utilizar el conocimiento de hechos pasados para afrontar el futuro, y reaccionar a las emergencias que se le presenten de un modo mucho más completo, seguro y competente en todos los aspectos".*
>
> — Kenneth Craik, The Nature of Explanation, 1943.

Siguiendo la línea de pensamiento de Craik, se propone un mecanismo de simulación que probará los distintos planes alternativos generados por el proceso ontogenético antes de que el agente ejecute alguno de estos en el "mundo real", o más precisamente, en el mundo donde se encuentra inmerso éste. El desarrollo y simulación de múltiples experimentos en paralelo hace posible que se llegue a buenas soluciones más rápidamente que si un control ejecutivo se encargase de hallar la solución óptima y después la impusiera como único plan deliberativo a seguir por el agente. Además, la obtención y puesta en ejecución de las múltiples soluciones buenas proporcionará la diversidad (*variedad indispensable*, en términos de Ashby (Ashby, 1956) y de Milsum (Milsum y Bertalanffy, 1978)) que aumenta la probabilidad de supervivencia del agente cuando cambia el medio ambiente.

A continuación, se enuncian los objetivos principales del mecanismo de simulación interna:

- El mecanismo provee al agente con la capacidad de evaluar escenarios de actuación futuros, sobre los cuales se sopesan las consecuencias de las acciones en el

largo plazo y no sólo las consecuencias inmediatas de las mismas. Por ejemplo, supóngase que el agente cognitivo es un "animal virtual" (animat) que debe aprender a buscar alimento en un entorno simulado y determinar qué estrategia le produce una mayor recompensa. Asúmase que dicho animat es un oso que busca miel en panales cubiertos de abejas. Si el oso sólo guiara su actuación por el refuerzo negativo que obtiene cada vez que intenta meter la mano en el panal y es picado por una abeja, no intentaría nunca más ejecutar dicha acción. No obstante, si el oso es capaz de simular internamente la situación y determinar, a través de ésta, que el refuerzo final es positivo (como consecuencia de obtener suficiente cantidad de miel), a pesar de las múltiples picaduras de las abejas, entonces su actuación estará guiada más hacia la consecución de una meta a largo plazo que hacia el refuerzo negativo inmediato. Claro está, si el balance de la simulación representa un refuerzo negativo, como por ejemplo la posibilidad de morir a causa de las múltiples picaduras, entonces la actuación del agente debe conducir hacia otra alternativa diferente.

- La simulación interna resulta particularmente útil debido a que el agente puede ensayar escenarios alternativos cuya ejecución en el mundo real no permiten el aprendizaje, esto es, aquellas situaciones que causan la "destrucción" o paralización del agente en el primer intento. Por ejemplo, si el animat decide ahora saltar por un precipicio, no tendrá oportunidad de aprender de sus errores, pues se moriría como consecuencia de saltar. Se observa entonces, que el mecanismo de simulación interna crea un entorno favorable y seguro donde el agente puede poner a prueba sus planes, conductas y acciones; y donde además puede "aprender" nuevas estructuras cognitivas eliminando posibles condiciones de riesgo.
- El mecanismo de simulación interna es lo suficientemente robusto como para operar aún cuando el agente dispone de modelos mentales parciales o con un alto nivel de incertidumbre acerca del mundo. Para esto, el sistema de simulación se apoya en los sub-sistemas subyacentes que sirven como una buena aproximación de la interacción del agente con su ambiente: los sistemas de razonamiento inferencial hacia adelante y hacia atrás (i.e., los Sistemas de Reglas de Producción, las Redes Neuronales, los Sistemas Inmunes Artificiales, y las Redes de Comportamientos), las memorias de largo plazo (i.e., la Memoria Episódica y Semántica), y el Sistema de Prospección y Anticipación.

La simulación de la interacción del agente con su ambiente resulta ser relativamente sencilla si se escoge el nivel de abstracción apropiado (Jakobi et al., 1995). Esta idea resulta atractiva desde un punto de vista intuitivo. Nuestro cerebro puede observar e imaginar conscientemente los resultados de nuestras acciones en un entorno, pero con frecuencia no genera modelos conscientes acerca de qué podemos hacer o cómo lo hacemos. Muy pocas personas podrían describir un sistema dinámico o un modelo neurofisiológico de ellas mismas conduciendo una bicicleta, sin embargo, la mayoría de los ciclistas pueden imaginarse a sí mismos conduciendo por una calle, incluso si nunca han visto tal calle. Esto puede ser explicado mediante la existencia de un modelo generalizado de las interacciones motrices del individuo con el entorno, el cual puede ser empleado para ensayar situaciones novedosas (Vaughan y Zuluaga, 2006).

En la presente propuesta se relaciona el proceso de simulación interna (sensoriomotriz) con la noción típica en psicología sobre la *imaginación*, la cual es vista como un

mecanismo que no requiere poner en práctica cada una de las hipótesis mentales generadas, y que además considera las consecuencias del comportamiento del agente sin tener una plena comprensión acerca de éste. La relación entre imaginación y simulación es expresada por Dawkins como sigue:

> *"We all know, from the inside, what it is like to run a simulation of the world in our heads. We call it imagination and we use it all the time to steer our decisions in a wise and prudent directions".*
> — Richard Dawkins, The evolved imagination, 1995.

Adicionalmente, la presente aproximación considera cada "plan ejecutable" como un postulado verdadero acerca de lo que el agente puede hacer, y los "planes no probados" como hipótesis. En este sentido es posible ver al agente cognitivo como un "científico Popperiano" que busca la verdad mediante la eliminación de las malas hipótesis identificadas a través del mecanismo de simulación interna:

> *"Los científicos pueden aniquilar su teoría a través de su crítica, sin ningún fatalismo asociado. En la ciencia, debemos permitir que nuestras hipótesis mueran en nuestro lugar".*
> — Karl Popper, La Responsabilidad de Vivir, 1995.

Interpretando esta famosa frase de Popper, desde una perspectiva casi literal, un agente puede verse a sí mismo "falleciendo" miles de veces en una simulación como resultado de planes infructuosos. Como consecuencia, el agente eliminará dichos planes sin arriesgar su integridad en el mundo real y dará prioridad a aquellos planes que probablemente puedan representar un mayor beneficio.

Cambiando de perspectiva, existe evidencia en el área de la neurofisiología (Clark y Grush, 1999; Hesslow, 1994; Hesslow, 2002) que indica que las estructuras neuronales involucradas en la *percepción* y en la iniciación del *comportamiento explícito* son también responsables del *comportamiento implícito* y la *imaginería mental*[2]. Como se ilustra en la figura 10.4.a, el comportamiento explícito de un agente cuando interactúa con el ambiente puede ser descrito como una secuencia de parejas estímulo-respuesta. En la situación inicial, un estímulo (E) dispara una respuesta (R), la cual cambia el estado del ambiente (desde la perspectiva del agente) en una nueva situación (E'), que actúa como un nuevo estímulo que dispara otra respuesta (R'), que a su vez causa E'' y R'', y así sucesivamente. Todos estos pasos se desarrollan mediante un proceso interno donde los estados sensoriales (e) causan respuestas (r).

La hipótesis neurofisiológica sobre el proceso de simulación interna (Hesslow, 1994; Hesslow, 2002) asume tres mecanismos que explican dicho fenómeno.

1. El primer supuesto es el *Comportamiento Implícito*, e.g., la habilidad para generar respuestas neuro-motrices que no se convierten en acciones corporales observables sino que se mantienen sólo como patrones de activación neuronales, de naturaleza totalmente interna.

[2]La imaginería mental es una actividad cognitiva que permite percibir algo en su ausencia. No es exclusivamente visual y, de hecho, nos permite crear imágenes, sonidos, olores y sensaciones en nuestra mente.

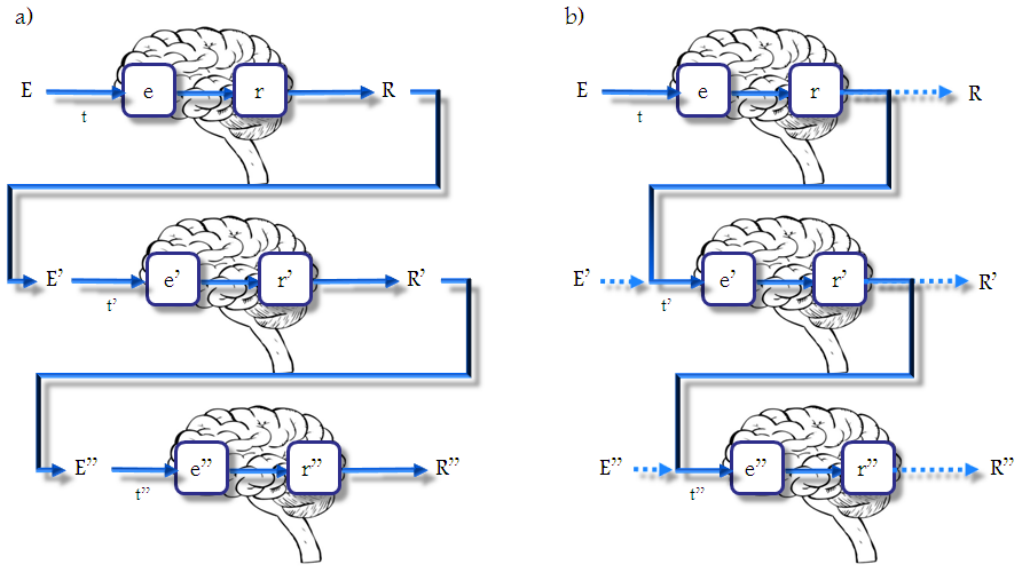


Figura 10.4: Actuación Real vs. Actuación Simulada. a) Secuencias de Estímulos-Respuestas durante el Comportamiento Explícito. E se refiere al estímulo o situación, e a la actividad neuro-sensorial, r a la actividad neuro-motriz, y R a la respuesta del comportamiento. b) Secuencia de interacciones sensoriomotrices internamente simuladas.

2. Se asume la existencia de una *Reactivación de Sensores* o mecanismo de imaginería. Este mecanismo permite la activación interna de las cortezas sensoriales del cerebro, de manera tal que se produce una *"experiencia simulada"* del estímulo, pero sin la presencia de ningún estímulo externo.
3. Finalmente, se asume la existencia de un *Mecanismo de Anticipación* capaz de predecir o simular las consecuencias sensoriales de una respuesta motriz (e.g., un movimiento corporal).

Cada uno de estos supuestos es soportado por la literatura en el área de la neurociencia. El comportamiento implícito y la imaginería (reactivación de sensores) han sido discutidos, por ejemplo, por (Jeannerod, 1994), y en trabajos posteriores, la imaginería por (Kosslyn et al., 1995). Finalmente, existe evidencia suficiente para considerar que el cerebelo es el candidato principal involucrado en la predicción y la anticipación (Miall y Wolpert, 1996; Miall, 1997; Thach, 1998; Wolpert et al., 1998).

Tomando en cuenta estos tres supuestos (comportamiento implícito, reactivación de sensores, y anticipación), resultaría posible *simular internamente* el comportamiento explícito que se ilustra en la parte a) de la figura 10.4, como un proceso de recreación de eventos interno, como se observa en la parte b) de la misma figura. En la situación inicial, un estímulo (E) causa una activación sensorial (e) la cual dispara la elaboración de una respuesta motriz (r). Pero en lugar de causar la respuesta explícita (R), la elaboración de la respuesta motriz podría causar una nueva activación sensorial (s') a través del mecanismo de anticipación de consecuencias y del mecanismo de reactivación sensorial. Esta

nueva activación (interna) de los sensores dispara otra elaboración de respuesta motriz (r'), la cual a su vez causa una nueva activación sensorial (s''), que dispara una elaboración de respuesta motriz (r''), y así sucesivamente. Así pues, en lugar de interactuar directamente con el ambiente, el agente simula internamente esta interacción.

Antes de describir el proceso de simulación interna conviene recordar grosso modo el proceso realizado en un ciclo cognitivo típico. En la figura 10.5 se presenta la ejecución de un ciclo cognitivo de manera simplificada. Resumiendo los pasos del ciclo cognitivo se tiene que: (1) el agente recibe una entrada sensorial del mundo percibido; (2) la MT comprueba la validez de la información almacenada en ese instante que forma el "contexto" (i.e., los perceptos, las metas, los motivadores, etc.); (3) se activan los mecanismos de razonamiento de las memorias declarativas (semántica y episódica) los cuales infieren información adicional sobre los perceptos; (4) el módulo de Atención selecciona la MAT que se hará con el control consciente del agente. La MAT activa la RC correspondiente y ésta a su vez activa un *Comportamiento* como resultado de ejecutar las dinámicas de dispersión de energía; (5) el *Comportamiento* elegido, activa los diferentes componentes (RP, RES, SIA y PMR) y se genera una acción integrada; (6) el módulo de Prospección y Anticipación genera una nueva predicción (anticipación) del estado del ambiente a partir de la acción integrada; (7) la acción integrada es enviada al módulo Efector donde éste activa los actuadores correspondientes y ejecuta la acción en el ambiente; y (8) la información del estado actual (unidades-MT) es actualizada por la Memoria de Trabajo.

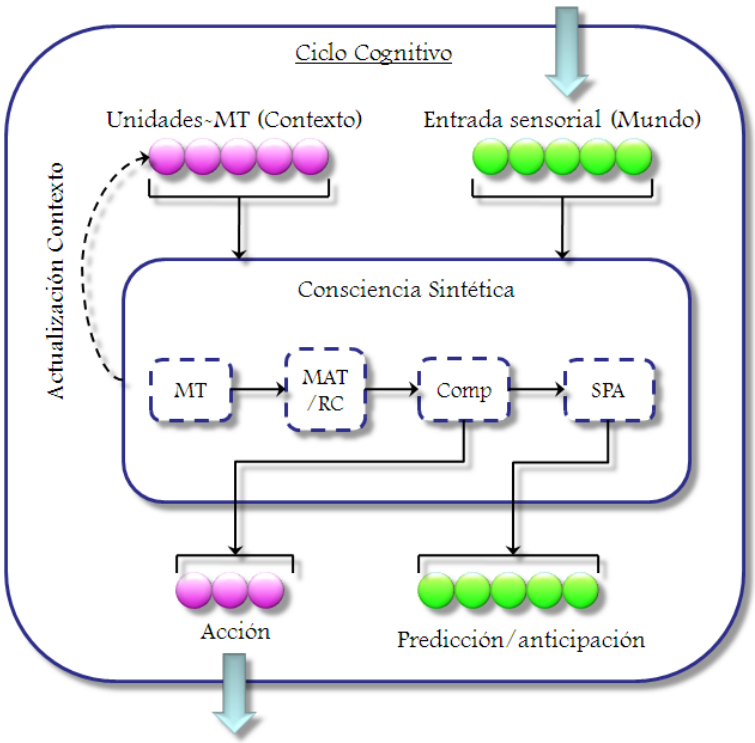


Figura 10.5: Ciclo Cognitivo simplificado que recibe entradas sensoriales del mundo.

Ahora, tomando como referencia la hipótesis sobre el fenómeno de la simulación interna descrita anteriormente, es posible proponer un mecanismo conformado por múlti-

ples ciclos cognitivos, cada uno de los cuales genera una predicción o anticipación del siguiente estado más probable del sistema, y donde dicha predicción sirve como nueva entrada sensorial al siguiente ciclo cognitivo. Obsérvese la figura 10.6.

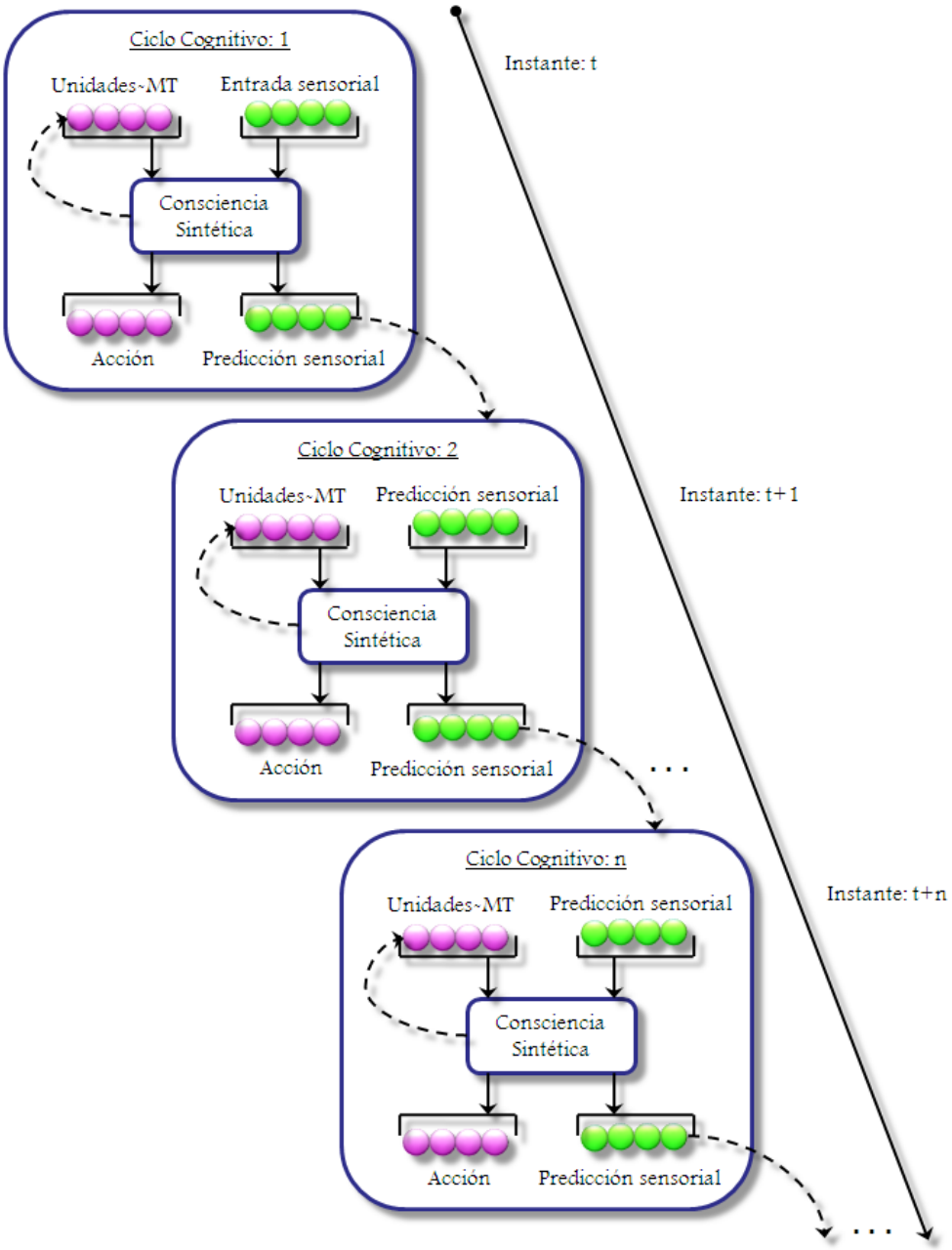


Figura 10.6: Mecanismo de Simulación Interna. En lugar de recibir una lectura sensorial del mundo percibido, cada ciclo cognitivo emplea la predicción generada en el ciclo anterior y la procesa como entrada sensorial. Las flechas punteadas representan la actualización de la información en cada ciclo cognitivo.

Como se ha mencionado anteriormente, el proceso de simulación interna le permite al agente determinar el grado de beneficio o perjuicio que produciría la ejecución de un plan en el ambiente. Por lo tanto, en la fase de planificación deliberativa cada agente ejecutará, en paralelo, un hilo de procesamiento que simulará la secuencia de tareas definidas por cada plan. Por tanto, es necesario determinar algún criterio de paro que establezca cuándo detener cada hilo de simulación. A continuación se proponen algunos de estos criterios:

1. *Gana el Primero en Terminar*: en el caso de establecerse como función objetivo la identificación del plan cuya simulación requiera menor cantidad de tiempo de ejecución o menor cantidad de ciclos cognitivos, se define entonces como criterio de paro la detención de todos los hilos restantes de la simulación y la ejecución del plan ganador (el que ha tardado menos tiempo/ciclos cognitivos en alcanzar la meta final). Este criterio puede ser empleado en las concreciones contextuales de la arquitectura donde la optimización del plan se base en la minimización del recurso "tiempo" o "duración" de una actividad, como por ejemplo, calcular la ruta mínima (número de desplazamientos mínimos) para visitar un conjunto de puntos de destino (e.g., el problema del vendedor viajero constreñido a la variable tiempo/distancia).
2. *Máxima Incertidumbre*: este criterio determina si existe información necesaria y suficiente para predecir el siguiente estado del sistema. Esto es, si en un determinado ciclo cognitivo en la simulación de un plan, se cumple para el módulo de Prospección y Anticipación que: (1) encuentra un clasificador que empareja con el estado actual simulado; y (2) el clasificador activado tiene una calidad de predicción que supera cierto umbral (por defecto $umbral_{sim} = 0.4$), lo cual significa que la anticipación se ha realizado con determinado grado de certeza; entonces el proceso de simulación continua con el siguiente ciclo cognitivo, de lo contrario, el hilo de simulación se detiene.
3. *Meta Final Alcanzada:* este criterio determina la detención de un hilo de simulación sólo cuando se ha alcanzado el estado final o meta que promete lograr el plan después de su ejecución. Este criterio se emplea, a diferencia del criterio 1, cuando se desea obtener el plan que representa el máximo beneficio para el agente. El proceso de selección del plan que representa el máximo beneficio será explicado al detalle en la sección 10.3.4.
4. *Máximo Número de Pasos*: en el caso de que la simulación de un plan tome más de cierto número máximo de pasos (ciclos cognitivos) entonces se descarta dicho plan como una solución favorable. La definición de este criterio depende del dominio del problema.

La decisión de escoger alguno de estos criterios depende principalmente del desempeño computacional del proceso ontogenético. El desempeño del proceso de confección de los planes está bajo la influencia del tamaño del repertorio (población) de planes. Un tamaño grande para el repertorio de planes garantiza una gran diversidad genética, lo que se asocia con una población robusta de planes que puede sobrevivir a intensos eventos de selección, mientras que, una baja diversidad genética (cuello de botella o consaguinidad) puede reducir la adaptabilidad y aumentar la posibilidad de extinción (en especial, las consecuencias pueden ser desastrosas si se extinguen planes potencialmente "buenos").

Supóngase, por ejemplo, que un tamaño aconsejable que mantiene la diversidad del repertorio sería de aproximadamente 100 planes, sin embargo, el problema radicaría en que la simulación interna y concurrente de 100 planes requeriría de grandes cantidades de procesamiento y memoria, perjudicando enormemente el desempeño del proceso de planificación. Por tanto, en este caso resultaría conveniente emplear el criterio 1 (*Gana el Primero en Terminar*) con el cual se garantizaría un acotamiento del espacio de búsqueda y por ende una disminución considerable en el tiempo de procesamiento. El inconveniente de este enfoque es que probablemente se descartarían planes que prometen alcanzar una mayor cantidad de beneficio total (máximos globales).

Por otro lado, si se escoge un tamaño más pequeño para el repertorio de planes, por ejemplo, entre 10 y 20 planes, podría emplearse un enfoque diferente donde se simularían todos los planes y luego se comprobaría cuál de ellos promete el mayor beneficio total. Para esto, se emplearía una combinación de los criterios 2, 3 y 4 (*Máxima Incertidumbre*, *Meta Final Alcanzada*, y *Máximo Número de Pasos*) para determinar cuándo detener la simulación de cada uno de los hilos. Este enfoque por un lado, soporta la búsqueda de un máximo global al evaluar exhaustivamente cada plan del repertorio, aunque por otro lado, sacrifica la diversidad del repertorio de planes, lo que puede causar una acelerada convergencia en un máximo local.

La decisión de cuál criterio emplear en cada concreción contextual de la arquitectura, depende del dominio del problema y de las características técnicas de las plataformas que se empleen para ejecutar los agentes cognitivos.

En los cuadros 10.2 y 10.3, se presentan los algoritmos empleados en cada hilo de simulación interna.

Método: razonar

```
1.  Variables
2.     unidadesMT     ↑↓ unidades activas en la memoria de trabajo.
3.     entrada        ↑ entrada sensorial del ciclo cognitivo actual.
4.     accion         ↑ acción integrada propuesta por el comportamiento activo.
5.  Inicio Procedimiento
6.     unidadesMT ← mt.comprobarValidezUnidadesMT()
7.     perceptos ← percepcion.construirPerceptos(entrada)
8.     perceptos'← memoriaDeclarativa.inferir(perceptos)
9.     unidadesMT ← mt.agregarUnidades(perceptos')
10.    mat ← atencion.seleccionarMAT(unidadesMT)
11.    rc ← mat.obtenerRC(unidadesMT)
12.    comp ← rc.seleccionarComportamiento(unidadesMT)
13.    listaAcciones ← comp.recomendarAcciones(unidadesMT)
14.    accion ← efector.integrarAcciones(listaAcciones)
15.    unidadesMT ← mt.agregarUnidades(accion)
16.    retornar accion
17. Fin Procedimiento
```

Algoritmo 10.2: Algoritmo global que procesa un ciclo cognitivo básico.

Método: activarSimulacionInterna

```
1.  Variables
2.     planes          ↑↓ conjunto de planes generados por el sistema.
3.  Inicio Procedimiento
4.     entrada ← percibirEntradaSensorial()
5.     encontradoMejorPlan ← false
6.     continuarSimu ← true //cada plan define una variable de este tipo.
7.     Mientras que !encontradoMejorPlan hacer:
8.        Para cada plan hacer:
9.           Si continuarSimu = true entonces
10.             accion ← razonar(entrada)
11.             entrada ← SPA.predecirNuevoEstado(entrada,
                accion) //se activa el Sistema de Prospección y Anticipación.
12.             Si criterio1 = true entonces
13.                encontradoMejorPlan ← validarPrimeroTerminar(SPA)
14.             Sino
15.                Si criterio3 = true entonces
16.                   encontradoMejorPlan ← validarMetaAlcanzada(SPA)
17.                Sino
18.                   Si criterio2 = true entonces
19.                      continuarSimu ← validarMaxIncertidumbre(SPA)
20.                   Fin Si
21.             Fin Si
22.          Fin Si
23.       Fin Para
24.    Fin Mientras
25.    Fin Procedimiento
```

Algoritmo 10.3: Algoritmo para el Mecanismo de Simulación Interna.

10.3.4. Selección y Ejecución del Mejor Plan

En la sección 10.3.2 se discutieron los operadores genéticos que actúan sobre el repertorio de planes permitiendo la exploración y explotación del espacio de búsqueda. Sin embargo, estos operadores actúan sólo sobre un conjunto específico de planes que ha sido determinado a través de un proceso de selección estocástica (en analogía con el proceso de selección natural). Este proceso de selección evalúa ciertas características de cada uno de los planes simulados y determina la "aptitud" correspondiente. Como ya se ha mencionado en capítulos anteriores donde se ha explicado el funcionamiento de una Máquina Darwiniana, la aptitud de una solución mide el nivel de desempeño que ha tenido ésta durante el transcurso del tiempo, proveyendo una métrica para determinar la probabilidad con la que será replicada en la siguiente generación.

De esta manera, la aptitud de cada plan es una medida acumulativa que determina cuán apropiado ha sido el plan en la consecución de una meta final. La aptitud de cada plan es evaluada en dos fases: (1) después de la planificación fuera de línea (off-line) donde interviene la simulación interna; y (2) durante la planificación en línea (on-line). Estas fases de evaluación se describen a continuación.

10.3.4.1. Evaluación de la Aptitud en la Planificación Fuera de Línea

Una vez se ha finalizado la simulación de un plan[3], se procede a estimar el beneficio o perjuicio que ocasionaría la ejecución de dicho plan en el ambiente. Así pues, la evaluación de la aptitud de cada plan depende de los siguientes factores:

Duración: este factor determina la duración de la simulación del plan como la suma de los tiempos de respuesta de todos los ciclos cognitivos (en unidades de milisegundos).

$$\textit{Duración}_{fl} = \begin{cases} \sum_{i=0}^{n} ck_{(i,t)} - ck_{(i,t-1)}, & \text{si } (ck_{(i,t)} \vee ck_{(i,t-1)} \in unidadesMT) \\ \sum_{i=0}^{n} TR_i, & \text{de lo contrario} \end{cases} \tag{10.5}$$

donde: n es el número total de ciclos cognitivos empleados en la simulación del plan. $ck_{(i)}$ y $ck_{(i-1)}$ son dos chunks de la memoria episódica que se encuentran activos en la memoria de trabajo. TR es el tiempo de respuesta del ciclo cognitivo.

La idea básica de cómo se estima el tiempo de duración de un ciclo cognitivo es la siguiente: por cada ciclo cognitivo simulado, el mecanismo de inferencia del módulo Declarativo extrae aquellos chunks episódicos que han sobrepasado el umbral de activación. Estos chunks representan los "recuerdos temporales" que tiene el agente sobre situaciones pasadas que resultan similares a la situación simulada. Por tanto, la duración de cada ciclo cognitivo se estima como la diferencia entre el registro temporal del ciclo cognitivo actual menos el registro temporal del ciclo cognitivo anterior (si existe), esto es, $ck_{(t)} - ck_{(t-1)}$. En el caso de que no se disponga de dicha información (e.g., ningún chunk episódico supere el umbral de activación), entonces se emplea una estimación del tiempo de respuesta del sistema, es decir, TR. El proceso de estimación de los tiempos de respuesta se discutirá en la sección 10.5.

Coste: este factor estima el coste que supondría ejecutar el plan en términos de la cantidad de recursos y energía empleada por el agente en cada ciclo cognitivo.

$$Coste_{fl} = \sum_{i=0}^{n} \sum_{j=0}^{r} E_{i,j} \tag{10.6}$$

[3]Téngase en cuenta que la evaluación de la aptitud en la planificación fuera de línea sólo se aplica en el caso de que "no" se esté empleando el criterio de paro de *Gana el Primero en Terminar* para la simulación. Esto se debe a que se requiere que todos los planes finalicen la simulación para poder evaluar su nivel de aptitud.

donde: n es el número total de ciclos cognitivos simulados. r es la cantidad de recursos o actuadores que emplea el agente para llevar a cabo el ciclo cognitivo i. $E_{i,j}$ es la cantidad de energía requerida por el recurso (actuador) j en el ciclo cognitivo i.

Por ejemplo, suponiendo que el agente es un robot que tiene que mover un conjunto de cajas desde una posición a otra de una habitación, la función del coste para cada ciclo cognitivo dependería del número de actuadores (e.g., el motor de desplazamiento, el motor de rotación, o la pinza con la que sujetas las cajas) involucrados en la acción resultante del proceso de toma de decisiones, y de la cantidad de energía que implicaría activar cada uno de dichos actuadores. La función de energía debe ser definida para cada dominio del problema. Por último, es importante aclarar que el $Coste$ es un término opcional en la estimación de la aptitud del plan, y sólo se incluirá en el caso de que se disponga de una función de energía.

Refuerzo Acumulado: este factor estima la cantidad de retroalimentación neta que acumularía el agente de llevarse a cabo el plan, en términos de la suma de los beneficios (refuerzos positivos) menos la suma de los perjuicios (refuerzos negativos) acumulados en cada ciclo cognitivo.

$$Refuerzo_{fl} = \sum_{i=0}^{n} RA_i \tag{10.7}$$

donde: RA es la cantidad de refuerzo acumulado (refuerzos positivos menos los refuerzos negativos) obtenido en cada ciclo cognitivo i.

El refuerzo acumulado (RA) se calcula a partir de los refuerzos que reciben los componentes involucrados en la ejecución de un ciclo cognitivo, como se discutió en la sección 10.2.

$$RA_i = RA_{(i,Comp)} + RA_{(i,MAT)} \tag{10.8}$$

donde: i es el ciclo conitivo actual, $RA_{(i,Comp)}$ es el refuerzo neto estimado que recibe el *Comportamiento* activo después de ejecutar la acción en el ambiente simulado. $RA_{(i,MAT)}$ es el refuerzo que recibe la MAT activa.

El refuerzo $RA_{(i,Comp)}$ es una predicción que hace el Sistema de Prospección y Anticipación de la cantidad de retroalimentación (positiva o negativa) que recibirá el *Comportamiento* después de ejecutarse la acción (como se explicó en la sección 8.6.2). Por otro lado, el refuerzo $RA_{(i,MAT)}$ corresponde a la evaluación de las funciones de aptitud FAAE y FAAF para la MAT activa (como se explicó en la sección 9.3.2.1).

Aptitud del Plan: finalmente, la función de la aptitud fuera de línea para el plan simulado (p) es calculada de la siguiente manera:

$$FAFL_p = \left(k1 \cdot Refuerzo_{(fl,p)}\right) \times \left(\frac{k2}{Coste_{(fl,p)}}\right) \times \left(\frac{k3}{\textit{Duración}_{(fl,p)}}\right) \quad (10.9)$$

donde: $k1$, $k2$, y $k3$ son factores que ajustan cada uno de los términos. Por ejemplo, por defecto se define $k3 = 1000$ con el fin de ajustar las unidades del tiempo estimado a segundos, en lugar de milisegundos. Estos factores deben ser definidos para cada concreción contextual de la arquitectura.

Como se puede observar, el $Refuerzo$ es inversamente proporcional a la *Duración* y al $Coste$ estimados. Esto con el objetivo de evitar situaciones donde la simulación acumule demasiados refuerzos debido a la ejecución monótona de una tarea. Por ejemplo, tomando como referencia el ejemplo anterior, si el robot aprendiera a recibir un refuerzo positivo por cada vez que levantase una caja de forma apropiada, es posible que entrará en un bucle monótono donde "soltase" y "levantase" continuamente la misma caja. Por tal motivo, la aptitud del plan no sólo mide la cantidad máxima de refuerzos positivos acumulados durante la simulación, sino la cantidad mínima de tiempo y costes que ha invertido el agente en ello.

10.3.4.2. Evaluación de la Aptitud en la Planificación En Línea

De manera similar al proceso de evaluación fuera de línea descrito anteriormente, la evaluación en línea valida el plan en curso en función de tres variables: la duración, el coste y el refuerzo acumulado. La evaluación del coste emplea la misma ecuación propuesta para la evaluación fuera de línea (ecuación 10.6), mientras que la evaluación de la duración y el refuerzo acumulado son ligeramente diferentes.

Duración: este factor determina la duración del plan en curso, en cada instante, como la suma de los tiempos de respuesta de todos los ciclos cognitivos ejecutados hasta ese momento (en unidades de milisegundos).

$$\textit{Duración}_{el} = \sum_{i=0}^{n} TR_i \quad (10.10)$$

donde: n es el número de ciclos cognitivos empleados por el plan hasta el instante de la evaluación. TR es el tiempo de respuesta para cada ciclo cognitivo determinado a través de un *temporizador*.

Refuerzo Acumulado: este factor determina la cantidad de retroalimentación neta que acumula el agente durante la ejecución del plan, en términos de la suma de los beneficios

(refuerzos positivos) menos la suma de los perjuicios (refuerzos negativos) acumulados en cada ciclo cognitivo.

$$Refuerzo_{el} = \sum_{i=0}^{n} RA_{(i,Comp)} + RA_{(i,MAT)} \tag{10.11}$$

donde: $RA_{(i,Comp)}$ es el refuerzo ambiental neto que recibe el *Comportamiento* activo después de ejecutar la acción en el ambiente. $RA_{(i,MAT)}$ es el refuerzo que recibe la MAT activa por la evaluación de su FAAF.

Aptitud del Plan en ejecución: finalmente, la función de la aptitud en línea para el plan en ejecución (p) es calculada de la siguiente manera:

$$FAEL_p = \left(k1 \cdot Refuerzo_{(el,p)}\right) \times \left(\frac{k2}{Coste_{(el,p)}}\right) \times \left(\frac{k3}{\textit{Duración}_{(el,p)}}\right) \tag{10.12}$$

donde: $k1$, $k2$, y $k3$ son factores que ajustan cada uno de los términos. Estos factores deben ser definidos para cada concreción contextual de la arquitectura.

10.3.4.3. Proceso de Selección y Ejecución de Planes

Una vez se han simulado todos los planes confeccionados por la Máquina Darwiniana, el mecanismo de Planificación procede a seleccionar, para su posterior ejecución, el mejor plan que ha resultado de la simulación (siempre y cuando el mejor plan seleccionado cumpla con ciertos criterios, o de lo contrario, se reinicia el proceso ontogenético). El proceso se detalla a continuación:

1. La *Estructura de Metas* determina la meta que tiene mayor prioridad sobre las demás (i.e, aquella que posee una mayor activación de nivel base - ANB), o una meta más global que contiene a las otras.
2. Se carga la *librería de planes*[4], si existe, o se genera aleatoriamente un nuevo repertorio de planes. Luego, se ejecuta una generación del proceso ontogenético sobre dicho repertorio de planes como se discutió en la sección 10.3.2.
3. Se inicia un hilo de procesamiento concurrente por cada plan generado por el proceso ontogenético, el cual es simulado internamente por el agente (como se describió en el algoritmo 10.3 de la sección anterior).

[4]La librería de planes es un repositorio donde son almacenados aquellos planes que han sido simulados y probados previamente por el agente. Para que un plan pueda ser almacenado en la librería debe haber demostrado un buen desempeño en experiencias pasadas del agente, lo cual se determina mediante una medida de calidad (e.g., cuando el nivel de aptitud del plan supera cierto umbral) que debe ser especificada para cada concreción contextual de la arquitectura. Estos planes pueden ser luego utilizados como mini-planes reactivos que definen una secuencia de acciones la cual no requiere nuevamente de un procesamiento deliberativo para su ejecución.

4. Una vez finaliza la simulación interna de cada plan:

   - Se evalúa la función de aptitud fuera de línea (FAFL) y se actualiza el valor de la aptitud del plan simulado (ps): $aptitud_{ps} = FAFL(ps)$.
   - Es posible que la simulación de un plan haya finalizado sin que éste haya logrado la meta final (e.g., en el caso de que la simulación se detenga debido a alguno de los criterios de paro: *Máxima Incertidumbre*, *Máximo Número de Pasos*, o *Gana el Primero en Terminar*). Por tanto, se valida si el plan simulado promete lograr la meta final para la cual ha sido diseñado. En el caso de que el plan prometa lograr la meta, se propone como un plan ejecutable, de lo contrario, el plan es descartado para su ejecución (aunque no es eliminado del repertorio de planes, con lo cual puede participar en futuras generaciones del proceso ontogenético). Finalmente, se termina el hilo de ejecución para dicha simulación.

5. Después de finalizada la simulación de todos los planes, se procede a seleccionar el mejor plan, esto es, el plan que ha obtenido la mayor aptitud después de la evaluación de la FAFL:

$$mejorPlan_i = \max_j(FAFL_{(j,i)}) \tag{10.13}$$

   donde i es la generación actual de planes producida por el proceso ontogenético, y j es el número total de planes simulados en cada generación.

6. A continuación, se valida si la aptitud del mejor plan seleccionado supera cierto umbral mínimo requerido: $aptitud_{mejorPlan} > umbral_{mp}$. La idea principal de esta validación es garantizar que el mejor plan seleccionado para su ejecución haya recolectado, durante la simulación, una cantidad mínima de beneficio. En este caso, se buscar optimizar (maximizar) el beneficio total acumulado durante la simulación. Adicionalmente, existe otro criterio "opcional" que busca minimizar el tiempo que puede tardar la ejecución del plan: $\textit{duración}_{mejorPlan} < t_{min}$ y cuya especificación depende del dominio del problema.

7. Si el plan no cumple con los criterios antes mencionados, se repite de nuevo el proceso ontogenético, es decir, se reinicia la MD y se aplican nuevamente los operadores genéticos sobre el repertorio actual de planes, produciendo como resultado, una generación de planes mejor adaptados. El proceso puede repetirse tantas veces como se requiera (desde el paso 2), o hasta que se cumpla un número máximo (definido a priori) de bucles de activación del proceso ontogenético.

8. Por otro lado, si el plan cumple con los criterios para ser ejecutado, entonces es activado y desempeñado por el agente. Durante la ejecución del plan (concretamente, en cada ciclo cognitivo ejecutado), se evalúa la aptitud de dicho plan (empleando la FAEL), con el objetivo de corregir, en tiempo de ejecución, la realización del plan.

   El plan en ejecución se mantiene activo sólo si cumple con los siguientes criterios en cada ciclo cognitivo:

$$(aptitud_{pe} > umbral_{pa}) \wedge (meta_{pe} \in meta_g(cca)) \tag{10.14}$$

Esto quiere decir, por un lado, que la aptitud del plan en ejecución (pe) debe mantenerse sombre un umbral mínimo ($umbral_{pa}$), el cual varía para cada concreción contextual de la arquitectura, con el objetivo de garantizar que el plan se mantenga activo; y, por otro, que la meta final del plan en ejecución (meta para la cual fue confeccionado el plan), debe ser una de las metas globales que se encuentra activa en la Estructura de Metas, durante el ciclo cognitivo actual (cca).

En el caso de que se cumplan estos dos criterios, la ejecución del plan se mantendrá vigente, de lo contrario, se finaliza el proceso de ejecución del plan (pe) y se procede a una *"re-planificación"* que estimará un nuevo plan que prometerá recaudar una mayor cantidad de beneficios netos (reflejados en la aptitud acumulada del plan), o prometerá alcanzar una meta global prioritaria del agente, o bien las dos condiciones anteriores.

10.3.5. Modelo Dinámico para el Mecanismo de Confección de Planes

La figura 10.7 ilustra, de manera simplificada, el diagrama de actividades para el Mecanismo de Confección y Validación de Planes. De este diagrama cabe destacar que una vez se selecciona el mejor plan, su ejecución estará condicionada, por un lado, a que su aptitud se mantenga por encima de un umbral $umbral_{pa}$ y, por otro, a que la meta global que promete alcanzar dicho plan permanezca como la meta activa prioritaria del sistema. En el caso de que alguna de estas dos validaciones (realizadas al final de cada ciclo cognitivo) no se cumpla, el sistema procede a activar el mecanismo de *re-planificación*, el cual reinicia la secuencia de actividades.

Por otro lado, la figura 10.8 ilustra el flujo de control que determina si el agente exhibirá una conducta *deliberativa a largo plazo* (planificadora), *deliberativa a corto plazo* (racional), o *reactiva*. En el caso de que se encuentre activa la conducta planificadora, el agente activará los procesos de planificación y simulación interna a fin de hallar una secuencia ordenada de tareas de alto nivel que deben llevarse a cabo para satisfacer un conjunto de metas globales. Por otro lado, si no se encuentra activa la conducta planificadora, el agente podrá exhibir dos tipos de actuación: *la deliberativa a corto plazo* donde la consciencia sintética del agente determinará cuál es el *Comportamiento* más adecuado para alcanzar una sub-meta específica, y la *reactiva* donde el agente manifestará una conducta de estímulo-respuesta frente a un estado observado.

10.4. Construcción y Propagación de Unidades de Conocimiento: Una Solución Basada en Memética

En esta sección se expone, por un lado, el mecanismo que le permite a cada agente construir unidades de conocimiento evolutivamente manipulables (memes) mediante los procesos epigenéticos, ontogenéticos, y filogenéticos; y, por otro, el mecanismo que permite trasmitir socialmente (a través del sistema multi-agente) dichas unidades de conocimiento.

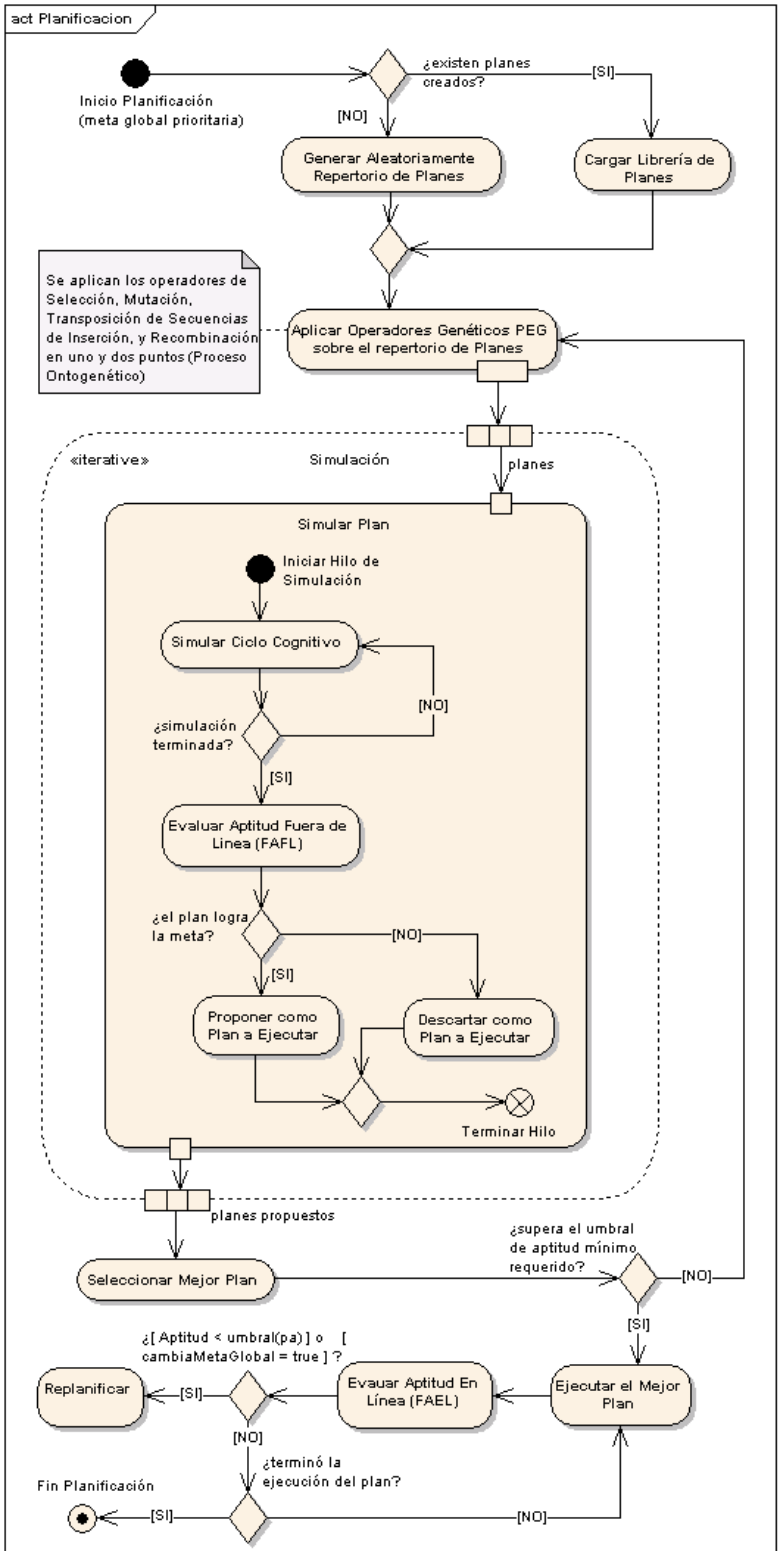


Figura 10.7: Diagrama de Actividades para el Mecanismo de Planificación.

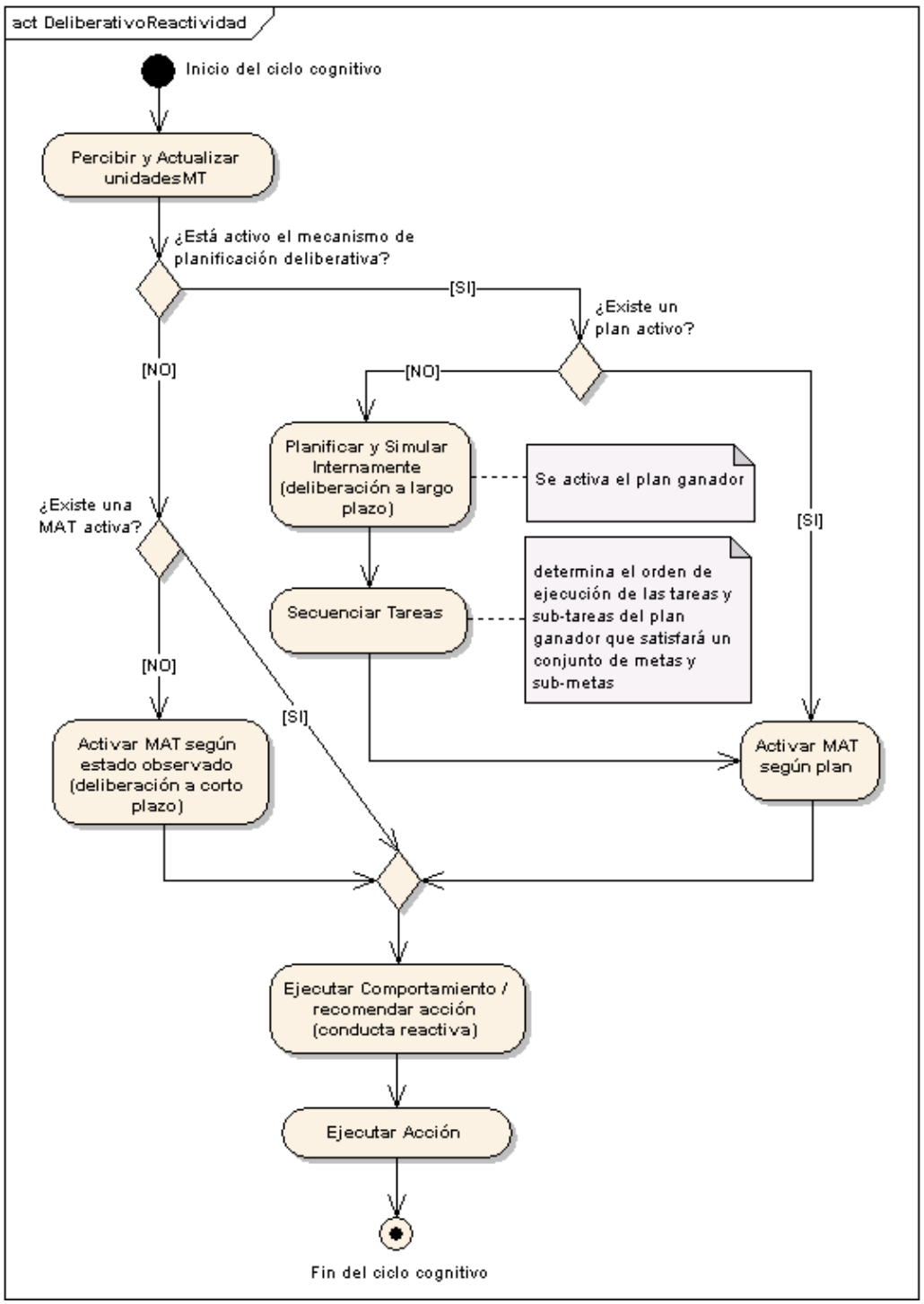


Figura 10.8: Diagrama de Actividades que muestra el flujo de control entre los procesos deliberativos vs. los procesos reactivos.

Antes de abordar el tema que compete a esta sección, se precisa cierta terminología que resulta necesaria para condensar o distinguir algunos conceptos que hasta el momento se habían tratado indistintamente.

Época de Aprendizaje: es un intervalo de tiempo, medido en términos de un número específico de ciclos cognitivos transcurridos, durante el cual el agente aprende (a partir de la interacción con su entorno) a desempeñar una tarea concreta. Durante una época de aprendizaje, el nivel de *aptitud* de las estructuras cognitivas involucradas puede incrementar o disminuir de acuerdo con la capacidad de la estructura cognitiva para identificar pautas de comportamiento.

Población de Individuos: en términos de la *computación evolutiva*, este concepto hace referencia al conjunto de representaciones o soluciones potenciales a un problema dado, en un instante de tiempo determinado. Es decir, las poblaciones de individuos suelen ser dinámicas, y cambian la información genética de sus individuos a través de la aplicación de operadores genéticos específicos. En METÁFORA, cada población de individuos depende del dominio sobre el cual opera la Máquina Darwiniana (MD) respectiva. Así por ejemplo, la población de individuos para la MD del componente SIA (Sistema Inmune Artificial) corresponde al repertorio de anticuerpos, para la MD del componente SA (Sistema Clasificador Anticipatorio) corresponde al repertorio de clasificadores, para la MD del mecanismo de generación de RCs (Redes de Comportamientos) corresponde al repertorio de cromosomas multigénicos, y para la MD del mecanismo de generación de Planes corresponde al repertorio de células.

Generación: se considera una fase evolutiva en la que se produce una población modificada de individuos (descendientes), como consecuencia de aplicar un conjunto de operadores genéticos específicos. Cabe resaltar entonces, que una generación puede darse en diferentes escalas temporales y en diferentes dominios, dependiendo de la MD a la cual se esté haciendo mención. Concretamente, se tiene la siguiente diferenciación:

- *Generación Epigenética:* esta generación actúa en la escala temporal más corta (medida en milisegundos o en una época de aprendizaje), y hace referencia a cada generación evolutiva de las MDs para los componentes SIA y SA.
- *Generación Ontogenética:* esta generación se desempeña en una escala temporal más prolongada que la anterior (medida en segundos o en un número específico épocas de aprendizaje), y se refiere a cada paso evolutivo donde son producidas, por medio de las MDs respectivas, nuevas y mejores estructuras cognitivas complejas como las RCs y los Planes. Esto bajo el supuesto de que los procesos evolutivos para las RCs y los Planes se ejecutan al mismo tiempo, de lo contrario, resultaría necesario hacer alguna distinción entre estos dos tipos de generaciones (e.g., Generación Ontogenética-RC y Generación Ontogenética-Plan)
- *Generación Filogenética:* esta generación actúa en la escala temporal más larga (medida en minutos/horas o en un número específico de épocas de aprendizaje), y hace referencia a cada ciclo evolutivo completo de la MD que produce nuevos repertorios co-adaptados y mejorados de memes.

Estructura Cognitiva/Individuo/Meme: estos tres términos hacen referencia a un mismo concepto: una unidad de conocimiento que puede ser transmitida y que es susceptible de ser evolucionada a través de los distintos procesos EOF (epigenéticos, ontogenéticos y filogenéticos). Hasta el momento, siempre se ha empleado el término "estructura cognitiva" para designar, de forma general, el concepto subyacente. Sin embargo, debido

a que los procesos evolutivos de METÁFORA siempre actúan sobre poblaciones de estructuras cognitivas, el término "individuo" tiene entonces equivalencia con el de estructura cognitiva. Por otro lado, debido a que cada estructura cognitiva posee una codificación genética (en términos de la computación evolutiva) la cual puede replicarse, recombinarse, y variar y, como consecuencia, modificar su estructura fenotípica, conviene entonces reemplazar el término "estructura cognitiva" por el de "meme".

10.4.1. Interacción entre la Epigénesis, la Ontogénesis y la Filogénesis

Hasta el momento se han definido, por un lado, los procesos *Epigenéticos* encargados de dirigir el aprendizaje de nuevo conocimiento procedimental y declarativo y, por otro, los procesos *Ontogenéticos* comprometidos en el desarrollo de estructuras cognitivas complejas como las RCs y los Planes. Estos dos procesos de desarrollo individual operan en un nivel *"intra-agente"* a través de múltiples procesos micro-evolutivos que dirigen las diferentes Máquinas Darwinianas. Ahora, resulta imprescindible pasar a un nivel de especificación *"inter-agente"* donde las diversas estructuras cognitivas desarrolladas individualmente puedan ser modificadas mediante un proceso macro-evolutivo que actúe a nivel poblacional[5]: esta es la función esencial del proceso *Filogenético*.

La filogénesis en biología cumple la función de determinar la historia evolutiva de las *especies*. Ajustando esta metáfora al proceso filogenético de la arquitectura, cada especie se corresponde con un tipo específico de estructura cognitiva que es susceptible al proceso de *"evolución"*. El proceso filogenético crea la línea base sobre la cual se efectúan los desarrollos epigenéticos y ontogenéticos subsecuentes. Es decir, en cada paso del proceso filogenético se establecen las diferentes estructuras de conocimiento innato con las cuales estará dotado cada agente, pero es a través de la epigénesis y de la ontogénesis que el agente adaptará su conducta global a las circunstancias y requisitos cambiantes del ambiente, lo cual implica que cada agente producirá un comportamiento adaptativo influido por su interacción con el entorno.

En este punto resulta importante resaltar tres aspectos: (1) los procesos EOF – epigenéticos, ontogenéticos, y filogenéticos – asumen la responsabilidad de *desarrollar* y *evolucionar* únicamente las estructuras cognitivas de los agentes (memes), y no su aspecto físico o corporal, ni arquitectónico (e.g., el número de módulos, o la función de cada uno de estos dentro de la arquitectura); (2) como consecuencia de lo anterior, conviene entonces hablar de *"procesos meméticos"* en lugar de *"procesos genéticos"*, debido a que los primeros son análogos a los segundos con la diferencia de que se aplican sólo en contextos cognitivos, mentales y culturales; y (3) la conducta de cada agente no está bajo la influencia de un *determinismo memético*, esto es, que la base cognitiva innata formada por el proceso filogenético no determina, de manera exclusiva, la conducta ni las habilidades futuras de cada agente, sino que contrariamente, existe cierta permeabilidad (proporcionada por los procesos epigenéticos y ontogenéticos) que permite que dicha base cognitiva innata sea refinada mediante la continua circulación de información entre el agente y el entorno. Así como en el cerebro humano, que aunque determinado desde el nacimiento por su número de células (neuronas), es capaz de evolucionar en inteligencia y comportamiento si el ambiente es estimulante.

[5] Concretamente, en un entorno multi-agente donde cada individuo de la población es un agente cognitivo.

Los procesos EOF se relacionan recíprocamente produciendo incrementos continuos de información. Para comprender mejor esta interacción resulta apropiado acudir de nuevo a una metáfora. Imagínese el proceso de desarrollo y evolución cognitiva de un agente como una pelota que desciende por los valles de un paisaje epigenético/ontogenético, cuyas características se moldean tanto por una base de conocimiento innata – filogenética – como por el medio ambiente. A medida que el tiempo transcurre, el paisaje se va reconfigurando, generando nuevas rutas y profundizando trayectorias recorridas previamente.

La accidentalidad del paisaje es determinada por la acción del proceso filógenético, el cual explora nuevas configuraciones geográficas en cada generación evolutiva, formando todo tipo de precipicios, montañas y valles. En la medida que los agentes interactúan con el ambiente disminuyen los rangos de variabilidad y plasticidad, es decir, en condiciones estables se esperaría que los factores que refuerzan esta perdida gradual de plasticidad tiendan a fijarse a través del aprendizaje (epigénesis) y del desarrollo (ontogénesis). Las interacciones de cada agente modifican el paisaje individual y definen los nuevos estados disponibles para la población (el paisaje adaptativo). De esta manera, el paisaje abre nuevas vías tanto epigenéticas/ontogenéticas como filogenéticas.

A medida que el agente se auto-define en su recorrido por el paisaje epigenético/ontogenético, su conocimiento sobre el entorno se especializa, y nuevas rutas cada vez más pandas se ramifican accediendo a las variaciones intraespecíficas e individuales. Esto implica que la pelota que desciende rápidamente por una colina buscará con el tiempo estabilizarse lentamente en los valles cercanos a través del aprendizaje y el desarrollo. Este mecanismo de *"adaptación"*, como es fácil de intuir, es el resultado de bucles continuos de retroalimentación negativa modulados por los procesos epigenéticos y ontogenéticos. Por el contrario, el proceso filogenético es guiado por la retroalimentación positiva, y su objetivo es trasladar al agente hasta otros paisajes en busca de óptimos globales, evitando así que el conocimiento adquirido por el agente se atasque en mesetas que representen soluciones locales, en otras palabras, concede un nuevo rango de plasticidad para la *"creación incremental de conocimiento"*.

La reducción en las irregularidades del paisaje epigenético (a través del aprendizaje de estructuras cognitivas básicas) acelera la ontogénesis, facilitando el acortamiento de las vías de desarrollo a causa de la presencia reiterativa de factores ambientales que influyen sobre la formación de estructuras cognitivas más complejas. La reducción de la redundancia se produce cuando son adjudicadas nuevas funciones a las estructuras cognitivas repetidas, de esta manera, el agente accede a nuevas tareas sin prolongar lo ontogenia excesivamente[6].

De este modo, durante la ontogénesis, los incrementos de información (conocimiento) se detectan como modificaciones de las rutas epigenéticas que permiten canalizar el desarrollo de estructuras cognitivas más complejas. Por ejemplo, el proceso epigenético especializa el conocimiento simbólico y sub-simbólico de cada *Comportamiento* de acuerdo con su función dentro del entorno, con lo cual el proceso ontogenético adquie-

[6]Este efecto del proceso epigenético se puede observar claramente, por ejemplo, en la interacción entre las meta-dinámicas y la Máquina Darwiniana del componente SIA, a través de la cual los anticuerpos menos adaptados y/o redundantes, son reemplazados por nuevos anticuerpos que pretenden explorar soluciones más prometedoras.

re gradualmente mayor cantidad de información que le permite determinar cómo ciertos sub-conjuntos de *Comportamientos* especializados pueden interactuar entre sí a fin de alcanzar una meta específica. Por otro lado, en la filogénesis, los incrementos de información (conocimiento) se reflejan como aumentos en la plasticidad fenotípica, o en otras palabras, como mejoras en las capacidades cognitivas (i.e., en las capacidades anticipatorias, decisorias, deliberativas, reactivas, etc.) que le permiten al agente afrontar los cambios del entorno. Por consiguiente, y como se propone en (Andrade, 2001), tanto el desarrollo como la evolución son considerados procesos abiertos, no programados, de incrementos continuos de información. La figura 10.9 ilustra cómo es realizado este proceso de incremento de información en METÁFORA.

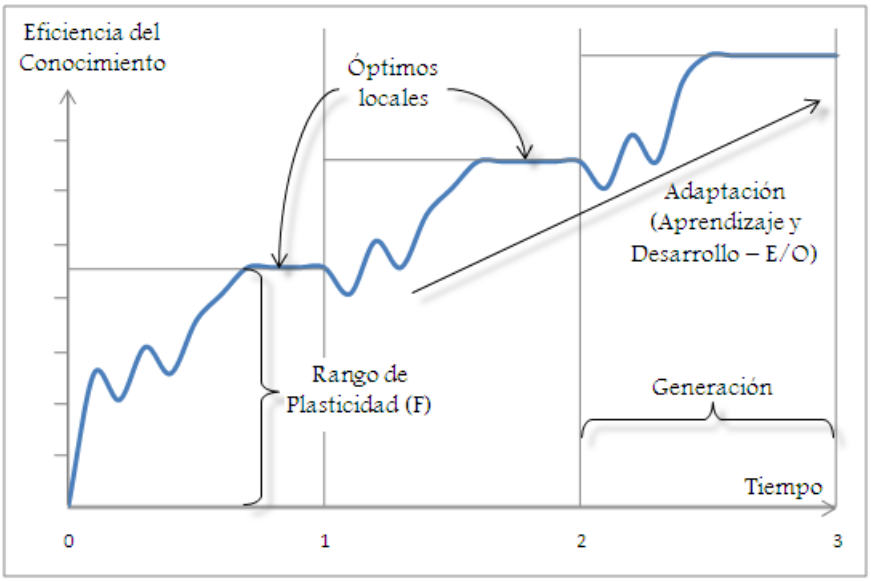


Figura 10.9: Incremento Continuo de Información. El proceso filogenético (F) extiende el rango de plasticidad en cada generación evolutiva permitiendo a los procesos epigenéticos (E) y ontogenéticos (O) re-dirigir la búsqueda hacia óptimos globales.

Así pues, en la presente tesis se propone que la interacción entre los procesos EOF puede ser vista como una espiral evolutiva, donde cada giro completo de la espiral ejecuta secuencialmente cuatro procesos: (1) la generación de las bases de conocimiento innato, (2) el proceso epigenético, (3) el proceso ontogenético, y (4) el proceso filogenético. Internamente, cada proceso realiza una serie de actividades las cuales pueden ejecutarse secuencialmente, en paralelo, o incluso a través de micro-espirales [7], todo ello con el fin de generar, incrementalmente, estructuras cognitivas que van desde las más simples hasta las más complejas, sirviendo cada una de éstas como punto de partida para el siguiente proceso de desarrollo/evolución en la espiral.

[7] Recuérdese que los procesos evolutivos (en todas las diferentes escalas) pueden requerir de múltiples bucles anidados de procesamiento hasta alcanzar una solución relativamente eficiente (de acuerdo con los criterios de optimalidad definidos para cada componente). Por ejemplo, para determinar qué anticuerpos responden mejor a las entradas sensoriales percibidas, el componente SIA necesita de múltiples ciclos cognitivos (generaciones epigenéticas) donde se activen continuamente las diferentes meta-dinámicas inmunológicas, el proceso de diversificación guiado por el Algoritmo Genético, y la selección competitiva de anticuerpos dirigida por el sistema de distribución de recompensas.

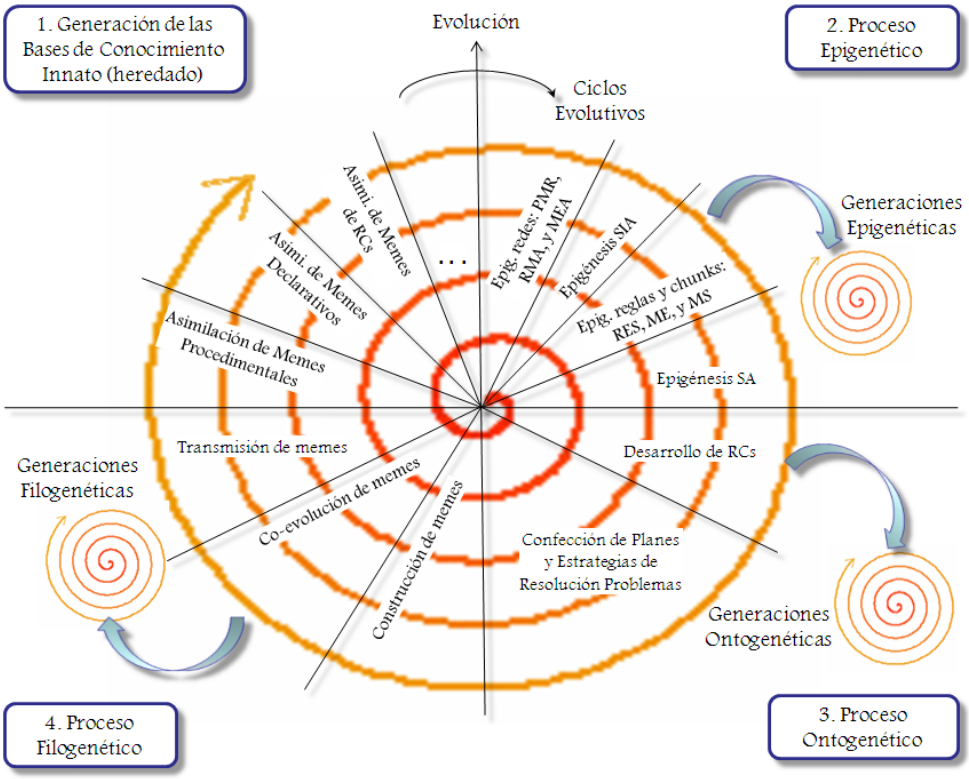


Figura 10.10: Modelo Evolutivo en Espiral para la Construcción de Conocimiento a través de la Integración EOF. PMR: red Perceptrón Multicapa con Retropropagación. RMA: Red de Memoria Asociativa. MEA: Memoria Episódica Abstracta. SIA: Sistema Inmune Artificial. RES: Reglas de Extracción Sub-simbólica. ME: Memoria Episódica. MS: Memoria Semántica. SA: Sistema Clasificador Anticipatorio. RC: Redes de Comportamientos.

Cada giro completo de la espiral se corresponde con un *"ciclo evolutivo"* del sistema. En la medida que transcurren estos ciclos, las estructuras cognitivas son aumentadas, refinadas, y evolucionadas incrementalmente, produciéndose de esta manera bases de conocimiento innato (heredado) que intentan modelar, cada vez con mayor aproximación, el mundo en el que se encuentran inmersos los agentes del sistema. En la figura 10.10 se ilustra el proceso propuesto.

Pasando al terreno de lo computacional, se propone que la integración de los procesos EOF sea orquestada a través de un *Algoritmo Memético* (AM). Como se discutió en la sección 6.3.2.4, los AM constituyen un paradigma de optimización híbrido fundamentado en la explotación sistemática de conocimiento, el cual combina diferentes metaheurísticas basadas tanto en búsquedas globales (poblaciones) como en búsquedas locales. La idea central de los AM es combinar las mejoras individuales de las soluciones en cada uno de los agentes junto con procesos de cooperación y competición de tipo poblacional.

Concretamente, se propone que los procesos epigenéticos y ontogenéticos se encarguen de coordinar las búsquedas locales, mientras que el proceso filogenético se ocupe de la búsqueda global. Cabe resaltar que, aunque comúnmente la *búsqueda local* en un

AM es dirigida por optimizadores locales (como las técnicas de seguimiento del gradiente), en METÁFORA se propone una combinación de estrategias tanto locales como poblacionales.

Estrategia	Componente	Descripción
Búsqueda Local	PMR	Aprendizaje Supervisado guiado por el Algoritmo de Retropropagación.
		Aprendizaje por Refuerzo guiado por el algoritmo estocástico de *Q-learning*.
	SIA	Emplea diferentes heurísticas que controlan la Exploración, la Explotación, y el Aprendizaje por Descubrimiento: un AG con operadores genéticos específicos, diversas meta-dinámicas, y un Sistema de Distribución de Recompensas (Bucket Brigade) que dirige la búsqueda.
	RES	Emplea un proceso pseudo-estocástico para extraer reglas simbólicas a partir de reglas sub-simbólicas, y para generalizar y especializar las reglas simbólicas extraídas.
	RMA y MEA	Igual que las redes neuronales PMR.
	ME y MS	Igual que el componente RES.
	SA	Emplea múltiples heurísticas: un Algoritmo de *Q-learning* para el Aprendizaje por Refuerzo, un Sistema de Distribución de Recompensas para el Aprendizaje Anticipatorio (Especialización), y un AG para la Generalización de conocimiento.
	RC y Planes	Proceso Ontogenético guiado por PEG. Emplea evaluaciones en línea y fuera de línea para dirigir el desarrollo de las estructuras cognitivas.
Búsqueda Global	Memes	Proceso Filogenético (co-evolución) guiado por PEG y por AG. Define Operadores Genéticos especializados para la optimización multi-objetivo. Adicionalmente, se apoya en los algoritmos de aprendizaje de cada componente para asimilar los memes evolucionados.

Tabla 10.1: Síntesis de las Estrategias de Búsqueda Local y Global empleadas para la Construcción de Conocimiento. PMR: red Perceptrón Multicapa con Retropropagación. SIA: Sistema Inmune Artificial. AG: Algoritmo Genético. RES: Reglas de Extracción Subsimbólica. RMA: Red de Memoria Asociativa. MEA: Memoria Episódica Abstracta. ME: Memoria Episódica. MS: Memoria Semántica. SA: Sistema Clasificador Anticipatorio. RC: Redes de Comportamientos. PEG: Programación de Expresiones Genéticas.

La tabla 10.1 categoriza las estrategias de búsqueda empleadas por los diferentes componentes de la arquitectura. Nótese que algunos métodos en la estrategia de búsqueda local emplean optimizadores basados en poblaciones, concretamente cuando se utilizan las Máquinas Darwinianas, evitando así la desventaja típica de este tipo de estrategia: dirigir la búsqueda hacia óptimos locales. No obstante, son considerados métodos de búsqueda local debido a que operan a un nivel intra-agente, lo que implica que el aprendizaje está condicionado a las experiencias y necesidades propias del agente.

Por otro lado, en las estrategias de búsqueda global, se emplea el *meme* como unidad de conocimiento transmisible, el cual sirve como puente entre los procesos EOF. En el contexto de METÁFORA, y tomando como referencia a (Blackmore, 2003) y (Dawkins, 1992), se tiene que un *meme* es una estructura cognitiva que cumple con los siguientes atributos:

- *Fidelidad*: los memes crean copias de sí mismos con cierta exactitud. Desde esta perspectiva, el meme se define como una unidad de imitación que puede ser comunicable con cierta fidelidad.
- *Fecundidad*: la frecuencia con la que se copia y se propaga un meme en el área de trabajo, determina su dominancia con respecto a los demás.
- *Longevidad*: un meme debe permanecer en el área de trabajo el tiempo suficiente para que sobre él actúen las presiones selectivas.

Considerando lo anterior, sólo los memes más exitosos (i.e., aquellos que cumplan en mayor medida con los tres atributos), tendrán una mayor probabilidad de ser copiados y por tanto de sobrevivir. De aquí se deriva entonces, una competencia interna entre todos los memes por ocupar el espacio mental de cada agente, y por ser luego propagados en el acervo memético mediante un proceso que, considerado en su sentido más amplio, puede llamarse *"imitación"*.

Así pues, los memes pueden ser considerados estructuras cognitivas auto-replicadoras, que parasitan la "mente" de cada agente, convirtiéndola en un vehículo de propagación por el cual viajan hasta otros agentes para luego reconstruirse a sí mismos, de la misma forma que un virus puede parasitar el mecanismo genético de una célula anfitriona y luego contagiarse. Esta imagen de los memes como ideas virales, había sido sugerida ya por otros autores (Delius, 1991; Blackmore, 2003; Dawkins, 1992), y es lo que básicamente distingue el concepto de "meme" del concepto simple de "estructura cognitiva" empleado en METÁFORA. Mientras que una estructura cognitiva típica (i.e., una regla, un chunk, un *Comportamiento*, una RC, un plan, etc.) se desarrolla localmente dentro de un acervo específico definido por cada agente; el meme extrae y encapsula la información de dicha estructura cognitiva; evoluciona globalmente a través de la replicación, la mutación, la recombinación y la fusión constante con otros memes; y finalmente, viaja de un agente a otro propagándose y haciéndose persistente.

10.4.2. Anatomía del Algoritmo Memético

En esta sección se describe el funcionamiento del AM, haciendo un especial énfasis sobre el proceso filogenético que hasta ahora se ha tratado de manera general. Antes

de entrar en materia, cabe enunciar algunas observaciones acerca del enfoque evolutivo utilizado por el AM.

Aunque en biología la teoría evolutiva más aceptada es el *darwinismo*, en el campo de la computación evolutiva son también válidos otros dos enfoques que, en general, hacen que el modelo darwiniano sea más elástico al no estar constreñidos por las restricciones biológicas y, en particular, resultan más apropiados para dirigir la evolución y transmisión de memes del AM propuesto. Estos dos enfoques son conocidos como el *enfoque lamarckiano* y el *efecto baldwiniano*, y su caracterización se expone a continuación:

Enfoque Lamarckiano: este enfoque argumenta que las características adquiridas por un individuo en el transcurso de su vida son heredadas a su descendencia (Lamarck, 1809). De acuerdo con esta aproximación, el aprendizaje tiene una gran influencia sobre el proceso evolutivo, de modo que el agente puede cambiar su código genético durante su interacción con el entorno. Computacionalmente, este enfoque evolutivo ha sido aplicado exitosamente en distintos problemas que involucran aprendizaje artificial (Gorges-Schleuter, 1997; Merz y Freisleben, 1997; Rosselló-Mir, 1999).

En METÁFORA, el esquema general de adquisición de conocimiento es en esencia lamarckiano, de manera que los memes producidos por cada generación filogenética pueden "modificar" su codificación genética antes de ser evaluados nuevamente en la siguiente generación. En otras palabras, los memes no sólo son refinados y mejorados por las dinámicas evolutivas del proceso filogenético, sino que además pueden variar genotípica y fenotípicamente en tiempo de ejecución (a través de los procesos epigenéticos y ontogenéticos) con el fin de adaptarse a los cambios ambientales.

Efecto Baldwiniano: esta aproximación es un concepto evolutivo que mantiene que sólo ciertos aspectos del aprendizaje pueden ser "transmitidos" a las siguientes generaciones. Principalmente, se basa en que aquellos individuos que sean más hábiles a la hora de aprender tendrán más posibilidades de sobrevivir, y del mismo, modo sus descendientes heredarán esa misma capacidad de aprendizaje (Baldwin, 1896).

En el enfoque baldwiniano, los individuos no transmiten genéticamente los conocimientos, sino las características genéticas que hacen que puedan aprender mejor aquellas habilidades adquiridas que los harán sobrevivir. Con el paso de las generaciones, es posible que algunas de estas habilidades sean "absorbidas" por el código genético, de modo que se convierten en conocimiento instintivo que no requiere de un aprendizaje explícito. Esta transmisión no se produce de manera directa, sino indirectamente, y por mera probabilidad.

En el campo de la computación evolutiva, el efecto baldwiniano se ha incorporado como método de aceleración de la convergencia (Ackley y Littman, 1992; Harvey, 1893; Belew, 1989; Hinton y Nowlan, 1997). En METÁFORA, el efecto baldwiniano se ha aplicado como un mecanismo de aceleración de los procesos epigenéticos y ontogenéticos. Más concretamente, se tiene que: (1) el componente SIA acelera la convergencia evolutiva a través del mecanismo de aprendizaje por contra-ejemplos y de la versión adaptada del algoritmo CHC (descrito en la sección 7.4.3.1); (2) el componente SA mejora el aprendizaje anticipatorio mediante los procesos de especialización y generalización de

clasificadores; (3) los *Comportamientos* aceleran el aprendizaje de reglas (simbólicas y sub-simbólicas) a través de los mecanismos de aprendizaje vertical; (4) el módulo de Orquestación de la Consciencia agiliza el proceso evolutivo de las RCs mediante la combinación de las funciones de evaluación en línea y fuera de línea; y (5) el módulo para la Generación de Planes orienta el proceso evolutivo mediante la simulación interna y la evaluación en tiempo de ejecución (en línea) que permite la re-planificación. Todos estos mecanismos aceleradores tienen como objetivo sesgar y alterar la forma del espacio de búsqueda, permitiendo a los diferentes tipos de memes evolucionar más rápidamente que si se usara un enfoque puramente darwiniano.

La diferencia esencial entre el efecto baldwiniano y el enfoque lamarckiano es que este primero no modifica, en tiempo de ejecución, el código genético de los individuos (memes), sino que lo hace sólo hasta que se produce una nueva generación filogenética, caso en el que se aplican los operadores genéticos sobre aquellos individuos que tuvieron un mejor desempeño, es decir, sobre aquellos que aprendieron a modelar de una forma más aproximada el comportamiento del entorno.

Retomando el propósito de esta sección, una vez que los procesos epigenéticos y ontogenéticos han tenido lugar, se procede a la evolución del acervo de memes. Como se puede observar en la figura 10.11, existe una población determinada de agentes cognitivos (cada uno de los cuales emplea la arquitectura cognitiva propuesta) quienes se comunican con un *"Agente Memético"* encargado de coordinar el proceso evolutivo de los memes. Este proceso consiste de tres fases las cuales se describen con detalle en los siguientes apartados: construcción y envío de memes, co-evolución de memes, y transmisión y asimilación de memes.

10.4.2.1. Fase 1: Construcción y Envío de Memes

Esta fase tiene como propósito la construcción, por parte de cada agente cognitivo, de los diferentes tipos de memes que serán luego enviados al agente memético para participar en el proceso de evolución.

Construcción de Memes a partir de los Comportamientos: por cada *Comportamiento* definido por el agente, el proceso de construcción de memes se realiza a través del componente SIA, esto es:

1. Se sincronizan todos los componentes del *Comportamiento* (PMR, SIA, RES, y RP) mediante los mecanismos de aprendizaje vertical, con el fin de obtener una base de conocimiento homogénea.
2. Dada la población actual de anticuerpos del componente SIA, $P_{(SIA,t)}$, se selecciona una sub-población $S_{(SIA,t)}$, la cual se compone de los anticuerpos con mayor *nivel de aptitud*. Por defecto, $S_{(SIA,t)}$ corresponde al 40 % de los mejores anticuerpos de $P_{(SIA,t)}$.
3. Por cada uno de los anticuerpos de $S_{(SIA,t)}$, se copia su información genética (i.e., la parte condición, la parte acción, y el nivel de aptitud del anticuerpo) y se inser-

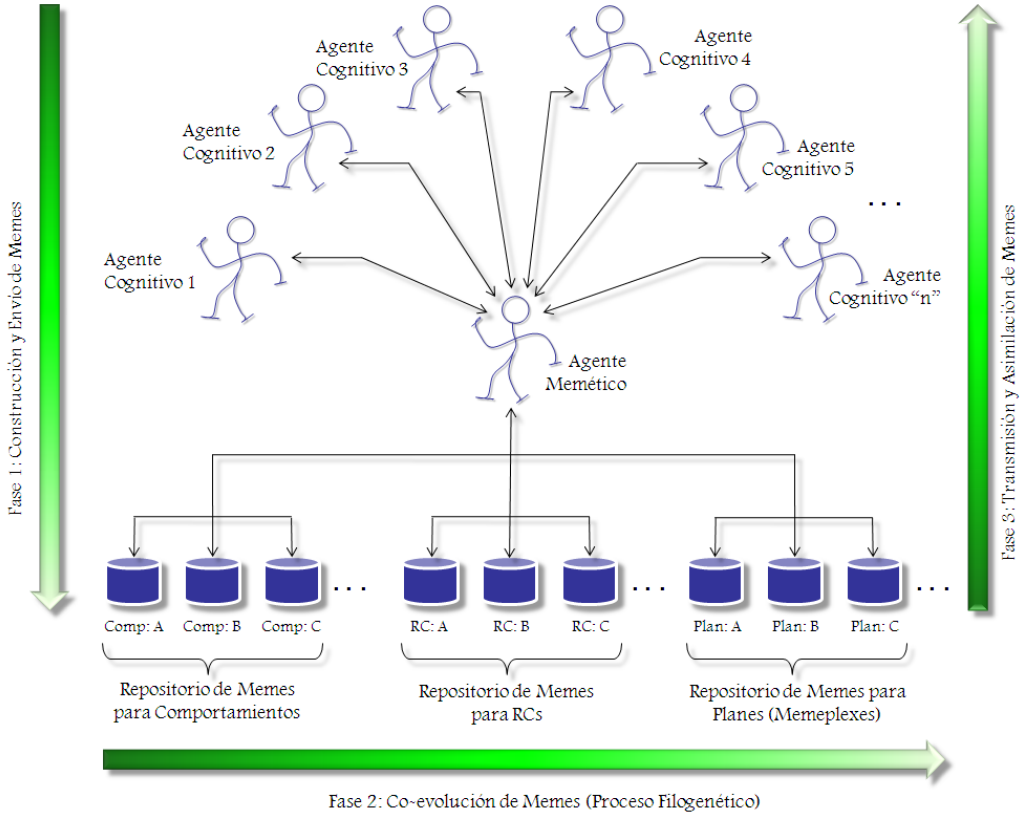


Figura 10.11: Proceso Evolutivo para los diferentes Acervos de Memes.

ta en un nuevo meme del tipo *Regla-Comportamiento*. Adicionalmente, el meme creado es etiquetado con el nombre del *Comportamiento* respectivo.

4. Una vez se han construido los memes de todos los *Comportamientos*, estos son enviados, a través de un canal de comunicación, al agente memético quien se encarga de almacenar cada meme en el repertorio de memes para los *Comportamientos*.

Construcción de Memes a partir del Sistema Anticipatorio: por cada SA asociado a un *Comportamiento* definido por el agente, se realiza lo siguiente:

1. Dada la población actual de clasificadores del SA, $P_{(SA,t)}$, se selecciona una subpoblación $S_{(SA,t)}$, la cual se compone de los clasificadores con mayor *nivel de aptitud*. Por defecto, $S_{(SA,t)}$ corresponde al 40 % de los mejores clasificadores de $P_{(SA,t)}$.
2. Por cada uno de los clasificadores de $S_{(SA,t)}$, se copia su información genética (i.e., la parte condición, la parte acción, la parte efecto, la parte marcador, y el nivel de aptitud o calidad del clasificador) y se inserta en un nuevo meme del tipo *Regla-SA*. Adicionalmente, el meme creado es etiquetado con el nombre del SA respectivo.

3. Una vez se han construido los memes de todos los SAs, estos son enviados, a través de un canal de comunicación, al agente memético quien se encarga de almacenar cada meme en el repertorio de memes para los SAs.

Construcción de Memes a partir de las RCs: por cada una de las diferentes tareas definidas para el agente, se realiza:

1. Dado el repertorio actual de MATs para una tarea, $P_{(RC,t)}$, se selecciona una sub-población $S_{(RC,t)}$, la cual se compone de las MATs con mayor *nivel de aptitud*. Por defecto, $S_{(RC,t)}$ corresponde al 40 % de las mejores MATs de $P_{(RC,t)}$.
2. Por cada una de las MATs de $S_{(RC,t)}$, se copia su información genética (es decir, el cromosoma multigénico que contiene la información genotípica de la RC respectiva), y se inserta en un nuevo meme del tipo *RC*. Adicionalmente, el meme creado es etiquetado con el nombre del repertorio de MATs respectivo.
3. Una vez se han construido los memes de todos los repertorios de MATs, estos son enviados al agente memético quien se encarga de almacenar cada meme en el repertorio de memes para las RCs.

Construcción de Memes a partir de los Planes (Memeplexes): por cada una de las diferentes categorías de planes definidas para el agente, se realiza:

1. Dado el repertorio actual de planes para una categoría específica, $P_{(P,t)}$, se selecciona una sub-población $S_{(P,t)}$, la cual se compone de los Planes con mayor *nivel de aptitud*. Por defecto, $S_{(P,t)}$ corresponde al 40 % de los mejores Planes de $P_{(P,t)}$.
2. Por cada uno de los Planes de $S_{(P,t)}$, se copia su información genética (es decir, la célula que contiene la información genotípica del Plan respectivo), y se inserta en un nuevo meme del tipo *Plan*. Adicionalmente, el meme creado es etiquetado con el nombre del repertorio de Planes respectivo.
3. Una vez se han construido los memes de todos los repertorios de Planes, estos son enviados al agente memético quien se encarga de almacenar cada meme en el repertorio de memes para los Planes.

10.4.2.2. Fase 2: Co-evolución de los Memes (Filogénesis)

> *"En la evolución no hace falta fracasar para extinguirse, sólo basta acertar con menos frecuencia".*
> – Charles Darwin.

Una vez que todos los agentes cognitivos han enviado al agente memético sus memes mejor adaptados, se procede a la evolución de los mismos a través del proceso filogenético. La idea básica que subyace al proceso filogenético se inspira en el conocido apotegma "divide y venceras" (*divide et impera*), por lo cual se propone una aproximación co-evolutiva donde la evolución aislada de cada repertorio de memes se enfoca hacia la

optimización de un objetivo en particular y, a su vez, coopera en la evolución de otros repertorios de memes que representan soluciones más complejas.

En este orden de ideas, retómese el concepto de *"memeplexe"*, el cual puede interpretarse como un complejo de memes "co-adaptados" que cooperan mutuamente. Así pues, en METÁFORA, los memes se auto-organizan jerárquicamente desde los más simples (reglas) hasta los más complejos (planes) a través del proceso co-evolutivo. En este caso, la evolución de un memeplexe (e.g., un plan) no depende únicamente de las presiones selectivas ejercidas dentro de su repertorio, sino que además, depende de la evolución de las RCs, los *Comportamientos*, y las reglas que lo conforman. Concretamente, la manera de evidenciar esta co-evolución es a través de la correlación existente entre las funciones de aptitud de cada meme. Por ejemplo, el nivel de aptitud de un Plan específico depende, en cierta medida, de la eficiencia de sus MATs asociadas (i.e., de sus RCs); a su vez, los niveles de aptitud de tales RCs dependen, parcialmente, de cuán especializados y apropiados sean los *Comportamientos* que las constituyen; y así mismo, el desempeño de dichos *Comportamientos* depende de la eficiencia de sus reglas (simbólicas y sub-simbólicas).

Teniendo en cuenta que las representaciones genéticas para los memes y los memeplexes ya han sido establecidas previamente[8], el siguiente aspecto a determinar son los operadores genéticos de la Máquina Darwiniana para el proceso filogenético. Cabe notar que la MD filogenética no requiere aplicar un operador de selección específico sobre los memes que han sido enviados al agente memético puesto que estos ya han pasado por un proceso de "selección elitista", objetivo que cumple la fase anterior.

La generación de nuevos memes durante la fase de reproducción filogenética se realiza mediante la manipulación de las estructuras cognitivas relevantes que se han identificado (i.e., los memes padres). A tal fin, en el diseño de un AM típico suelen ser empleadas cualquiera de las plantillas genéricas definidas para ello, por ejemplo, RRR (*Random Respectful Recombination*), RAR (*Random Assorting Recombination*), RTR (*Random Transmitting Recombination*), entre otras (Radcliffe, 1994). En cualquier caso, sobra decir que el rendimiento del AM se verá beneficiado si en lugar de manipular a ciegas la información genética de los memes, se hace de manera dirigida empleando conocimiento del problema. Desde un punto de vista general, esta inclusión de conocimiento del problema en la manipulación de los memes tiene dos vertientes: la selección de las características parentales que serían transmitidas a la descendencia (a través del operador de recombinación), y la selección de las características no-parentales que serían también incluidas en la misma (mediante los operadores de transposición y mutación).

En relación a la selección de la información contenida en los padres que debe transmitirse a los hijos a través del operador de recombinación, la evidencia experimental (Mathias y Whitley, 1992; Oguz y Ercan, 2005) aconseja conservar aquellas características comunes a ambos padres. Una vez hecho esto, el descendiente puede completarse de diferentes maneras. Debido a que los dominios y representaciones genéticas de cada

[8]En secciones anteriores donde se han detallado tanto las Máquinas Darwinianas para los procesos epigenéticos y ontogenéticos, como los métodos de codificación genética para las estructuras cognitivas involucradas.

meme son diferentes[9], se propone que el operador de recombinación filogenético se base en la plantilla RRR, gracias a la flexibilidad e independencia del dominio que ofrece, aunque con la incorporación de ciertas adaptaciones (según el tipo de meme), las cuales se exponen a continuación:

Operador de Recombinación para Memes del Tipo Regla: este operador genético se emplea sobre los memes del tipo regla, esto es, sobre aquellos memes que tienen una estructura cromosómica "lineal", los cuales han sido extraídos de los componentes SIA y SA. Concretamente, este operador actúa de la misma forma que el operador de *cruce uniforme HUX* junto con el mecanismo de *prevención de incesto*, descritos ambos en el algoritmo 7.5.

Operadores de Recombinación para Memes del Tipo RC y Plan: este operador genético actúa sobre los memes que tienen una estructura cromosómica no-lineal (i.e., arbórea), en concreto, sobre aquellos memes que han sido extraídos de estructuras cognitivas como las RCs y los Planes. Para este tipo de memes, se aplican dos operadores de recombinación distintos: *Recombinación Uniforme* y *Recombinación Genética*.

El operador de *recombinación uniforme* produce dos nuevos individuos (memes), conservando el equilibrio entre la exploración y la exploración del espacio de búsqueda. Concretamente, este operador emplea la plantilla RRR (i.e., copia en los nuevos memes los alelos – elementos – comunes de los padres y genera aleatoriamente el resto de alelos), y aplica las restricciones de recombinación propuestas por la PEG para rellenar los segmentos de las cabeceras y las terminaciones con alelos de sus respectivos dominios (como se explicó en la sección 9.3.1.5). Adicionalmente, los niveles de aptitud de los memes hijos son calculados como el promedio de las aptitudes de los padres. El algoritmo ejecutado por el operador se presenta en el cuadro 10.4.

Por ejemplo, considérese dos cromosomas padres correspondientes a dos memeplexes diferentes del tipo RC como los de la expresión 10.2. El resultado de recombinar estos dos padres, podría generar dos nuevos memeplexes como los de la expresión 10.3, donde los alelos comunes a los dos padres se muestran sombreados, y el resto corresponden a alelos generados aleatoriamente.

Por otro lado, el operador de *recombinación genética* (Ferreira, 2001) selecciona aleatoriamente dos genes completos del cromosoma, y los intercambia de posición. Por ejemplo, considérese nuevamente los memes padres de la expresión 10.2. Suponiendo que el *Gen 2* es el que se ha seleccionado aleatoriamente para su intercambio, entonces los cromosomas de los memes hijos serían como los representados en la expresión 10.4. En este caso, los cromosomas que se presentan sombreados corresponden a la información genética heredada del memeplexe padre 1, mientras que aquellos que no lo están corresponden al memeplexe padre 2.

Nótese que este operador, a diferencia del operador de recombinación uniforme, intercambia la información completa del gen sin ninguna alteración, evitando "destruir"

[9]Por ejemplo, se emplean representaciones lineales vs. no-lineales (arbóreas); los genes funcionales, estructurales, y homeóticos emplean diversos dominios genéticos; los segmentos cabecera y terminación de los genes estructurales están sujetos a restricciones y emplean también diferentes dominios; etc.

Método: recombinarUniforme

```
1.  Variables
2.     SubPob          ↑↓ sub-población de memes elegidos para recombinarse.
3.     SubPobTemp      ↑ sub-población de memes recombinados.
4.  Inicio Procedimiento
5.     SubPobTemp[] ← new Poblacion[SubPob.tamanio]
6.     Para cada pareja de memes padres de SubPob (a, b) hacer:
7.        Para cada alelo de ''a'' hacer:
8.           c ← new meme, d ← new meme
9.           Si a.alelo == b.alelo entonces:
10.             c.alelo ← a.alelo
11.             d.alelo ← a.alelo
12.          De lo contrario
13.             Si alelo ∈ cabecera entonces:
14.                c.alelo ← generarAleatorio(dominioCabecera)
15.                d.alelo ← generarAleatorio(dominioCabecera)
16.             De lo contrario
17.                c.alelo ← generarAleatorio(dominioTerminacion)
18.                d.alelo ← generarAleatorio(dominioTerminacion)
19.             Fin Si
20.          Fin Si
21.       Fin Para
22.       c.aptitud ← calcularAptitudPromedio(a,b)
23.       d.aptitud ← calcularAptitudPromedio(a,b)
24.       SubPobTemp ← agregar(c)
25.       SubPobTemp ← agregar(d)
26.    Fin Para
27.    retornar SubPobTemp
28. Fin Procedimiento
```

Algoritmo 10.4: Algoritmo para el operador de Recombinación Uniforme.

Cromosoma 1 (Memeplexe Padre):

Gen 1							Gen 2							Gen 3						
0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6
C_1	D	P_1	D	P_2	P_3	...	C_2	P_7	P_6	P_2	P_3	P_4	...	C_3	P_7	P_8	P_9	P_8	P_7	...

Cromosoma 2 (Memeplexe Padre):

Gen 1							Gen 2							Gen 3						
0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6
C_1	D	D	P_3	D	P_1	...	C_2	P_8	P_2	P_2	P_5	D	...	C_3	P_1	P_2	P_4	P_6	P_7	...

Expresión Genética 10.2: Cromosomas Padres

el conocimiento aprendido por los propios genes. El algoritmo para este operador no es presentado debido a su simplicidad.

Cromosoma 3 (Memeplexe Hijo):

Gen 1							Gen 2							Gen 3						
0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6
C_1	D	P_5	T	P_1	P_2	…	C_2	P_3	P_5	P_2	P_6	P_8	…	C_3	P_1	D	P_4	P_5	P_7	…

Cromosoma 4 (Memeplexe Hijo):

Gen 1							Gen 2							Gen 3						
0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6
C_1	D	P_2	P_1	D	D	…	C_2	P_6	P_8	P_2	P_1	D	…	C_3	T	P_4	P_6	P_3	P_7	…

Expresión Genética 10.3: Descendencia producida por Recombinación Uniforme

Cromosoma 3 (Memeplexe Hijo):

Gen 1							Gen 2							Gen 3						
0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6
C_1	D	P_1	D	P_2	P_3	…	C_2	P_8	P_2	P_2	P_5	D	…	C_3	P_7	P_8	P_9	P_8	P_2	…

Cromosoma 4 (Memeplexe Hijo):

Gen 1							Gen 2							Gen 3						
0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6
C_1	D	D	P_3	D	P_1	…	C_2	P_7	P_6	P_2	P_3	P_4	…	C_3	P_1	P_2	P_4	P_6	P_7	…

Expresión Genética 10.4: Descendencia producida por Recombinación Genética

Operadores de Transposición para Memes del Tipo RC y Plan: este operador únicamente actúa sobre los memeplexes cuya Máquina Darwiniana emplea PEG como mecanismo evolutivo. La razón para esto, es que los memes del tipo regla no están compuestos por bloques de construcción explícitos que se puedan transponer, como sí lo están los cromosomas multigénicos de los memeplexes. Concretamente, el AM emplea dos operadores de este tipo: el operador de *transposición de secuencias de inserción* (previamente descrito en le sección 9.3.1.5), y el operador de *transposición genética* que se explica a continuación.

El operador de *transposición genética* activa y traslada "genes completos" al inicio del cromosoma (en el caso de las RCs) o de la célula (en el caso de los planes). A diferencia del operador de transposición de secuencias de inserción, en la transposición genética el transposón (en este caso el gen completo) es eliminado del lugar de origen luego de ser trasladado, con el fin de mantener constante la longitud del cromosoma o de la célula. La idea básica que subyace al funcionamiento de este operador, es la selección aleatoria de un gen (excepto el primero, obviamente) el cual luego es trasladado. Considérese el cromosoma de la expresión 10.5 para una RC. Supóngase ahora, que el Gen 3 se ha activado como transposón, situándose al inicio del cromosoma y desplazando a los demás genes, como se observa en la expresión 10.6.

Cromosoma x (Memeplexe):

Gen 1							Gen 2							Gen 3						
0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6
C_1	D	D	P_3	D	P_1	...	C_2	P_1	P_2	P_4	P_6	P_7	...	C_3	P_7	P_6	P_2	P_3	P_4	...

Expresión Genética 10.5: Cromosoma Sin Variación

Cromosoma x (Memeplexe):

Gen 1							Gen 2							Gen 3						
0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6
C_3	P_7	P_6	P_2	P_3	P_4	...	C_1	D	D	P_3	D	P_1	...	C_2	P_1	P_2	P_4	P_6	P_7	...

Expresión Genética 10.6: Variación producida por Transposición Genética

Cabe resaltar que, en la mayoría de las ocasiones, los genes (memes) intercambiados/transpuestos por los operadores de *recombinación genética* y *transposición genética*, son tan diferentes que producen cambios considerables en el fenotipo del memeplexe. Por ejemplo, si los memes que conforman un plan cambian de posición dentro del mismo memeplexe (a través de la transposición genética), o se intercambian entre sí con memes de otro memeplexe (a través de la recombinación genética), se generaría un plan con una secuencia de tareas completamente diferente al memeplexe inicial aunque preservando la identidad de cada uno de sus bloques de construcción, los propios memes. En todo caso, el potencial transformador de estos dos operadores se revela cuando son producidas alteraciones fenotípicas aunque con cierta conservación de la información genotípica, (i.e., la información de los genes/memes constituyentes).

Operadores de Mutación para Memes del Tipo Regla: este operador aplica una variación aleatoria sobre la información genética de los memes del tipo regla. La idea básica de este operador consiste, en primer lugar, en determinar probabilísticamente si el meme será mutado o no (con una probabilidad variable como se explica más adelante). En el caso de que el meme deba ser mutado, se calcula aleatoriamente una posición (alelo) del cromosoma memético, la cual se muta por su valor contrario (e.g., si el valor del alelo es "1" se muta a "0" y viceversa.).

Nótese que el algoritmo para este operador de mutación resulta muy similar al descrito en la sección 7.4.3.1 para el componente SIA, con la diferencia de que no se emplea un alfabeto ternario durante la mutación $\{0, 1, \#\}$. Las razones para esto son, por un lado, que el propósito del operador de mutación es "especializar" cada meme del tipo regla y por tanto no debe incluir el alelo comodin $\#$ (el cual sí que es empleado en la generalización de una regla) y, por otro, porque los componentes PMR, RES y RP emplean únicamente alfabetos binarios (0, 1) en la representación sub-simbólica de sus reglas.

Operadores de Mutación para Memes del Tipo RC y Plan: este operador actúa de la misma manera que el operador de mutación definido por las MDs basadas en la PEG. Concretamente, para la mutación de memes del tipo RC se toman en cuenta las consideraciones descritas en la sección 9.3.1.5 y, para los memes del tipo Plan se toman en cuenta las observaciones de la sección 10.3.2.

Algoritmo para la Máquina Darwiniana Filogenética: el diseño del algoritmo para esta MD se fundamenta en los siguientes criterios:

Minimización de la Epístasis: se habla de epístasis[10] cuando los elementos básicos de información a partir de los cuales se construyen las soluciones (y que son manipulados por los operadores genéticos) interactúan de manera no aditiva sobre la función de aptitud (Davidor, 1991). La existencia de una interacción de este tipo impide que se pueda descomponer la función objetivo en términos optimizables de manera independiente[11]. Obviamente, cuanto menor sea esta interacción no aditiva, mayor será la relevancia que, en términos absolutos, tengan los elementos de información manipulados por los operadores, y menos propensa será la función de aptitud a extraviar la búsqueda. Este criterio es aplicado en METÁFORA en la medida que la acción de los elementos de información de un meme determinado (i.e., una regla, un *Comportamiento*, una RC o un Plan) no afecta directamente a otros elementos de información del mismo meme. Por ejemplo, las modificaciones que se hagan sobre la estructura de un elemento de información de un meme RC (i.e., sobre la estructura de uno de sus *Comportamientos*), afecta de forma independiente al *Comportamiento* en cuestión y no a otros *Comportamientos* del mismo meme, permitiendo que la aptitud de dicho elemento pueda ser evaluada de forma independiente. No obstante, la eficiencia y optimalidad de todo el meme depende de la interacción no aditiva entre los múltiples elementos de información que lo componen.

Maximización de la correlación de bondad: asumiendo un cierto operador genético, se mide la correlación existente entre la adecuación de los memes progenitores y los memes descendientes. Si la correlación es alta, los individuos (memes) buenos tendrían una descendencia buena por lo general (Radclife y Surry, 1994). Dado que el proceso de selección (fase de construcción y envío de memes) se encarga de elegir a los memes más adecuados, la búsqueda se iría desplazando gradualmente hacia las zonas más prometedoras del espacio de búsqueda. Este criterio se puede apreciar, por ejemplo, en el propósito "conservativo" de los operadores de *recombinación uniforme* y de *recombinación HUX* por mantener las características parentales en la descendencia.

Mantenimiento del equilibrio entre Convergencia y Diversidad: estos dos factores influyen sobre la efectividad de la MD filogenética. Por un lado, la Convergencia centra la búsqueda en regiones prometedoras mediante la presión selectiva y, por otro, la Diversidad evita la convergencia prematura, esto es, la rápida convergencia hacia zonas que no contienen el óptimo global (Eiben y Smith, 2003). Cuando los operadores de recombinación disminuyen la varianza de la población y por tanto la diversidad, la descendencia

[10] En términos biológicos, la epístasis se define como la interacción génica entre diferentes genes para una determinada característica. Este fenómeno sucede cuando la acción de un gen se ve modificada por la acción de uno o varios genes del mismo cromosoma.

[11] Este impedimento ha de entenderse en el sentido de que una optimización separada de los elementos básicos de información (alelos de un meme) conduciría a un óptimo local de la función objetivo.

producida conduce hacia la convergencia prematura a óptimos locales. Por otro lado, cuando los operadores de recombinación aumentan la varianza de la población y por tanto la diversidad, el algoritmo conduce hacia una búsqueda demasiado aleatoria, con lo que el proceso filogenético no convergería adecuadamente. Así pues, el objetivo de este criterio es mantener un equilibrio entre la convergencia y la diversidad de la población, como se ilustra en la figura 10.12.

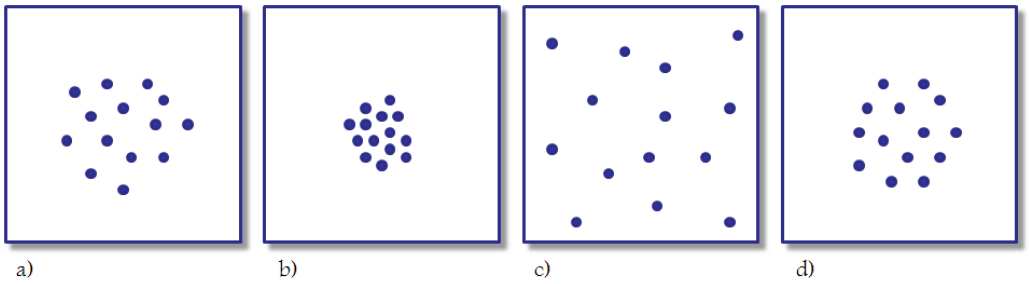


Figura 10.12: Comportamiento de los operadores genéticos basado en la varianza de la población. a) Población de memes padres. b) Descendencia con varianza baja (alta convergencia). c) Descendencia con varianza alta (alta diversidad). d) Descendencia con varianza apropiada (equilibrio entre convergencia y diversidad).

Con el fin de mantener el equilibrio entre los factores de convergencia y diversidad, se proponen las siguientes pautas en el diseño de los operadores de la MD filogenética:

- *Diversidad en la Mutación:* una probabilidad de mutación alta no necesariamente soluciona el problema de la convergencia prematura, aunque una probabilidad baja sí que influye en dicho problema (Eiben y Smith, 2003). Por tanto, se propone una probabilidad de mutación "variable" para cada cromosoma memético hijo. Esto implica entonces la aplicación de dos criterios: (1) con el fin de proveer una diversidad elevada al inicio del proceso filogenético, se determina una probabilidad de mutación alta ($p_{mm} = 0.9$)[12] para todos los tipos de memes hijos. En el caso de que el meme hijo sea elegido para su mutación, se modifican aleatoriamente 3 de sus alelos. Luego, en cada generación filogenética, la probabilidad de mutación disminuirá gradualmente (con un decremento del 1 %) hasta llegar a una $p_{mm} = 0.1$, donde unicamente se muta un alelo del meme hijo; y (2) con el objetivo de introducir un mayor cambio en aquellos memes "malos" (poco eficientes) en comparación con los "buenos" (muy eficientes), la probabilidad de mutación en cada generación filogenética varía de la siguiente manera: para los memes malos $p_{mm} = p_{mm} + 0.1$, y para los memes buenos $p_{mm} = p_{mm} - 0.1$.
- *Diversidad en la Recombinación:* se definen dos criterios que permiten la selección de los memes padres de forma que se mantenga la diversidad de la población: (1) la prohibición de incesto, esto es, dos memes padres se cruzan únicamente si su distancia Hamming[13] está por encima de cierto umbral (por defecto,

[12] Si un número generado aleatoriamente está por debajo de esta probabilidad, entonces es mutado, de lo contrario, permanecerá sin variación.

[13] Recuérdese que la distancia Hamming es una métrica definida por la Teoría de la Información, la cual se ha empleado en las Máquinas Darwinianas definidas en secciones anteriores para indicar el nivel de afinidad

$umbral_{inc} = L/2$, con L igual a la longitud del cromosoma); y (2) los operadores de recombinación generan dos memes hijos que comparten las mismas características parentales pero donde el resto de sus alelos son generados aleatoriamente (siguiendo el principio de la plantilla RRR) con el fin de mantener la diversidad.

- *Convergencia por Búsqueda Dirigida:* con el fin de balancear la diversidad a través de una convergencia dirigida, se proponen cuatro criterios que emplean conocimiento del dominio del problema: (1) la selección elitista de los mejores memes para su reproducción (como se describió en la sección 10.4.2.1); (2) el proceso filogenético ejecutará tantas generaciones como sean necesarias hasta alcanzar una aptitud promedio de la población de memes hijos superior en un 5 % a la aptitud promedio de la población inicial de memes padres. En cada generación, los memes del tipo regla (para el componenente SIA) son evaluados empleando el Sistema de Anticipación y Prospección, los memes del tipo RC son evaluados empleando únicamente la función del aspecto funcional (sección 9.3.2.1), y los memes del tipo Plan son evaluados empleando la evaluación fuera de línea y la simulacion interna (sección 10.3.4.1); (3) una vez se cumple el criterio anterior, el algoritmo verifica que todos los memeplexes (i.e., las RCs y los Planes) sean estructuralmente válidos (i.e., obtengan el máximo valor para la evaluación del aspecto estructural, como se describió en el apartado 9.3.2.1). Aquellos memeplexes que no sean válidos estructuralmente serán reemplazados por el mejor memeplexe hijo; y (4) debido a las variaciones producidas por los operadores de recombinación genética y transposición genética, el algoritmo verifica que los cromosomas meméticos hijos del tipo RC y Plan no definan *Comportamientos* redundantes, esto es, que cada gen del cromosoma se ocupe de definir un *Comportamiento* diferente de la RC respectiva. Para esto, el algoritmo recorre cada gen del cromosoma cotejando sí su primer alelo se corresponde con un identificador secuencial del *Comportamiento* asociado, y en el caso de no ser así, modifica el alelo por el identificador correspondiente. Por ejemplo, supóngase que como resultado del proceso de reproducción se ha generado un cromosoma memético del tipo RC como el siguiente: **C3... C1... C3 ... C2 ...**, el cual, una vez validado por el algoritmo sería transformado a: **C1... C2... C3 ... C4 ...**

El diagrama de actividades de la figura 10.13 esquematiza el flujo de control para los diferentes operadores que aplica el proceso filogenético. Cabe resaltar que la última actividad (*Preparar la población de nuevos Memes*), además de aplicar los diferentes criterios para la convergencia dirigida, también *"reemplaza"* la población inicial de padres por la nueva población de memes hijos producidos, de manera tal que si el proceso filogenético continúa lo hará sobre dicha base.

10.4.2.3. Fase 3: Transmisión y Asimilación de los Memes

Después de que los memes y memeplexes han sido co-evolucionados en la fase anterior, estos son transmitidos a los diferentes agentes cognitivos para su asimilación. Debido

entre dos cromosomas. Cuanto mayor sea la distancia, más diferencias existirán entre los dos cromosomas. La distancia se calcula como el número de diferencias entre el conjunto de alelos de dos cromosomas (e.g., la distancia Hamming entre 1011101 y 1001001 es 2.).

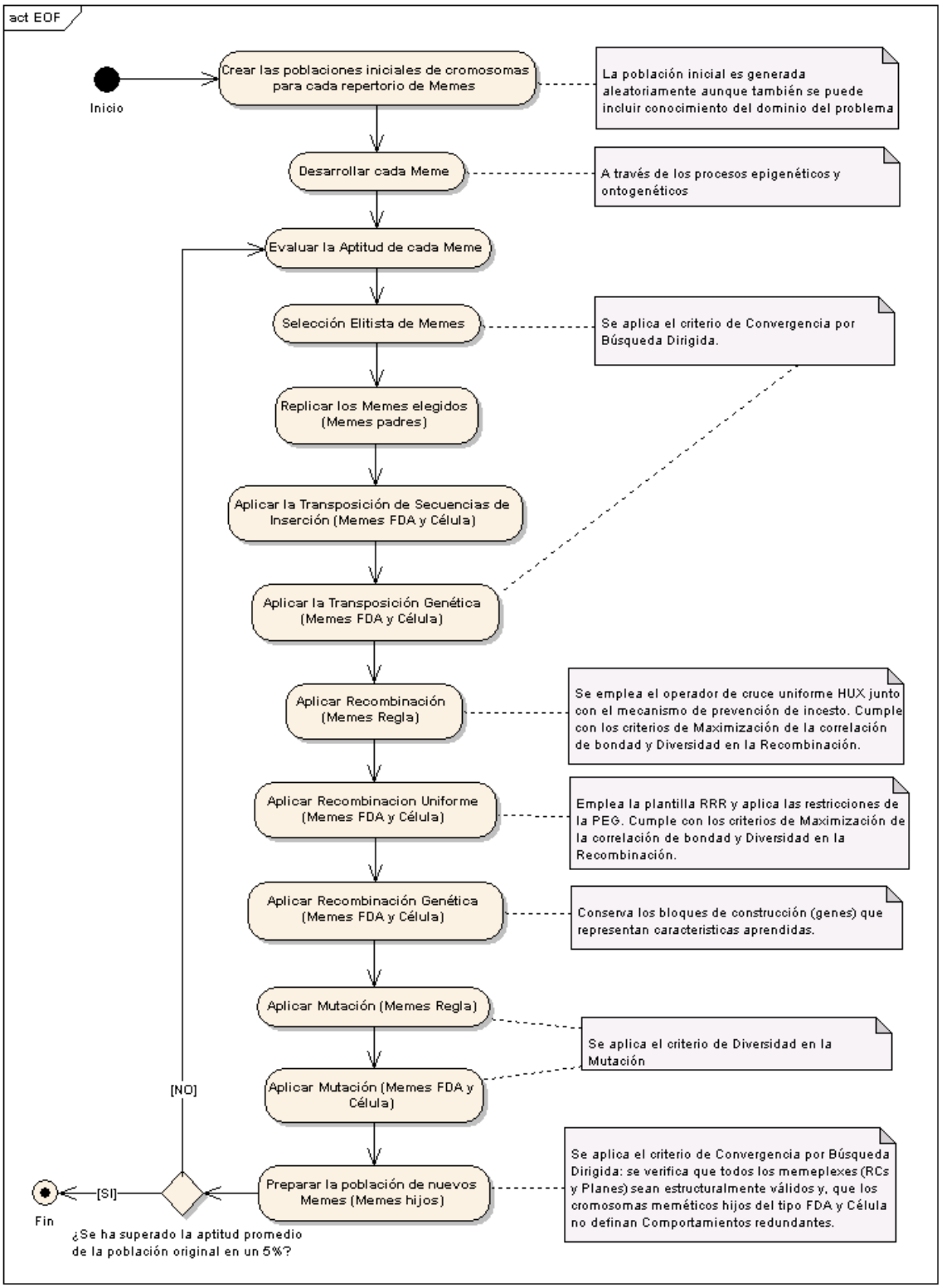


Figura 10.13: Diagrama de Actividades para el Proceso Filogenético.

a la diversidad de tipos de memes utilizados conviene precisar algunas consideraciones sobre el diseño de los mecanismos para la asimilación de nuevo conocimiento, objetivo que atañe a la presente sección.

1. Asimilación de los Memes del tipo Regla-Comportamiento: por cada uno de los *Comportamientos* que define el agente y su respectivo conjunto de memes co-evolucionados, se realiza:

1.1. Incorporación de los Memes en el Componente SIA: en este paso, el nuevo conjunto de memes co-evolucionados (cuyo tamaño debe corresponderse con la misma cantidad de memes extraídos durante la fase 1[14]), sustituye un conjnto de anticuerpos menos adaptados seleccionados de la población original, planteándose para ello una adaptación al modelo de *hacinamiento (crowding)* propuesto originalmente por (DeJong, 1975). Así pues, la idea básica del mecanismo de sustitución propuesto consiste en reemplazar de la población original un anticuerpo (meme) que sea lo suficientemente semejante a, y con menor nivel de aptitud que, el meme que lo sustituirá. De esta manera los nuevos memes tienden a competir únicamente con los anticuerpos de la población original con quienes tienen cierta semejanza, evitando así la convergencia prematura y la eliminación destructiva de conocimiento (debido al reemplazo de anticuerpos potencialmente buenos). A continuación, se exponen las pautas a tomar en consideración en el modelo de hacinamiento:

- Por cada uno de los memes nuevos, se seleccionan aleatoriamente de la población de anticuerpos (Pob) tres sub-poblaciones ($SubPob_1$, $SubPob_2$, $SubPob_3$)[15]. El tamaño de cada sub-población corresponde por defecto al 10 % del tamaño de Pob.
- Por cada sub-población, se halla el anticuerpo que tenga menor nivel de aptitud y luego es agregado a una lista temporal ($listaTemp$).
- Luego se recorre $listaTemp$ y, por cada anticuerpo de la lista, se halla el nivel de semejanza entre éste y el meme nuevo que será insertado, utilizando para ello la métrica de la distancia Hamming. El anticuerpo que se asemeje más al meme nuevo es reemplazado por éste último.

En el algoritmo 10.5 se sintetiza el proceso propuesto para la sustitución de memes por hacinamiento. Nótese que se emplea el término genérico "individuo" para designar la estructura cognitiva que será sustituida, en este caso, un anticuerpo. El algoritmo se ejecuta por cada uno de los *Comportamientos* definidos por el agente.

[14]Por defecto, la cantidad de memes co-evolucionados equivale al 40 % del tamaño de la población de anticuerpos.

[15]Se puede emplear un número mayor de sub-poblaciones con el fin de explorar en paralelo en más espacios de búsqueda. Sin embargo, la idea fundamental es, por un lado, no emplear toda la población de anticuerpos pues esto degradaría el desempeño del algoritmo y, por otro, emplear un número no muy grande de sub-poblaciones (entre 3 y 5) cada una con un tamaño no superior al 10 % del tamaño de la población de anticuerpos sobre la cual se harán las sustituciones.

Método: sustituirMemes

```
1.  Variables
2.     Pob            ↑↓ población original de individuos.
3.     memes          ↑ conjunto de memes nuevos que serán insertados.
4.     SubPob         vector de sub-poblaciones seleccionadas aleatoriamente de Pob.
5.     listaTemp      lista de los individuos candidatos a ser reemplazados.
6.     semejanza      nivel de semejanza entre un individuo y el nuevo meme.
7.     distancia      distancia Hamming entre un individuo y el nuevo meme.
8.     id             identificador del individuo seleccionado para su sustitución.
9.  Inicio Procedimiento
10.    Para cada meme ``i'' de ``memes'' hacer:
11.       Para cada una de las sub-poblaciones ``j'' hacer:
12.          SubPob[j] ← obtenerSubPoblacionhAleatoria(Pob)
13.          listaTemp ← agregar(buscarIndvMenorAptitud(SubPob[j]))
14.       Fin Para
15.       Para cada uno de los elementos ``k'' de listaTemp hacer:
16.          distancia ← calcularHamming(listaTemp.k, memes.i)
17.          Si distancia < semejanza entonces:
18.             semejanza ← distancia
19.             id ← k.identificador
20.          Fin Si
21.       Fin Para
22.       reemplazarIndividuo(Pob, listaTemp.id, memes.i) //sustituye
          en Pob, el individuo seleccionado de listaTemp por el nuevo meme.
23.    Fin Para
24. Fin Procedimiento
```

Algoritmo 10.5: Algoritmo para la Sustitución de Memes por Hacinamiento.

Supóngase, por ejemplo, un meme nuevo como el siguiente: "1101001101". Considérese ahora que se seleccionan los anticuerpos con menor aptitud de tres sub-poblaciones (determinadas de forma aleatoria de una población de 1.000 anticuerpos), los cuales se presentan en la tabla 10.2. Debido a que el anticuerpo 833 tiene la mayor afinidad con el meme nuevo (8 semejanzas), entonces éste es elegido para su sustitución a pesar de no ser el anticuerpo con menor aptitud de los tres.

Rango de la Sub-población	Id	Anticuerpo	Aptitud	Semejanza
125 - 320	203	0001111011	308	4
432 - 689	527	0100011000	78	5
721 - 965	833	1001011101	205	8

Tabla 10.2: Ejemplo de Sustitución por Hacinamiento. Los rangos de las sub-poblaciones se seleccionan aleatoriamente de la población de anticuerpos. *Id* representa el índice del anticuerpo con menor aptitud de la sub-población respectiva.

Una vez que ha finalizado el proceso de inserción de los memes nuevos, estos son consolidados y asimilados por los componentes restantes en tiempo de ejecución, esto es, durante las siguientes etapas de aprendizaje y desarrollo dirigidas por los procesos epigenéticos y ontogenéticos respectivamente. Estos dos procesos se exponen a continuación.

1.2. Consolidación de los Memes a través del Aprendizaje Vertical Ascendente: el conjunto de memes nuevos que ha sido agregado a cada componente SIA, es extraído a través del proceso de extracción de reglas sub-simbólicas (explicado en la sección 7.5.2). Con el transcurso del tiempo, el agente especializará y/o generalizará tales memes mediante la interacción con el entorno. El objetivo de la consolidación es, por tanto, la adquisición explícita de nuevo conocimiento que puede ser explicado a través de representaciones simbólicas, en contraste con el aprendizaje implícito de representaciones sub-simbólicas llevado a cabo durante el proceso filogenético.

1.3. Asimilación de los Memes a través del Aprendizaje Vertical Descendente: por último, después de que el conocimiento simbólico ha sido refinado por el proceso anterior, se ejecuta un aprendizaje descendente (como se explicó en la sección 7.8), donde el componente RES orienta la asimilación de nuevo conocimiento en el componente PMR en forma de pautas neuronales. Adicionalmente, el componente SIA refina, mediante la retroalimentación, aquellos memes nuevos que han sido insertados dentro de su población en forma de anticuerpos.

2. Asimilación de los Memes del tipo Regla-SA, RC, y del tipo Plan: el proceso de asimilación para este tipo de memes se efectúa de manera muy similar que para los memes de Comportamiento, por lo que se emplea también el algoritmo 10.5. La única diferencia con el proceso de asimilación descrito anteriormente es que los memes del tipo Regla-SA, RC, y Plan no requieren de sincronización con otros elementos, razón por la cual no definen mecanismos para el aprendizaje ascendente o descendente.

10.4.3. Ciclo de Vida de los Agentes Cognitivos

Cada agente cognitivo atraviesa por una serie de estados que determinan, en particular, los diferentes procesos cognitivos llevados a cabo internamente y, en general, su ciclo de vida determinado por los procesos EOF. A continuación, se exponen brevemente cada uno de estos estados cuya interacción se esquematiza en la figura 10.14 mediante un diagrama de transición de estados.

- **En Inicialización:** en este estado se crea el agente cognitivo y se generan sus diferentes repertorios de estructuras cognitivas. Adicionalmente, los diversos parámetros que determinan el comportamiento de cada sub-sistema, módulo, y componente del agente son cargados desde un fichero donde se han almacenado previa-

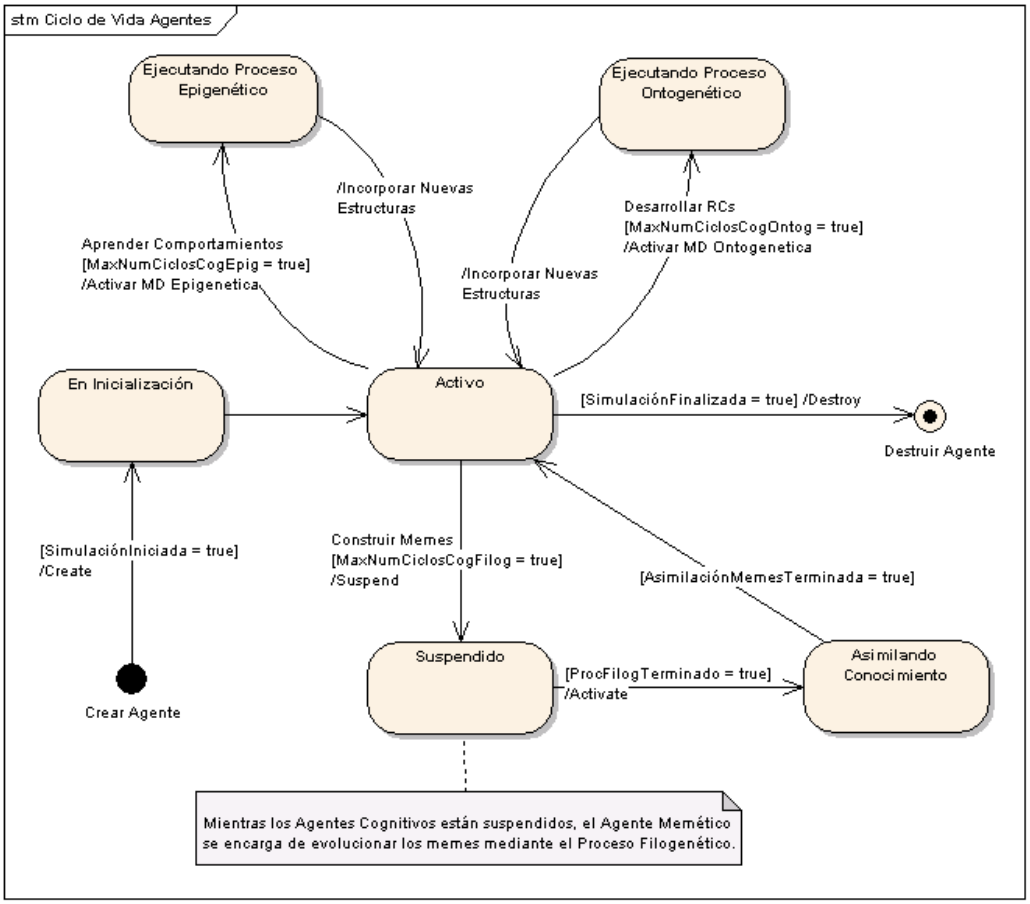


Figura 10.14: Diagrama de Transición de Estados para el Ciclo de Vida de un Agente Cognitivo.

mente las características de la configuración[16]. No obstante, cabe recordar que la inicialización de ciertas estructuras cognitivas se realiza a través de un mecanismo estocástico; este es el caso, por ejemplo, de los repertorios de anticuerpos (SIA), los repertorios de clasificadores (SA), los repertorios de MATs (RCs), y los repertorios de Planes.

- **Activo:** mientras que el agente se encuentre en este estado ejecutará múltiples ciclos cognitivos los cuales involucran, comúnmente, procesos como la interpretación de la información sensorial, la recuperación del conocimiento declarativo, la orquestación de la atención selectiva, la activación de las metas y motivadores, la activación de los mecanismos de razonamiento simbólico y sub-simbólico, la generación del estado anticipado, y la ejecución de la acción integrada, entre otros; en términos generales, durante este estado el agente ejecuta ciclos cognitivos continuos de la forma: $\langle$*percepción–procesamiento–actuación*$\rangle$.

[16]Normalmente, este fichero debe ser creado por el diseñador de la concreción de la arquitectura, empleando para ello una interfaz de usuario del módulo meta-cognitivo que le permite definir los parámetros de configuración.

- **Ejecutando Proceso Epigenético:** en este estado se inicializa el proceso epigenético por el cual se refinan, a través del aprendizaje, las estructuras cognitivas de cada *Comportamiento*, de cada componente SA, y de cada componente de memoria declarativa. La condición para que se efectúe la transición a este estado es que se haya alcanzado un número máximo de ciclos cognitivos (épocas de aprendizaje) que enmarcan una generación epigenética completa. Cabe resaltar que, aunque este estado represente la creación de un nuevo hilo de procesamiento, el hilo principal (aquél descrito en el estado anterior) se mantiene en ejecución.
- **Ejecutando Proceso Ontogenético:** este estado inicializa el proceso por el cual las estructuras cognitivas más complejas (RCs y Planes) son desarrolladas y evolucionadas en tiempo de ejecución. La condición que determina la transición a este estado establece la culminación de un cierto número de ciclos cognitivos correspondientes a una generación ontogenética completa. De la misma manera que el estado anterior, durante la activación de este estado se mantiene en ejecución, además, el hilo principal de procesamiento.
- **Suspendido:** en este estado la actuación del agente es suspendida, esto es, se detiene el hilo principal que procesa cada ciclo cognitivo típico. La condición que determina la transición a este estado valida que se haya alcanzado un número máximo de ciclos cognitivos correspondientes a una generación filogenética completa. Antes de suspender la actuación del agente, éste efectúa el proceso de construcción y envío de Memes (como se explicó en la sección 10.4.2.1). El agente se mantiene en estado *Suspendido* hasta que el proceso filogenético haya terminado de co-evolucionar todos los repertorios de memes.
- **Asimilando Conocimiento:** durante este estado, el agente es re-activado con el fin de iniciar el proceso de asimilación de nuevos memes (como se describió en la sección 10.4.2.3). Después de que el conocimiento ha sido asimilado por el agente, el estado *Asimilando Conocimiento* efectúa una transición hacia el estado *Activo*, donde se re-inicializa el hilo principal de procesamiento.

Es importante destacar el hecho de que los agentes cognitivos no son destruidos y creados nuevamente como parte del proceso reproductivo que intenta simular la filogénesis, la cual, en términos biológicos, sí que implica hechos como el destruir los organismos más viejos y/o débiles (muerte) y el crear una nueva descendencia (nacimiento de nuevos organismos) en cada generación reproductiva. Por el contrario, la filogénesis se aplica, como se ha podido intuir durante del desarrollo de la presente propuesta, únicamente sobre las estructuras cognitivas pertinentes, quienes sí que definen ciclos de vida circunscritos a las dinámicas evolutivas de cada Máquina Darwiniana. Por tanto, la creación o destrucción de un agente cognitivo está sujeta únicamente a las condiciones de inicio o terminación de la simulación total del sistema.

10.5. Tiempos de Respuesta

Con el fin de proveer una métrica fiable que permita estimar el desempeño general del modelo cognitivo propuesto, se plantea un mecanismo justificable y sistemático coordinado por el módulo Metacognitivo, el cual determina una aproximación de los tiempos

de respuesta (TR) requeridos por cada ciclo cognitivo. La estimación de los tiempos de respuesta es empleada en aquellos casos donde el sistema no dispone de datos históricos sobre la duración real de ciertos acontecimientos (e.g., los tiempos de respuesta para una simulación donde se ejecutan múltiples agentes en la misma máquina no serán los mismos que los que arrojaría la misma simulación pero ejecutando únicamente un agente en cada máquina, es decir, en un enfoque puramente concurrente). Adicionalmente, los TRs resultan útiles en la evaluación de la duración de un plan simulado, como se describió en la sección 10.3.4.1.

10.5.1. Tiempos de Respuesta para el Módulo Comportamental

Como es de intuir, el tiempo de respuesta global ($TR_{(c)}$) para este módulo es una función que involucra los tiempos de respuesta (en milisegundos ms) de cada componente del *Comportamiento* que se encuentra activo.

$$TR_{(c)} = f(TR_{(PMR)}, TR_{(SIA)}, TR_{(RES)}, TR_{(RP)})(ms) \tag{10.15}$$

En el caso de que se establezca la *Selección Estocástica* como mecanismo de integración (véase la sección 7.7), el tiempo de respuesta del *Comportamiento* activo correspondería únicamente a la función TR del componente activo. Por el contrario, si se establece como mecanismo de integración la *Combinación Basada en Pesos* o la *Agregación por Votación* entonces la función TR correspondería a la suma de los tiempos de respuesta de cada componente activo del *Comportamiento*.

El TR de cada componente corresponde a una función donde intervienen tres aspectos variables:

$$TR_{(i,c)} = f(TP_{(i,c)}, TD_{(i,c)}, TA_{(i,c)})(ms) \tag{10.16}$$

donde: i es el componente activo del *Comportamiento*, c es el ciclo cognitivo actual, TP es el tiempo que toma el procesamiento perceptual. TD es el tiempo estimado para la toma de decisiones. TA es el tiempo que toma la ejecución de la acción.

A partir de la función 10.16 se puede formular, de forma general, una ecuación que refleje la manera en que se relacionan tales aspectos variables. Por defecto, se considera una función aditiva entre dichas variables, es decir:

$$TR_{(i,c)} = TP_{(i,c)} + TD_{(i,c)} + TA_{(i,c)}(ms) \tag{10.17}$$

Debido a que los tiempos de respuesta para cada aspecto del componente pueden variar dependiendo de factores como: la plataforma tecnológica empleada para la simulación de los procesos cognitivos, el tipo de procesamiento y la complejidad algorítmica correspondiente a cada componente, la cantidad de información que es procesada, entre otros; se propone la utilización de un conjunto de parámetros cuyos valores por defecto están soportados por diferentes tipos de experimentos en el área de la neurociencia y la neurocomputación.

Por defecto, y tomando como referencia los estudios de (Tanaka y Sagi, 1998; Rubinstein, 2005), los tiempos de respuesta para el procesamiento perceptual en los componentes que se encuentran en el nivel emergente del *Comportamiento* se determinan en 50 ms, esto es, $TP_{(PMR)} = 50ms$ y $TP_{(SIA)} = 50ms$. Como consecuencia, los tiempos de respuesta del nivel cognitivista se definen como:

$$TP_{(RES)} = TP_{(RP)} = TP_{(PMR)} + 50ms \tag{10.18}$$

Esto se debe a que existe evidencia sobre el hecho de que antes de producirse el procesamiento consciente de la información perceptual se efectúa una gran cantidad de pre-procesamiento subconsciente (Marcel, 1983). Por tanto, es razonable asumir que los procesos que acceden conscientemente a la información perceptual (desde el nivel cognitivista) requieren más tiempo que el acceso subconsciente de la misma información (desde el nivel emergente). Asimismo, ha sido demostrado experimentalmente que una intención iniciada de forma subconsciente la cual es luego transformada en una intención consciente, puede tomar desde cien hasta varios cientos de milisegundos (Libet, 1985). Tomando en cuenta estos dos puntos de vista se han definido los tiempos de respuesta para el procesamiento perceptual en cada componente.

Por otro lado, el valor del TR para el proceso de toma de decisiones llevado a cabo por los componentes del nivel emergente (i.e., $TD_{(PMR)}$ y $TD_{(SIA)}$) se asume como un valor constante. Esto se debe a que el proceso subconsciente para la toma de decisiones es preferiblemente directo (se produce directamente desde un estado hacia una acción concreta) y, por consiguiente, es más rápido. De acuerdo con (Libet, 1985), los procesos cognitivos que toman más de 150 ms pueden ser considerados procesos de toma de decisiones "conscientes", con lo cual se propone, para los componentes del nivel emergente, un valor inferior a este límite experimental de la siguiente manera:

$$TD_{(PMR)} = TD_{(SIA)} = 100ms \tag{10.19}$$

En cuanto a la determinación de los tiempos de respuesta del proceso de toma de decisiones en el nivel cognitivista (i.e., $TD_{(RES)}$ y $TD_{(RP)}$) se propone una métrica basada en los principios citados por (Anderson, 1993b; Sun et al., 2001). Concretamente, se asume que el TR correspondiente a emplear una *regla de producción* o un *chunk de acción* es inversamente proporcional a la posibilidad (no la probabilidad) de que dicha regla o chunk sea requerido en el ciclo cognitivo actual, basándose para esto en su utilización en situaciones pasadas. La justificación cognitiva para esto es que cuanto más alta sea la ANB de un chunk o una regla simbólica (lo cual indicaría que ésta ha sido recurrente y, por tanto, más accesible a través de la MT), el TR requerido será más bajo puesto que no se requiere de un tiempo adicional de procesamiento para las dinámicas típicas de selección, activación e inferencia de dicha regla o chunk. Esto es:

$$TD_{(RES)} = TD_{(RP)} = \left(t1 \times \frac{1}{ANB_{(regla)}}\right) + \left(t2 \times \frac{1}{ANB_{(chunk)}}\right)(ms) \tag{10.20}$$

donde: $t1$ y $t2$ son constantes (por defecto iguales a 100 ms), que determinan la diferencia entre los TRs de las reglas y los chunks.

Esta fórmula indica dos clases de estímulos desencadenantes (priming effects): el estímulo desencadenante para la regla aplicada y el efecto desencadenante para el chunk seleccionado. Como se describió en 7.5, la activación de nivel base (ANB) representa la posibilidad de necesitar una unidad de información basada en la historia de su uso pasado (Anderson, 1993b). Por tanto, la $ANB_{(regla)}$ y la $ANB_{(chunk)}$ representan la posibilidad de que sea requerida (en el ciclo cognitivo actual) una regla o un chunk respectivamente.

Por último, el TR relacionado con la ejecución de la acción es variable y depende de una multitud de factores, entre ellos la modalidad de respuesta utilizada (e.g., respuestas verbales, respuestas motoras, respuestas internas, etc.). Por ejemplo, tomando como referencia los estudios de (Anderson y Lebiere, 1998) se propone que los TRs para las acciones motoras sean por defecto: $TA_{(PMR)} = TA_{(SIA)} = TA_{(RES)} = TA_{(RP)} = 100ms$; mientras que para las respuestas internas (acciones que no producen respuestas motoras, e.g., sumar dos números), se propone que los TRs sean: $TA_{(PMR)} = TA_{(SIA)} = TA_{(RES)} = TA_{(RP)} = 50ms$. Adicionalmente, estos tiempos de respuesta relacionados con la ejecución de acciones pueden variar dependiendo del dominio del problema, razón por la cual pueden ser determinados individualmente por el diseñador de la concreción contextual de la arquitectura.

Este mecanismo para la determinación de los tiempos de respuesta puede ser extendido para tratar una gran variedad de cuestiones relacionadas con el desempeño del sistema. Por ejemplo, los conocidos pros y contras de los factores de *velocidad* vs. *precisión* pueden ser modelados utilizando este mecanismo. Para esto, se requiere hacer algunos supuestos. Asúmase que se emplea el mecanismo de modulación de acciones por *selección estocástica*. Ahora, supóngase que $TD_{(RES)}$ y $TA_{(RP)}$ es una función que crece monótonamente para $t1$ y $t2$. Asúmase también que la temperatura (grado de aleatoriedad) en la selección de acciones y/o de reglas es una función en términos de $t1$ y $t2$ tal que:

$$\tau_{(NC)} = g1(t1, t2)(ms) \tag{10.21}$$

donde: $\tau_{(NC)}$ es la temperatura del nivel cognitivista (i.e., para los componentes RES y RP). $g1$ es una función que decrece monótonamente en términos de $t1$ y $t2$ (los cuales pueden variar según el dominio del problema).

Bajo estos supuestos, si $t1$ y $t2$ son menguados, esto es, si el tiempo para la toma de decisiones (i.e., $TD_{(RES)}$) es reducido, entonces, la temperatura para la toma de decisiones se incrementará y como consecuencia la precisión del desempeño del sistema disminuirá. De aquí la dualidad entre velocidad y precisión. De manera similar, en el nivel emergente, se asume que los tiempos de respuesta para la toma de decisiones son variables, y por tanto, la temperatura para la selección de acciones es una función que se formula en términos del componente emergente seleccionado ($TD_{(PMR)}$ o $TD_{(SIA)}$):

$$\tau_{(NE)} = g2(TD_{(comp)})(ms) \tag{10.22}$$

donde: $\tau_{(NE)}$ es la temperatura del nivel emergente. $g2$ es una función que decrece monótonamente en términos de $TD_{(comp)}$. $comp$ ser alguno de los componentes del nivele emergente (PMR o SIA.

Por tanto, si $TD_{(comp)}$ es reducido, la temperatura incrementará y consecuentemente la precisión del desempeño del sistema decrecerá[17].

10.5.2. Tiempos de Respuesta para el Módulo Declarativo

Primero que todo, considérese el nivel cognitivista del módulo Declarativo. El tiempo de recuperación de un *chunk* es inversamente proporcional a su activación de nivel base (ANB) (Sun et al., 2001). Esto es, el tiempo de recuperación de un *chunk* se define como:

$$t_{(chunk)} = t2 + \big(t3 \times \frac{1}{ANB_{(chunk)}}\big)(ms) \tag{10.23}$$

donde: $t2$ y $t3$ son dos constantes temporales que por defecto tienen un valor igual a 50 ms y 100 ms respectivamente.

De la misma manera, el tiempo de aplicación de una *regla asociativa* es inversamente proporcional a su activación de nivel base, esto es:

$$t_{(regla)} = t4 + \big(t5 \times \frac{1}{ANB_{(regla)}}\big)(ms) \tag{10.24}$$

donde: $t4$ y $t5$ son dos constantes temporales que deben ser ajustadas para cada dominio del problema, y que por defecto tienen un valor de 50 ms y 100 ms respectivamente.

La aplicación de una *regla asociativa* puede implicar también, como resultado, la recuperación de un conjunto de *chunks*. Por tanto, en este caso, resultaría necesario también sumar los tiempos relativos a la recuperación de los *chunks* asociados (como se describe en la ecuación 10.23).

Debido a que todas las *reglas asociativas* seleccionadas son ejecutadas en paralelo en el nivel cognitivista del módulo Declarativo (i.e., las memorias semánticas y episódicas), el TR resultante de aplicar una *regla asociativa* es el máximo tiempo que tomaría aplicar una *regla asociativa* y sus correspondientes *chunks* asociados:

$$TR_{(MD)} = \underset{x \to r}{\text{máx}}(t_{(regla)} + t_{(chunk)})(ms) \tag{10.25}$$

donde: $TR_{(MD)}$ es el tiempo de respuesta para cada componente del módulo Declarativo (memorias semánticas y/o memorias episódicas del nivel cognitivista). $x \to r$ son todas las reglas asociativas aplicables dado un estado percibido "x".

Ahora, considérese el nivel emergente del módulo Declarativo. Debido a que las redes neuronales empleadas en los componentes RMA y MEA (memorias semánticas y episódicas respectivamente) realizan un procesamiento directo e implícito entre las entradas y las salidas de la red, esto es, que durante el procesamiento de la información

[17] En el caso de emplear el mecanismo de *Combinación Basada en Pesos* o el mecanismo de *Agregación por Votación*, la temperatura (τ) puede ser especificada de manera similar, como una función de la velocidad de respuesta del sistema.

no se realiza ningún tipo de interpretación o traducción de las entradas percibidas ni de las acciones ejecutadas sino que éstas son procesadas simplemente como información sub-simbólica; se definen por tanto los tiempos de respuesta como valores constantes determinados por el número de elementos a procesar (i.e., el número de neuronas en cada capa). Se tiene entonces que el TR para cada red neuronal activa es:

$$TR_{(RN)} = (t6 \times nn_{(ce)}) + (t7 \times nn_{(co)}) + (t8 \times nn_{(cs)})(ms) \tag{10.26}$$

donde: $t6$, $t7$, y $t8$ son constantes temporales que dependen del dominio del problema y cuyos valores por defecto son: $t6 = 5ms$, $t7 = 10ms$, y $t8 = 5ms$ ($t7$ requiere de más tiempo debido a que participa en los procesos de activación sináptica hacia adelante y en la retropropagación del error). $nn_{(ce)}$ es el número de neuronas en la capa de entrada. $nn_{(co)}$ es el número de neuronas en la capa oculta. $nn_{(cs)}$ es el número de neuronas en la capa de salida.

Cabe mencionar, que este esquema de estimación es compatible con el descrito para el componente PMR del módulo Comportamental. Esto es, que la sumatoria de los diferentes tiempos de respuesta que define el componente PMR ($TP_{(PMR)}$, $TD_{(PMR)}$, y $TA_{(PMR)}$), es equivalente a la sumatoria de los tiempos de respuesta de cada capa de la red neuronal (como se propone para $TR_{(RN)}$). Así pues, para efectos de simplificación, los TRs del componente PMR se han tomado como valores constantes aunque bien podría utilizarse la ecuación 10.26 para estimar dichos TRs.

10.5.3. Tiempos de Respuesta para el Módulo de Consciencia

El TR de este módulo está condicionado por los procesos que acceden a los contenidos de la Memoria de Trabajo y por los procesos relacionados con la orientación de la Atención. Debido a que el acceso a los contenidos de la MT se realiza a través de procesos subconscientes y automáticos que por lo general operan imbricadamente en el transcurso de múltiples ciclos cognitivos (e.g., la recuperación y almacenamiento temporal de conocimiento desde las memorias de largo plazo), resultaría redundante, en la mayoría de los casos, el estimar un tiempo de respuesta diferente para cada uno de los procesos involucrados. Por tanto, y tomando como referencia la evidencia aportada por (Baddeley, 1991), se propone un valor constante para el TR de los procesos que acceden directamente (de forma subconsciente y automática) a los contenidos de la MT, tal que: $TR_{(MT)} = 50ms$.

Por otro lado, el TR que corresponde al proceso de orientación de la Atención se basa en los tiempos requeridos por las dinámicas de dispersión de activación de la MAT que se encuentra activa en cada ciclo cognitivo. Así pues, el TR para el proceso atencional es un valor que varía proporcionalmente con el número de *Comportamientos* que contiene la RC asociada a la MAT. Tomando como referencia los resultados presentados por (Maes, 1989), se propone que cada ciclo de ejecución del mecanismo atencional, esto es, un ciclo completo de dispersion de energía de activación dentro de la RC, tomaría un tiempo de respuesta:

$$t_{(RC)} = t9 \times nc(ms) \tag{10.27}$$

donde: $t9$ es una constante temporal con un valor por defecto igual a 5 ms. nc es el número de *Comportamientos* de la RC activa.

Recuérdese que la RC puede ejecutar internamente múltiples ciclos de ejecución hasta que finalmente obtiene un *Comportamiento* ejecutable (i.e., un *Comportamiento* es ejecutable si cumple con: tener el mayor nivel de activación, satisfacer todas sus precondiciones, y superar un umbral de activación). Por tanto, en cada ciclo cognitivo puede que se requieran de múltiples ciclos de ejecución orquestados por la RC activa (en el caso de no cumplirse alguna de las condiciones para que un *Comportamiento* sea ejecutable). Empíricamente, se estima que en el mejor de los casos el algoritmo de la RC requerirá únicamente de un ciclo de ejecución, y en el peor de las casos, de diez. Se tiene entonces que:

$$\begin{aligned} TR_{(RC)} &= \frac{(1 \times t_{(RC)}) + (10 \times t_{(RC)})}{2}(ms) \\ TR_{(RC)} &= 5.5 \times t_{(RC)}(ms) \end{aligned} \tag{10.28}$$

10.5.4. Una Nota General sobre la Estimación de los TRs

Nótese que, por motivos de simplicidad y uniformidad, el TR para cada ciclo cognitivo es calculado como una serie de unidades de tiempo, cada una de las cuales es determinada como la suma de los tiempos de respuesta requeridos para el procesamiento perceptual, la extracción de información de las memorias de largo plazo, la orientación de la atención, la toma de decisiones, y la ejecución de las acciones. Esto es, los tiempos de respuesta son calculados paso a paso y sin ningún tipo de interacción entre pasos. Sin embargo, en la realidad, pueden existir interacciones más complejas (e.g., el tiempo adicional que toman las dinámicas epigenéticas del SIA cada cierto número de ciclos cognitivos) las cuales no son consideradas en el esquema simplificado expuesto anteriormente.

Adicionalmente se puede observar que el mecanismo de estimación de TRs propuesto define, para cada sub-sistema que participa durante la ejecución de un ciclo cognitivo simple, los elementos que influyen de manera más relevante sobre los tiempos de respuesta respectivos. No obstante, en los TRs de cada sub-sistema intervienen en realidad muchos otros elementos que pueden incrementar o disminuir los tiempos de respuesta totales, pero que se han descartado debido a que su inclusión no representaría cambios significativos sobre la estimación final. Por otro lado, se puede notar que existe cierta redundancia en los elementos involucrados en la estimación de los diferentes TRs. Esto se debe, por un lado, a que los sub-sistemas comparten ciertos elementos comunes (e.g., la información perceptual) y, por otro, a que los TRs de ciertos sub-sistemas dependen de los TRs de otros (e.g., existe una dependencia evidente entre los TRs del nivel emergente y los TRs del nivel cognitivista, en especial cuando los procesos de aprendizaje del primero son supervisados por el segundo).

Nótese también, que el mecanismo de estimación de los TRs considera únicamente la ejecución de ciclos cognitivos simples, esto es, ciclos de la forma $\langle$*percepción–procesamiento–actuación*$\rangle$, desestimando por tanto la ejecución de los hilos de procesamiento correspondientes a los procesos ontogenéticos y filogenéticos. El motivo principal para esto, es que tales procesos no se ejecutan siempre en cada ciclo cognitivo simple sino

que, por el contrario, se activan solamente cada cierto número de ciclos cognitivos (denominados épocas de aprendizaje). Por tanto, en aquellos casos donde se desee estimar el tiempo total requerido por el agente para llevar a cabo una tarea, se deberá primero calcular el tiempo total correspondiente a la sumatoria de los TRs de los múltiples ciclos cognitivos simples, y luego se agregará, como un valor constante, el tiempo correspondiente a la ejecución de cada uno de los procesos ontogenéticos y filogenéticos que tuvieron lugar durante la realización de la tarea.

En este orden de ideas, a continuación se estima el TR de un ciclo cognitivo simple tanto para un escenario optimista (i.e., el TR total para el mejor de los casos), como para un escenario pesimista (i.e., el TR total para el peor de los casos). Entonces:

$$\begin{aligned} TR_{(total,opt)} &= \big(TP_{(ce)} + TD_{(ce)} + TA_{(ce)}\big) + TR_{(RN)} + t_{(RC)}(ms) \\ TR_{(total,opt)} &= 50ms + 100ms + 50ms + 50ms + 50ms = 300ms \end{aligned} \tag{10.29}$$

donde: ce corresponde cualquiera de los componentes del nivel emergente del *Comportamiento*.

En el escenario optimista, se considera la activación de únicamente un componente del nivel emergente del *Comportamiento*, debido a que el procesamiento de la información se realiza de forma subconsciente y automática (como se discutió anteriormente), produciendo un TR más pequeño. En cuanto a la extracción de conocimiento de las memorias de largo plazo solamente se consideran las memorias del nivel emergente por la misma razón que el anterior punto. Y por último, con respecto al módulo de Consciencia, se considera que la RC activa contiene diez *Comportamientos* y que sólo requiere de un ciclo de ejecución para determinar el *Comportamiento* ejecutable. Se estima entonces que el ciclo cognitivo simple en el mejor de los casos tomaría aproximadamente 300 ms.

$$\begin{aligned} TR_{(total,pes)} &= \big(2 \times TR_{(ce)}\big) + \big(2 \times TR_{(cc)}\big) + \big(TR_{(RN)} + TR_{(MD)}\big) + TR_{(RC)}(ms) \\ TR_{(total,pes)} &= (2 \times 200ms) + (2 \times 350ms) + (50ms + 100ms) + (10 \times 50ms) \\ TR_{(total,pes)} &= 1.750ms \end{aligned} \tag{10.30}$$

donde: ce es un componente del nivel emergente del *Comportamiento*. cc es un componente del nivel cognitivista del *Comportamiento*.

Por otro lado, en el escenario pesimista, se considera que todos los componentes del *Comportamiento* estarían activos y que se utilizaría alguno de los mecanismos de integración de acciones (e.g., la combinación basada en pesos o la agregación por votación). En cuanto al módulo declarativo, se asume que se extraería conocimiento de los dos niveles (emergente y cognitivista). Y por último, se considera que la RC activa requeriría de diez ciclos de ejecución para determinar el *Comportamiento* ejecutable. Se estima entonces que el ciclo cognitivo simple tomaría en el peor de los casos (i.e., cuando operan concurrentemente los procesos conscientes y subconscientes), aproximadamente 1.750 ms.

En el caso de requerirse una estimación más aproximada de los tiempos de respuesta, tendría que tomarse en consideración algunos aspectos de interacción e imbricación

entre los diferentes tipos de procesos (e.g., el procesamiento perceptual, la extracción de conocimiento de las memorias de largo plazo, la toma de decisiones, y la ejecución de acciones) como lo sugieren (Kieras y Meyer, 1998; Byrne y Anderson, 1998), entre otros. Según estos autores, los procesos de percepción, decisión, y actuación en un sistema cognitivo real forman un flujo secuencial que va desde un estadio hasta el siguiente. Sin embargo, el proceso de actuación que se efectúa sobre una tarea puede solaparse con los procesos de percepción y decisión efectuados sobre otra tarea: en otras palabras, estos procesos pueden ser concurrentes (Kieras y Meyer, 1998).

Las diferentes modalidades de la actuación (e.g., los movimientos de la mano izquierda vs. los movimientos de la mano derecha) pueden ser también paralelizados de forma significativa. Este argumento se aplica tanto si las acciones de las diferentes modalidades se ejecutan sobre la misma tarea como si no. En este sentido, se necesitaría tratar al detalle muchos otros aspectos: por ejemplo, el TR para la percepción y el TR para la decisión podrían, o no, solaparse (dependiendo del dominio del problema). Si el TR para la actuación de un ciclo cognitivo tarda demasiado, probablemente la actuación del siguiente ciclo tenga que esperar hasta que finalice la actuación del ciclo anterior (en especial si están comprometidos recursos motores compartidos). Como consecuencia, el tiempo de preparación para las acciones motoras es amalgamado con el tiempo de decisión, lo que puede ser o no apropiado, y así sucesivamente. Por derecho propio, estos aspectos pueden ser muy interesantes en el campo de estudio de las ciencias cognitivas. Sin embargo, la exploración al detalle de estos probablemente no arroje información adicional sobre la interacción entre el conocimiento explícito e implícito, o acerca de la interacción entre los diferentes módulos de METÁFORA, aspectos en los cuales se enfoca el mecanismo propuesto para la estimación de los tiempos de respuesta.

Cabe mencionar que, los tiempos de ejecución pueden estimarse, *grosso modo*, en función del orden de complejidad algorítmica de los diferentes componentes y módulos de la arquitectura. Una vez efectuado el análisis de eficiencia algorítmica, se determina que los diferentes componentes de METÁFORA poseen ordenes de complejidad constantes, logarítmicos, lineales, y polinomiales, sin alcanzar ordenes exponenciales o factoriales, con lo cual además de ser algoritmos factibles y tratables computacionalmente para un número creciente de parámetros de entrada, son algoritmos cuyos tiempos de respuesta pueden crecer (o no) proporcionalmente en función de dichos parámetros. Es por este motivo que se han empleado funciones constantes y crecientes lineales para describir los TRs por defecto. No obstante, vale la pena resaltar que los TR pueden ser reajustados en tiempo de ejecución; esto es, el sistema continuamente captura un registro del tiempo que transcurre durante la ejecución de cada proceso atómico en un ciclo cognitivo; luego, estos tiempos son promediados y cotejados con los valores que se han definido por defecto para cada tipo de TR; y por último, en el caso de detectarse un desvío significativo en las nuevas mediciones, los TRs son actualizados con los tiempos registrados.

Como comentario final, el esquema propuesto provee entonces un valor aproximado de los tiempos de respuesta que tardaría la ejecución de cada ciclo cognitivo simple del sistema. Como consecuencia, este valor estimativo puede ser empleado como un indicador para la evaluación del desempeño global del sistema, así como un criterio que utiliza el mecanismo de simulación interna para validar la viabilidad de los planes (como se discutió en la sección 10.3.4).

10.6. Discusión

A continuación se exponen algunas observaciones que pueden servir tanto para clarificar ciertos aspectos del funcionamiento de los mecanismos del módulo Metacognitivo como para precisar algunas cuestiones que podrían ser tomadas en cuenta en extensiones posteriores de la arquitectura.

Sobre el Mecanismo de Regulación: en aras de la simplificación, se ha optado por no describir este mecanismo durante el desarrollo del presente capítulo puesto que, actualmente, su funcionalidad principal se restringe solamente a permitir el ajuste de los parámetros de configuración de todos los componentes de la arquitectura a través de una interfaz de usuario. Es por tanto responsabilidad del diseñador de la concreción contextual, el definir adecuadamente la configuración inicial de los diferentes parámetros, aunque de no ser así, el sistema hará uso de los parámetros por defecto[18]. No obstante, sería interesante que esta función la complementase el agente de forma autónoma, empleando para ello algún mecanismo de meta-control, basado en meta-heurísticas, que permitiese ajustar tales parámetros en tiempo de ejecución, simulando así un nivel *auto-reflexivo*. Nuevamente, podría proponerse para ello algún modelo de computación evolutiva.

Sobre el Mecanismo de Sincronización: este mecanismo no ha sido descrito explícitamente puesto que su funcionalidad ha estado inmersa en el comportamiento de otros módulos. Concretamente, los módulos Comportamental y Declarativo emplean este mecanismo para sincronizar los componentes de los niveles cognitivista y emergente. Esta sincronización se realiza a través del aprendizaje vertical (ascendente y descendente), como se enunció en la secciones 7.8, 8.2.4, y 8.2.5.

Sobre el Mecanismo de Planificación:

- El problema de la planificación independiente del dominio es una cuestión muy compleja. Los planificadores clásicos proponen, por lo general, modelos con debilidades claramente identificables como lo son, por ejemplo, su capacidad restringida para acceder a toda la información del entorno, o el no considerar que el mundo está cambiando continuamente y que puede ser modificado por la actuación de otros agentes. Estas son, entre otras, las características que hacen que los planificadores clásicos no sean la mejor opción para este tipo de problemas (Sapena, 2005). En contraposición, el mecanismo de planificación propuesto para METÁFORA se enmarca dentro de un esquema de planificación práctica que integra estrategias heurísticas y reglas empíricas para la resolución de los problemas. Por un lado, el *proceso ontogenético* emplea estrategias heurísticas con el fin de descubrir la secuencia de tareas que subyacen al mejor plan posible, mientras que, la *simulación interna*, valida el conjunto de planes generados empleando para ello conocimiento empírico sobre la posibilidad de que ciertos cambios ambientales sucedan y sobre *la actuación presumible de otros agentes*. Estos son aspectos que la planificación clásica no suele considerar.

[18] Recuérdese que a lo largo del desarrollo de la tesis se han propuesto valores por defecto para cada uno de los parámetros de los distintos componentes.

- Con el fin de realizar una evaluación más precisa de la viabilidad y efectividad de cada plan, podría incluirse, además del proceso de simulación interna, un planificador clásico (e.g., SHOP2) el cual generaría un plan candidato que sería cotejado contra el plan generado por el proceso ontogenético y, dependiendo del enfoque establecido, los dos planes podrían unificarse en uno solo (empleando algún mecanismo de integración), o podría utilizarse el planificador clásico para supervisar y evaluar (en las fases tempranas de aprendizaje) el plan generado por el mecanismo propuesto.

- El mecanismo propuesto considera una ejecución secuencial de las tareas que conforman el plan. Esto puede convertirse en un inconveniente cuando el dominio del problema puede definir un orden no secuencial para la ejecución de las tareas. Para remediar esta situación, el mecanismo propuesto emplea la re-planificación en tiempo de ejecución. No obstante, la funcionalidad del mecanismo de planificación podría ser extendida a través de la inclusión de relaciones como las que define el lenguaje formal para la implementación de Redes de Tareas Jerárquicas, TAEMS (Horling et al., 1999). Estas relaciones (e.g., *facilitates*, *enables*, *hinders*, *produces*, etc.) permiten que las distintas tareas de un plan establezcan interacciones y condiciones entre sí determinando cuándo se deben ejecutar unas u otras. Concretamente, estas relaciones podrían ser incluidas dentro del conjunto de *nodos función* que definiría el gen homeótico de cada plan. De esta manera, la ejecución del plan no sería secuencialmente rígida sino que, a través del gen homeótico, se podrían representar planes más complejos que definirían, por ejemplo, ejecuciones cíclicas de las tareas, bifurcaciones condicionales que ejecutarían diferentes ramas del árbol jerárquico de tareas, etc.

- Cabe resaltar una vez más que el mecanismo de planificación corresponde a un proceso deliberativo que se ejecuta concurrentemente junto con el hilo de procesamiento principal que controla los ciclos cognitivos simples. Esto quiere decir que, durante la planificación pueden ejecutarse paralelamente múltiples ciclos cognitivos simples, los cuales se encargarían de procesar la información percibida de forma reactiva (o deliberativa siempre y cuando no se incluya la propia planificación que, en términos computacionales, resulta más costosa que el resto de los procesos). Es por este motivo que la simulación interna se emplea principalmente durante los procesos cognitivos de orden superior (como la planificación), mientras que los ciclos cognitivos más simples, o incluso la orquestación de la consciencia sintética, emplean como mecanismo de predicción la generación de un estado anticipado del mundo (como se discutió en la sección 8.6). No deja de ser interesante, sin embargo, el extender la funcionalidad del mecanismo de simulación para que permitiese al agente ensayar la consecuencia de futuras acciones (durante la ejecución de cada ciclo cognitivo simple), en un marco temporal que no se restrinja solamente al estado inmediatamente posterior, y que a su vez, fuese un proceso que no aumentase significativamente los tiempos de respuesta del agente.

- Es evidente que la simulación en paralelo de los múltiples planes candidatos puede ocasionar problemas de desempeño, en especial si debe considerarse una gran cantidad de información relacionada con el ambiente simulado. Al igual que se propuso en la discusión del capítulo anterior, la solución apunta hacia la utiliza-

ción de una plataforma multi-agente, donde en este caso cada agente se encargaría de simular un plan específico. En este sentido, la creciente disponibilidad de sistemas de computación concurrente, generalmente basados en clusters, permite el paralelizar con cierta facilidad este tipo de situaciones.

- Ampliando la idea que soporta al mecanismo de planificación, y desde una perspectiva más ambiciosa, podría plantearse un modelo para la generación de *conductas innovadoras*. Esto es, durante el proceso de planificación el agente encuentra la solución a un problema a partir de la combinación de elementos conocidos previamente, sobre los cuales descubre un orden subyacente que no conocía. Este mismo principio se puede aplicar a muchas otras conductas del agente, principalmente a aquellas que pertenecen al terreno de la *imaginación* y la *creatividad*. En este sentido, el proceso ontogenético sería el encargado de realizar tales combinaciones de elementos, pero esta vez, no sólo combinaría tareas como lo hace en la planificación, sino conceptos, ideas, o en el sentido más general, memes; así pues, sería el *mecanismo descubridor*, mientras que la simulación interna sería el laboratorio donde las nuevas ideas serían puestas a prueba, donde se validarían los *"qué pasaría si..."*

Parte III

Resultados

Capítulo 11

Concreción Contextual de la Arquitectura

"For now, we assume that self-evolving robots will learn to mimic human traits, including, eventually, humor. And so, I can't wait to hear the first joke that one robot tells to another robot".
– Lance Morrow.

El presente capítulo describe las características de la plataforma computacional que sirvió de apoyo a la concreción contextual de la arquitectura cognitiva. A pesar de que la funcionalidad de la arquitectura fue probada inicialmente en un entorno de vida artificial donde diversos animales artificiales (animats) competían entre sí por los recursos del ambiente, asegurando así su supervivencia (Romero y De Antonio, 2009a); un estudio más completo sobre la eficiencia, el desempeño, y la robustez de la misma fue llevado a cabo sobre un entorno robótico que propició el análisis objetivo de los resultados. La especificación de las características computacionales de dicho entorno robótico, así como la configuración de los sensores y actuadores empleados por el agente cognitivo, son objeto del presente capítulo.

11.1. Definición de la Plataforma Robótica

"All the world's a stage, And all the men and women merely players".
— William Shakespeare, As You Like It, 1623.

player/stage es una plataforma de software libre, desarrollada por un grupo internacional de investigadores en el área de robótica (*The Player Project*), basada en un modelo cliente/servidor, la cual provee un conjunto de interfaces que permiten construir aplicaciones robóticas que pueden ser usadas tanto en entornos simulados como en entornos reales

(empleando para ello robots móviles del tipo Pioneer, Khepera, entre otros[1]). Esta plataforma robótica define tres sistemas: *Player*, *Stage*, y *Gazebo*. El último de ellos, Gazebo, es un simulador 3D extraoficial, el cual no fue empleado en la experimentación que se llevó a cabo en la presente tesis.

Player es un servidor que permite controlar los dispositivos de un robot móvil y obtener información de sus sensores. Se puede considerar como una capa software que abstrae los detalles del hardware del robot, creando independencia del mismo. Los algoritmos de control del robot funcionan como clientes de Player (a través de sockets TCP/IP). De esta manera, es posible controlar el robot enviando mensajes que sigan el protocolo de comunicación de Player o (que será lo habitual) llamando a funciones de las librerías de Player, abstrayendo así los detalles de la comunicación. De este modo, Player es considerado un sistema operativo para robots (Robot OS) puesto que oculta todos los detalles de hardware. Para lograr dicho objetivo, Player define una serie de interfaces estándar, cada una de las cuales es una especificación de las distintas formas en las que se puede interactuar con alguna clase de dispositivos del robot.

Stage es un entorno en 2D donde se simulan los aspectos cinemáticos para un conjunto de robots móviles (los aspectos dinámicos como la tridimensionalidad son atendidas por Gazebo). Este sistema proporciona un mundo virtual para los robots móviles, los sensores, y varios objetos que los robots pueden detectar y manipular. Stage es normalmente usado en conjunción con Player para servir como mundo simulado a los programas de control, aunque puede ser usado también como biblioteca de simulación autónoma. Para los algoritmos de control (en este caso, los procesos cognitivos orquestados por METÁFORA) será transparente el que se esté trabajando con un robot real o con Stage. Player y Stage fueron desarrollados originalmente en el Robotic Research Lab de la USC (University of South California), pero en la actualidad son el resultado del esfuerzo conjunto de una amplia comunidad de desarrolladores [2].

Stage fue diseñado y pensado para soportar aplicaciones multi-agente, motivo por el cual dispone de modelos simples y no muy pesados computacionalmente, que intentan emular cada dispositivo con una fidelidad aceptable. De esta manera, el diseño de Stage permite crear desde simulaciones simples basadas en las rejillas bidimensionales empleadas comúnmente por los algoritmos de Inteligencia Artificial, hasta simulaciones de gran calidad. Así pues, el propósito de Stage es, por un lado, ser lo suficientemente realista como para que los mismos programas de control utilizados durante la simulación puedan ser empleados luego en robots reales y, por otro, ofrecer la suficiente robustez y eficiencia como para soportar la simulación de grandes poblaciones de robots sin que esto afecte significativamente sus tiempos de respuesta.

[1] Algunos modelos comerciales pueden encontrarse en: `http://www.mobilerobots.com/ResearchRobots/ResearchRobots.aspx`

[2] El proyecto de código abierto puede descargarse del repositorio: `http://playerstage.sourceforge.net`.

11.2. Modelo Cliente/Servidor para la concreción contextual

En esta sección se describirá brevemente el modelo cliente/servidor para la plataforma *player/stage* y su integración con el modelo cognitivo de METÁFORA. Primero que todo, es necesario precisar el propósito de algunos componentes que utiliza Player para almacenar información de configuración sobre los dispositivos del robot. En caso de utilizar Stage, estos no serán reales, sino simulados.

- *Interfaz (Interface)*: es una serie de funcionalidades que proporciona un determinado dispositivo. Especifica la manera en que operan entre sí un conjunto determinado de sensores, actuadores, y algoritmos.
- *Controlador (Driver)*: es una implementación concreta de una interfaz. Cada controlador se comunica directamente con un dispositivo de hardware, ocultando así los detalles del control y la comunicación. Las entradas y las salidas que define un controlador deben corresponderse con una o más interfaces.
- *Dispositivo (Device)*: define un tipo específico de controlador, un tipo de interfaz y una dirección (que puede ser remota) de dónde localizar el dispositivo. Cada robot puede definir más de una instancia del mismo tipo de dispositivo, y para poder interactuar con éstas es necesario subscribirse primero al tipo de dispositivo.

Por ejemplo, la interfaz *laser* define un formato para un conjunto de rangos de datos (distancias) que son detectados a través de un dispositivo láser. Por su parte, el controlador *sicklms200* provee el control de un dispositivo hardware particular, el SICK LMS200. Este controlador se encarga de la comunicación de bajo nivel con este hardware a través del puerto serial. El controlador sabe como formatear los datos recuperados para adaptarse a la interfaz *laser*. Ahora, los diferentes controladores que provee Stage para este propósito (*sicklms200*, *urglaser*, *sickpls*, etc.), cumplen todos con el comportamiento que define la interfaz *laser*. De esta manera la implementación desde el cliente se hace totalmente transparente: esto es la clave que permite hacer el código reutilizable, tanto a nivel de hardware como de software.

Cabe mencionar, que pueden haber dispositivos compartiendo el mismo controlador (pero no la misma interfaz), y que pueden haber dispositivos compartiendo la misma interfaz (pero no el mismo controlador). La documentación de Player incluye una referencia de todas las interfaces, los controladores, y los dispositivos que vienen implementados en la distribución actual del software, aunque por supuesto, el usuario puede definir los suyos propios.

Player utiliza una estructura Cliente/Servidor con el fin de comunicar datos e instrucciones entre el programa de control (en este caso sería el modelo cognitivo de METÁFORA) y el hardware. Como se mencionó anteriormente, cada dispositivo hardware del robot debe ser subscrito como un cliente del servidor Player empleando para ello un mecanismo denominado *proxy*. Los ficheros de configuración asociados con el robot (o con la simulación) se encargan de informarle al servidor Player qué dispositivos están adscritos a él, de manera que cuando se invoca el comando de ejecución se inicializa el servidor Player y se establecen todas las conexiones necesarias con los dispositivos hardware. La

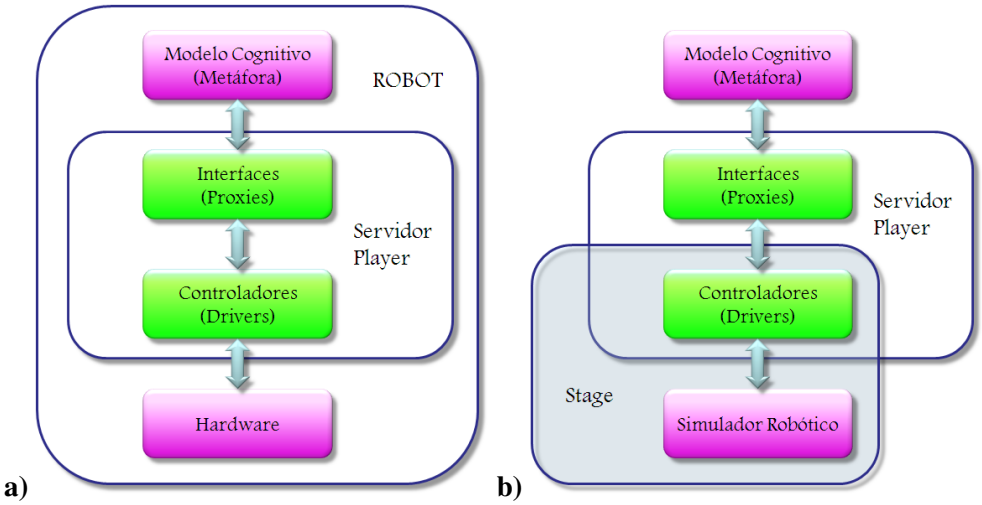


Figura 11.1: Estructura de Control del Servidor Player. a) Cuando se emplean robots reales. b) Cuando se emplea el simulador Stage. Pueden existir múltiples proxies conectados al servidor en cualquier momento.

figura 11.1.a muestra un diagrama de la estructura de Player cuando se implementa en un robot real. En este caso, nótese que tanto la implementación del modelo cognitivo propuesto como el servidor Player y todas las interfaces, controladores y dispositivos, estarían emplazados y serían ejecutados dentro del robot.

Por otro lado, en Player/Stage la invocación del comando de ejecución iniciaría el servidor Player y se cargarían, en una ventana de simulación, todos los elementos definidos en el fichero de configuración *simulacion.world*, ejecutándose de esta manera la simulación robótica en el ordenador y no en el hardware. La figura 11.1.b muestra un diagrama de la estructura básica de Player/Stage. En este caso, el componente software del modelo cognitivo podía situarse en una ubicación remota (más específicamente y como se detallará más adelante, empleando para ello un agente software que se ejecutaría preferiblemente en otra máquina). El agente cognitivo (que emplearía la arquitectura de METÁFORA) tendría control sobre el robot a través de la subscripción que realizaría a los diferentes proxies del cliente Player.

Considérese ahora la figura 11.2. En el lado del cliente se ejecutaría tanto el modelo cognitivo como el cliente Player. El modelo cognitivo, en este caso, corresponde a un agente software implementado en Java, el cual representaría una instancia específica del modelo cognitivo propuesto para METÁFORA. Por otro lado, en el cliente Player, se implementaría una clase que utilizaría librerías específicas siguiendo un modelo de programación basado en "proxies". Cada proxy es una clase de C++ y representa al propio robot o bien a uno de sus dispositivos. El cliente Player interactuaría con objetos locales (los proxies) que actuarían de "intermediarios" de los dispositivos del robot. Estos proxies pueden ser remotos y, por tanto, estar situados en cualquier máquina de la red. Así pues, desde el cliente Player se invocarían los métodos de los objetos proxy para controlar al robot u obtener información del mismo, ocultando los detalles de la comunicación por la red.

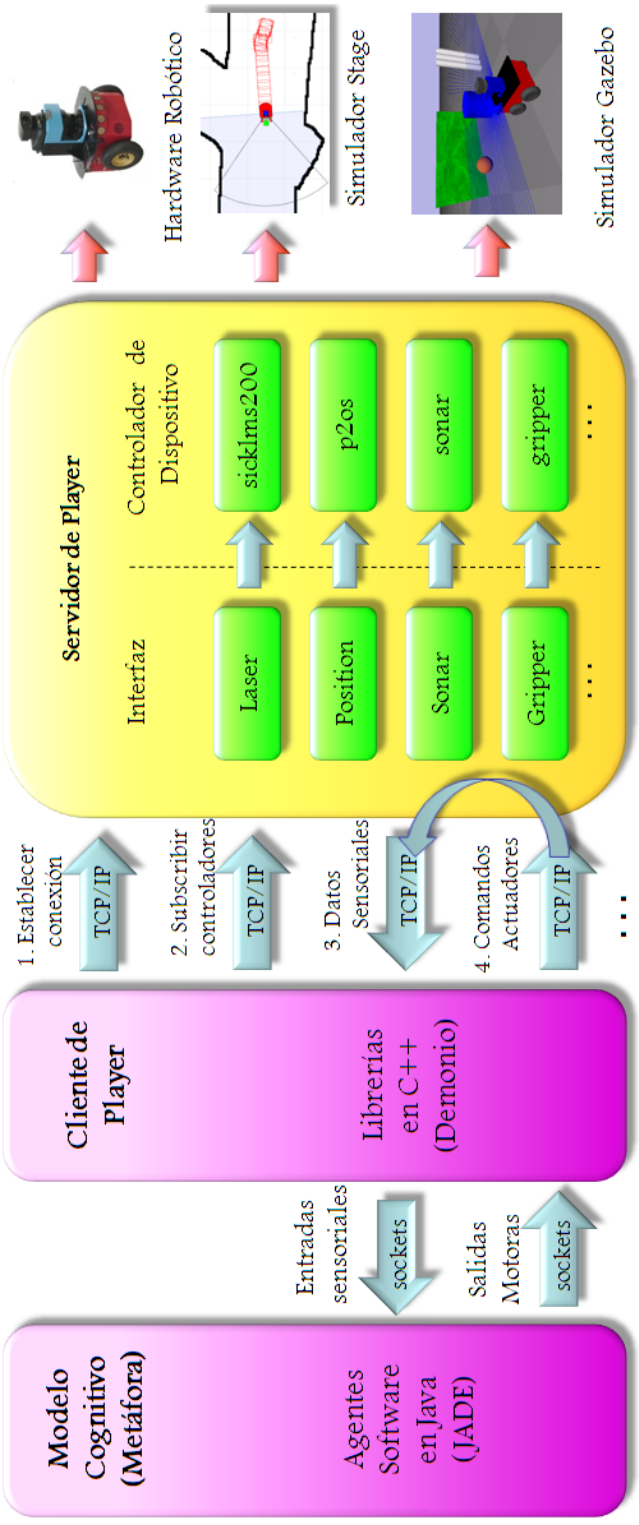


Figura 11.2: Modelo Cliente/Servidor de la Plataforma Robótica Player/Stage.

Como se puede observar en la figura 11.2, la implementación del modelo cognitivo es completamente independiente de la implementación del cliente Player, lo que evita el acoplamiento con *player/stage* y permite, a su vez, el poder "conectar" el modelo cognitivo con cualquier otro tipo de plataforma robótica o, incluso, con cualquier otra concreción contextual de METÁFORA. La comunicación entre el modelo cognitivo y el cliente Player se realiza a través de un modelo basado en *sockets* (Tanembaum, 2001), esto debido a que es una forma sencilla, veloz, y eficiente de transmitir los datos entre los dos tipos de lenguajes de programación (Java y C++).

La comunicación entre el cliente Player y el servidor Player, la cual también se basa en un modelo de sockets (a través del protocolo TCP/IP), sigue normalmente los siguientes pasos:

1. El cliente Player establece la conexión con el robot, a través de un objeto de la clase PlayerClient.
2. El cliente Player subscribe, a la instancia de la clase PlayerClient, los controladores (y por ende los dispositivos) del robot con los que va a interactuar.
3. La comunicación entre el cliente Player y el servidor Player entra en un bucle en el que se leerá información de los sensores y se enviarán instrucciones a los actuadores. Player se comunica con el robot a intervalos regulares, un número determinado de veces por segundo. Por defecto, la lectura de los sensores se realizará cada 100 milisegundos, no obstante, este intervalo dependerá del tiempo de respuesta que tarde la ejecución específica de cada ciclo cognitivo, es decir, el robot no enviará al cliente Player la nueva lectura de los sensores hasta que no haya recibido una respuesta (motora, generalmente) correspondiente al ciclo cognitivo anterior.

Por último, cabe notar, que en la figura 11.2 se representa la manera en que el servidor Player abstrae el control de los dispositivos de la operación de los mismos. Esto es, el servidor Player podrá comunicar el cliente Player, de forma indiferente, con un hardware robótico (robot real), con el simulador Stage, o con el simulador Gazebo. Como se ha mencionado anteriormente, para los escenarios experimentales propuestos basta con emplear el simulador Stage puesto que provee la funcionalidad necesaria para evaluar los aspectos funcionales de la arquitectura. Sin embargo, algunos experimentos serán evaluados además utilizando un hardware robótico específico (un robot Pioneer) con el fin de cotejar los tiempos de respuesta entre la simulación y la ejecución real.

11.3. Vista de Implementación y de Despliegue

En esta sección se describen, por un lado, los componentes software para el modelo cognitivo y, por otro, la distribución física de tales componentes en los nodos hardware (ordenadores que operan en un modelo cliente/servidor).

En la figura 11.3 se presenta el diagrama de componentes para el modelo cognitivo. Obsérvese que el sistema se compone de ocho sub-sistemas correspondientes con cada uno de los módulos que propone la arquitectura cognitiva: *Módulo de Percepción*, *Módulo de Orquestación de la Consciencia*, *Módulo Comportamental*, *Módulo Declarativo*,

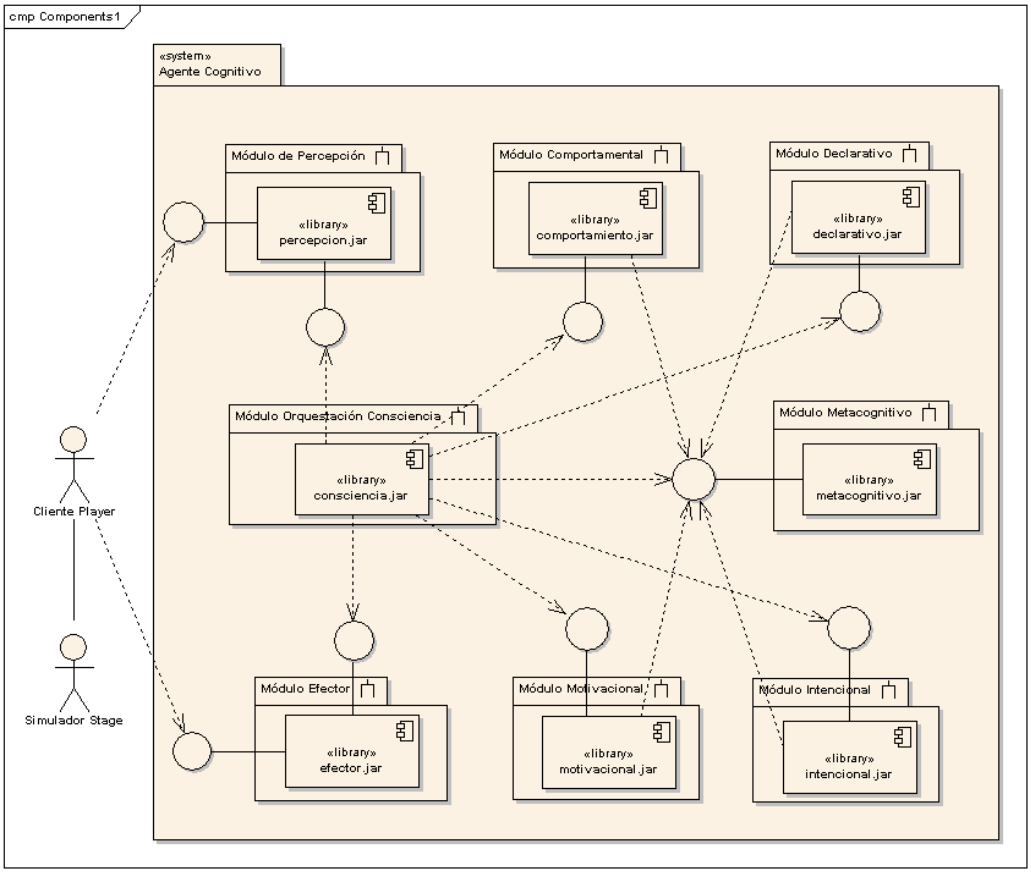


Figura 11.3: Diagrama de Componentes para el Modelo Cognitivo.

Módulo Intencional, *Módulo Motivacional*, *Modulo Metacognitivo*, y *Módulo Efector*. Internamente, cada sub-sistema define una estructura jerárquica de paquetes y clases las cuales se encuentran encapsuladas en un componente reutilizable de código Java (i.e., en un fichero .jar). Por simplicidad del diagrama, únicamente se representa gráficamente los componentes software omitiendo la estructura de paquetes y las clases que los conforman. Adicionalmente, la funcionalidad asociada a cada componente software se corresponde con las responsabilidades y colaboraciones definidas para cada módulo de la arquitectura, motivo por el cual no se describe al detalle cada componente.

Nótese además que los componentes no interactúan directamente sino a través de interfaces específicas, con lo cual se evita el acoplamiento estructural y permite, en todo caso, el fácil reemplazamiento de los componentes software, haciendo la arquitectura más flexible y escalable. Externamente, representado como un actor, se encuentra el cliente Player. Este actor es un hilo de procesamiento (demonio), escrito en código C++, que se ejecuta indefinidamente mientras la simulación esté activa. Este actor interactúa cíclicamente con el Agente Cognitivo a través de los componentes de *percepción* y *efector*, solicitando la captura de información sensorial y enviando comandos motores al servidor Player donde se encuentra el simulador Stage. Los distintos componentes software del sistema actúan como librerías que son invocadas por el Agente Cognitivo, que en este

caso se implementa como un agente software soportado por la plataforma multi-agente JADE.

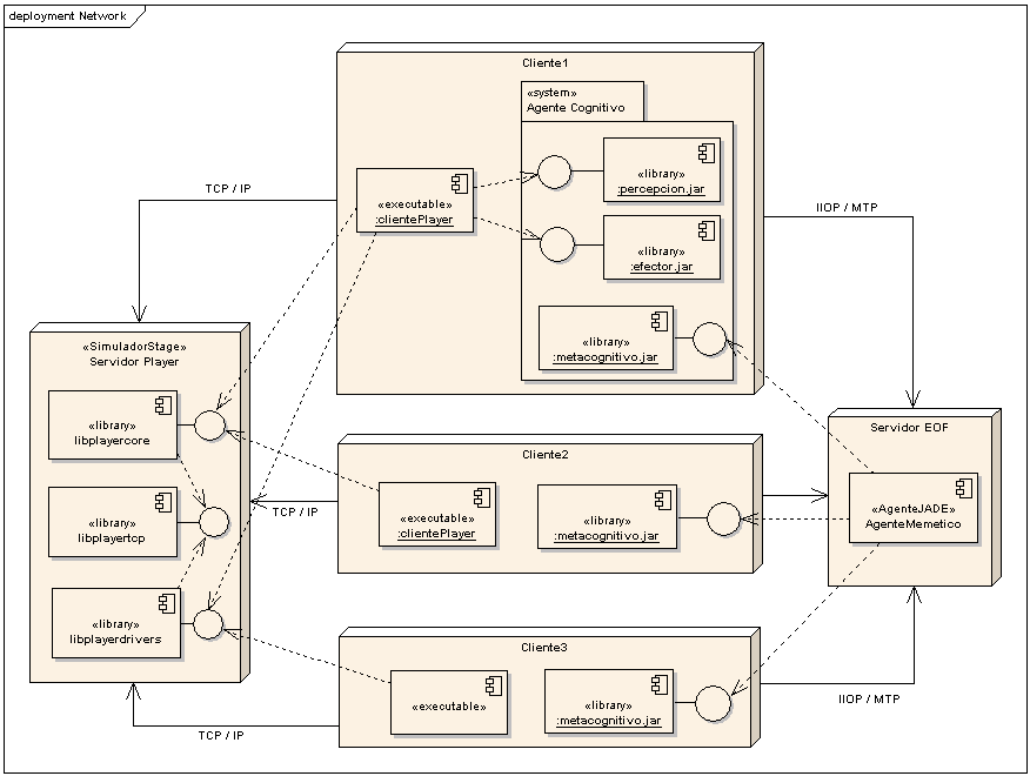


Figura 11.4: Diagrama de Despliegue.

En la figura 11.4 se presenta el modelo de despliegue de los componentes software. A continuación se describen cada uno de los nodos del diagrama:

- *Nodo Servidor Player*: en este nodo se ejecutará la simulación, utilizando para ello los componentes que definen el servidor Player y el simulador Stage:
  - *libplayercore*: es una librería escrita en código C++ que gestiona los mensajes en cola, ofrece soporte para la carga de los plugins, y analiza los ficheros de configuración de Player.
  - *libplayerdrivers*: es una librería escrita en código C++ que gestiona todos los controladores incluidos en la distribución de Player utilizada.
  - *libplayertcp* es también una librería escrita en C++ que da soporte a la comunicación de datos, en un modelo cliente/servidor, a través del protocolo TCP/IP.
- *Nodo Cliente*: este tipo de nodo representa cada máquina física donde serán desplegados y ejecutados los componentes software para un agente cognitivo específico. Esto es, por cada robot empleado durante la simulación deberá existir, idealmente,

un ordenador que procese la información percibida, y que genere una respuesta motora como consecuencia[3]. En este nodo se despliegan los componentes[4]:

- *clientePlayer*: este componente, escrito en código C++, corresponde al hilo de procesamiento que sirve de puente entre el servidor Player y el agente cognitivo, transmitiendo las peticiones de captura de información sensorial y la ejecución de comandos motores.
- *percepción* y *efector*: estos componentes software sirven para comunicar el modelo cognitivo interno del agente (cognitivo) con el cliente Player.
- *metacognitivo*: este componente se encarga de controlar los mecanismos del módulo Metacognitivo.

- *Nodo Servidor EOF*: en este nodo se despliega el componente *AgenteMemetico*, el cual se comunica con el componente *metacognitivo* de cada agente cognitivo a fin de coordinar la ejecución de los procesos epigenéticos, ontogenéticos, y filogenéticos. Durante esta comunicación entre agentes son extraídos, evolucionados, y asimilados, un conjunto selecto de memes.

Como se ha mencionado anteriormente, tanto el agente memético como cada uno de los agentes cognitivos, están soportados por la plataforma JADE. JADE (*Java Agent DEvelopment Framework*) es una plataforma software basada en Java (y por tanto de código abierto) para el desarrollo de sistemas multi-agente, que se ajusta a los estándares *FIPA* (*Foundation for Intelligent Physical Agents*) para la construcción de agentes inteligentes[5]. Incluye dos productos:

- Una plataforma de agentes que cumple los estándares FIPA.
- Un framework de desarrollo de agentes Java basado en un conjunto de librerías específicas.

La plataforma JADE puede estar distribuida en varias máquinas y proporciona soporte para la implementación de los agentes como "threads" (hilos independientes de procesamiento) agrupados dentro de contenedores [6].

Una de las características más importantes que los agentes JADE proporcionan es la habilidad de comunicarse. El paradigma de comunicación adoptado es el paso de mensajes síncrono. Así pues, los mensajes intercambiados entre los agentes cognitivos y el agente memético siguen el formato especificado por el lenguaje ACL, definido por el estándar internacional FIPA, para la interoperabilidad de agentes [7].

[3]Por motivos de simplificación, solamente se representa al detalle el Cliente 1, sin embargo, los múltiples clientes definidos para la simulación definirán la misma estructura. También por simplificación, se han definido sólo tres nodos cliente (con sus respectivos agentes robóticos), sin embargo, se definirían tantos nodos cliente como robots se utilicen en la simulación

[4]Por razones de simplificación del diagrama, no son representados todos los ocho componentes del modelo cognitivo.

[5]La página oficial de FIPA puede ser accedida en: `http://www.fipa.org/`

[6]Para mayor información sobre JADE (documentación, código fuente, etc.), el lector puede consultar la web oficial: `http://jade.tilab.com/`

[7]La especificación que ofrece el estándar FIPA para ACL puede ser accedida en: `http://www.fipa.org/specs/fipa00061/SC00061G.pdf`

Por último, cabe mencionar que la plataforma JADE permite a cada agente llevar a cabo múltiples tareas de forma concurrente. Para ello, JADE trata a cada agente como un hilo único de ejecución, y sus tareas se implementan como objetos de la clase *Behavior*. Estas tareas (behaviors) operan de forma concurrente y cooperativa sin establecer preferencias en la ejecución (como sí lo hacen los threads de Java), evitando de esta manera los típicos problemas de sincronización y de acceso compartido a los recursos. El tratamiento de cada uno de los múltiples procesos cognitivos que define METÁFORA se realiza a través de una instancia de una de las siguientes clases: *Behavior* (para comportamientos genéricos que ejecutan diferentes acciones dependiendo de un estado), *CyclicBehaviour* (para aquellas tareas que se ejecutan indefinidamente y de manera cíclica), y *OneShotBehaviour* (para aquellos comportamientos que sólo se ejecutan una vez). La ejecución de los múltiples comportamientos (behaviors) se planifica temporalmente siguiendo una política *round robin*.

11.4. Construcción de Perceptos y Formateo de Acciones

En esta sección se describen los dispositivos y los respectivos proxies de *player/stage* que dispondrá cada robot, necesarios tanto para la detección de las entradas sensoriales como para la ejecución de las acciones motoras. Adicionalmente, se exponen, por un lado, los mecanismos de interpretación y conversión de datos sensoriales que realiza el módulo de Percepción con el fin de crear el conjunto de *perceptos* utilizados por el modelo cognitivo y, por otro, los mecanismos de traducción que lleva a cabo el módulo Efector con el fin de formatear las acciones producidas por el modelo cognitivo a representaciones específicas interpretables por los actuadores del robot.

11.4.1. Detección de Obstáculos a través del Proxy Sonar

El robot está provisto de un anillo sonar ultrasonido (Shirley, 1989) conformado por 16 sensores dispuestos alrededor suyo. Cada sensor se configura para permitir, un rango de detección de 15^{o}, y unas distancias mínima y máxima de detección de 0.0 y 5.0 metros, respectivamente. El orden de lectura de los sensores del sonar se realiza en sentido de las manecillas del reloj, desde el sensor 0 hasta el sensor 15, donde la lectura para cada sensor representa la distancia más cercana desde el centro del robot a un objeto (si lo hubiese) o, en su defecto, la distancia máxima de detección definida (5.0 m). La figura 11.5 representa la disposición de los sensores.

La clase SonarProxy provee acceso a la colección de sensores y a sus lecturas específicas (en metros), cada una con una precisión de 8 bytes (i.e., valores de punto flotante). Una vez realizadas las lecturas de los sensores, el módulo de Percepción del agente robótico, a través de su interprete, procede a convertir y formatear dichos datos a bloques de información significativa que utilizarán los diferentes *Comportamientos* del agente (e.g., el *Comportamiento* ⟨`Evadir-Obstáculos`⟩).

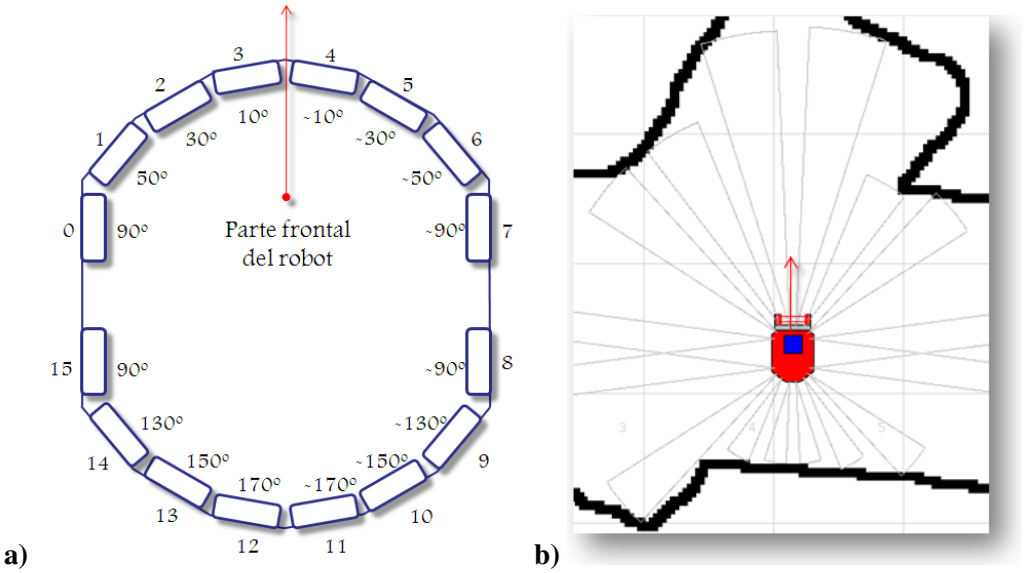


Figura 11.5: Sensores Ultrasonido. a) Disposición angular de los sensores alrededor del robot (enumerados del 0 al 15). b) Ejemplo de las lecturas de los sensores del Sonar en Stage.

11.4.1.1. Conversión y Formateo de las lecturas del Sonar

Con el fin reducir el tiempo de procesamiento que requiere validar los 16 sensores del sonar, se reduce el número de lecturas a la mitad, conservando aquellas lecturas más bajas. Esto es, se crea un vector de 8 posiciones tal que:

- Posición 0: mín $(sensor[15], sensor[0])$
- Posición 1: mín $(sensor[1], sensor[2])$
- Posición 2: mín $(sensor[3], sensor[4])$
- Posición 3: mín $(sensor[5], sensor[6])$
- Posición 4: mín $(sensor[7], sensor[8])$
- Posición 5: mín $(sensor[9], sensor[10])$
- Posición 6: mín $(sensor[11], sensor[12])$
- Posición 7: mín $(sensor[13], sensor[14])$

Luego, el módulo de Percepción recorre la colección de lecturas del sonar y determina cuál de ellas supera un umbral de detección correspondiente a la distancia mínima a la que debe estar un objeto para ser considerado un obstáculo. En los experimentos realizados se estableció que: $umbral_{(deteccion)} = 0.7m$. Tomando como referencia este umbral, el módulo de Percepción genera, a partir de los datos detectados, dos tipos de perceptos: simbólicos para el nivel cognitivista, y sub-simbólicos para el nivel emergente. Por ejemplo, supóngase que el sonar ha detectado las lecturas de la tabla 11.1:

S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
0.845	1.236	0.587	0.659	0.623	0.722	1.654	3.127	5.0	5.0	2.754	2.041	1.296	0.678	0.542	0.623

Tabla 11.1: Lectura del Sonar en un instante de tiempo t.

Una vez han sido capturadas las lecturas del sonar, éstas son validadas para conformar el vector de la tabla 11.2 (siguiendo el criterio de escoger la lectura mínima para cada par de sensores).

0	1	2	3	4	5	6	7
0.623	0.587	0.623	0.722	3.127	2.754	1.296	0.542

Tabla 11.2: Vector conformado por las lecturas mínimas del Sonar.

El vector resultante es luego transformado y formateado, por el Interprete, a sus correspondientes perceptos simbólicos y sub-simbólicos. Durante la transformación de las lecturas sin procesar del sonar a los perceptos sub-simbólicos, se asigna un valor de "1" a aquellas distancias que están por debajo del umbral de detección (i.e., cuando el sensor ha detectado un obstáculo cerca), de lo contrario se asigna un valor de "0" (i.e., cuando no se ha detectado obstáculos o cuando sí se han detectado pero están lo suficientemente lejos). La transformación correspondiente al ejemplo propuesto sé presenta en la tabla 11.3.

Perc. 1	Perc. 2	Perc. 3	Perc. 4	Perc. 5	Perc. 6	Perc. 7	Perc. 8
1	1	1	0	0	0	0	1

Tabla 11.3: Representación Sub-simbólica de los Perceptos correspondientes a la Detección de Obstáculos.

Estos perceptos sub-simbólicos serán las entradas que recibirán las redes neuronales (componentes PMR) y los sistemas inmunes (componentes SIA) del nivel emergente del *Comportamiento*. Ahora, tomando como referencia nuevamente el umbral de detección, se construyen los perceptos simbólicos para el nivel cognitivista, como se aprecia en el cuadro 11.1. Así pues, a partir de los perceptos construidos es posible inferir entonces que existe un obstáculo (o varios) situado(s) al lado izquierdo y adelante del robot.

Adicionalmente, el módulo de Percepción puede construir un nuevo percepto que determina si alguno de los sensores 2, 3, 4, o 5 (sensores frontales del robot) ha detectado un obstáculo lo suficientemente cerca del robot como para ocasionar una colisión inminente. Estos perceptos indicarán al agente robótico que debe adoptar una conducta reactiva (e.g., detenerse y retroceder). Para esto, se valida si la lectura de dichos sensores está por debajo de un umbral de colisión ($umbral_{(colision)} = 0.4m$). Por ejemplo, en el caso de que la distancia detectada esté por debajo del umbral, se generaría un percepto sub-simbólico con un valor de "1", y un percepto simbólico como el que se muestra en el cuadro 11.2.

Percepto 1:

⟨elemento, obstáculo⟩
⟨proximidad, cerca⟩
⟨ubicación, izquierda⟩

Percepto 2:

⟨elemento, obstáculo⟩
⟨proximidad, cerca⟩
⟨ubicación, izquierda-adelante⟩

Percepto 3:

⟨elemento, obstáculo⟩
⟨proximidad, cerca⟩
⟨ubicación, adelante⟩

Percepto 4:

⟨elemento, obstáculo⟩
⟨proximidad, lejos⟩
⟨ubicación, derecha-adelante⟩

Percepto 5:

⟨elemento, obstáculo⟩
⟨proximidad, lejos⟩
⟨ubicación, derecha⟩

Percepto 6:

⟨elemento, obstáculo⟩
⟨proximidad, lejos⟩
⟨ubicación, derecha-atrás⟩

Percepto 7:

⟨elemento, obstáculo⟩
⟨proximidad, lejos⟩
⟨ubicación, atrás⟩

Percepto 8:

⟨elemento, obstáculo⟩
⟨proximidad, cerca⟩
⟨ubicación, izquierda-atrás⟩

Cuadro 11.1: Ejemplo de Perceptos Simbólicos construidos a partir de las lecturas del Sonar.

En resumen, el módulo de Percepción genera, a partir de la información detectada por el Sonar, nueve perceptos así: ocho perceptos que determinan si se han detectado obstáculos cerca o no, y un percepto que determina si existe un obstáculo adelante del robot que impida la maniobrabilidad.

Percepto 9:

⟨elemento, obstáculo⟩
⟨proximidad, **muy-cerca**⟩
⟨ubicación, adelante⟩

Cuadro 11.2: Ejemplo del Percepto Simbólico que advierte una colisión.

11.4.2. Detección de Objetos a través de los Proxies Laser y Fiducial

Con el fin de detectar e identificar los diferentes tipos de objetos que se encuentran en el entorno circundante, el robot emplea dos tipos de proxies: *Laser* y *Fiducial*. En primer instancia, el proxy *Laser* provee acceso a un dispositivo láser el cual realiza un barrido, en sentido contrario a las manecillas del reloj, de un conjunto de mediciones espaciadas con cierta regularidad. Cada medición, o punto de escaneo, es tratado como si fuese registrado por un sensor diferente.

Específicamente, el proxy *Laser* permite definir: el rango de datos que será detectado mediante la indicación de los ángulos de inicio y fin del escaneo, la resolución angular del barrido, y el número de lecturas incluidas en la detección. En los experimentos realizados se empleó por defecto un barrido de 181 lecturas, desde el ángulo 0º hasta el ángulo 180º, con una resolución igual a 1. Esta resolución indica que el ángulo entre cada punto de escaneo (θ), es igual a 1º. Adicionalmente, se define por defecto una distancia máxima de alcance del escáner láser igual a 10 metros. En la figura 11.6.a se representa el área de barrido del escáner láser (región de color azul).

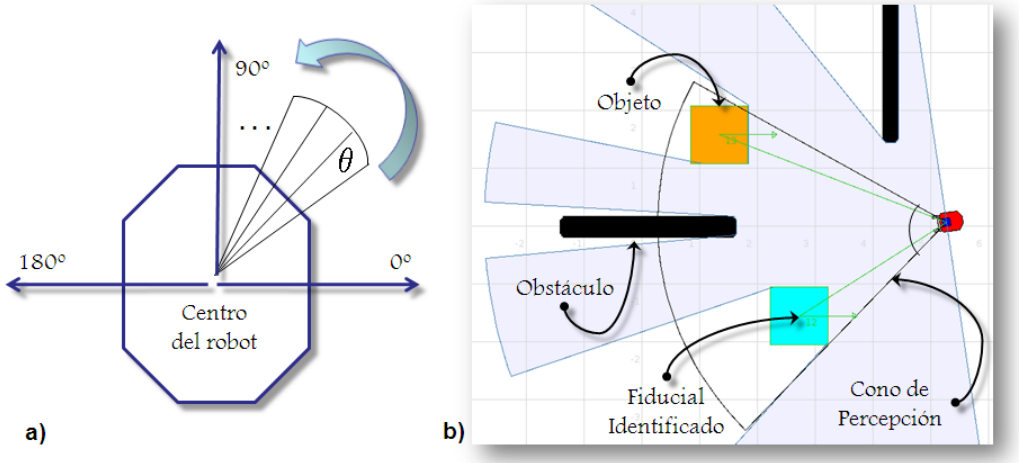


Figura 11.6: Sensores Laser y Fiducial. a) Lecturas del escáner láser y sus posiciones en el robot (181 lecturas). b) Representación de los sensores Laser y Fiducial del robot en Stage. La región azul corresponde al área de detección barrida por el escáner láser, y el cono de percepción al área de detección barrida por los sensores fiducial.

A pesar de que el proxy *Laser* puede ser empleado (al igual que el proxy *Sonar*) para la detección de obstáculos, se ha optado por emplear únicamente el sonar para este tipo de tarea. La razón para esto es que el *Laser* provee muchos más puntos de lectura que el sonar[8], requiriéndose por tanto un mayor tiempo de procesamiento en cada ciclo cognitivo (incluso si la resolución del escáner láser es baja). Por tanto, el motivo principal por el cual se emplea el proxy *Laser* se debe a éste es requerido por el proxy *Fiducial* para la identificación de los objetos específicos del entorno.

El proxy *Fiducial* provee acceso a los dispositivos que detectan ciertos marcadores codificados (fiducials) situados en el entorno. Un fiducial es un punto fijo en una imagen[9], de manera que el proxy *Fiducial* simula un software de procesamiento de imágenes que localiza puntos específicos durante el barrido que realiza el escáner láser del robot. A partir de estos puntos específicos el robot puede identificar el objeto correspondiente.

En la figura 11.6.b se muestra el barrido que realiza el proxy *Fiducial*. Por motivos de claridad, en la figura, el ángulo de barrido del proxy *Fiducial* (representado como un cono de percepción) es de menor apertura y alcance que el barrido producido por el escáner láser. No obstante, en los experimentos realizados se emplea el mismo ángulo de barrido y la misma distancia de alcance para los dos proxies: 180^{o} de apertura y 10 metros de alcance.

Adicionalmente, en la figura puede observarse que los objetos que se encuentran dentro del cono de percepción de los sensores fiducial (a excepción de los obstáculos), retornan un código específico o marcador asociado al objeto. Por ejemplo, el objeto de color naranja retorna un marcador igual a 13, mientras que el objeto de color cyan retorna un marcador igual a 12. Esto quiere decir que por cada objeto (identificable) del entorno, los sensores fiducial detectan el código que ha sido asociado al objeto (por el diseñador del entorno robótico[10]), el ángulo de orientación del objeto con respecto al centro del robot, y el vector de orientación de dicho objeto. A partir de esta información detectada es posible calcular, mediante funciones trigonométricas, la distancia desde el centro del robot al objeto y la posición relativa del objeto con respecto al robot (i.e., a la izquierda, a la derecha, etc.).

11.4.2.1. Conversión y Formateo de las lecturas del Fiducial

La información detectada por el proxy *Fiducial* para cada objeto se presenta en la tabla 11.4. Cabe decir que las coordenadas detectadas corresponden a la posición relativa

[8]Esto es, 181 mediciones si se sitúa un escáner láser en la parte delantera del robot y 361 mediciones si se sitúa además un láser en la parte posterior.

[9]En el campo de la física, las gráficas 3D por ordenador, y la fotografía, los "fiducials" cumplen la función de puntos de referencia. Esto es, son puntos específicos dentro de una escena que sirven para: hacer seguimiento de los objetos en movimiento, crear referencias que pueden utilizar otros objetos, o realizar mediciones sobre tales objetos.

[10]Para que un objeto pueda ser reconocible por los sensores fiducial, el diseñador debe especificar, en los ficheros de configuración de player, un código único de identificación para dicho objeto. En la simulación robótica propuesta sólo serán reconocibles aquellos objetos con los que puede interactuar el robot (e.g., efectuar acciones como levantar, soltar, empujar, etc.), motivo por el cual los muros del entorno son considerados como obstáculos y por tanto, no definen un código reconocible por los sensores fiducial.

del objeto con respecto al robot, y que el valor de Ángulo corresponde al ángulo de rotación del robot (yaw angle). Obsérvese la figura 11.7.

Identificador	Coordenada X	Coordenada Y	Ángulo
12	2.625	0.935	-135º

Tabla 11.4: Ejemplo de la información detectada por los sensores fiducial para un objeto identificable.

Una vez detectada la información que provee el fiducial sobre el objeto que se encuentra dentro del cono de percepción del robot, el módulo de Percepción procede a construir los respectivos perceptos. Concretamente, se generan los siguientes perceptos del tipo sub-simbólico:

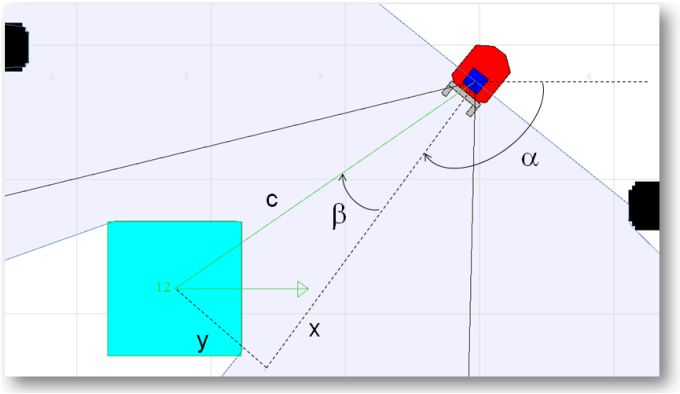


Figura 11.7: Ubicación relativa de un Objeto con respecto a la orientación del robot.

1. *Identificador:* este percepto corresponde al número único de identificación del objeto percibido, el cual puede estar en el rango de 0 - 59 (para la simulación propuesta, se define hasta un máximo de 60 objetos identificables por el robot que pueden estar situados en el entorno simulado). Esta información es luego convertida a un formato sub-simbólico (i.e., mediante una codificación binaria), con lo cual se requiere de 6 bits para ello (e.g., 59 = 111011).
2. *Tipo de Objeto*: en la simulación propuesta se emplean cuatro tipos específicos de objetos identificables, con lo cual se requiere de 2 bits para representar este percepto: cajas (00), depósitos (01), cargador de batería (10), y otros robots (11). La manera en que el robot interactúa con estos tipos de objetos se explicará más adelante en la descripción de los escenarios experimentales.
3. *Ubicación*: este percepto indica la ubicación relativa del objeto con respecto al robot. Para esto, el módulo de Percepción calcula, mediante funciones trigonométricas, el ángulo de la ubicación relativa del objeto (β) a partir del ángulo de rotación del robot (α) y de la posición relativa del objeto (x, y) con respecto al centro del detector (obsérvese la figura 11.7). Una vez calculado el ángulo β, se aplican las siguientes reglas:

Si $\beta <=$ 15º Y $\beta >=$ -15º entonces el objeto está adelante del robot
Si $\beta >$ 15º entonces el objeto está a la izquierda del robot
Si $\beta <$ -15º entonces el objeto está a la derecha del robot

De esta manera, se necesitan dos bits para representar la ubicación del objeto con respecto al robot: adelante (00), a la izquierda (01), y a la derecha (10). Como se puede observar, el percepto nunca indicará que el objeto está situado detrás del robot puesto que, si lo estuviese, el robot no lo identificaría ya que solamente se emplea un escáner láser en la parte delantera del mismo.

4. *Distancia*: este percepto indica si el objeto detectado se encuentra lo suficientemente cerca del robot como para poder interactuar con él. Para esto, se han definido tres umbrales de aproximación, tal que: $umbral_{(lejos)} = 8.0m$, $umbral_{(cerca)} = 3.0m$, y $umbral_{(mcerca)} = 0.5m$. Así pues, por cada objeto identificado se calcula la distancia hasta el robot (la hipotenusa "c" del triángulo rectángulo de la figura 11.7) y se aplican, luego, las siguientes reglas:

Si $d >= umbral_{(lejos)}$ entonces el objeto está lejos del robot
Si $d >= umbral_{(cerca)}$ Y $d < umbral_{(lejos)}$ entonces el objeto está cerca
Si $d >= umbral_{(mcerca)}$ Y $d < umbral_{(cerca)}$ entonces el objeto está muy cerca

De esta manera, se necesitan dos bits para representar la ubicación del objeto: lejos (00), cerca (01), y muy cerca (10).

Un ejemplo de la transformación de la información detectada por el proxy *Fiducial* a un percepto sub-simbólico es presentada en la tabla 11.5. Nótese que el percepto únicamente incluiría la fila correspondiente a la codificación sub-simbólica, por tanto, la finalidad de la fila de codificación simbólica es simplemente orientativa. Así pues, el percepto sub-simbólico sería la concatenación de los valores binarios resultantes de la transformación, en este caso: "1100011001".

Codificación	Identificador	Tipo de Objeto	Ubicación	Distancia
simbólica	12	Depósito	Derecha	4.5 m
sub-simbólica	1100	01	10	01

Tabla 11.5: Ejemplo de la construcción de un percepto sub-simbólico a partir de los datos detectados por el Fiducial y procesados por el módulo de Percepción.

El percepto simbólico generado a partir de la misma información detectada por el proxy *Fiducial* se presenta en el cuadro 11.3. Nótese que los perceptos simbólicos almacenan información adicional sobre los objetos identificados. Esta información puede ser empleada por el robot para: (1) realizar posteriores inferencias sobre el estado de los objetos, y (2) proveer información específica que pueda servir en la demostración de la línea de razonamiento que ha seguido el agente robótico.

El módulo de Percepción creará tantos perceptos simbólicos y sub-simbólicos como objetos haya identificado el robot a través del proxy *Fiducial*.

Percepto 10:

⟨elemento, objeto⟩
⟨tipo-objeto, depósito⟩
⟨identificador, 12⟩
⟨ubicación, derecha⟩
⟨distancia, cerca⟩
⟨posición-x, 2.625⟩
⟨posición-y, 0.935⟩
⟨ángulo-robot, -135°⟩
⟨ángulo-orientación, -23°⟩
⟨distancia-al-centro, 2.783⟩

Cuadro 11.3: Ejemplo de un Percepto Simbólico construido a partir de las lecturas del Fiducial.

11.4.3. Manipulación de la Pinza a través del Proxy Gripper

En los diferentes escenarios de experimentación, la manera en que el robot interactúa con los objetos es a través de una pinza o gripper que tiene acoplado justo en frente suyo. El proxy *Gripper* provee esta funcionalidad simulando un accesorio de manipulación que posee dos grados de libertad. El dispositivo del *Gripper* consta de dos palas las cuales se pueden desplazar horizontalmente, empleando para ello dos motores reversibles DC. Cuando estas palas se juntan lo suficiente y se halla un objeto de tamaño y forma apropiadas en medio de éstas, el proxy *Gripper* simula la acción de sujetar el objeto, levantarlo, y/o transportarlo, como se observa en la figura 11.8.

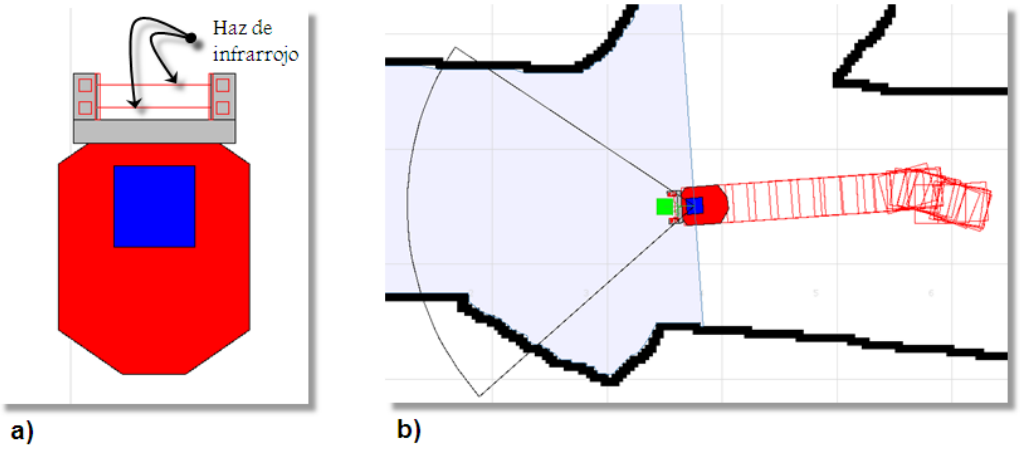


Figura 11.8: Proxy Gripper. a) Ubicación de la pinza en el robot y representación de los haces de luz infrarroja. b) Simulación en Stage del robot levantando una caja.

Adicionalmente, la pinza posee dos sensores de infrarrojos, uno interno y otro externo, los cuales son empleados para determinar si un objeto se encuentra lo suficientemente

cerca o no, como para poder ser sostenido por la pinza. Además, estos sensores de infrarrojos proveen información acerca de si se encuentra o no un objeto dentro de las pinzas o si éstas están totalmente cerradas, con lo cual es posible determinar si la pinza ha podido levantar con éxito un objeto.

Así pues, el proxy *Gripper* es utilizado tanto en el proceso de percepción (proveyendo información sobre el estado de la pinza y de los objetos que se hallan internamente en el área de sujeción) como en el proceso de actuación (permitiendo controlar los servomotores de las palas). A continuación, se describe la información codificada para cada uno de estos procesos.

11.4.3.1. Conversión y Formateo de las lecturas del Gripper

El módulo de Percepción construye cuatro tipos de perceptos (simbólicos y subsimbólicos) a partir de la información del proxy *Gripper*:

1. *Número de haces interrumpidos*: el proxy *Gripper* provee información sobre el número de haces de infrarrojos que son interrumpidos por la presencia de un objeto en el área de sujeción. Esta información corresponde a un estado entre 0 y 3, y su correspondiente codificación sub-simbólica sería:
   - "0" (00), si ninguno de los dos haces es interrumpido;
   - "1" (01), si el haz de infrarrojo interno es interrumpido;
   - "2" (10), si el haz de infrarrojo externo es interrumpido; y
   - "3" (11), si ambos haces son interrumpidos indicando que es posible sujetar el objeto.
2. *Estado de la pinza*: el proxy *Gripper* permite conocer el estado actual de la pinza. Los diferentes estados de la pinza y su correspondientes representaciones subsimbólicas serían:
   - Pinza Abierta (00),
   - Pinza Cerrada (01), y
   - Pinza en movimiento (cerrándose o abriéndose) (01).
3. *Cargando objeto*: este percepto, que puede ser inferido a partir de los sensores descritos en los puntos anteriores, indica si el robot está cargando y transportando un objeto entre sus pinzas (i.e., cuando los dos haces de infrarrojos son interrumpidos y la pinza está cerrada). Si el robot está cargando un objeto, el percepto sub-simbólico será "1", de lo contrario "0".
4. *Perdida de foco del Objeto*: este percepto indica si el robot ha iniciado la sujeción del objeto y por alguna razón éste cae nuevamente y los sensores infrarrojos no lo detectan justo en frente del robot. En este caso, el percepto sub-simbólico correspondiente a la perdida del foco del objeto es igual a "1", o en caso contrario igual a "0".

Así, por ejemplo, si el robot carga exitosamente un objeto, el percepto sub-simbólico construido sería como el que se presenta en la tabla 11.6, y su representación correspondería a la concatenación de tales características: "110110". El percepto simbólico correspondiente se presenta en el cuadro 11.4. Nótese que el percepto simbólico incluye una nueva pareja ⟨`propiedad, valor`⟩ que indica el tipo de objeto que está cargando el robot. En la simulación robótica propuesta, el robot solamente podrá levantar objetos del tipo "caja", razón por la cual esta propiedad es meramente informativa en el nivel simbólico, y se considera implícita en el nivel sub-simbólico.

Haces interrumpidos	Estado Pinza	Cargando Objeto	Perdida de Foco
11	01	1	0

Tabla 11.6: Ejemplo de un percepto sub-simbólico construido a partir de la información detectada por los sensores de la pinza.

Percepto n:

⟨`elemento, pinza`⟩
⟨`haces-interrumpidos, 3`⟩
⟨`estado-pinza, cerrada`⟩
⟨`cargando-objeto, si`⟩
⟨`perdida-foco, no`⟩
⟨`tipo-objeto, caja`⟩

Cuadro 11.4: Ejemplo de un Percepto Simbólico construido a partir de las lecturas del proxy Gripper.

11.4.3.2. Traducción de los actuadores del Gripper

Como se ha mencionado anteriormente, el proxy *Gripper* permite mover horizontalmente sus dos palas con el fin de sujetar o soltar un objeto. De esta manera, el agente cognitivo puede ejecutar tres tipos de acciones:

1. *Abrir*: esta acción abre las palas de la pinza y corresponde al valor sub-simbólico "00". Después de ejecutarse esta acción, el objeto sostenido por la pinza (si existe) es liberado.
2. *Cerrar*: esta acción cierra las palas de la pinza y corresponde al valor sub-simbólico "01". Si existe un objeto en el área de sujeción, éste es agarrado por la pinza, de lo contrario, la pinza se cierra completamente.
3. *Parar*: esta acción le indica al dispositivo del gripper que detenga su ejecución, bien sea si se está abriendo o cerrando. Corresponde al valor sub-simbólico "10".

En el caso de que el robot haya identificado un objeto justo en frente suyo y, como consecuencia, ordene al proxy *Gripper* ejecutar la acción de cerrar la pinza, pero ésta se cierre por completo; se advertirá, en el siguiente ciclo cognitivo, que el objeto no ha sido sujetado exitosamente y que se ha perdido el foco de éste. Esto resulta útil puesto que es probable que al intentar cerrar las palas de la pinza, éstas empujen el objeto hacia adelante evitando su sujeción, y ya que el objeto se encuentra tan cerca del robot, los sensores fiducial no podrán leer correctamente el identificador del objeto (debido a la distorsión de la imagen del fiducial), con lo cual el robot no reconocerá que el objeto está adelante suyo. En este caso, el robot deberá (idealmente) explorar su entorno circundante inmediato con el fin de encontrar el objeto que ha perdido de foco e intentar nuevamente sujetarlo.

11.4.4. Determinación del Desplazamiento y Localización del Robot a través del Proxy Position2dProxy

Con el propósito de determinar la localización del robot en cada instante (en forma de coordenadas cartesianas) y poder calcular, a partir de esta ubicación, las posiciones absolutas de los objetos identificados, el módulo de Percepción utiliza el proxy Position2dProxy (P2d de ahora en adelante). Este proxy, además, permite al módulo Efector del agente cognitivo controlar los actuadores correspondientes al desplazamiento del robot, esto es, la velocidad lineal y la velocidad angular o de rotación.

El proxy P2d puede emplear distintos mecanismos para determinar la posición del robot. Uno de los sistemas más utilizados en robótica móvil para este cometido es la *odometría*. Los robots móviles usan la odometría para estimar (y no determinar) su posición relativa a su localización inicial. Existe evidencia suficiente (K. y Kleeman, 1997; Zunino y Christensen, 2002; Martinelli et al., 2003) de que la odometría proporciona una buena precisión a corto plazo, es barata de implantar, y permite tasas de muestreo muy altas. Sin embargo, la idea fundamental de la odometría es la integración de información incremental del movimiento a lo largo del tiempo, lo cual conlleva una inevitable acumulación de errores. En concreto, la acumulación de errores de orientación, causa grandes errores en la estimación de la posición, los cuales van aumentando proporcionalmente con la distancia recorrida por el robot.

Es por este motivo que se ha decidido que el proxy P2d no utilice un mecanismo de odometría sino que, en su lugar, utilice un sistema de navegación satelital a través de un dispositivo GPS. Esto permitirá al robot obtener datos más precisos durante la simulación. No obstante, los experimentos realizados con el robot real sí que emplearon un sistema de odometría. El uso de un dispositivo GPS en robótica móvil no suele ser muy realista, pero para efectos de la experimentación resulta conveniente.

11.4.4.1. Conversión y Formateo de las lecturas del proxy P2d

A través del proxy P2d es posible determinar la localización y la orientación del robot en el plano cartesiano definido por el fichero mundo (world). Esta información es luego codificada en perceptos sub-simbólicos y simbólicos así:

1. *Coordenada X*: el proxy P2d provee un dato de 4 bits (i.e., un valor float) para la coordenada absoluta X del centro del robot. Esta información es truncada a sólo un dígito significativo después del punto decimal, y luego es multiplicada por diez para obtener un valor entero. Por ejemplo, si la coordenada X es 2.5467, primero se trunca a 2.5 y luego se multiplica por diez: 25. Luego, este número es codificado en binario: 25_{10} = $11001_{(2)}$.
2. *Coordenada Y*: se realiza el mismo procedimiento descrito para la coordenada X.
3. *Ángulo de Rotación*: el proxy P2D provee el ángulo de rotación del robot (con respecto al eje horizontal) en radianes. A partir de esta lectura, el módulo de Percepción ejecuta los siguientes pasos para la traducción de la información a los correspondientes perceptos: (1) convierte de radianes a grados la medición del ángulo, (2) aproxima el valor al número entero más cercano, (3) divide este valor entre 30, y (4) convierte el resultado a binario. El objetivo del paso (3) es codificar el ángulo de rotación del robot empleando una cantidad mínima, pero significativa, de bits; esto debido a que entre mayor sea la longitud de las cadenas sub-simbólicas, mayor tiempo tardarán los componentes del nivel emergente en "aprender" y "generalizar" dicha pauta. Es por este motivo que se determina, antes de la conversión a binario, el rango correspondiente (cada uno definido en 30º) en el que se encuentra el ángulo de rotación (e.g., $[0^o - 29^o] \rightarrow rango0$, $[30^o - 59^o] \rightarrow rango1$, etc.). Por ejemplo, el ángulo 1.756 rad se convertiría en: 1.756 rad $\times 180/\pi = 100.6114^o \rightarrow 100^o \rightarrow rango3 \rightarrow 0011_{(2)}$. Nótese que se utilizan 4 bits para la conversión del rango del ángulo de rotación puesto que existen 12 rangos consecutivos (rango 0 - rango 11), ya que: $11_{(10)} = 1011_{(2)}$. No obstante, en el percepto simbólico, sí que es almacenado el valor real del ángulo de rotación del robot.

Por ejemplo, a partir de la lectura del proxy P2d (x = 2.3456, y = 1.9873, α = 2.2343), el módulo de Percepción construiría el percepto sub-simbólico de la tabla 11.7 y el percepto simbólico del cuadro 11.5.

Coordenada X	Coordenada Y	Ángulo de Rotación
10111	10011	0100

Tabla 11.7: Ejemplo de un percepto sub-simbólico construido a partir de la información detectada por el proxy P2d.

11.4.4.2. Traducción de los actuadores del proxy P2d

Concretamente, a través del proxy P2d el agente cognitivo puede controlar la velocidad lineal y angular del robot. Así pues, el módulo Efector del agente traduce dos tipos de acciones:

1. *Velocidad Lineal*: el control de este actuador es muy sencillo puesto que el modelo cognitivo hace referencia sólo a cuatro tipos de velocidad, los cuales son luego tra-

Percepto n:

⟨elemento, robot⟩
⟨coordenada-x, 2.3⟩
⟨coordenada-y, 1.9⟩
⟨ángulo-rotación, 128.01⟩
⟨rango-ángulo-rotación, 4⟩

Cuadro 11.5: Ejemplo de un Percepto Simbólico construido a partir de las lecturas del proxy P2d.

ducidos, por el módulo Efector, a una acción específica que determina la magnitud escalar a la que debe desplazarse el robot[11]. Los diferentes tipos de acciones son:

- *Velocidad de Detención*: esta acción le indica al robot que debe detenerse. Se codifica sub-simbólicamente como "00", y corresponde a una velocidad de 0.0 m/s.
- *Velocidad de Navegación*: esta acción hace referencia a la velocidad normal con la que se desplaza el robot. Su codificación sub-simbólica es "01" y corresponde a una velocidad de 0.5 m/s.
- *Velocidad de Retroceso*: esta acción le indica al robot que debe retroceder a una velocidad de -0.5 m/s y se codifica sub-simbólicamente como "10".
- *Velocidad de Aproximación*: esta acción le indica al robot que debe disminuir la velocidad a 0.15 m/s (especialmente cuando se está aproximando hasta algún objeto con el que desea interactuar). Se codifica sub-simbólicamente como "11".

2. *Velocidad Angular*: este actuador controla la velocidad de rotación del robot en términos de un ángulo de giro (en radianes) por unidad de tiempo (en segundos). Concretamente, el módulo Efector recibe una de las siguientes acciones:

   - *Rotar a la Izquierda*: esta acción le indica al robot que debe girar hacia la izquierda a una velocidad de 40^o/seg $\cong$ 0.698 rad/seg. Su codificación sub-simbólica es "000".
   - *Rotar a la Derecha*: esta acción le indica al robot que debe girar hacia la derecha a una velocidad de -40^o/seg $\cong$ -0.698 rad/seg. Su codificación sub-simbólica es "001".
   - *Alinearse a la Izquierda*: esta acción le indica al robot que debe girar lentamente hacia la izquierda (con el fin de alinearse con un objeto específico) a una velocidad de 10^o/seg $\cong$ 0.174 rad/seg. Su codificación sub-simbólica es "010".

[11] Se utiliza una magnitud escalar debido a que el robot sólo puede avanzar o retroceder, pero no desplazarse lateralmente. Esto se debe a que las ruedas del robot no son holonómicas pues sólo definen un grado de libertad. En las simulaciones en 3D sí que sería necesario considerar más grados de libertad, y por tanto, definir una magnitud vectorial para la velocidad.

- *Alinearse a la Derecha*: esta acción le indica al robot que debe girar lentamente hacia la derecha (con el fin de alinearse con un objeto específico) a una velocidad de -10º/seg ≅ -0.174 rad/seg. Su codificación sub-simbólica es "011".
- *No Rotar*: esta acción le indica al robot que no debe girar, es decir, que su velocidad angular debe ser 0.0 rad/seg. Su codificación sub-simbólica es "100".
- *Rotar Aleatoriamente*: esta acción le permite al robot girar aleatoriamente hacia la izquierda o hacia la derecha, con una velocidad angular entre 20º/seg y -20º/seg, o lo que es lo mismo, entre 0.349 rad/seg y - 0.349 rad/seg. Su codificación sub-simbólica es "101". Esta acción resulta útil cuando el robot decide deambular y explorar su entorno.

Por ejemplo, cuando el agente cognitivo decide avanzar y reorientar su desplazamiento hacia un objeto que se halla a 10 metros de distancia y 45º a la izquierda, enviaría, probablemente, las señales de la tabla 11.8 a los correspondientes actuadores.

Actuador 1	**Actuador 2**
Velocidad de Navegación	Rotar a la Izquierda
01	000

Tabla 11.8: Ejemplo de formato de una acción que controla los actuadores de desplazamiento del robot.

Capítulo 12

Evaluación de la Arquitectura Cognitiva

"Ninguna cantidad de experimentación puede probar definitivamente que tengo razón; pero un solo experimento puede probar que estoy equivocado".
– Albert Einstein.

12.1. Motivación y Alcance de la Evaluación

Hasta el momento, se han propuesto y desarrollado un conjunto de hipótesis sobre el diseño de un modelo cognitivo artificial, capaz de auto-organizar sus propias estructuras cognitivas mediante procesos darwinianos que operan a diferentes escalas jerárquicas y temporales. Por tanto, es objeto del presente capítulo el llevar a cabo la etapa final del método deductivo-experimental que se ha seguido durante el desarrollo de la presente tesis, donde se verificarán las hipótesis propuestas a través de la experimentación, el análisis comparativo, y el subsecuente análisis de resultados.

Como se ha podido ver a lo largo de los capítulos anteriores, METÁFORA propone una arquitectura compleja, compuesta de múltiples componentes y mecanismos interoperantes, cuya evaluación debe ser acotada de manera apropiada con el fin de probar aquello que representa el aporte más sustancial e innovador en la presente tesis. Por tanto, el objetivo de la evaluación apunta hacia la validación de dos aspectos principales: (1) el desempeño de los componentes involucrados en los tres enfoques evolutivos propuestos: la *epigénesis*, la *ontogénesis*, y la *filogénesis* de estructuras cognitivas; y (2) la funcionalidad global de METÁFORA en contraste con otras arquitecturas cognitivas relevantes. Por consiguiente, se proponen dos tipos de estudio en la evaluación de la arquitectura:

- *Estudio Cuantitativo*: este estudio contrasta un conjunto de hipótesis con los datos reales obtenidos durante la experimentación. Para realizar esta verificación de las hipótesis es preciso someter los datos a un análisis estadístico, de manera que

se pueda descartar el hecho de que los resultados obtenidos se deben al azar o a algún factor no considerado. En concreto, se empleará un análisis estadístico inferencial a través del cual se medirá el desempeño, eficiencia, y robustez de aquellos componentes de la arquitectura bajo observación. El desarrollo de este estudio corresponde a los experimentos que se presentan en las secciones siguientes.

- *Estudio Cualitativo*: a través de este estudio se realiza una comparación funcional de las características de METÁFORA en relación a otras arquitecturas cognitivas relevantes (véase el anexo A). En este análisis se establecerán ciertos elementos de juicio a partir de los cuales se elaborará un análisis comparativo (*benchmarking*). Los resultados obtenidos en este estudio se presentan en las conclusiones finales de la presente tesis (ene l capitulo 13).

En cuanto al estudio cuantitativo, serán descartados de la experimentación (al menos dentro del alcance de este primer estudio) aquellos elementos de la arquitectura que no influyan de forma directa sobre los procesos EOF que se pretenden evaluar. La razón para esto es que el considerar la evaluación de todos y cada uno de los componentes de la arquitectura implicaría una tarea demasiado exhaustiva, la cual, en algunos casos, arrojaría resultados obvios que no aportarían información significativa sobre la eficiencia de aquellos componentes y mecanismos más innovadores propuestos en METÁFORA. Es por esto que la evaluación de aquellos componentes cuya eficiencia ha sido demostrada por otros autores en diversos estudios anteriores[1], resultaría redundante para el objeto del presente estudio. Por tanto, la experimentación consistirá en probar, de forma incremental, los distintos niveles de organización adaptativa de la arquitectura, empezando por el nivel más bajo (el epigenético), hasta llegar al más alto (el filogenético), pasando por el nivel intermedio (el ontogenético). Durante este ascenso por los niveles de organización, la interacción entre los distintos componentes y mecanismos intervinientes será validada de forma incremental, esto es, mientras que en el nivel inferior se realizarán pruebas particularizadas de los componentes más atómicos de la arquitectura, en los niveles superiores la experimentación tendrá como objetivo evaluar, de forma integrada, el desempeño de los sub-sistemas compuestos por elementos y mecanismos del nivel de organización inmediatamente inferior. Así pues, grosso modo, los elementos a evaluar durante la experimentación serán:

- *Nivel Epigenético:* en este nivel será probados y analizados los componentes involucrados en el aprendizaje de reglas, esto es, los componentes que conforman cada *Comportamiento* de METÁFORA (RP, RES, SIA, y PMR), y el Sistema Clasificador Anticipatorio (SA). Debido a que el aporte más innovador en este nivel de la arquitectura ha sido el diseño del componente SIA (especialmente por su cualidad de híbrido), el análisis llevado a cabo en este nivel tendrá como objetivo comparar el desempeño de dicho sistema con el de otros sistemas de aprendizaje equivalentes.
- *Nivel Ontogenético*: en este nivel se evaluará el desempeño del algoritmo evolutivo propuesto para auto-configurar la funcionalidad y la estructura de las Redes de Comportamientos.

[1] Por ejemplo, lo apropiadas que resultan ser las redes neuronales asociativas para el modelado de las memorias semánticas (Rumelhart y McClelland, 1986; Sun et al., 2001; Dodd, 2005).

- *Nivel Filogenético*: y por último, en este nivel se evaluará el desempeño del algoritmo memético propuesto, empleando para ello un escenario multi-agente (y/o multi-robot en este caso).

12.2. Marco Experimental

Debido a la necesidad de llevar a cabo una rigurosa metodología de comparación estadística entre los diferentes algoritmos propuestos, se describen en esta sección los métodos de inferencia estadística empleados en el estudio experimental, así como la clasificación de los distintos experimentos realizados.

12.2.1. Selección de Métodos de Análisis Estadístico

El teorema de *"No-Free-Lunch"* (NFL) (Wolpert y Macready, 1997) postula que por cada par de algoritmos de búsqueda/optimización, hay tantos problemas en el que el primer algoritmo es mejor que el segundo, como problemas en el que el segundo algoritmo es mejor que el primero. Una consecuencia de ello es que si no se introduce ningún conocimiento suplementario del dominio del problema en el algoritmo, es tan probable que funcione peor que la búsqueda aleatoria, como que funcione mejor. Por tanto, no es posible la elección entre dos algoritmos solamente basándose en lo bien que han funcionado con anterioridad sin conocer nada sobre el dominio del problema (hipótesis del teorema de NFL) y, por ende, se requiere tener un conocimiento parcial del problema, conocimiento que permite el diseño de algoritmos con características específicas que los hacen adecuados para la resolución de dicho problema.

Una vez identificado el conocimiento parcial del problema y la necesidad de disponer de algoritmos para su resolución, se plantea la cuestión de decidir cuándo un algoritmo es mejor que otro. En el caso del uso de metaheurísticas o algoritmos evolutivos, esto se debe hacer atendiendo a criterios de eficiencia y/o eficacia (García et al., 2007). Cuando no se dispone de resultados teóricos que permitan comparar el comportamiento de los algoritmos, es necesario entonces centrarse en el análisis riguroso de los resultados empíricos.

En la literatura especializada, se pueden encontrar diferentes propuestas para utilizar pruebas estadísticas en el análisis de experimentos (Bartz-Beielstein, 2006). La mayoría de los trabajos validan los resultados a partir de estadísticos básicos, como medias y varianzas, junto con técnicas estadísticas paramétricas (Czarn et al., 2004; Mori et al., 2005). Sin embargo, es bien conocido que las técnicas estadísticas paramétricas requieren de una serie de condiciones sobre la muestra de resultados que no siempre se cumplen: *independencia*, *normalidad* y *homogeneidad* de las varianzas (Zar, 1999; Sheskin, 2003).

En (Sheskin, 2003), la distinción que se hace entre pruebas paramétricas y no paramétricas se basa en el nivel de medida representado por los datos que van a ser analizados. De esta manera, una prueba paramétrica es aquella que utiliza datos reales (de tipo nominal, ordinal, o que representan un orden estableciendo un ranking) pertenecientes a un intervalo con una distribución de probabilidad determinada. Esto no implica que

siempre que se dispone de este tipo de datos, haya que usar una prueba paramétrica. Puede darse el caso de que una o más suposiciones que hacen las pruebas paramétricas se violen, haciendo que el análisis estadístico pierda credibilidad. Por otro lado, las pruebas estadísticas que no se basan en ninguna suposición en cuanto a la distribución de probabilidad a partir de la que fueron obtenidos los datos se denominan pruebas no paramétricas. Como normal general, una prueba no paramétrica es menos restrictiva que una paramétrica, aunque menos robusta que una paramétrica que se realiza sobre datos que cumplen todas las condiciones necesarias. Por tanto, para utilizar las pruebas paramétricas es necesario que se cumplan las siguientes condiciones (Zar, 1999; Sheskin, 2003):

- *Independencia*: en estadística, dos sucesos son independientes cuando el que haya ocurrido uno de ellos no modifica la probabilidad de ocurrencia del otro.
- *Normalidad*: una observación es normal cuando su comportamiento sigue una distribución normal o de Gauss con una determinada media μ y varianza σ. Por tanto, una prueba de normalidad sobre una muestra indica la presencia o no de esta condición sobre los datos observados. Existen varios tipos de prueba de normalidad, entre los más conocidos el contraste x^2 de Pearson y la prueba de Kolmogorov-Smirnov. Para los experimentos realizados en esta tesis se ha decidido emplear la prueba de *Shapiro-Wilk* (Shapiro y Wilk, 1965) puesto que permite contrastar fácilmente (de forma numérica y gráfica) el ajuste de los datos obtenidos a una distribución normal. Esta prueba calcula el nivel de simetría y la *curtosis* (o forma de la curva)[2] como resultado del análisis de los datos observados, para después calcular su diferencia con respecto a los de una distribución Gaussiana, obteniendo el valor p[3] a partir de la suma de los cuadrados de las discrepancias. En los experimentos, se aplica la prueba de normalidad con una nivel de significancia $\alpha = 0.05$ (5 % de error), de manera que un valor p mayor que α indica que la condición de normalidad es satisfecha.
- *Heterocedasticidad*: esta propiedad indica que existe una violación de la hipótesis de igualdad de varianzas. Por tanto, para poder llevar a cabo una prueba paramétrica se debe comprobar que existe *Homocedasticidad*. En este caso, la prueba de Levene (Levene, 1974) se utiliza para comprobar si k muestras presentan o no esta homogeneidad en las varianzas.

Cuando la naturaleza de los resultados observados cumplen con estos criterios se suele emplear una prueba paramétrica, usualmente el procedimiento ANOVA de un factor,

[2]En teoría de la probabilidad y estadística, la curtosis es una medida de la forma o apuntamiento de las distribuciones. Así, las medidas de curtosis (también llamadas de concentración central) tratan de estudiar la mayor o menor concentración de frecuencias alrededor de la media y en la zona central de la distribución. Si el ajuste es bueno, los puntos se deben distribuir aproximadamente según una recta a 45º en un gráfico Q-Q.

[3]En estadística, una hipótesis nula es una hipótesis construida para anularse o refutarse, con el objetivo de apoyar una hipótesis alternativa. Cuando se la utiliza, la hipótesis nula se presume verdadera hasta que una evidencia estadística en la forma de una prueba de hipótesis indique lo contrario. Por su parte, el valor p es el mínimo nivel de significancia tal que se puede rechazar la hipótesis nula. El valor p hace las veces de nivel de significancia crítico: para niveles menores que p la hipótesis nula se acepta, y para niveles mayores que p, la hipótesis se rechaza. Por tanto, el valor p asociado a un resultado observado es la probabilidad de obtener un valor como el observado o más extremo si la hipótesis nula es cierta.

o la prueba *t-student* para comparar dos muestras independientes (Sheskin, 2003). No obstante, como se argumenta en (García et al., 2007) y como se demostrará en el desarrollo de cada experimento, el análisis de resultados para un conjunto de algoritmos de búsqueda heurística, principalmente si se trata de algoritmos evolutivos, no suele cumplir con los tres criterios para desempeñar una prueba paramétrica, razón por la cual no se describirá al detalle este tipo de análisis y el proceso de inferencia estadística se centrará únicamente en las pruebas no paramétricas.

En otro orden de ideas, el estudio no paramétrico de los resultados obtenidos puede contemplar un análisis comparativo del desempeño de un conjunto de algoritmos utilizados en la solución de uno o varios problemas, para lo cual se puede seguir una de las siguientes alternativas: el *análisis de un único escenario* y el *análisis de múltiples escenarios*. El primero corresponde al estudio del desempeño de múltiples algoritmos sobre un único problema o escenario. El segundo supone el estudio de múltiples algoritmos sobre más de un problema simultáneamente, considerándose el hecho de que cada problema tiene un nivel de dificultad y que los resultados obtenidos entre los diferentes problemas no son comparables. El análisis de un único escenario es habitualmente utilizado para probar los algoritmos de Inteligencia Computacional y su uso se encuentra documentado en la literatura especializada (Dietterich, 1998). No obstante, aunque con frecuencia no se satisfacen las condiciones requeridas para el empleo de estadísticas paramétricas, un estudio estadístico paramétrico podría obtener conclusiones similares que si se utilizase uno no paramétrico. Sin embargo, en un análisis de múltiples problemas, una prueba paramétrica podría arrojar conclusiones erróneas (García y Herrera, 2008).

Para el análisis de un único escenario se empleará principalmente un análisis de las curvas de aprendizaje de los algoritmos observados, utilizando para ello el promedio de los resultados obtenidos en múltiples ejecuciones del mismo experimento. Esto es, gracias a que los distintos niveles organizativos de la arquitectura (epigenético, ontogenético, y filogenético) emplean algoritmos de enfoque estocástico (i.e., los procesos aleatorios de las diferentes máquinas darwinianas), los resultados en cada experimento pueden variar ligeramente, requiriéndose por tanto un método que refleje el comportamiento medio del sistema durante la ejecución de múltiples experimentos (todos ellos con las mismas condiciones iniciales) sobre el mismo escenario. Por tanto, a fin de obtener una muestra representativa y fiable de los resultados, se siguen los siguientes pasos:

1. Cada experimento se ejecuta un cierto número de épocas de aprendizaje, donde cada época implicará un número variable de ciclos cognitivos.
2. Luego, el mismo experimento es replicado un cierto número de veces (e.g., 50 veces). Cabe notar que siempre se emplean las mismas condiciones iniciales del experimento a excepción de las poblaciones de individuos las cuales se generan aleatoriamente para cada caso.
3. Después, los resultados son promediados para cada ciclo cognitivo o iteración, empleando para ello algún tipo de media confiable.
4. Por último, los resultados obtenidos son representados en una curva de aprendizaje con el fin de analizar su convergencia.

A continuación se describen los tres tipos de medias empleadas para promediar los datos obtenidos en cada experimento:

- *Media Aritmética*: esta media, también llamada simplemente promedio, es igual a la suma de todos los valores de un conjunto finito de números dividida entre el número de sumandos, tal que:

$$\overline{x} = \frac{\sum_{i=1}^{n} a_i}{n} = \frac{a_1 + \ldots + a_n}{n} \tag{12.1}$$

  Una de las limitaciones de la media es que se ve afectada por valores extremos: valores muy altos tienden a aumentarla mientras que valores muy bajos tienden a reducirla, lo que implica que puede dejar de ser una medida representativa de la población.

- *Media Geométrica*: corresponde a la raíz n-ésima del producto de todos los resultados analizados, tal que:

$$\overline{x}_g = \sqrt[n]{\prod_{i=1}^{n} a_i} = \sqrt[n]{a_1 \cdot a_2 \cdot \ldots a_n} \tag{12.2}$$

  La media geométrica es menos sensible que la media aritmética a los valores extremos, aunque es de significado estadístico menos intuitivo.

- *Media Armónica*: esta media es igual al recíproco, o inverso, de la media aritmética de los recíprocos de los resultados analizados, tal que:

$$\overline{x}_h = \frac{n}{\sum_{i=1}^{n} \frac{1}{a_i}} = \frac{n}{(\frac{1}{a_1} + \ldots + \frac{1}{a_n})} \tag{12.3}$$

  La media armónica resulta poco influida por la existencia de determinados valores mucho más grandes que el conjunto total de resultados observados, siendo en cambio sensible a valores mucho más pequeños que el conjunto. Esta media se suele utilizar para promediar niveles de rendimiento y desempeño de algoritmos (como es el objetivo de la mayor parte de los experimentos realizados en este capítulo).

Por otro lado, un estadístico que permite medir la precisión en las respuestas para un algoritmo dado es el *Error Cuadrático Medio* (Pielke, 1984). Por precisión se entiende el grado de correspondencia entre pares individuales de valores esperados y valores observados, donde los valores observados corresponden a los resultados obtenidos para cada algoritmo estudiado. Como se observa en la ecuación 12.4, el error cuadrático medio estima las diferencias en promedio entre los valores esperados y los valores observados.

$$E_{cm} = \sqrt{\frac{\sum_{k=1}^{n} (\phi_k - \phi_{(k,obs)})^2}{n}} \tag{12.4}$$

donde: n es el número de muestras analizadas. ϕ_k es el valor esperado para la iteración k. $\phi_{(k,obs)}$ es el valor observado para la iteración k.

El E_{cm} utiliza el método de suma de los cuadrados para que no se anulen los errores negativos con los errores positivos. Otra propiedad interesante del E_{cm} es que puede descomponerse como la suma de dos componentes: la varianza de las estimaciones más su sesgo al cuadrado (Zar, 1999); con lo cual, el algoritmo con menor E_{cm} coincidirá con el de menor varianza e indicará, además, una menor desviación (determinada por su sesgo) de los resultados obtenidos por cada algoritmo con respecto a los valores verdaderos o esperados.

En otro orden de ideas, las pruebas de comparaciones múltiples se basan en analizar un conjunto de hipótesis (familia de hipótesis, compuesta por m hipótesis) y rechazar el mayor número posible de ellas controlando que no se produzcan errores del Tipo I[4] (Sheskin, 2003). En la literatura especializada, existen numerosas técnicas de comparaciones múltiples, como por ejemplo, los procedimientos de Tukey, Scheffe, Dunnet, etc. (Zar, 1999; Sheskin, 2003). Sin embargo, la mayoría de ellos requieren un diseño (denominado ANOVA) anterior a su aplicación. La familia de procedimientos derivados del test de Bonferroni no requieren dicha condición, por lo que pueden ser aplicados en un conjunto de hipótesis cualquiera. En el caso de los experimentos realizados, se empleará el conjunto de hipótesis obtenido por la prueba de Friedman.

Para los experimentos propuestos que emplean un análisis de múltiples problemas, se suponen k algoritmos que son probados en n escenarios diferentes. Con el fin de obtener un valor de ranking medio R_j para el algoritmo j sobre todos los escenarios, se hará uso de la prueba de Friedman (Sheskin, 2003). Esta prueba se trata de un equivalente no paramétrico a la prueba de medidas repetidas ANOVA. Esa prueba calcula el orden de los resultados observados por algoritmo (para el algoritmo j con k algoritmos en total) para cada escenario, asignando al mejor de ellos el orden 1, y al peor el orden k. Bajo la hipótesis nula, que se forma a partir de suponer que todos los resultados de cada algoritmo son equivalentes y, por tanto, sus rankings son similares, el estadístico de Friedman es:

$$X_F^2 = \frac{12n}{k(k+1)}\left(\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right) \tag{12.5}$$

donde: el estadístico se distribuye acorde a X_F^2 con $k-1$ grados de libertad. $R_j = 1/n \sum_i r_i^j$. n es el número de escenarios.

Si la hipótesis nula es cierta y el número de muestras es moderadamente grande, la distribución de X_F^2 se aproxima a una chi-cuadrado con k - 1 grados de libertad; de forma que se rechaza la hipótesis nula para valores de F superiores al valor crítico para el nivel de significación fijado. En el caso de rechazar la hipótesis nula, lo cual indica que existen diferencias significativas entre los diferentes resultados en cada escenario, debe

[4]El error de tipo I o también llamado error de tipo alfa (alfa es la probabilidad de que ocurra este error), es el error que se comete cuando el investigador rechaza la hipótesis nula siendo ésta verdadera en la población. Es equivalente a encontrar un resultado falso positivo, debido a que el investigador llega a la conclusión de que existe una diferencia entre las hipótesis cuando en realidad no existe.

aplicarse un método de análisis estadístico a posteriori. Con el fin de brindar un mayor soporte al análisis estadístico, se propone emplear dos tipos de pruebas a posteriori de la familia Bonferroni, cuyos resultados serán luego cotejados. Estas dos pruebas son: el procedimiento original de *Bonferroni-Dunn* y el procedimiento *Hochberg*.

El procedimiento de Bonferroni-Dunn (Zar, 1999) es similar a la prueba de Tukey para ANOVA y se utilizará para comparar un algoritmo frente a los demás. La calidad de dos algoritmos es significativamente diferente si la correspondiente media de rankings es tan diferente como su diferencia crítica:

$$dc = q_\alpha \sqrt{\frac{k(k+1)}{6n}} \tag{12.6}$$

donde: el valor de q_α es el valor crítico para una múltiple comparación no paramétrica.

Formalmente, $q_\alpha = R_i - R_j$, donde R_i es el valor del ranking medio obtenido por la prueba de Friedman para el algoritmo de control, y R_j el ranking para el algoritmo a comparar. El valor de dc de cada hipótesis se puede utilizar para calcular la probabilidad de error (valor p) mediante la distribución normal (Zar, 1999). En total, se dispone de una familia de hipótesis, cada una asociada a su correspondiente valor p.

Por otro lado, el procedimiento de Hochberg (Hochberg, 1998) es una prueba decremental que define diferentes niveles de rechazo por hipótesis. Sea $p_1, \ldots, p_m$ los valores p ordenados (del más pequeño al más grande), $\alpha'_1, \ldots, \alpha'_m$ los niveles de significancia asociados, y $H_1, \ldots, H_m$ las correspondientes hipótesis, tal que, si $p_{(m-1)}$ es menor que $\alpha'_{(m-1)}$, entonces todas las hipótesis restantes son rechazadas, si no, $H_{(m-1)}$ es aceptada y $p_{(m-2)}$ es comparada con $\alpha_{(m-2)}$; y así sucesivamente. Este procedimiento es mucho más potente que el procedimiento de Bonferroni-Dunn, aunque este último es mucho más apropiado para ser visualizado a través de representaciones gráficas.

Así pues, la comparación de múltiples tipos de algoritmos será desempeñada, primero, a través de un método estadístico que evalúe las diferencias entre las medias muestrales relacionadas (prueba Friedman), esto es, los resultados obtenidos por cada algoritmo. Luego, una vez que esta prueba rechace la hipótesis de equivalencia de las medias, se procederá a identificar las diferencias concretas entre los algoritmos mediante la aplicación de los procedimientos estadísticos a posteriori (Hochberg y Bonferroni-Dunn), los cuales permiten comparar un algoritmo de control con dos o más algoritmos. En la figura 12.1 se presenta un diagrama de flujo del método empleado para realizar los análisis estadísticos[5].

12.2.2. Caracterización de los Experimentos

Como se ha comentado anteriormente, el propósito de la experimentación apunta hacia la evaluación del desempeño de los tres enfoques evolutivos propuestos: la epigénesis, la ontogénesis, y la filogénesis. En la figura 12.2 se representa una clasificación de los distintos tipos de experimentos que serán llevados a cabo, así como el propósito general de cada uno de ellos.

[5]Para realizar los diferentes análisis estadísticos se empleó: el paquete estadístico SPSS (la versión PASW Statistics 18) y la hoja de cálculo de Microsoft Excel.

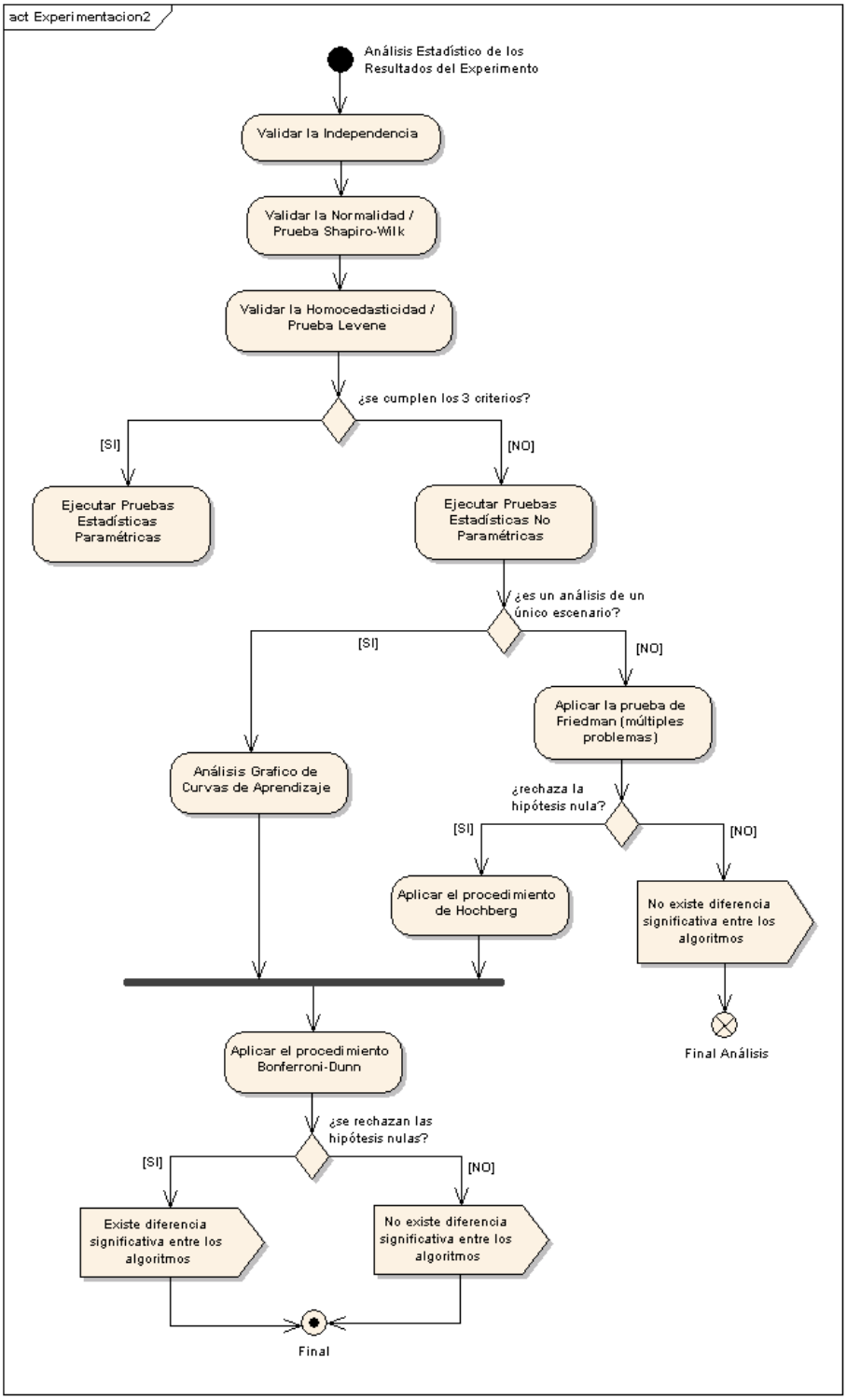


Figura 12.1: Flujo de Ejecución de las Pruebas Estadísticas.

Por consiguiente, los experimentos estarán agrupados según el nivel organizativo de la arquitectura que se desea evaluar, para lo cual se activarán o desactivarán determinados módulos y componentes de procesamiento de la arquitectura según convenga.

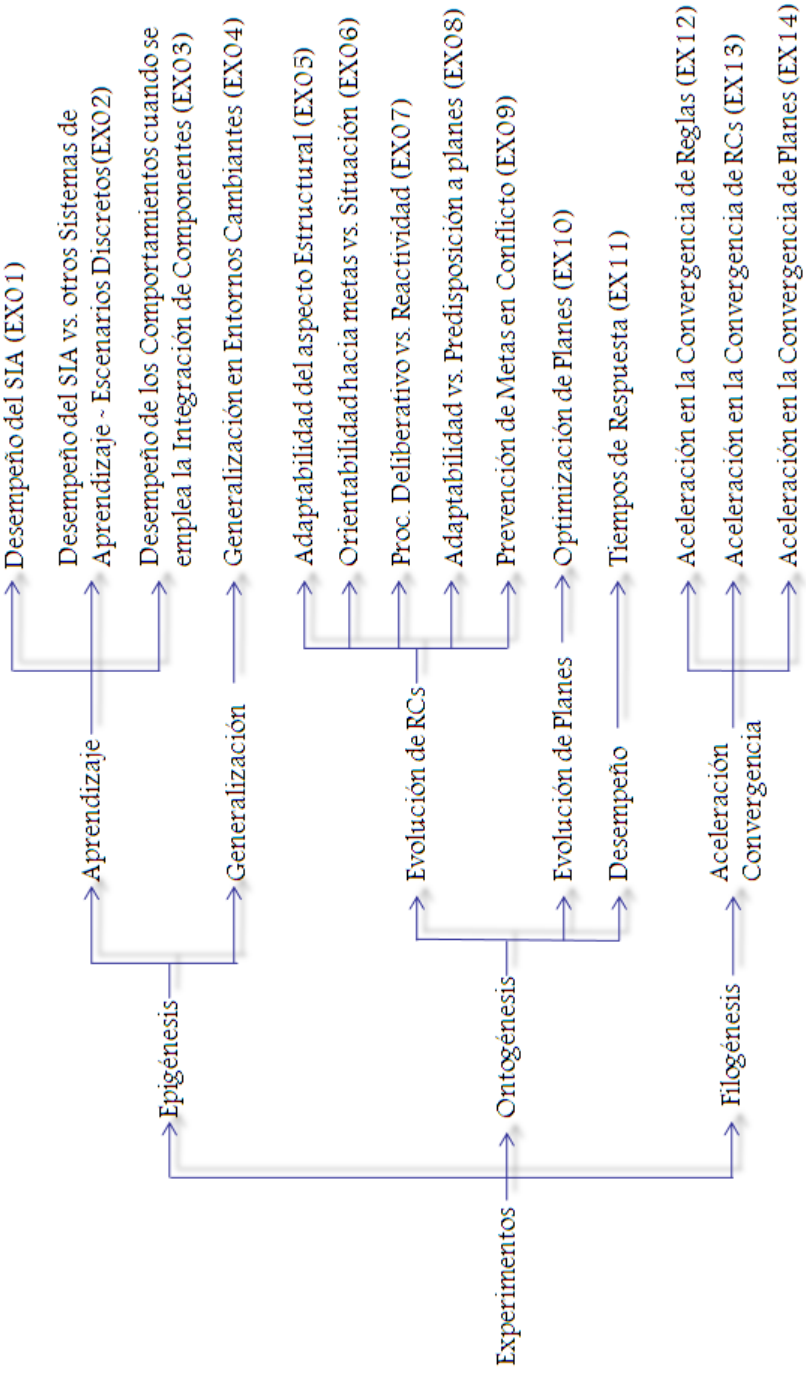


Figura 12.2: Estructura Taxonómica de los Experimentos.

Adicionalmente a esta estructura taxonómica, los experimentos pueden ser categorizados de acuerdo con el tipo de entorno empleado para su evaluación. Concretamente se tienen tres tipos:

- *Experimentos simulados empleando P/S*: en este tipo de experimentos se emplea *player/stage* para evaluar los comportamientos robóticos que permite simular dicha plataforma.
- *Experimentos simulados sin P/S*: debido a que la plataforma player/stage posee ciertas restricciones de simulación (e.g., el apilamiento vertical de objetos), se realizará entonces, para algunos de los experimentos, una implementación concreta de un escenario simulado a través del cual podrán ser evaluadas las características de los *Comportamientos* bajo observación.
- *Experimentos con Robots reales*: en esta prueba se utilizarán robots reales del tipo Pioneer, para contrastar los tiempos de respuesta entre la simulación y la ejecución real de ciertos *Comportamientos* más básicos (i.e., evadir-obstáculos, explorar, etc.).

12.3. Experimentos para el Nivel Epigenético

En esta sección se describen los diferentes experimentos realizados y su correspondiente análisis de resultados para el Nivel Epigenético de la arquitectura. Como se observa en la figura 12.3, a fin de efectuar los experimentos correspondientes a este nivel, se requiere solamente de cuatro módulos de la arquitectura. Incluso, en aquellos experimentos donde es analizado por separado el componente SIA, el resto de componentes del módulo Comportamental son desactivados (PMR, RP, y RES). Por tanto, el mecanismo de integración del módulo Efector solamente estará activo mientras reciba múltiples acciones recomendadas del módulo Comportamental, es decir, únicamente en aquellos casos donde estén habilitados dos o más componentes en los *Comportamientos* bajo estudio.

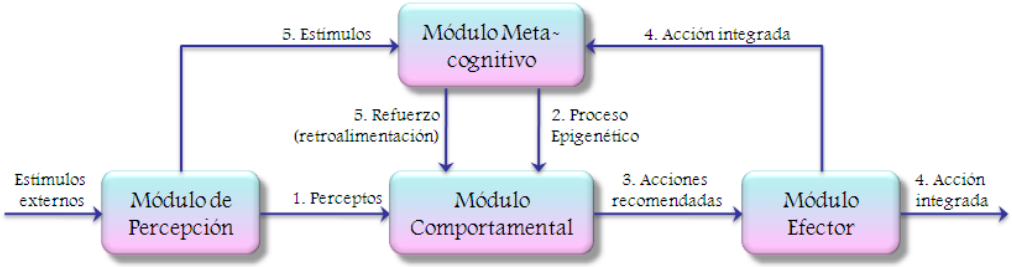


Figura 12.3: Módulos de METÁFORA activos durante los experimentos del Nivel Epigenético.

12.3.1. Experimento EX01: Análisis de Desempeño del SIA

12.3.1.1. Descripción

Este experimento tiene como propósito medir la tasa de convergencia del aprendizaje para el componente SIA de un *Comportamiento*. De todos los componentes de procesa-

miento propuestos para el *Comportamiento*, el análisis de desempeño se ha enfocado en el componente SIA debido a los siguientes motivos:

- El componente SIA es uno de los aportes más innovadores de la presente tesis. Representa un componente cognitivo híbrido que reúne las fortalezas de la teoría inmunológica, los algoritmos genéticos, y los sistemas clasificadores (como se discutió en la sección 7.4). Por tanto, resulta necesario validar el desempeño del componente en escenarios donde se requieran procesos de aprendizaje.
- Debido a que los componentes restantes para cada *Comportamiento* corresponden a sistemas cuyo desempeño ha sido ampliamente discutido en la literatura especializada, no se dedicará un análisis individual para validar su convergencia en escenarios de aprendizaje. Por ejemplo, en cuanto al componente PMR el lector puede consultar los resultados obtenidos en (Faulkner y Wei, 2000; Sun et al., 2001; Ratha y Jain, 2001; Alsmadi et al., 2009; Isa et al., 2010), entre otros. Por otro lado, en cuanto a los sistemas de reglas de producción, el lector puede consultar a (Clark y Karmiloff-Smith, 1993; Smolensky, 1988; Sun et al., 2001; Lehman et al., 2006; Roy, 2005; Mataric, 2001), entre otros.

En este experimento no se emplea la simulación bajo la plataforma player/stage debido a que el componente SIA será probado, de manera aislada al resto de la arquitectura, en un escenario de aprendizaje con estados discretos; esto con el fin de facilitar la obtención de resultados significativos y mensurables respecto a las tasas de aprendizaje y la velocidad de convergencia del algoritmo.

En este orden de ideas, se propone un entorno simplificado (no gráfico) que consiste de una rejilla de 5 x 5 casillas. En este escenario, el robot móvil (R) debe recolectar 4 cajas (C) que se encuentran situadas en el entorno, minimizando el número de pasos (desplazamientos) que debe ejecutar para recolectar todas las cajas (en este caso, el agente debe optimizar la trayectoria a solamente 4 pasos). Adicionalmente, dentro del entorno pueden hallarse obstáculos o muros (M) que no permiten el paso del robot. Por el contrario, si no existe ningún elemento en la casilla, se considera como un espacio vacío (“ ”) al cual puede desplazarse el robot. En cada instante de tiempo, en cada casilla podrá existir uno y sólo un objeto (R, C, M, o “ ”) de los descritos anteriormente.

El robot tiene una percepción de 360º sobre su entorno más inmediato (las 8 casillas alrededor de él) y está provisto de sensores para detectar qué tipo de elemento se encuentra en cada posición, sin embargo, en principio, el robot no reconoce los diferentes elementos detectados por los sensores, es decir, no sabe qué refuerzo se puede producir como consecuencia de desplazarse hasta alguna casilla específica. Por tanto, el desplazamiento hacia cada una de estos elementos generará un refuerzo que dirigirá el aprendizaje, el cual bien puede ser positivo (recoger una caja, con $r = 1$), negativo (intentar atravesar un muro, con $r = -1$), o nulo (un desplazamiento a un espacio vacío, con $r = 0$). El sistema por tanto, debe aprender a moverse siguiendo la ruta de las cajas, minimizando el número de desplazamientos y evitando los obstáculos.

Cada elemento del ambiente es codificado sub-simbólicamente de la siguiente manera:

- ⟨caja⟩ = “11”
- ⟨muro⟩ = “01”
- ⟨espacio-vacío⟩ = “00”

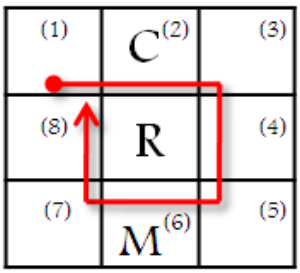


Figura 12.4: Lectura de los elementos en las casillas circundantes al robot.

Por tanto, la entrada sensorial que percibe el robot es una cadena de 16 bits, 2 bits por cada una de las 8 posiciones circundantes observadas. La lectura de los sensores se realiza en sentido de las manecillas del reloj, empezando por la casilla 1 y finalizando en la casilla 8, como se observa en la figura 12.4. Por ejemplo, la entrada sensorial de la figura quedaría codificada como: “00**11**0000000**01**0000”, e indicaría que el robot ha percibido una caja en la casilla 2, y un muro en la casilla 6. Por otro lado, las acciones del sistema corresponden a un desplazamiento hasta una de las 8 posiciones circundantes al robot, donde el desplazamiento hasta una caja implica el recogerla, y el desplazamiento a través de un muro no es permitido (i.e., el robot conserva su posición inicial sin producir movimiento alguno). La codificación sub-simbólica de las acciones es:

- ⟨mover-casilla-1⟩ = “000”
- ⟨mover-casilla-2⟩ = “001”
- ⟨mover-casilla-3⟩ = “010”
- ⟨mover-casilla-4⟩ = “011”
- ⟨mover-casilla-5⟩ = “100”
- ⟨mover-casilla-6⟩ = “101”
- ⟨mover-casilla-7⟩ = “110”
- ⟨mover-casilla-8⟩ = “111”

Por último, la hipótesis inicial que se desea comprobar es:

Ho: “el componente SIA demuestra una convergencia decreciente no prematura en su curva de aprendizaje”

12.3.1.2. Diseño de los Escenarios

Para este experimento se emplean tres tipos de escenarios. El primero de ellos, es un ambiente muy básico de navegación donde el robot tiene que recoger las 4 cajas dispuestas diagonalmente. El segundo escenario, similar al anterior, introduce nuevos elementos en el ambiente: los muros. En el tercer y último escenario, con un grado de complejidad mayor, la ubicación de las cajas es modificada por un patrón de distribución en forma de "V".

Los escenarios son probados cada uno en el orden descrito, empleando el conocimiento incremental adquirido por el robot. Esto es, que el robot emplea el conocimiento aprendido en el escenario 1 para actuar en el escenario 2, y el conocimiento adquirido en los dos escenarios anteriores para actuar luego en el escenario 3 (véase la gráfica 12.5). El diseño de complejidad incremental de los escenarios tiene por objetivo evaluar dos aspectos importantes:

- *Convergencia*: se estudia la tasa de convergencia en el aprendizaje de cada escenario (del más simple al más complejo).
- *Generalización*: se estudia la capacidad del SIA para relacionar el conocimiento adquirido incrementalmente en cada escenario. En este estudio, se compara la convergencia en el escenario más complejo cuando el SIA utiliza conocimiento aprendido previamente en los escenarios más simples, contra la convergencia en el mismo escenario complejo pero sin hacer uso de conocimiento previo.

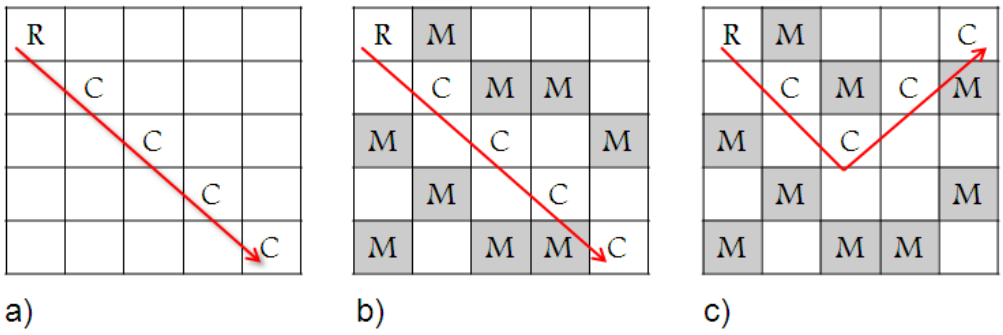


Figura 12.5: Escenarios para el Experimento 1. a) Distribución diagonal de cajas. b) Inclusión de Muros en el escenario. c) Distribución de cajas en forma de "V".

Por cada escenario, el algoritmo del SIA es ejecutado 50 épocas de aprendizaje, lo que corresponde a un *experimento*. Esto es, en cada época de aprendizaje, el robot ejecuta tantos pasos sean necesarios para recoger las 4 cajas situadas en el entorno. Al finalizar cada época de aprendizaje, el escenario es reinicializado (aunque conservándose el conocimiento aprendido por el SIA), situando al robot y a las cajas en sus posiciones iniciales. Adicionalmente, por cada escenario son ejecutados 30 experimentos; luego, por cada época de aprendizaje, se calcula el número promedio (i.e., la media aritmética, la media geométrica, y la media armónica) de pasos ejecutados por el robot durante los 30 experimentos; y finalmente se representan gráficamente los resultados.

Por último, en cuanto a la configuración de los parámetros para los experimentos realizados, se toman los valores por defecto propuestos en la sección 7.4. La población inicial de anticuerpos se define en 100 individuos, la cual puede aumentar dinámicamente hasta 300 anticuerpos.

12.3.1.3. Análisis e Interpretación de los Resultados

Determinación del Tipo de Prueba: con respecto al tipo de estudio de inferencia estadística utilizado para analizar los resultados de los experimentos, se concluye que no es posible aplicar una prueba paramétrica debido a que no se cumplen las tres condiciones para ello:

- *Independencia*: los experimentos son independientes puesto que son ejecuciones del algoritmo que emplean una población de anticuerpos inicial generada aleatoriamente para cada caso.
- *Normalidad*: la tabla 12.1 demuestra que ninguno de los experimentos en los diferentes escenarios cumplen con la prueba de normalidad aplicada (Shapiro-Wilk), debido a que el valor p de confianza es inferior al nivel mínimo de significancia ($\alpha = 0.05$) en todos los casos.

	Escenario 1	**Escenario 2**	**Escenario 3**
Valor p	0.043	0.037	0.011

Tabla 12.1: Prueba de Normalidad para EX01

- *Homocedasticidad*: la tabla 12.2 indica que ninguno de los experimentos llevados a cabo en los tres escenarios cumplen con la prueba de heterocedasticidad aplicada (Levene), puesto que las varianzas (basadas en las medias armónicas) de los resultados no son homogéneas, esto es, son inferiores a α.

	Escenario 1	**Escenario 2**	**Escenario 3**
Valor Levene	0,038	0.021	0.024

Tabla 12.2: Prueba de heterocedasticidad para EX01

Convergencia del SIA con y sin aprendizaje previo: dado que no se cumplen los criterios para realizar una prueba paramétrica, se rechaza la hipótesis nula y se procede al análisis de la convergencia en cada escenario. En la gráfica 12.6 se presentan tres curvas de aprendizaje (basadas en los promedios de los resultados obtenidos en el escenario 1) que representan la convergencia del sistema. Las curvas corresponden a la Media Geométrica, la Media Geométrica, y la Media Armónica para los 30 experimentos en cada escenario. La Media que presenta menor sesgo para este tipo de tendencias es la

Media Armónica. En la gráfica se observa que para el experimento realizado, después de promediar las 30 ejecuciones del SIA en el escenario 1, el robot aprende a recoger las 4 cajas empleando para ello solamente 4 pasos. Se evidencia además que el aprendizaje converge aproximadamente después de 16 épocas.

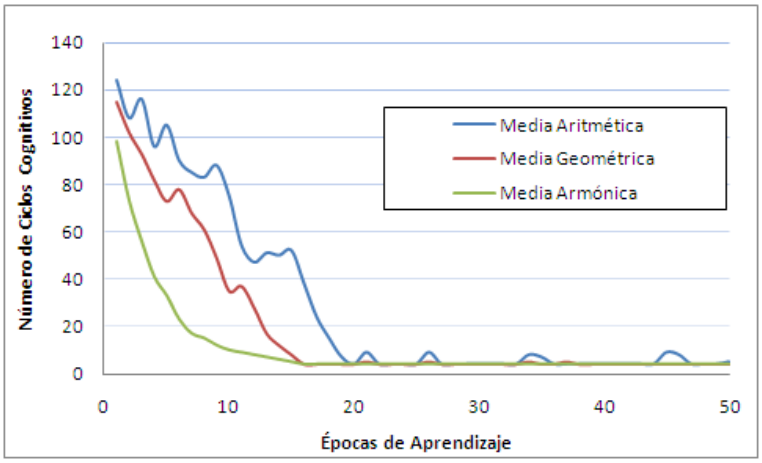


Figura 12.6: Curva de aprendizaje para el SIA (Escenario 1).

En las gráficas 12.7 y 12.8 se presentan los resultados para el escenario 2 y 3 respectivamente. Como se puede observar, en cada uno de los escenarios el SIA logra converger nuevamente, aunque esta vez empleando un número mayor de épocas de aprendizaje (para aquellos casos donde el SIA no emplea el conocimiento previamente adquirido). Esto es, el sistema converge después de 33 épocas de aprendizaje (en promedio) en el escenario 2, y después de 41 épocas de aprendizaje (en promedio) en el escenario 3. Intuitivamente, la razón por la cual el SIA requiere de más épocas de aprendizaje en los escenarios 2 y 3 se debe a la complejidad misma de los escenarios, es decir, mientras que en el escenario 1 el agente debe aprender un patrón de desplazamiento en diagonal muy simple que involucra únicamente el reconocimiento de las cajas, en los escenarios 2 y 3 el robot debe identificar nuevos elementos en su entorno (los muros) y aprender nuevos patrones de desplazamiento (en el caso del escenario 3).

Nótese en las gráficas 12.7.b y 12.8.b que la convergencia del aprendizaje se acelera cuando el SIA emplea el conocimiento adquirido en los escenarios previos, a una tasa porcentual de aproximadamente el 80 % en comparación con los experimentos donde el SIA no empleó conocimiento previo (como se puede observar en la columna de relación de desempeño de la tabla 12.3). Esta aceleración en el aprendizaje de nuevos escenarios se debe principalmente a la capacidad del SIA para *generalizar* el conocimiento previamente adquirido; por ejemplo, en uno de los experimentos realizados se identificó un anticuerpo creado en el escenario 1, el cual fue luego reutilizado en los escenarios 2 y 3: $\langle 000\#000\#11\#00\#00 : 100 \rangle$. Este anticuerpo representa una regla generalizada tal que, si el robot identifica una caja en la posición 5 (casilla inferior izquierda) entonces, sin importar que en las casillas 2, 3, y 7 existan muros o no (condición representada por los comodines #), el robot se desplazará hasta la casilla 5 y recogerá la caja. Un estudio más detallado sobre la generalización de conocimiento del SIA es presentado en el experimento 4.

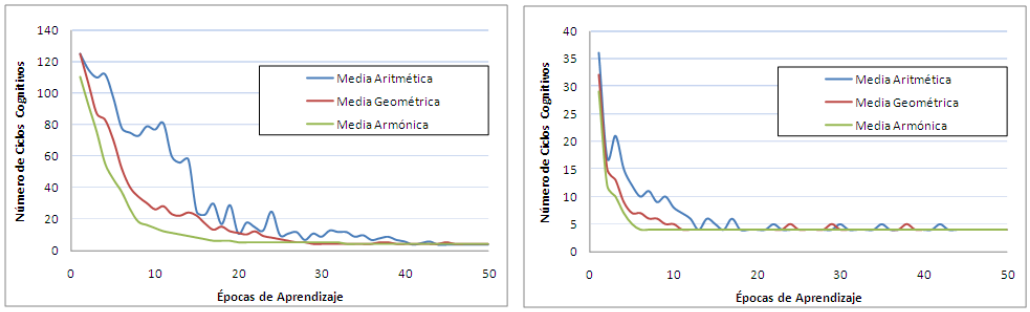


Figura 12.7: Curva de aprendizaje para el SIA - Escenario 2. a) Aprendizaje sin conocimiento previo. b) Aprendizaje a partir del conocimiento previo adquirido en el escenario 1.

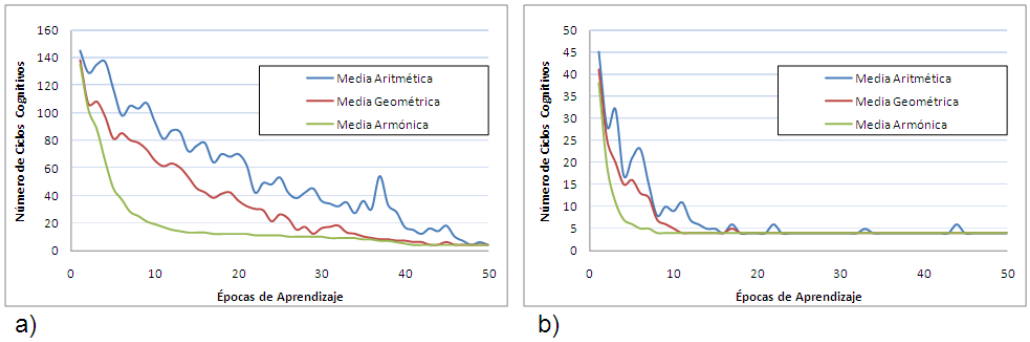


Figura 12.8: Curva de aprendizaje para el SIA - Escenario 3. a) Aprendizaje sin conocimiento previo. b) Aprendizaje a partir del conocimiento previo adquirido en los escenarios 1 y 2.

Escenario	Núm. épocas sin aprendizaje previo (A)	Núm. épocas con aprendizaje previo (B)	Relación de Desempeño [1 - (B/A)] %
Escenario 1	16	-	-
Escenario 2	33	6	81 %
Escenario 3	41	8	80 %

Tabla 12.3: Relación de Desempeño del SIA con y sin aprendizaje previo (empleando la media armónica)

Convergencia del SIA empleando Contraejemplos: con el propósito de comprobar el incremento en la eficiencia del proceso de aprendizaje del SIA cuando se activa el mecanismo de contraejemplos, se analizan comparativamente las curvas de convergencia de los tres escenarios propuestos. En la gráfica 12.9 se presentan las curvas de aprendizaje para cada escenario (empleando únicamente la media armónica de los resultados). Nótese que, en cada caso, la curva de aprendizaje del SIA cuando el mecanismo de contraejem-

plos está activado (CCE), converge más rápidamente que la curva de aprendizaje del SIA cuando dicho mecanismo está desactivado (SCE).

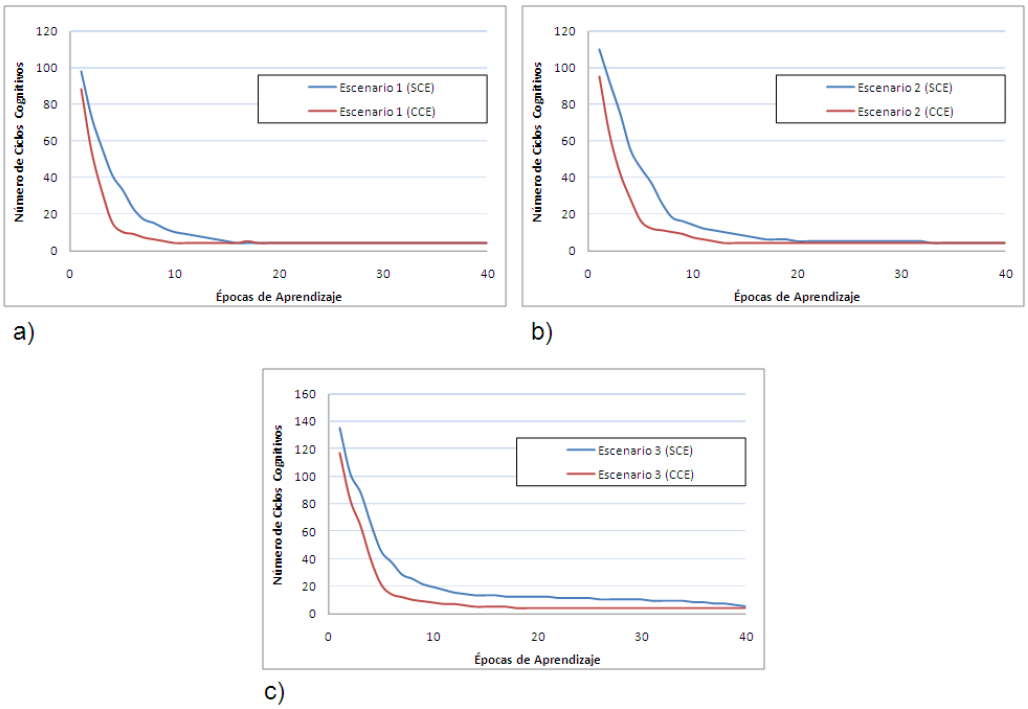


Figura 12.9: Comparativo entre las curvas de aprendizaje del SIA con el mecanismo de contraejemplos activado (CCE) y desactivado (SCE). a) Escenario 1. b) Escenario 2. c) Escenario 3.

En la tabla 12.4 se presentan las relaciones de desempeño del SIA para cada escenario propuesto. Nótese que cuando el SIA incorpora el mecanismo de contraejemplos se reduce el número de épocas de aprendizaje requeridas en los tres escenarios propuestos. Así pues, es importante destacar que el desempeño del SIA mejora significativamente cuando utiliza el mecanismo de contraejemplos, convergiendo más prontamente que la versión del SIA que no utiliza contraejemplos, como se puede observar en la columna de relación de desempeño de la tabla 12.4.

Escenario	Núm. épocas - SCE (A)	Núm. épocas - CCE (B)	Mejora del desempeño [1 - (B/A)] %
Escenario 1	16	9	43.7 %
Escenario 2	33	16	51.5 %
Escenario 3	41	18	56 %

Tabla 12.4: Relación de Desempeño del SIA (empleando la media armónica) con contraejemplos (CCE) vs. sin contraejemplos (SCE).

Otro criterio razonable para determinar qué versión del algoritmo (CCE o SCE) produce una convergencia más rápida hacia un estado estacionario es tomar aquella que cometa, en promedio, el menor error durante el proceso de aprendizaje. Para esto, se emplea el *Error Cuadrático Medio* de la ecuación 12.4. Para este experimento, n se define en 50 muestras, que se corresponde con las 50 épocas de aprendizaje ejecutadas por el SIA en cada escenario.

Por otro lado, en cada ciclo cognitivo se calcula un error en proporción al número de bits erróneos de la acción producida por el SIA. Por ejemplo, si la acción del SIA es "110" y la salida esperada es "010", el error sería: $1/3 = 0.3333$. Debido a que por cada época de aprendizaje el SIA es ejecutado un número variable de iteraciones, entonces el error de cada época es el promedio de los errores producidos en cada iteración, tal que:

$$e_k = \sum_{j=1}^{m} \frac{e_{(i,j)}}{m} \tag{12.7}$$

donde: m es el número de iteraciones ejecutadas en cada época de aprendizaje. e_i es el error entre la salida deseada y la salida del SIA producida en cada ejecución.

Así pues, se puede deducir a partir de los errores cuadráticos medios presentados en la tabla 12.5 que, en los casos donde el SIA incorpora el mecanismo de contraejemplos, éste no sólo converge más rápidamente hacia un estado estacionario (empleando menos épocas de aprendizaje), sino que además produce un E_{cm} menor en comparación con los casos donde no se utilizó dicho mecanismo.

Escenario	$E_{cm}(SCE)$	$E_{cm}(CCE)$
Escenario 1	1.39	0.88
Escenario 2	1.43	0.94
Escenario 3	1.65	1.06

Tabla 12.5: Error Cuadrático Medio calculado para cada escenario de aprendizaje.

Tiempo de Procesamiento de los Métodos del SIA: esta parte del experimento tiene como objetivo diagnosticar el desempeño de los diferentes métodos del SIA desde el punto de vista del tiempo de procesamiento (*CPU time*). Para esto, se empleó la herramienta JProbe 8.3[6]. En la figura 12.10 se presenta un diagrama de clases para el SIA, donde es posible observar los métodos que se han definido para la clase principal *SistemaInmune* (los métodos y atributos del resto de las clases no se presentan por cuestiones de simplicidad).

Los métodos de la Clase *SistemaInmune* que estarán bajo observación durante el diagnóstico, se listan a continuación[7]:

[6] El sitio we oficial de JProbe es: `www.quest.com/jprobe/`.

[7] Recuérdese que estos métodos corresponden con las dinámicas descritas para el SIA en la sección 7.4

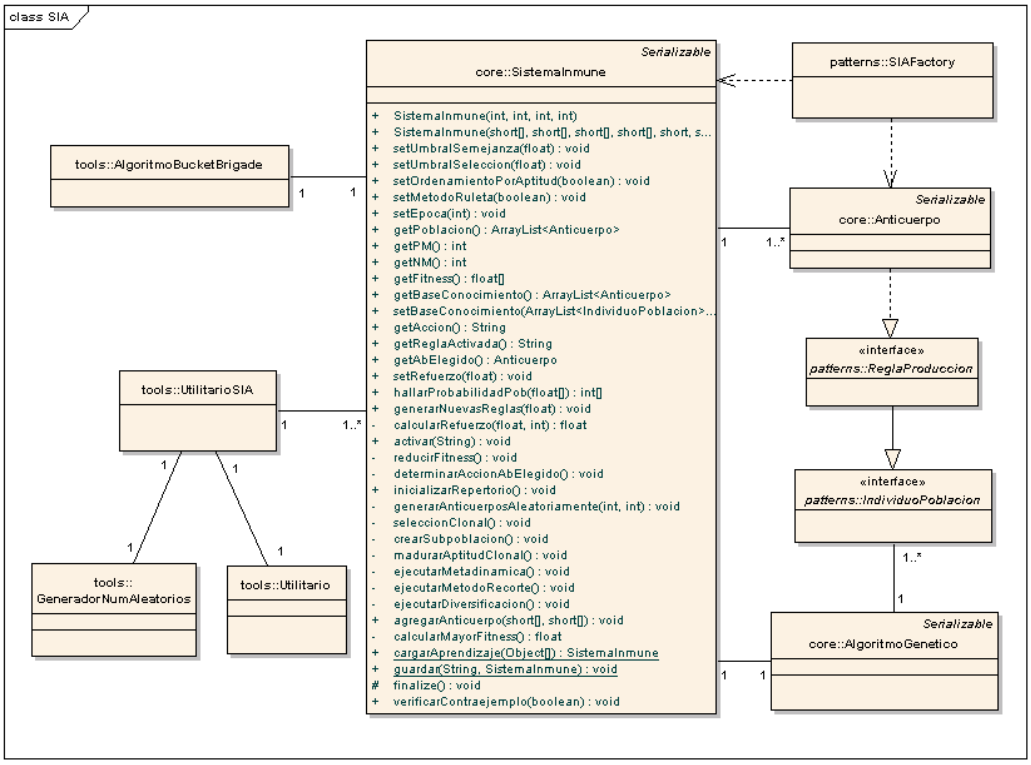


Figura 12.10: Diagrama de Clases simplificado para el SIA.

- *ejecutarDiversificiacion()*: invoca métodos del Algoritmo Genético.
- *mutarAnticuerpo()*: invoca métodos del Algoritmo Genético
- *ejecutarMetadinamica()*
- *ejecutarMetodoRecorte()*
- *madurarAptitudClonal()*
- *seleccionClonal()*
- *verificarContraejemplo()*
- *seleccionEstocastica()*
- *apostar()*
- *impartirImpuestoApuesta()*: invoca métodos del algoritmo de distribución de recompensas Bucket Brigade
- *realizarSubasta()*: invoca métodos del algoritmo Bucket Brigade

En la figura 12.11 se presentan los resultados del diagnóstico realizado con JProbe. Como puede observarse, el método que más consume tiempo de procesamiento (un 90 %) es *ejecutarMetadinamica()*. Esto se debe a que dicho método se encarga de restingir el aumento dinámico de anticuerpos, simulando una estrategia de "olvido-recuerdo" donde aquellos anticuerpos similares a otros pero con niveles de aptitud mucho menores, serán eliminados del repertorio de anticuerpos, mientras que aquellos con niveles más altos de

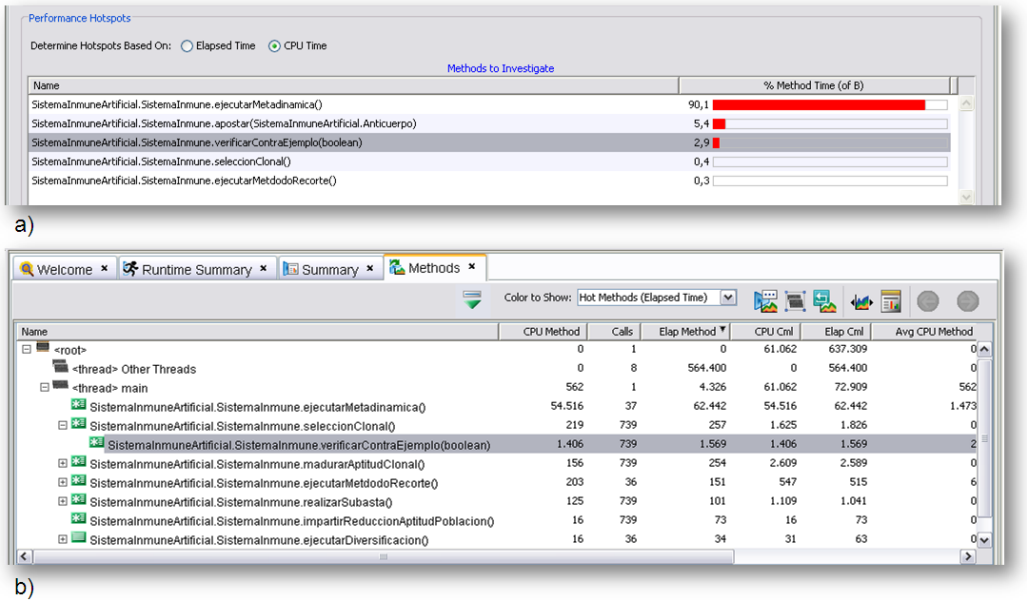


Figura 12.11: Diagnóstico de Desempeño del SIA utilizando JProbe. a) Porcentaje de los tiempos de procesamiento para cada método. b) Resumen de las estadísticas obtenidas por método.

aptitud serán conservados para su posterior maduración clonal. Debido a que el tamaño del repertorio cambia dinámicamente durante la ejecución del SIA, el tiempo de procesamiento que requiere este método también aumenta en proporción. Por otro lado, obsérvese que la inclusión del método *verificarContraejemplos()* no influye significativamente sobre el desempeño del sistema (requiriendo menos del 3 % del tiempo de procesamiento). Por último, los métodos que no se presentan en el reporte emplean menos del 0.9 % del tiempo de procesamiento, motivo por el cual no influyen significativamente sobre el rendimiento de sistema.

Conclusión del Experimento:

- Se comprueba, a partir del análisis de resultados de este experimento sencillo pero contundente, la capacidad del SIA para converger (en todos los diferentes escenarios propuestos) hacia un estado estacionario mediante un aprendizaje guiado por refuerzos ambientales.
- Se evidencia además una característica determinante en el proceso de aprendizaje: "la generalización de conocimiento como mecanismo de adaptación frente a cambios ambientales". Por tanto, se demuestra que el desempeño del sistema puede ser incrementado sustancialmente si el SIA generaliza las estructuras cognitivas aprendidas en experiencias pasadas, y las utiliza luego en la resolución de situaciones nuevas.

- Finalmente, se demuestra el aporte significativo que hace el mecanismo de contraejemplos en la mejora y aceleración de la convergencia del SIA, sin que esto afecte drásticamente en el desempeño del sistema.

12.3.2. Experimento EX02: Comparación del Desempeño del SIA vs. Otros Sistemas de Aprendizaje - Escenarios Discretos

12.3.2.1. Descripción

Este experimento tiene como objetivo comparar el desempeño del SIA contra otros sistemas de aprendizaje de reglas sub-simbólicas propuestos por otros autores. Con el fin de garantizar cierta uniformidad en el análisis comparativo, se proponen los siguientes criterios para la selección de los sistemas de aprendizaje contra los cuales se realizará la prueba de desempeño:

- Las entradas sensoriales, así como las salidas efectoras de cada sistema, deben poderse expresar en un lenguaje sub-simbólico representado por un alfabeto binario (0,1), o ternario en el caso de que se emplee un comodín para la generalización de pautas (#).
- Los algoritmos propuestos deben emplear un *aprendizaje por refuerzo* con el fin de que en cada paso pueda ser evaluada la acción ejecutada.
- El sistema debe ajustarse a un ciclo continuo de "percepción - procesamiento - actuación". Adicionalmente, el tiempo de respuesta de cada algoritmo para un ciclo completo debe restringirse a un intervalo entre 30 y 50 milisegundos.
- Los sistemas analizados deben emplear algoritmos basados en poblaciones de individuos, cuya mejora progresiva de la aptitud debe corresponder a un proceso estocástico.

Los algoritmos seleccionados para el presente experimento que cumplen con los criterios antes mencionados, se describen brevemente a continuación:

- *Sistema Clasificador de Aprendizaje (LCS)*: este sistema, propuesto por Holland (Holland et al., 1989), es un modelo de aprendizaje automático basado en clasificadores (reglas sub-simbólicas). El sistema define un algoritmo de distribución de recompensas (el *bucket brigade*) como mecanismo de selección de clasificadores, y un algoritmo genético que permite evolucionar la población total de clasificadores.
- *Sistema Clasificador Extendido (XCS)*: este sistema es una extensión propuesta por (Wilson, 2000) al LCS convencional. Algunas de las características diferenciadoras de un XCS más relevantes son: (1) la aptitud del clasificador se basa en la precisión que tiene para predecir el pago (refuerzo) y no en la predicción en sí misma; (2) el algoritmo genético actúa solamente sobre el sub-conjunto de acciones y no sobre toda la población de clasificadores; (3) define un mecanismo para la subsumción de clasificadores en lugar de emplear la eliminación directa; y (4),

en contraste con los LCS típicos, los XCS pueden ser aplicados tanto en entornos markovianos como no markovianos[8].

- *Sistema Clasificador Anticipatorio (ACS)*: este sistema, propuesto por Stolzmann (Stolzmann, 1998) y extendido luego por Butz, Goldberg y Stolzmann (Butz et al., 2002a), integran la teoría cognitiva del comportamiento anticipado con el diseño de los sistemas clasificadores típicos (LCS). El comportamiento algorítmico del ACS fue descrito en la sección 8.6.2.

Así pues, el aspecto que se desea evaluar en este experimento es el desempeño del SIA contra los otros tres sistemas propuestos, para lo cual se analizará la velocidad de convergencia de los algoritmos respectivos en cuatro escenarios diferentes. Por último, las hipótesis sujetas al análisis de resultados del presente experimento son:

Ho (nula): "El desempeño de los cuatro algoritmos analizados es igual."
H1 (alternativa): "Existe una diferencia significativa entre el desempeño del SIA y los demás sistemas de aprendizaje propuestos."

12.3.2.2. Diseño de los Escenarios

Para poder comparar adecuadamente el desempeño del SIA contra los sistemas de aprendizaje seleccionados, el estudio comparativo se dividió en dos partes: (1) un análisis comparativo de desempeño empleando un único escenario (el presentado en la figura 12.12); y (2) un análisis comparativo de desempeño empleando múltiples escenarios (los tres escenarios descritos en el experimento 1 y el escenario de la figura 12.12). La justificación para dividir este estudio en dos partes se debe a que, por un lado, el primer análisis permite visualizar (a través de una curva de aprendizaje) la velocidad de convergencia de cada algoritmo en la solución de un escenario complejo y, por otro, el segundo análisis permite comparar y cuantificar la diferencia en el desempeño de los algoritmos cuando se emplean múltiples escenarios simultáneamente (del más simple al más complejo).

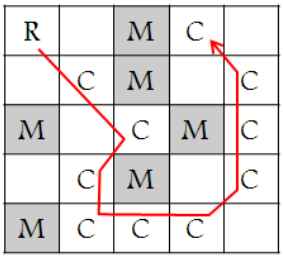


Figura 12.12: Escenario de Aprendizaje 4 (recorrido óptimo: 10 desplazamientos).

Puesto que todos los sistemas de aprendizaje participantes en este estudio emplean un algoritmo genético, se propone una configuración paramétrica estándar para todos

[8]El código fuente del XCS se encuentra disponible en `http://www.coboslab.psychologie.uni-wuerzburg.de/code/`.

ellos que garantiza un punto de partida uniforme a partir del cual se podrán identificar las diferencias sustanciales en el desempeño de cada algoritmo. Para la definición de estos parámetros (véase la tabla 12.6) se tomaron en cuenta dos criterios: (1) la configuración paramétrica sugerida en (Eiben y Smith, 2003) para algoritmos genéticos, que permite un equilibrio entre las propiedades de diversidad y convergencia; y (2) los resultados empíricos obtenidos y divulgados previamente en (Romero y De Antonio, 2008a; Romero y De Antonio, 2009a). En cuanto a los parámetros adicionales que define el XCS pero que no se presentan en la tabla 12.6 (i.e., la tasa de aprendizaje, el factor de descuento, el umbral de subsumción, etc.), se utilizaron los valores empíricos sugeridos en (Butz y Wilson, 2000).

Parámetro	**Valor**
Número de Experimentos	30
Número de Épocas	100
Población máxima de individuos (todos)	300
Impuesto de Vida (SIA y LCS)	0.005
Impuesto de Apuesta (SIA y LCS)	0.003
Probabilidad de Cruce (todos)	0.3
Probabilidad de Mutación (todos)	0.5
Umbral de semejanza (SIA)	20 %
Núm. de clonaciones por refuerzo positivo (SIA)	4
Núm. de clonaciones por refuerzo negativo (SIA)	1
Núm. de mutaciones por refuerzo positivo (SIA)	1
Núm. de mutaciones por refuerzo negativo (SIA)	4

Tabla 12.6: Configuración paramétrica de los Sistemas de Aprendizaje

12.3.2.3. Análisis e Interpretación de los Resultados

Análisis de Convergencia para un único escenario: en este estudio se compara el desempeño del SIA contra los otros 3 sistemas de aprendizaje. Para esto, se emplea un análisis gráfico de las curvas de convergencia para cada algoritmo, como se presenta en la gráfica 12.13. Para cada curva se emplean como muestras las medias armónicas de las 30 ejecuciones del experimento. Como se puede observar en la gráfica, las curvas de aprendizaje para los sistemas SIA, XCS, y ACS resultan ligeramente diferentes entre sí, mientras que, la curva de aprendizaje para el LCS demuestra claramente una convergencia más lenta.

En la tabla 12.7 se presentan el número de épocas de aprendizaje requeridas por cada sistema aprendizaje hasta converger a un estado estacionario, y las relaciones de desempeño del SIA en comparación con cada uno de los sistemas de aprendizaje estudiados. Como se puede observar, el SIA demuestra una tasa de desempeño más alta que los sis-

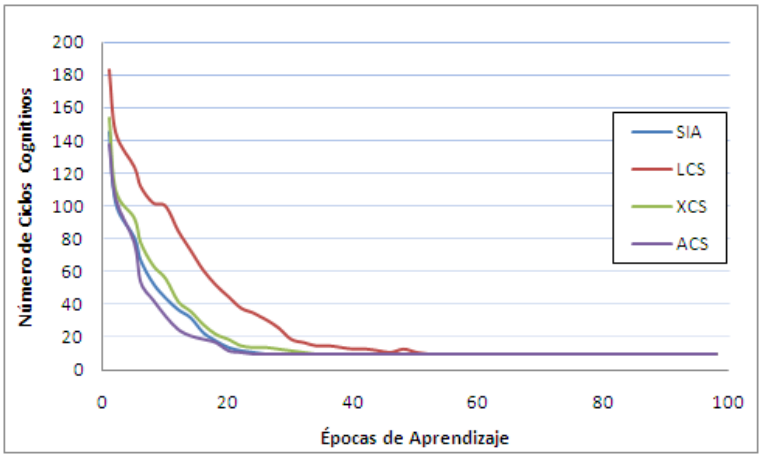


Figura 12.13: Curvas de aprendizaje para los sistemas SIA, LCS, XCS, y ACS.

temas LCS y XCS, aunque por otro lado, un poco más baja que la del sistema ACS. La razón para que el ACS haya convergido más prontamente que el SIA se debe principalmente a la capacidad anticipatoria del ACS que le permite predecir las consecuencias de ejecutar una acción determinada en cada estado observado. No obstante, es posible determinar si la diferencia en el desempeño de cada sistema es significativa empleando un análisis de múltiples escenarios.

Sistema (S)	Núm. Épocas - convergencia	Relación de Desempeño [1 - (SIA / S) %]
SIA	26	–
LCS	53	49.05 %
XCS	33	21.21 %
ACS	24	-8.33 %

Tabla 12.7: Relación de Desempeño del SIA vs. los otros sistemas de aprendizaje.

Análisis de Convergencia para múltiples escenarios: en este análisis se compara el desempeño de los cuatro sistemas de aprendizaje en múltiples escenarios (escenarios 1-4), empleando para ello una prueba no paramétrica. Tomando el supuesto nuevamente de que los experimentos son independientes (debido a la generación aleatoria de la población de individuos), se procede a verificar los criterios de normalidad y homocedasticidad. La tabla 12.8 muestra los resultados de la prueba de normalidad Shapiro-Wilk donde el símbolo "*" indica que no se cumple la normalidad y el valor entre paréntesis se trata del valor p de confianza necesario para rechazar o aceptar la hipótesis de normalidad. La tabla 12.9 muestra los resultados aplicando la prueba de Levene, en donde el símbolo "*" indica que las varianzas para un determinado escenario no son homogéneas.

Sistema	Escenario 1	Escenario 2	Escenario 3	Escenario 4
SIA	(.07)	(.08)	*(.02)	*(.01)
LCS	*(.03)	*(.04)	*(.01)	*(.00)
XCS	(.11)	(.10)	*(.04)	*(.01)
ACS	(.08)	(.08)	*(.00)	*(.02)

Tabla 12.8: Prueba de Normalidad Shapiro-Wilk

Sistema	Escenario 1	Escenario 2	Escenario 3	Escenario 4
SIA	(.09)	*(.01)	*(.02)	*(.00)
LCS	(.08)	(.08)	*(.00)	*(.01)
XCS	*(.04)	*(.02)	*(.02)	*(.00)
ACS	(.13)	(.09)	*(.01)	*(.00)

Tabla 12.9: Prueba de Heterocedasticidad Levene

En el apéndice 17 se presentan los histogramas de frecuencias y los gráficos Q-Q (con una distribución normal de contraste) para los resultados del aprendizaje de cada sistema en los cuatro escenarios propuestos. Así pues, a partir de los resultados obtenidos (tanto gráfica como numéricamente) queda patente el incumplimiento de las condiciones de normalidad y homocedasticidad necesarias para el uso de una prueba paramétrica. Se procede entonces con un análisis estadístico no paramétrico.

A continuación se aplica la prueba de Friedman para comprobar si existe una diferencia significativa en los resultados de cada escenario. La prueba se ha dividido en dos grupos atendiendo el grado de dificultad de los escenarios: el primer grupo contiene los escenarios 1 y 2 que consisten en el aprendizaje de un patrón de desplazamiento diagonal, y el segundo grupo contiene los escenarios 3 y 4 que requieren el aprendizaje de un patrón de desplazamiento más complejo. Los valores marcados con un símbolo "*" determinan el mayor valor entre el valor Friedman y el estadístico X^2, indicando, en el caso de que el valor Friedman sea el mayor, que la hipótesis nula debe ser rechazada. La tabla 12.10 resume los resultados de la prueba.

Escenario	Valor Friedman	Valor en X^2
Escenario 1 y 2	(1.42)	*(5.23)
Escenario 3 y 4	*(7.65)	(5.23)
Todos los escenarios	*(13.12)	(5.23)

Tabla 12.10: Prueba de Friedman

La prueba de Friedman advierte la existencia de diferencias significativas entre los resultados observados en los escenarios del grupo 2 y en todos los escenarios a la vez.

Por tanto, un análisis estadístico a posteriori en el caso de los escenarios 1 y 2 no va a ser necesario, puesto que los resultados son similares. En la tabla 12.11 se presentan los rankings obtenidos para cada algoritmo mediante la prueba de Friedman.

Algoritmo	Ranking
SIA	3.51
LCS	8.36
XCS	4.25
ACS	3.14

Tabla 12.11: Rankings de los algoritmos obtenidos mediante la prueba Friedman

Mediante el procedimiento descrito en la sección 12.2.1, se calcula la diferencia de rankings entre el algoritmo SIA (algoritmo de control) con respecto al resto, obteniendo el valor dc de cada comparación. Con este valor, se calcula el valor p_i, con $i = 1, ..., k$, para cada una de las k hipótesis. En este caso, existen 3 hipótesis, donde cada una afirma que no existe una diferencia significativa entre los resultados de desempeño del SIA y los del sistema de aprendizaje a comparar (i.e., SIA vs. LCS, SIA vs. XCS, y SIA vs. ACS). La Tabla 12.12 presenta la estimación de los valores p_i correspondientes a cada hipótesis, junto con los valores ajustados por cada procedimiento de comparaciones múltiples utilizado (Bonferroni-Dunn y Hochberg), en orden incremental con respecto al valor de p_i.

i	SIA vs.	dc	p	Bonferroni-Dunn	Hochberg
1	LCS	4.543	0.0023	0.01	0.0020
2	XCS	3.237	0.0049	0.01	0.0042
3	ACS	1.763	0.0245	0.01	0.0226

Tabla 12.12: Pruebas Posteriori (considerando $\alpha = 0.05$)

A partir de los resultados presentados en la tabla 12.12 se puede señalar lo siguiente:

- La prueba Bonferroni-Dunn rechaza las hipótesis con $p < 0.01$, es decir, se rechazan las hipótesis 1 y 2 con lo cual se deduce, para esta prueba, que el desempeño del SIA es mejor que el de los sistemas LCS y XCS.
- La prueba Hochberg rechaza las hipótesis con $p < \alpha(0.05)$, esto es, se rechazan las tres hipótesis, lo cual indica que el SIA tiene un desempeño significativamente mejor en comparación con cada uno de los otros tres sistemas.

Conclusión del Experimento: A partir de los resultados obtenidos en las pruebas no paramétricas, se puede concluir que el SIA tiene claramente un mejor desempeño que los algoritmos LCS y XCS en todos los escenarios propuestos. Esta diferencia en el desempeño resultó más evidente en los escenarios más complejos, puesto que todos los sistemas

en general necesitaron un mayor número de épocas de aprendizaje para encontrar la solución óptima, pero era el SIA quien convergía más prontamente en comparación con LCS y XCS.

Por otro lado, el SIA demostró tener un mejor desempeño que el ACS en la prueba Hochberg, mientras que, en la prueba de Friedman, el ACS obtuvo un ranking mejor que el del SIA aunque no con una diferencia considerable. En cuanto a la prueba Bonferroni-Dunn, se acepta la hipótesis nula de que "no existen" diferencias entre el desempeño del SIA y del ACS, con un nivel de significancia del 0.05, sin embargo, si se emplea un nivel de significancia mayor (e.g., $\alpha = 0.10$), la hipótesis es rechazada y se podría concluir que el desempeño del SIA es mejor que el del ACS. No obstante, si se analizan los resultados para cada escenario por separado es posible notar que el SIA converge más rápidamente que el ACS en los escenarios 2 y 3, mientras que el ACS lo hace para los escenarios 1 y 4. Por tanto, sería conveniente emplear una prueba no paramétrica más potente que la de Bonferroni-Dunn (e.g., Holm, Hommel y Rom, etc. (Sheskin, 2003)), y probablemente más escenarios, a fin de detectar un mayor número de diferencias entre los algoritmos.

En términos generales, el SIA, a pesar de no incorporar un mecanismo anticipatorio, demuestra un desempeño eficiente y una capacidad de convergencia superior (en la mayoría de los casos) que los sistemas comparados. Por otro lado, se deduce que la potencia del SIA como sistema cognitivo de procesamiento sub-simbólico se debe a que cumple con tres principios básicos:

- *Es adaptativo*: el SIA modifica su comportamiento interno para ser cada vez más apto (esto se refiere a la supervivencia diferencial de los mejores anticuerpos guiada por el proceso darwiniano) frente al ambiente cambiante en el cual se encuentra inmerso.
- *Es robusto*: aunque fallen algunos de sus componentes, su comportamiento global es estable y tolerante a fallos, es decir, nunca colapsa por completo. Incluso, en caso de sufrir daños como el olvido de conocimientos (eliminación de anticuerpos) causado por la acción del *método de recorte* y las *meta-dinámicas* del algoritmo, el sistema puede auto-repararse, lo cual implica la recuperación de material constitutivo como las estructuras cognitivas que ya había adquirido previamente. Por ejemplo, cada vez que el SIA es situado en un ambiente diferente, éste tiende a olvidar el conocimiento previo que no es utilizado con mucha frecuencia en el nuevo ambiente; no obstante, el algoritmo se apoya en dos estrategias para mitigar el impacto del olvido: (1) emplea una memoria inmune limitada de largo plazo donde almacena los anticuerpos que han tenido un desempeño excepcional, protegiéndolos contra las dinámicas de eliminación y utilizándolos como material de primera para la maduración clonal; y (2) en el caso de que un anticuerpo potencialmente bueno sea eliminado antes de ser almacenado en la memoria inmune, éste puede ser nuevamente re-descubierto gracias a las dinámicas de selección y maduración clonal y al proceso evolutivo dirigido por la máquina darwiniana.
- *Es efectivo*: el SIA tiene un conjunto de objetivos que alcanza exitosamente. Es decir, el algoritmo orienta siempre su comportamiento hacia un estado estable (convergencia).

12.3.3. Experimento EX03: Desempeño de los Comportamientos cuando se emplea la Integración de Componentes

12.3.3.1. Descripción

El propósito de este experimento es demostrar que la *integración* e *hibridación* de los componentes de enfoque cognitivista y emergente definidos por un *Comportamiento*, mejora sustancialmente el desempeño global del sistema y lo hace más tolerante a fallos. Concretamente, en este experimento se compara el desempeño por separado de los componentes PMR, SIA, RP, y RES, contra el desempeño del sistema cuando emplea la salida integrada de dichos componentes en la resolución de dos tipos diferentes de tareas. Los mecanismos de integración que serán probados son la *Combinación Basada en Pesos* y la *Agregación por Votación*, ambos descritos en la sección 7.7.

Las hipótesis sujetas al análisis de resultados del presente experimento son:

Ho (nula): "El desempeño del Comportamiento cuando emplea la integración de componentes no presenta diferencia en comparación con el desempeño de los componentes por separado."
H1 (alternativa): "Existe una diferencia significativa en el desempeño del Comportamiento cuando emplea la integración de componentes."

12.3.3.2. Diseño de los Escenarios

Para este experimento se proponen dos escenarios diferentes empleando la plataforma *player/stage*, los cuales se describen a continuación:

- **Escenario 1:** el robot se encuentra dentro de un recinto cerrado y debe aprender a no colisionar contra los muros. Adicionalmente, el robot debe aprender a explorar todo su entorno en un número limitado de ciclos cognitivos. Para esto, el robot es provisto con dos *Comportamientos*: ⟨`Evadir-Obstáculos`⟩ y ⟨`Explorar--Entorno`⟩.
- **Escenario 2:** el robot debe buscar una caja que se encuentra ubicada en alguna posición del entorno, y una vez localizada, debe aprender a ejecutar las acciones correctas para recoger dicha caja. Los *Comportamientos* definidos para esta tarea son: ⟨`Identificar-Caja`⟩ y ⟨`Recoger-Caja`⟩.

A continuación, se describe la configuración de las entradas y las salidas para cada uno de los *Comportamientos* (véase la sección 11.4 que describe los tipos de perceptos utilizados en la simulación):

- ⟨`Evadir-Obstáculos`⟩ (EO): como entradas, recibe las 16 lecturas del sonar ultrasonido y, como salidas, controla los actuadores de desplazamiento y de rotación (proxies Sonar y Position2d).

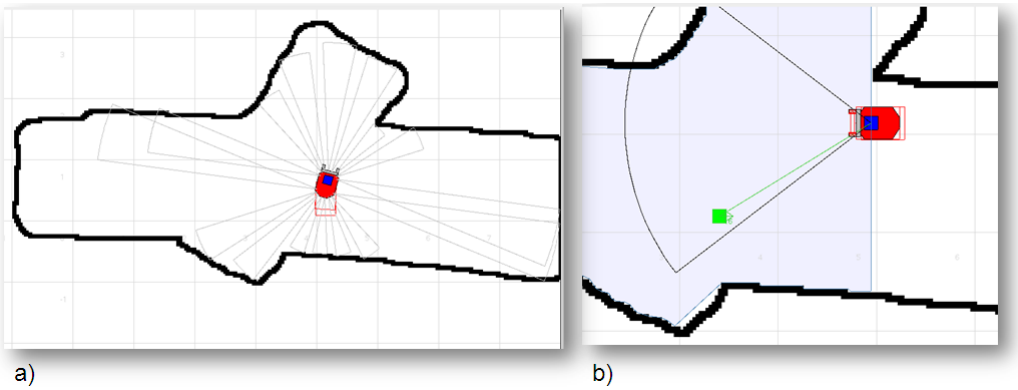


Figura 12.14: Escenarios de prueba para el Experimento EX03. a) Escenario 1. b) Escenario 2.

- ⟨`Explorar-Entorno`⟩ (EE): utiliza la misma configuración que Evadir Obstáculos.
- ⟨`Identificar-Caja`⟩ (IC): como entradas, recibe un vector con máximo diez lecturas de los objetos identificados por los proxies Fiducial y Laser y, como salidas, controla los actuadores de desplazamiento y de rotación.
- ⟨`Recoger-Caja`⟩ (RC): como entradas, recibe la distancia y la orientación relativa de la caja que debe debe recoger el robot (después de haberla hallado) y, como salidas, controla los actuadores de desplazamiento, de rotación, y de la pinza (gripper).

Con respecto al componente PMR de cada *Comportamiento*, la tabla 12.13 resume los parámetros de configuración. Nótese que el número de neuronas en la capa de entrada y de salida se corresponde directamente con el número de entradas y salidas definidas anteriormente. Los demás parámetros se han seleccionado tomando como referencia los estudios de (Kuzmin, 2002), y realizando ciertos ajustes para el problema propuesto. Cabe mencionar que para este experimento se han configurado los componentes PMR con un aprendizaje por refuerzo, esto es, que la red neuronal emplea el algoritmo de *Q-learning* y ajusta los pesos sinápticos en tiempo de ejecución (sin previo entrenamiento de la red). En cuanto al componente SIA, se define el mismo número de entradas y de salidas definidas para el componente PMR, y se emplea la configuración paramétrica descrita en el experimento EX02.

Por otro lado, con el objeto de comprobar la manera en que se integra la línea de razonamiento del componente RP con la de los demás, se define un conjunto mínimo de reglas simbólicas. Como se ha discutido en capítulos anteriores, el objetivo de estas reglas es proveer conocimiento experto que guíe las fases iniciales de aprendizaje del sistema. No obstante, la responsabilidad de construir las estructuras cognitivas durante la interacción del agente con el ambiente recaerá principalmente en los componentes que definen mecanismos de aprendizaje y generalización de conocimiento (i.e., PMR, SIA, y RES). Por tanto, se definen únicamente las siguientes cuatro reglas para el *Comportamiento* ⟨`Evadir-Obstáculos`⟩ (por motivos de simplificación, sólo se presentan las

Parámetro	EO	EE	IC	RC
Num. neuronas capa entrada	16	16	10	5
Num. neuronas capa oculta	30	30	28	8
Num. neuronas capa salida	5	5	5	6
Umbral de Activación	0.8	0.8	0.8	0.8
Tasa de Aprendizaje	0.05	0.05	0.05	0.05
Factor de Descuento	0.95	0.95	0.95	0.95

Tabla 12.13: Configuración Paramétrica de los componentes PMR

condiciones que detectan un obstáculo en una posición dada, para el resto de posiciones se asume que no se detecta ningún obstáculo y por tanto no se presentan):

SI
⟨ubicación, adelante⟩⟨elemento, obstáculo⟩ Y
⟨ubicación, der-adelante⟩⟨elemento, obstáculo⟩
ENTONCES
⟨velocidad, detenerse⟩ Y
⟨rotación, rotar-izquierda⟩

SI
⟨ubicación, izq-adelante⟩⟨elemento, obstáculo⟩ Y
⟨ubicación, adelante⟩⟨elemento, obstáculo⟩
ENTONCES
⟨velocidad, detenerse⟩ Y
⟨rotación, rotar-derecha⟩

SI
⟨ubicación, izq-adelante⟩⟨elemento, obstáculo⟩ Y
⟨ubicación, adelante⟩⟨elemento, obstáculo⟩ Y
⟨ubicación, der-adelante⟩⟨elemento, obstáculo⟩
ENTONCES
⟨velocidad, retroceder⟩ Y
⟨rotación, no-rotar⟩

SI
⟨ubicación, izq-atrás⟩⟨elemento, obstáculo⟩ Y
⟨ubicación, der-atrás⟩⟨elemento, obstáculo⟩ Y
⟨ubicación, atrás⟩⟨elemento, obstáculo⟩
ENTONCES
⟨velocidad, avanzar⟩ Y
⟨rotación, no-rotar⟩

Con el objetivo de dirigir el aprendizaje de reglas en cada *Comportamiento* propuesto, se definen las funciones de refuerzo respectivas:

⟨Evadir-Obstáculos⟩

$$R_{(EO)} = \begin{cases} -1, & \text{Si el robot colisiona} \\ -1 & \text{Si } sum_t < sum_{(t-1)},\ sum_t = \sum_{j=0}^{16} d_j \\ 1 & \text{de lo contrario} \end{cases} \tag{12.8}$$

donde: sum es la sumatoria de todas las distancias hasta los obstáculos circundantes al robot. d es la distancia desde el robot hasta un obstáculo detectado por el sensor j. t y $t-1$ son los instantes de tiempo actual y anterior respectivamente.

Intuitivamente, en la medida que el robot se aleje de los obstáculos con el transcurso del tiempo, recibirá más refuerzos positivos que negativos.

⟨Explorar-Entorno⟩:

$$R_{(EE)} = \begin{cases} -1, & \text{Si el robot se detiene} \\ 0.5 \cdot d + 0.5 \cdot \sum_{j=0}^{16} so & \text{de lo contrario} \end{cases} \tag{12.9}$$

donde: d es la distancia que el robot se ha desplazado desde el instante de tiempo anterior $(t-1)$ hasta el instante actual (t). so (sin obstáculos) es una condicional que valida si se ha identificado un obstáculo o no, esto es, $so = 1$ si el sensor no ha detectado un obstáculo (i.e., si la lectura corresponde a la distancia máxima de alcance del sonar: 5m), o $so = 0$ si la lectura es menor a dicha distancia máxima.

De esta manera, en la medida que el robot esté en constante movimiento (explorando), y alejado de los muros circundantes, recibirá más refuerzos positivos.

⟨Identificar-Caja⟩:

$$R_{(IC)} = \begin{cases} 1, & \text{Si el robot identifica correctamente el tipo de caja} \\ -1, & \text{de lo contrario} \end{cases} \tag{12.10}$$

Debido a que en el entorno pueden encontrarse dispersos múltiples tipos de objetos (cajas), el robot deberá aprender a identificarlos correctamente a través de un proceso de clasificación[9]:

[9]En este experimento, el robot sólo tendrá que identificar si el objeto detectado es un obstáculo o una caja (la única que se encuentra en el entorno). Sin embargo, en el experimento EX04 se describe con más detalle esta tarea de identificación.

⟨Recoger-Caja⟩:

$$R_{(RC)} = \begin{cases} 1, & \text{Si el robot recoge la caja exitosamente} \\ -1, & \text{Si el robot deja caer la caja} \\ -1 & \text{Si } pos_t > pos_{(t-1)},\ pos_t = 0.5 \cdot \frac{1}{d} + 0.5 \cdot \frac{1}{a} \\ 1 & \text{de lo contrario} \end{cases} \tag{12.11}$$

donde: d es la distancia entre la pinza del robot y la caja. a es el ángulo de orientación (en radianes) entre el centro del robot y el centro de la caja. t y $t-1$ son los instantes de tiempo actual y anterior respectivamente.

Intuitivamente, con el transcurso del tiempo, si el robot se alinea con el centro de la caja (reduciendo el ángulo de orientación) y reduce la distancia de aproximación, la magnitud del refuerzo positivo será más alta.

Por último, se menciona a continuación el procedimiento empleado para llevar a cabo el experimento:

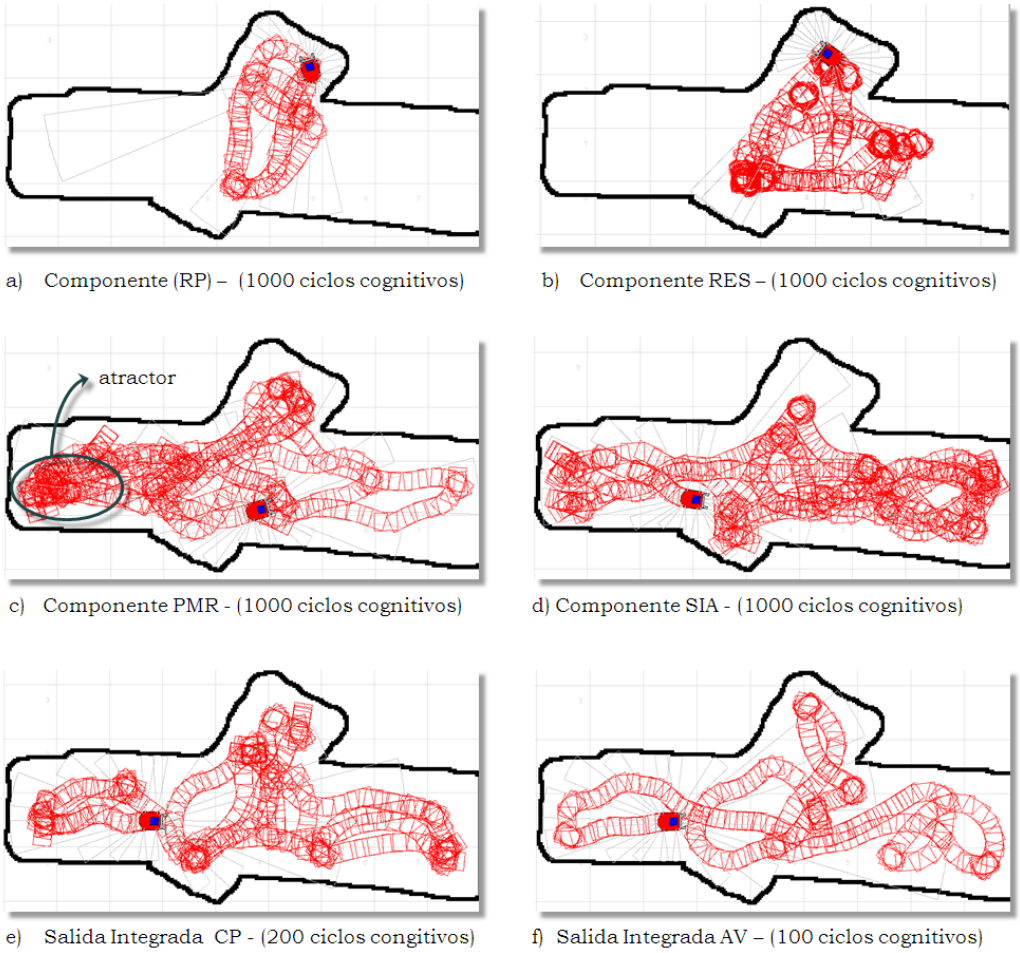
1. Se ejecutaron 50 épocas de aprendizaje por cada una de los dos escenarios.
2. Por cada época de aprendizaje se ejecutó un máximo de 200 ciclos cognitivos.
3. Al finalizar cada época, se cuantificó el número de refuerzos negativos obtenidos por cada componente (por separado), y los obtenidos por el *Comportamiento* en general (cuando se empleó el mecanismo de integración).
4. El experimento se ejecutó luego 50 veces empleando la misma configuración inicial para cada componente, y los resultados fueron promediados (a través de la media armónica).

Así pues, la función objetivo del experimento es que cada componente minimice incrementalmente el número de refuerzos negativos obtenidos en cada época hasta alcanzar un error mínimo del 5 %, esto significa, que por cada época de 1000 ciclos cognitivos se espera que las acciones de los componentes (por separado e integrados) no superen un máximo de 50 refuerzos negativos.

12.3.3.3. Análisis de Resultados

Análisis de Desempeño de un único Escenario: en esta parte del experimento se pretende analizar, utilizando el escenario 1, el desempeño de los *Comportamientos* Evadir Obstáculos y Explorar Entorno, para lo cual se empleará un análisis gráfico de las curvas de aprendizaje para cada *Comportamiento*. De esta manera se podrá observar gráficamente la velocidad de convergencia de cada componente por separado y de todos los componentes integrados.

En la figura 12.15 se presentan las diferentes rutas de navegación que describe el robot en el escenario 1, tanto para la activación individual de cada componente como para la activación de salidas integradas. De esta figura se puede señalar lo siguiente:



a) Componente (RP) – (1000 ciclos cognitivos)
b) Componente RES – (1000 ciclos cognitivos)
c) Componente PMR - (1000 ciclos cognitivos)
d) Componente SIA - (1000 ciclos cognitivos)
e) Salida Integrada CP - (200 ciclos congitivos)
f) Salida Integrada AV – (100 ciclos cognitivos)

Figura 12.15: Rutas de Navegación empleando diferentes configuraciones de aprendizaje. CP: Combinación de Pesos. AV: Agregación por Votación.

- Figura 12.15.a): se observa cómo el robot, restringido únicamente al uso de las cuatro reglas simbólicas del componente RP, exhibe un patrón de comportamiento monótono después de 1000 ciclos cognitivos. Esta monotonicidad depende en cierto grado del número de reglas predeterminadas que posee el componente RP (cuantas más reglas disponga el componente, menos monótono será su comportamiento), pero por otro lado, su capacidad restringida para adquirir nuevo conocimiento lo convierte en un componente más limitado. Por tanto, la principal fortaleza de este componente radica en su capacidad para razonar sobre un conocimiento experto previamente modelado, más no en su capacidad para aprender.

- Figura 12.15.b): el robot emplea el componente RES, el cual extrae un conjunto de reglas sub-simbólicas a partir del componente PMR que aprende "on-line". Nótese que, aunque el robot presenta ahora un patrón de comportamiento más explorativo en comparación con el componente RP, sigue describiendo una ruta de navegación monótona. Esto se debe a que el componente RES depende en gran medida de las pautas que aprende primero la red PMR, y que luego generaliza y especializa el

componente RES, causando que el proceso de aprendizaje se haga más lento. Sin embargo, cabe resaltar que la potencia del componente RES reside en su capacidad de convertir el conocimiento sub-simbólico en reglas simbólicas explicativas y comunicables.

- Figura 12.15.c): el robot emplea el componente PMR a través del cual describe un ruta de navegación mucho más explorativa. Aproximadamente durante los primeros 300 ciclos cognitivos, la red PMR tiende a caer en zonas de atracción (o atractores cíclicos), esto es, la red neuronal encuentra soluciones locales (como moverse en círculos sin avanzar, con lo cual evita colisionar contra los muros aunque incumpliendo con el propósito de explorar el entorno) las cuales luego son reemplazadas por soluciones globales (como bordear los muros circundantes sin colisionar y explorando a la vez el entorno).
- Figura 12.15.d): el robot emplea el componente SIA y describe una ruta de navegación más explorativa que evita en gran medida los atractores locales. En comparación con el componente PMR (que también terminó explorando todo el entorno después de los 1000 ciclos cognitivos), el SIA hace una exploración más uniforme. Así pues, es posible observar la capacidad de auto-organización de los componentes PMR y SIA, la cual les permite adaptarse a las condiciones del entorno en tiempo de ejecución.
- Figuras 12.15.e) y 12.15.f): es estas figuras se observa la ruta de navegación del robot cuando emplea los mecanismos de integración de acciones, durante los primeros 100 y 200 ciclos cognitivos. Se evidencia claramente que el proceso de aprendizaje de reglas converge más rápidamente cuando se emplea la salida integrada de los cuatro componentes en paralelo, provocando por un lado, que el entorno sea explorado por completo empleando muchos menos ciclos cognitivos (entre 100 y 200 ciclos) y, por otro, haciendo que el sistema sea más robusto y tolerante a fallos: el robot recibe menos refuerzos negativos, las soluciones encontradas son menos propensas a caer en óptimos locales, la ruta de exploración es más uniforme, las acciones incorrectas de un componente son compensadas por las acciones correctas de los demás componentes, etc. ...

En la gráfica 12.16 se presentan las curvas de aprendizaje para cada uno de los componentes por separado, así como las curvas de aprendizaje de las salidas integradas. Obsérvese que no se presenta la curva para el componente RP puesto que éste no emplea un mecanismo de aprendizaje que le permita reducir el número de refuerzos negativos incrementalmente. No obstante, este componente sí que participó en la conformación de las salidas integradas. De la gráfica se puede observar claramente que la curva de aprendizaje convergió más rápidamente para los casos donde se empleó un mecanismo de integración de acciones. Adicionalmente, en la tabla 12.14 se presenta un resumen de las estadísticas obtenidas en cada caso, donde queda demostrada la aceleración en la convergencia cuando se emplean dichos mecanismos de integración de acciones (i.e., los mecanismos CP y AV necesitaron sólo 12 y 13 épocas respectivamente, para alcanzar la convergencia, y redujeron asimismo hasta casi un 50 % del error cuadrático medio). Nótese por último que la aceleración en la convergencia de los mecanismos de integración CP y AV no presentan diferencias significativas entre ellos.

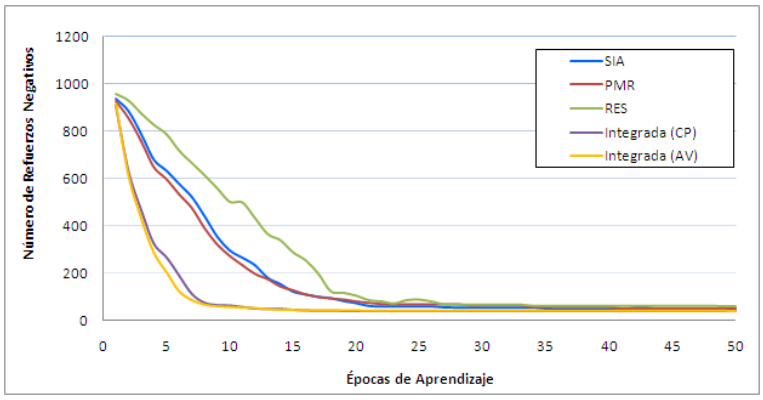


Figura 12.16: Curvas de aprendizaje para los componentes de ⟨`Evadir-Obstáculos`⟩. CP: Combinación de Pesos. AV: Agregación por Votación.

Componente	NEC	ECM
SIA	36	299.93
PMR	41	285.5
RES	50	368.29
Integración CP	12	186.11
Integración AV	13	180.66

Tabla 12.14: Estadísticas de Convergencia. NEC: Número de Épocas hasta alcanzar la convergencia (media armónica). ECM: Error Cuadrático Medio.

Análisis de Desempeño de múltiples Escenarios: en esta parte del experimento se analizan simultáneamente los resultados de desempeño de los cuatro *Comportamientos* (EO, EE, IC, y RC), para cada uno de sus componentes por separado e integrados[10]. Para esto, se emplea una prueba no paramétrica.

A pesar de que los resultados analizados cumplen con el criterio de Independencia, no cumplen con los criterios de normalidad y homocedasticidad como se puede observar en las tablas 12.15 y 12.16 respectivamente. Por tanto, se puede proceder con una prueba no paramétrica.

A continuación se aplica la prueba de Friedman para comprobar si existe una diferencia significativa en los resultados de desempeño de todos los *Comportamientos* analizados. Las tablas 12.17 y 12.18 resumen los resultados de la prueba.

La prueba de Friedman advierte la existencia de diferencias significativas entre los resultados observados en todos los *Comportamientos* a la vez, y por tanto, se rechaza la hipótesis nula. Se procede entonces con las pruebas a posteriori para realizar comparaciones múltiples.

[10]Recuérdese que el componente RP no se analiza por separado puesto que no define un proceso de aprendizaje.

Componente	EO	EE	IC	RC
SIA	*(0.3)	(0.6)	*(.03)	*(.00)
PMR	*(.04)	(0.6)	*(.04)	*(.01)
RES	(.07)	(0.7)	*(.04)	*(.03)
Integración CP	*(.01)	*(.03)	*(.02)	*(.00)
Integración AV	*(.02)	*(.02)	*(.02)	*(.01)

Tabla 12.15: Prueba de Normalidad Shapiro-Wilk. Las casillas con "*" no cumplen con la normalidad. Con $\alpha = 0.05$.

Componente	EO	EE	IC	RC
SIA	*(.01)	*(.04)	*(.00)	*(.01)
PMR	*(.02)	*(.01)	*(.02)	*(.00)
RES	(.06)	*(.03)	(.07)	*(.01)
Integración CP	*(.00)	*(.02)	*(.03)	*(.01)
Integración AV	*(.02)	*(.00)	*(.01)	*(.00)

Tabla 12.16: Prueba de Heterocedasticidad Levene. Las casillas con "*" indica que las varianzas no son homogéneas. Con $\alpha = 0.05$.

	Valor Friedman	Valor en X^2
Todos los comportamientos	*(18.07)	(6.32)

Tabla 12.17: Prueba de Friedman. La casilla con "*" indica el mayor valor.

Algoritmo	Ranking
SIA	7.23
PMR	7.18
RES	12.87
Integración CP	3.36
Integración AV	3.07

Tabla 12.18: Rankings de los algoritmos obtenidos mediante la prueba Friedman

Debido a que el mejor ranking obtenido por la prueba Friedman corresponde al método de integración por agregación de votos AV (3.07), se emplea éste como algoritmo de control y sus resultados de desempeño se compararán contra los resultados de los

tres componentes por separado[11]. La Tabla 12.19 presenta la estimación de los valores p_i correspondientes a cada hipótesis (AV vs. SIA, AV vs. PMR, y AV vs. RES), junto con los valores ajustados por cada procedimiento de comparaciones múltiples utilizado (Bonferroni-Dunn y Hochberg), en orden incremental con respecto al valor de p_i.

i	AV vs.	dc	p	Bonferroni-Dunn	Hochberg
1	RES	5.653	0.0017	0.01	0.0015
2	PMR	2.029	0.0089	0.01	0.0080
3	SIA	1.983	0.0097	0.01	0.0092

Tabla 12.19: Pruebas Posteriori (considerando $\alpha = 0.05$)

A partir de los resultados presentados en la tabla 12.19 se puede señalar lo siguiente:

- La prueba Bonferroni-Dunn rechaza todas hipótesis con $p < 0.01$, es decir, se rechazan las 3 hipótesis, con lo cual se deduce que el desempeño de los *Comportamientos* es mejorado significativamente cuando se emplea el mecanismo de integración AV.
- La prueba Hochberg rechaza las hipótesis con $p < \alpha(0.05)$, esto es, se rechazan las tres hipótesis, lo cual indica que el mecanismo de integración AV tiene un desempeño significativamente mejor en comparación con cada uno de los tres componentes por separado.

Análisis del grado de Asimilación y Consolidación de Conocimiento: en esta parte del experimento se analiza el grado de asimilación y consolidación de nuevo conocimiento a través de los procesos de *Rectificación Ascendente* (sub-simbólico $\rightarrow$ simbólico) y *Orientación Descendente* (simbólico $\rightarrow$ sub-simbólico) descritos en la sección 7.7.

En la tabla 12.20 se presentan las gráficas de variación en el tiempo de los factores de peso para cada componente, de acuerdo con el mecanismo de integración empleado (*Combinación de Pesos* y *Agregación por Votación*)[12] para el *Comportamiento* ⟨`Evadir-Obstáculos`⟩. Recuérdese que cada componente del *Comportamiento* influye sobre la salida integrada de forma diferente de acuerdo con un factor de peso, el cual varía progresivamente dependiendo de los niveles de positividad y negatividad acumulados por cada uno de dichos componentes.

De la tabla 12.20 merece la pena resaltar algunos puntos. Por un lado, obsérvese que el comportamiento del robot en las épocas iniciales de aprendizaje es dirigido por el conocimiento experto almacenado en el componente RP, mientras que progresivamente, los

[11] Obsérvese que no existe una diferencia significativa de rankings entre los métodos de integración CP y AV. Por tanto, si se empleara el método CP como algoritmo de control, las pruebas a posteriori arrojarían resultados muy similares.

[12] A pesar de que estas gráficas se presentan por separado, por motivos de claridad, corresponden a la variación simultánea de los factores de peso de cada componente registrados durante 30 épocas de aprendizaje.

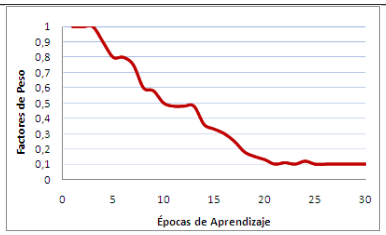
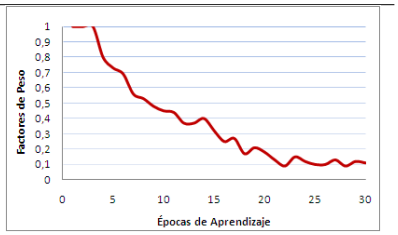
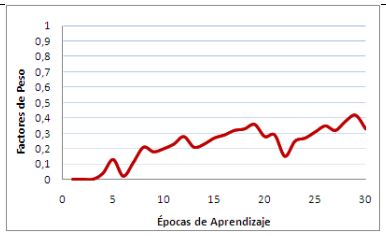
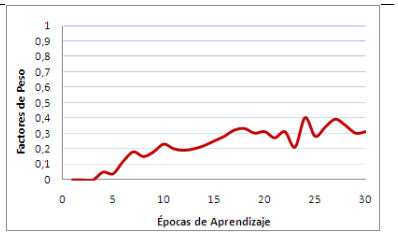
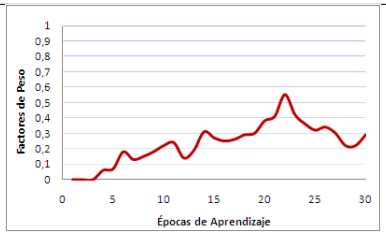
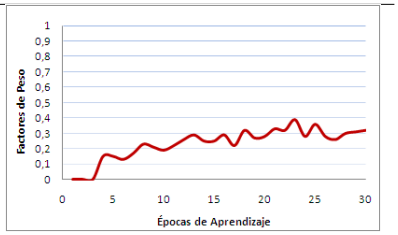
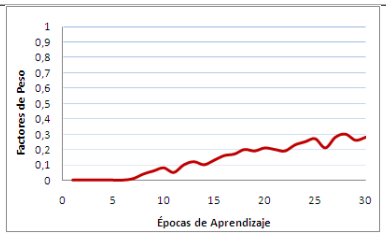
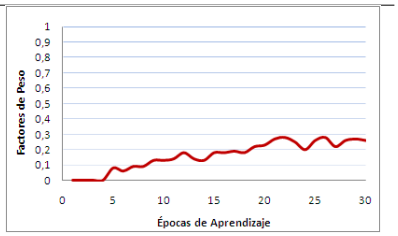
Comp.	Combinación de Pesos	Agregación por Votación
RP		
SIA		
PMR		
RES		

Tabla 12.20: Variación de los Factores de Peso durante el aprendizaje (empleando la media armónica) para el comportamiento ⟨`Evadir-Obstáculos`⟩.

componentes PMR y SIA construyen nuevas pautas sub-simbólicas a través del aprendizaje por refuerzo. Aproximadamente, entre las épocas de aprendizaje 1 y 5, los componentes PMR y SIA asimilan las reglas simbólicas almacenadas en RP a través del proceso de Orientación Descendente (Asimilación); y desde más o menos la época 5 en adelante, los componentes emergentes PMR y SIA descubren nuevas pautas cognitivas que son consolidadas paulatinamente en el componente cognitivista RES a través de un proceso de Rectificación Ascendente. El proceso de consolidación en RES requiere de más tiempo que el aprendizaje implícito llevado a cabo por los componentes emergentes, puesto que durante la consolidación se desempeñan además los procesos de especialización y generalización de las reglas sub-simbólicas extraídas. Por último, es importante resaltar que aunque los componentes PMR, SIA y RES describen una tendencia de crecimiento progresivo, sus curvas presentan cierta fluctuación debido a la continua "competencia" por hacer dominante su acción recomendada en cada ciclo cognitivo.

Por otro lado, en la tabla 12.21 se presenta un estadístico del número de reglas nuevas (diferentes de las reglas iniciales de RP) aprendidas por los componentes SIA, PMR, y RES después de 50 épocas de aprendizaje.

Componente	EO	EE	IC	RC
SIA	26	5	6	16
PMR	21	7	5	18
RES	12	3	1	5

Tabla 12.21: Nuevas pautas cognitivas construidas por cada componente después del aprendizaje (media armónica aproximada al entero más cercano).

A continuación, se presentan algunas de las reglas extraídas desde nivel emergente por el componente RES para el comportamiento ⟨Evadir-Obstáculos⟩. Por motivos de simplificación, solamente se exponen aquellas condiciones de la regla que detectan un obstáculo a través de un sensor específico, con lo cual debe hacerse el supuesto que los demás sensores no identifican ningún obstáculo.

Regla Generalizada:

SI
⟨ubicación, adelante⟩⟨elemento, obstáculo⟩
ENTONCES
⟨velocidad, aproximarse-lento⟩ Y
⟨rotación, rotar-izquierda⟩

Regla Generalizada:

SI
⟨ubicación, atrás⟩⟨elemento, obstáculo⟩
ENTONCES
⟨velocidad, avanzar⟩ Y
⟨rotación, no-rotar⟩

Regla Especializada:

SI
⟨ubicación, adelante⟩⟨elemento, obstáculo⟩ Y
⟨ubicación, derecha⟩⟨elemento, obstáculo⟩ Y
⟨ubicación, atrás⟩⟨elemento, obstáculo⟩
ENTONCES
⟨velocidad, detenerse⟩ Y
⟨rotación, rotar-izquierda⟩

Regla Especializada:

SI
⟨ubicación, izquierda⟩⟨elemento, obstáculo⟩ Y
⟨ubicación, adelante⟩⟨elemento, obstáculo⟩ Y
⟨ubicación, atrás⟩⟨elemento, obstáculo⟩
ENTONCES
⟨velocidad, detenerse⟩ Y
⟨rotación, rotar-derecha⟩

Regla Especializada:

SI
⟨ubicación, izquierda⟩⟨elemento, obstáculo⟩ Y
⟨ubicación, derecha⟩⟨elemento, obstáculo⟩ Y
⟨ubicación, atrás⟩⟨elemento, obstáculo⟩
ENTONCES
⟨velocidad, avanzar⟩ Y
⟨rotación, no-rotar⟩

Análisis de Tolerancia a Fallos: en esta parte del experimento se pone a prueba la robustez del *Comportamiento* en situaciones donde uno de sus componentes falla. Para llevar a cabo este experimento se configura inicialmente el componente RP del *Comportamiento* ⟨Evadir-Obstáculos⟩ con diez reglas de producción erróneas (e.g., avanzar cuando tiene un obstáculo en frente, girar a la derecha cuando hay un obstáculo en la misma dirección, etc.). Adicionalmente, los componentes del *Comportamiento* son inicializados con los siguientes factores de peso: $FP_{(RP)} = 1.0$, $FP_{(SIA)} = 0.0$, $FP_{(PMR)} = 0.0$, y $FP_{(RES)} = 0.0$; y se utiliza el mecanismo de integración por Combinación de Pesos.

En la gráfica 12.17 se observa claramente que, con el transcurso de las épocas de aprendizaje, el *Comportamiento* varía adaptativamente los factores de peso de sus componentes. Debido a que el componente RP no posee un mecanismo de aprendizaje que le permita corregir las reglas erróneas configuradas inicialmente, su factor de peso disminuye rápidamente mientras que el resto de los componentes aumentan dicho factor progresivamente. Después de la época de aprendizaje no. 9 se observa que el componente RP tiene una participación casi nula en el proceso de toma de decisiones llevado a cabo por el *Comportamiento*, quien delega tal responsabilidad en los otros tres componentes ajustados por el proceso de aprendizaje.

Análisis de los Tiempos de Respuesta: en esta parte del experimento se comparan los TR de cada uno de los componentes del *Comportamiento* contra el TR de la salida integrada. Concretamente para este análisis se presentan los resultados del *Comportamiento* ⟨Evadir-Obstáculos⟩, no obstante, el resto de *Comportamientos* obtuvieron resultados muy similares los cuales no se presentan en este apartado por motivos de simplifi-

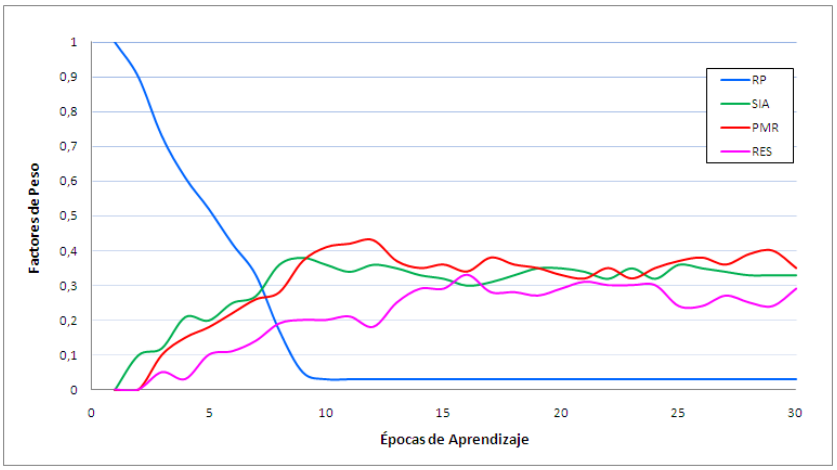


Figura 12.17: Progresión de los Factores de Peso de cada componente en la prueba de Tolerancia a Fallos.

cación. En la tabla 12.22 se presenta la configuración de hardware y del sistema operativo empleado en el desarrollo de los experimentos:

Característica	RC
Procesador (CPU)	Intel Pentium 4
Velocidad	2.60 Ghz
Cores	2
Memoria RAM	1.5 Gb
S.O.	Linux 2.622.17-01- default i686
Sistema	openSUSE 10.3 (i586)

Tabla 12.22: Configuración de Hardware y S.O.

En la gráfica 12.18 se presentan los TR para cada uno de los componentes del *Comportamiento*, así como los TR de la salida integrada. Nótese que los TR más bajos son para los componentes RP y PMR, los cuales se encuentran en el intervalo de los 11 ± 2 milisegundos. La razón para que estos dos componentes empleen menos tiempo de procesamiento se debe, por un lado, a que en el componente RP se definieron únicamente cuatro reglas de producción, con lo cual el proceso de inferencia se redujo altamente y, por otro, a que la red PMR posee una topología con un número reducido de neuronas, con lo cual la actualización de los pesos sinápticos en cada capa se lleva a cabo más rápidamente.

Seguidamente y en orden ascendente, se encuentra el componente SIA con un TR en el intervalo de los 15 ± 3 milisegundos. No obstante, obsérvese que cada 10 épocas de aprendizaje se producen unos picos en la curva, los cuales se deben principalmente a la activación de tres procesos del algoritmo: el método de diversificación, el método de

recorte, y las meta-dinámicas. Estos métodos requieren de una mayor cantidad de tiempo de procesamiento en comparación con el resto de procesos que se ejecutan siempre en todos los ciclos cognitivos (la selección clonal, la maduración clonal, la distribución de recompensas, etc.).

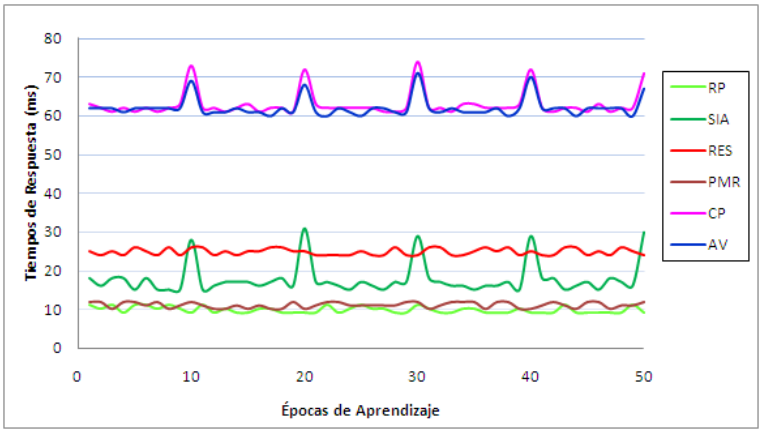


Figura 12.18: Tiempos de Respuesta para cada componente por separado e integrado (se emplean las medias armónicas).

En cuanto al componente RES, se observa un TR en el intervalo de tiempo de los 24 ± 3 milisegundos. Este componente emplea un mayor tiempo de procesamiento debido a la activación de los mecanismos de extracción, inferencia, especialización y generalización de reglas en cada ciclo cognitivo. Por otro lado, respecto a las salidas integradas del *Comportamiento* se observa un TR en el intervalo de 62 ± 3 milisegundos, lo cual se corresponde aproximadamente con una relación aditiva de los TR de cada componente implicado.

En la gráfica 12.19 se presenta un comparativo entre los TR de la salida integrada cuando se emplea el simulador *player/stage* vs. los TR de la salida integrada cuando se utilizan los robots reales (Pioneer). Concretamente, en los TR de los robots reales se evidencia un aumento aproximado entre el 30 - 34 % de los TR obtenidos en la simulación. Esto se debe principalmente a que durante la simulación no se considera el TR relativo al tiempo que toma la ejecución motriz de las acciones, esto es, la comunicación con cada uno de los dispositivos (servomotores de velocidad, rotación, etc.). La tabla 12.23 muestra un resumen de las estadísticas obtenidas para cada caso.

Finalmente, dados los resultados obtenidos se puede argumentar que los TR (tanto para la simulación como para los robots reales) se encuentran dentro de los límites estimados en la sección 10.5.1. No obstante, cabe resaltar que estos resultados pueden variar ligeramente dependiendo de la configuración de hardware empleada en la simulación, o incluso, de la configuración y calibración de los dispositivos de los robots reales.

Conclusiones del Experimento: a partir de los resultados presentados en este experimento, se concluye lo siguiente:

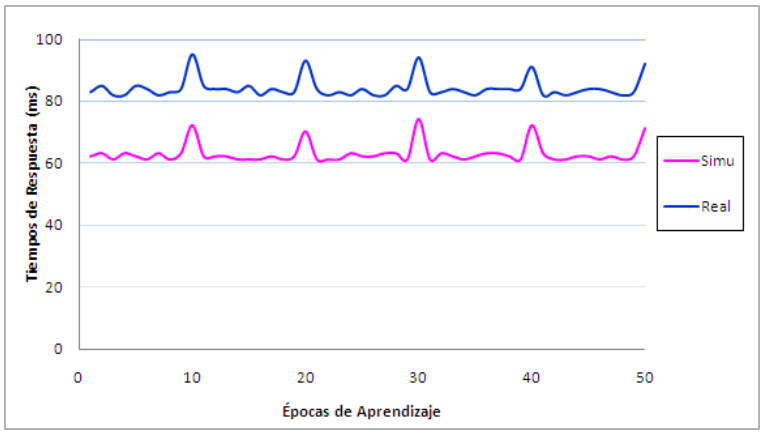


Figura 12.19: Tiempos de Respuesta de la salida integrada: Simulación vs. Robots Reales. (se emplean las medias armónicas)

Estadística	CP	AV	Robot Real
Mínimo (ms)	61.4	60.1	82.7
Máximo (ms)	73.2	72.9	92.4
Media ($\bar{x}$) (ms)	62.77	61.75	84.13
Desviación Estándar (σ)	2.93	2.96	2.72
Varianza muestral (σ^2)	8.62	8.80	7.39

Tabla 12.23: Estadísticas para los Tiempos de Respuesta - EX03

- El enfoque híbrido propuesto mejora significativamente el desempeño global de los *Comportamientos* haciendo que converjan de manera más eficiente, esto es, que el aprendizaje converge más rápidamente hacia un óptimo global evitando la convergencia prematura en óptimos locales.
- Se demuestra que el mecanismo de integración permite a los *Comportamientos* ser más tolerantes a fallos, compensando las acciones equivocadas de un componente con las acciones correctas de los demás. Esto es, bien sea que se emplee un método de agregación por votación o de combinación de pesos, con el transcurso del tiempo, las salidas de los componentes tienden a sincronizarse, con lo cual si durante el proceso de selección de acciones un componente determinado envía una acción incorrecta, los demás componentes mitigarán los efectos de dicha acción recomendando una acción correcta y unificada que ganará por mayoría (de votos o de activación acumulada).
- Los mecanismos de integración de salidas son procesos *"adaptativos"* los cuales ajustan, en tiempo de ejecución, los factores de peso que determinan la relevancia que tienen las acciones recomendadas por cada uno de los componentes, en función de cuan eficientes han sido estos durante la interacción con el ambiente. Esta variación en los factores de peso produce distintos modos operacionales en el agente,

los cuales se corresponden en líneas generales con las nociones psicológicas acerca de los modos de conducta intuitivos (reactivos), modos de conducta deliberativos, y modos de conducta combinados (teleoreactivos).

- Se demuestra a través de la interacción entre los procesos de Rectificación Ascendente y Orientación Descendente, que las reglas sub-simbólicas aprendidas por el nivel emergente son consolidadas luego en el nivel cognitivista, y que las reglas simbólicas refinadas por el nivel cognitivista son asimiladas después por el nivel emergente, proceso éste que se mantiene en un ciclo continuo mientras que el agente interactúe con su entorno.
- A partir de los resultados de los TRs, y según lo discutido en el capitulo 7, se demuestra que los *Comportamientos* pueden procesar la información de forma tanto reactiva como deliberativa. Si se desea que el *Comportamiento* exhiba una conducta puramente reactiva, se activan únicamente los componentes de procesamiento inconsciente, esto es, los del nivel emergente, salvo que el SIA debe configurarse en el modo de *estímulo/respuesta* y no en el modo de red inmune inferencial (sección 7.4.3.2). Por otro lado, si desea que el *Comportamiento* exhiba una conducta puramente deliberativa se deben activar los componentes del nivel cognitivista (o de procesamiento consciente) junto con el SIA en el modo de red inmune inferencial, o bien con todos los componentes activos y empleando alguno de los mecanismos de integración de acciones.
- Se concluye además que los TR permanecen dentro de intervalos de tiempo apropiados, tomando como referencia los TR sugeridos por las ciencias cognitivas como se discutió en la sección 10.5.1.
- Por todo lo anterior, se puede argumentar que el ensamblaje de múltiples componentes de naturaleza tanto cognitivista como emergente, produce un comportamiento integrado y robusto cuya eficiencia de procesamiento resulta ser algo más que la simple suma de sus partes.

12.3.4. Experimento EX04: Clasificación y Generalización de Conocimiento en Entornos Cambiantes

12.3.4.1. Descripción

Este experimento tiene como objetivo analizar el grado de *"plasticidad"* de un *Comportamiento* para re-estructurar sus pautas cognitivas cuando se presentan cambios ambientales y señales de ruido en los datos. Por tanto, esta plasticidad se medirá en términos de la capacidad del *Comportamiento* para clasificar adaptativamente los datos y generalizar el conocimiento adquirido.

Para llevar a cabo este experimento, se emplearon los mismos *Comportamientos* descritos para el experimento EX03, con la única diferencia de que el *Comportamiento* ⟨`Identificar-Caja`⟩ es extendido con la capacidad de identificar diferentes tipos

de objetos y no solamente cajas rectangulares (por lo cual se denominará de ahora en adelante ⟨`Identificar-Objeto`⟩) de acuerdo con las siguientes características[13]:

1. *Número de puntos (NP)*: el ambiente es diseñado con tres grupos de objetos que están categorizados según el número de puntos. Los tres grupos son de cuatro, ocho y doce puntos respectivamente. Se utilizan dos bits para la codificación subsimbólica, tal que:
   - Grupo de objetos con cuatro puntos: 00
   - Grupo de objetos con ocho puntos: 01
   - Grupo de objetos con doce puntos: 10
2. *Número de ángulos rectos (NAR)*: corresponde al número total de ángulos rectos internos de la geometría identificada del objeto. Se utilizan dos bits codificados así:
   - Sin ángulos rectos: 00
   - Entre uno y cuatro ángulos rectos: 01
   - Entre cinco y ocho ángulos rectos: 10
   - Entre nueve y doce ángulos rectos: 11
3. *Número de ángulos agudos (NAA)*: se utilizan dos bits codificados de igual forma que para el número de ángulos rectos.
4. *Número de ángulos obtusos (NAO)*: se utilizan dos bits codificados de igual forma que para el número de ángulos rectos y agudos
5. *Cantidad máxima de ángulos rectos continuos (MAR)*: codifica la cantidad máxima de ángulos rectos continuos que constituyen la geometría del objeto. Puesto que en el ambiente simulado se encuentran objetos hasta con doce puntos, en teoría, podría haber objetos hasta con doce ángulos rectos continuos, por lo que se utilizan cuatro bits para codificar este número.
6. *Cantidad máxima de ángulos agudos continuos (MAA)*: igual que el anterior, se utilizan cuatro bits.
7. *Cantidad máxima de ángulos obtusos continuos (MAO)*: igual que los dos anteriores, se utilizan cuatro bits.

Para la simulación robótica se han diseñado diez tipos diferentes de objetos que el robot tendrá que identificar y clasificar. Por consiguiente, el nivel cognitivista recibirá siete entradas o perceptos simbólicos (uno por cada característica identificada del objeto), mientras que el nivel emergente recibirá veinte entradas sub-simbólicas (que se corresponden con la suma de bits para cada característica). En cuanto a las salidas, el robot deberá aprender a clasificar cada uno de los diez objetos presentados, por tanto, las salidas del nivel cognitivista corresponden a diez, y para el nivel emergente corresponden a cuatro bits.

[13] Téngase en cuenta que para estimar los valores descriptivos de cada característica del objeto a partir de las lecturas de los sensores del láser y/o del sonar, se requiere de un pre-procesamiento perceptual, soportado por un análisis trigonométrico y de geometría cartesiana.

El propósito del experimento consiste entonces en presentar primero los diez objetos originales al robot, los cuales debe aprender a clasificar, y luego, introducir ciertas distorsiones en la geometría de dichos objetos para comprobar si el robot los identifica y clasifica correctamente a pesar del ruido introducido.

12.3.4.2. Diseño de los Escenarios

Para llevar a cabo este experimento, se emplea la interfaz gráfica del simulador robótico propuesto en (Romero y Niño, 2006), puesto que el proxy Fiducial de la plataforma *player/stage* no soporta la asignación de múltiples marcadores a un mismo objeto (e.g., un marcador por cada vértice). Así pues, se propone el escenario inicial (original) de la figura 12.20 y los escenarios modificados de las figuras 12.21 y 12.22. Como puede observarse en las figuras, cada escenario dispone de un conjunto de almacenes donde se simula el almacenaje de los objetos de acuerdo con su clasificación.

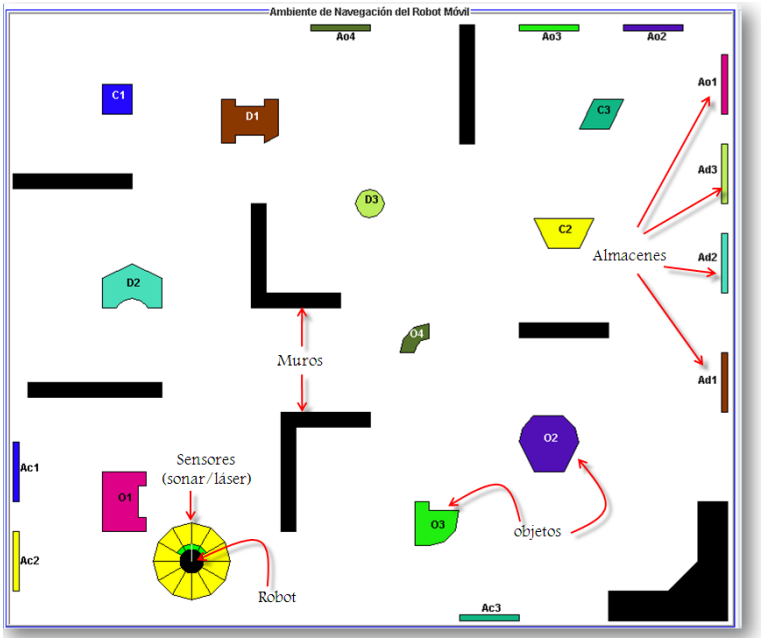


Figura 12.20: Escenario original para la identificación de objetos.

Los escenarios modificados presentan una estructura semejante al escenario original, por cuanto se encuentran los mismos obstáculos y la ubicación de los almacenes permanece constante, sin embargo, para centrarse en el análisis de las habilidades de generalización del conocimiento y la eficiencia de la memoria asociativa del *Comportamiento* ⟨`Identificar-Objeto`⟩, fueron alteradas las geometrías de los objetos originales, de modo que las lecturas de detección cambian de acuerdo con el escenario. Como se puede observar, mientras que el escenario 1 incorpora distorsiones moderadas en los objetos, el escenario 2 incorpora distorsiones más drásticas que se pretende sean generalizadas igualmente. El experimento desarrollado para este análisis, se llevó a cabo de la siguiente manera:

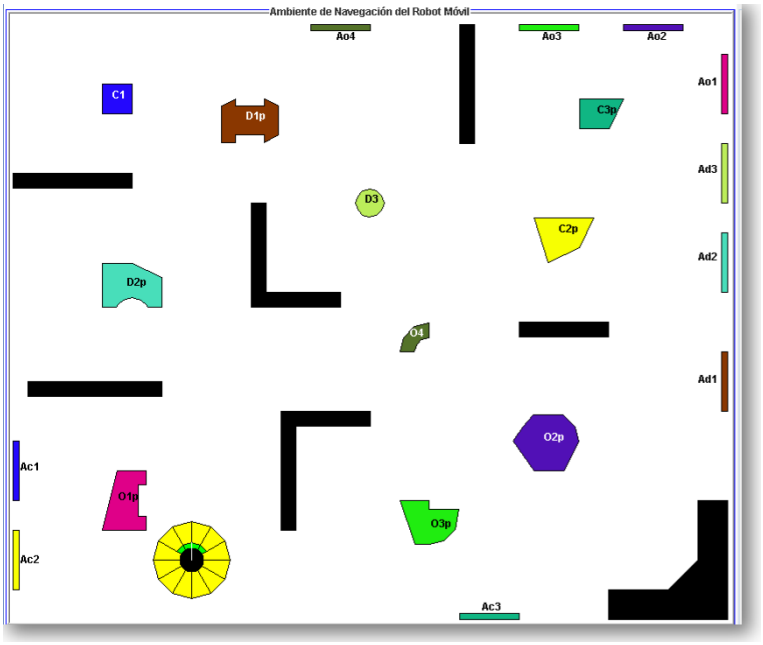


Figura 12.21: Escenario Modificado 1 para la identificación de objetos. Se observa que las geometrías de los objetos han sufrido distorsiones moderadas con respecto a las geometrías originales.

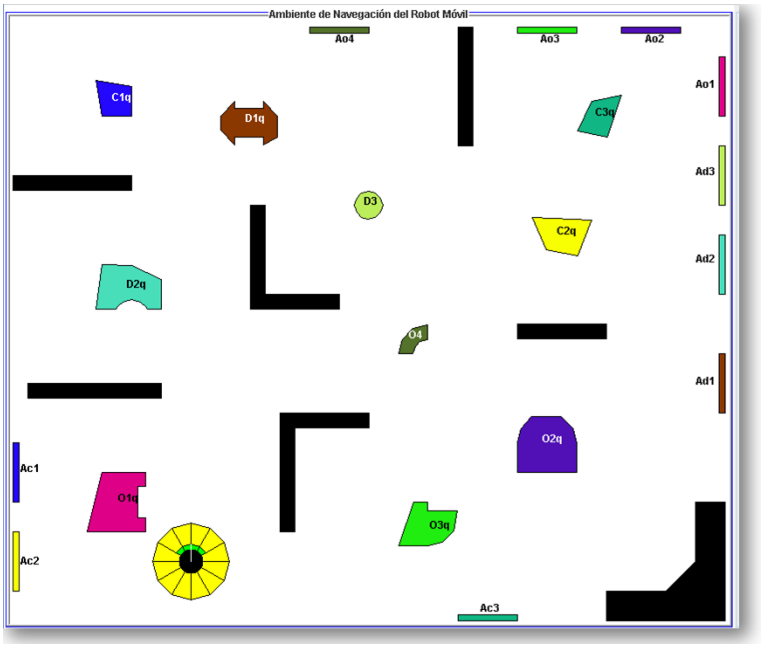


Figura 12.22: Escenario Modificado 2 para la identificación de objetos. Se observa que las geometrías de los objetos han sufrido distorsiones drásticas con respecto a las geometrías originales.

1. El robot realiza el proceso normal de exploración del ambiente y de detección de los perímetros de los diez objetos del escenario original.
2. Por cada uno de los objetos identificados, el robot clasifica el objeto y determina en cuál almacén debe ser depositado. Si la clasificación es correcta, el robot recibe un refuerzo positivo, de lo contrario, recibe un refuerzo negativo. El experimento se repite el número de veces que sean necesarias hasta que el agente aprenda a clasificar todos los objetos sin recibir ningún refuerzo negativo.
3. Una vez el robot posee el conocimiento que le permite clasificar los diez objetos del escenario original, se sitúa en el escenario modificado 1.
4. El robot navega nuevamente por el escenario, a fin de explorar y detectar los nuevos contornos de los objetos distorsionados.
5. El robot identifica y clasifica el objeto observado, y se determina a cuál almacén debe ser llevado.
6. El mismo procedimiento anterior se lleva a cabo para el escenario modificado 2.

12.3.4.3. Análisis de Resultados

Para este experimento se emplea un análisis de las curvas de aprendizaje descritas por el robot en cada escenario.

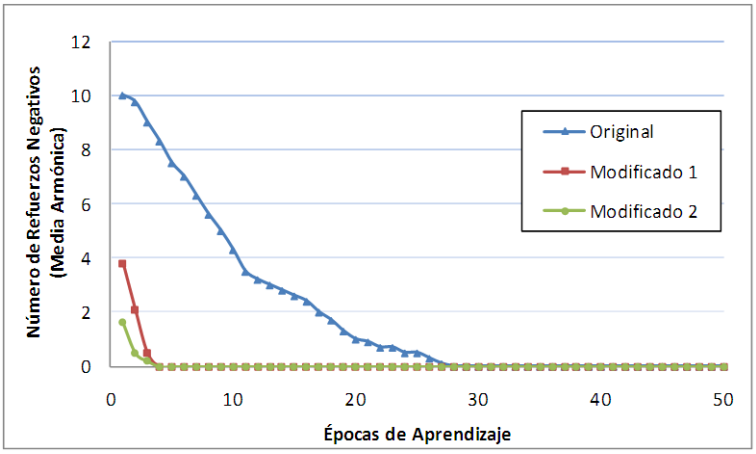


Figura 12.23: Curvas de aprendizaje para los escenarios original y modificados.

En la gráfica 12.23 se observa, para el escenario original, que el *Comportamiento* requiere de aproximadamente 28 épocas de aprendizaje (en promedio) para converger hacia un estado estable donde deja de recibir refuerzos negativos por sus acciones. Por otro lado, cuando el robot es situado en el escenario modificado 1, la capacidad de generalización es muy buena, logrando que el agente haga una re-identificación de los objetos muy rápida (empleando para ello menos de cuatro épocas de aprendizaje en promedio). En cuanto al escenario modificado 2, se requiere nuevamente de un proceso de re-aprendizaje, pero esta vez se realiza mucho más rápido, casi de manera instantánea (requiriéndose más o menos una época de aprendizaje en promedio), a pesar de que el

nivel de ruido causado por las distorsiones en las geometrías de los objetos es mucho más alto. De esta manera se deduce que el número de refuerzos negativos es reducido en un 85 % para el escenario modificado 1, y en un 92 % para el escenario modificado 2. Esta velocidad en la convergencia en los escenarios modificados se debe principalmente a la plasticidad cognitiva del agente para reutilizar y generalizar el conocimiento aprendido en las experiencias pasadas (en los escenarios original y modificado 1).

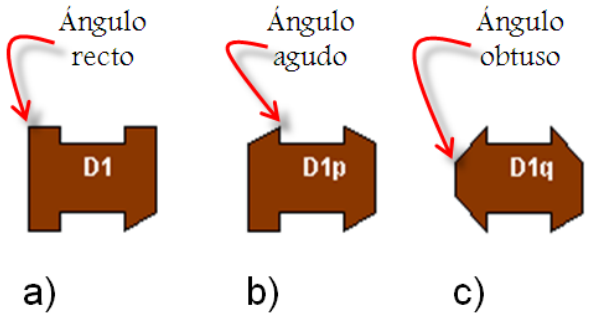


Figura 12.24: Objeto de doce vértices. a) objeto original. b) objeto distorsionado del escenario modificado 1. c) objeto distorsionado del escenario modificado 2.

A continuación, se analiza un caso particular de identificación y generalización de una pauta cognitiva a través de los tres escenarios de aprendizaje. Para esto, considérese la figura 12.24 que presenta las geometrías original y distorsionadas para el objeto "D1" (cuya categoría se codifica con el número 4, y en binario como 100, lo cual corresponde a la acción clasificatoria que debe tomar el *Comportamiento* para dicho objeto). En la tabla 12.24 se puede apreciar la regla de clasificación con mayor aptitud identificada por el *Comportamiento* en cada escenario, y se puede observar además, el proceso de adaptación y generalización que sufre dicha regla en la medida que se introducen cambios y ruido en la información percibida.

Escenario	Condición							Acción	Tipo
	NP	NAR	NAA	NAO	MAR	MAA	MAO		
Original	12	9-12	1-4	1-4	10	1	1	D1	simbólico
	10	11	01	01	1010	0001	0001	100	sub-simb.
Modificado 1	12	5-12	1-4	1-4	pares	1	1	D1	simbólico
	10	1#	01	01	###0	0001	0001	100	sub-simb.
Modificado 2	12	0-12	1-4	1-4	pares	1	0-3	D1	simbólico
	10	##	01	01	###0	0001	00##	100	sub-simb.

Tabla 12.24: Reglas de clasificación aprendidas para el Objeto D1. NP: Número de Puntos. NAR: Número de Ángulos Rectos. NAA: Número de Ángulos Agudos. NAO: Número de Ángulos Obtusos. MAR: Cantidad Máxima de Ángulos Rectos. MAA: Cantidad Máxima de Ángulos Agudos. MAO: Cantidad Máxima de Ángulos Obtusos.

Así pues, la regla generalizada que se obtiene en el escenario modificado 2 después de las fases de aprendizaje incremental, indica que cualquier objeto con las siguientes

características debe ser clasificado como un objeto de categoría "D1": un objeto de doce puntos o vértices; que tenga entre cero y doce ángulos rectos; con cuatro ángulos agudos o menos; con cuatro ángulos obtusos o menos; con un número "par" máximo de ángulos rectos consecutivos (0, 2, 4, ... 12); con máximo un ángulo agudo consecutivo; y con no más de tres ángulos obtusos consecutivos.

Conclusiones del Experimento: el anterior experimento revela que el agente, empleando su *Comportamiento* para identificar objetos, es capaz de adaptarse eficientemente a las condiciones dinámicas del ambiente; además, es robusto ante los cambios producidos en las pautas cognitivas, consiguiendo asociar características aprendidas previamente con situaciones similares aunque distorsionadas. Se demuestra por tanto, el nivel de plasticidad cognitiva de los *Comportamientos* para auto-organizar y generalizar las pautas cognitivas previamente aprendidas, y emplearlas en situaciones donde la información sensorial es imprecisa o contiene un nivel de ruido considerable.

12.4. Experimentos para el Nivel Ontogenético

En esta sección se describen los diferentes experimentos realizados y su correspondiente análisis de resultados para el Nivel Ontogenético de la arquitectura. En la figura 12.25 se presentan los módulos de METÁFORA que estarán activados a fin de llevar a cabo los experimentos. Obsérvese que el nivel ontogenético, a diferencia del epigenético, incluye los módulos de Orquestación de la Consciencia, Intencional, y la Memoria de Trabajo, a través de los cuales coordina los procesos atencionales y articula la activación de los *Comportamientos*. Por tanto, esta nueva configuración de los componentes de la arquitectura le permitirá al agente exhibir conductas tanto reactivas (procesamiento inconsciente) como deliberativas (procesamiento consciente).

Es importante resaltar que los experimentos realizados en esta sección se orientan únicamente a evaluar la adaptabilidad del proceso ontogenético, con lo cual debe asumirse que los *Comportamientos*, las memorias declarativas, y los componentes del sistema anticipatorio han adquirido, aprendido, consolidado y refinado previamente su conocimiento sobre el entorno, a través del proceso epigenético.

Por último, cabe mencionar que en el análisis de resultados de los experimentos de esta sección no se emplearán pruebas no paramétricas, como sí se hizo en los experimentos de la sección anterior. Esto se debe a que no se dispone de la implementación de algoritmos equivalentes al proceso ontogenético propuesto contra los cuales elaborar comparaciones multi-escenario. Por tanto, el análisis de resultados se enfocará únicamente a la evaluación de la convergencia evolutiva del proceso ontogenético, en cuanto al nivel de adaptabilidad de los diseños de las Redes de Comportamientos (aspecto funcional y estructural), y de los Planes deliberativos.

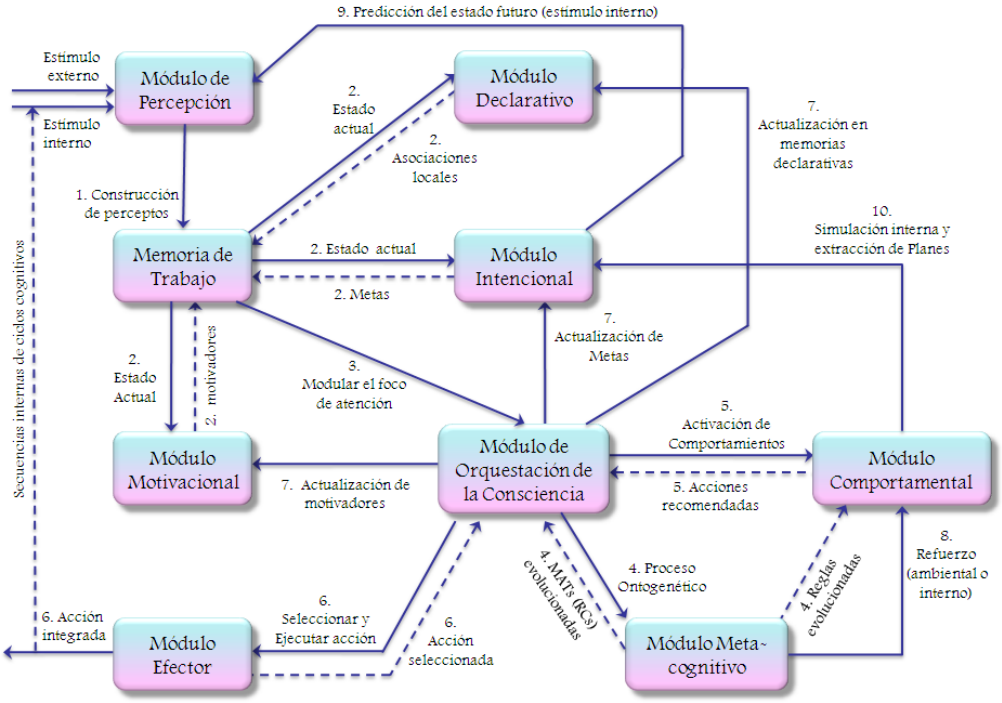


Figura 12.25: Módulos de METÁFORA activos durante los experimentos del Nivel Ontogenético.

12.4.1. Experimento EX05: Análisis de Adaptabilidad del Aspecto Estructural de las RCs

12.4.1.1. Descripción

Este experimento tiene como propósito evaluar el nivel de adaptabilidad del proceso ontogenético en la evolución de las estructuras de las RCs, dadas unas metas específicas. Para llevar a cabo este experimento, el robot tendrá que explorar su entorno en busca de un conjunto de cajas que se encuentran esparcidas por doquier, y una vez halladas éstas deberá llevarlas al almacén. Paralelamente, el robot deberá evadir los obstáculos que se encuentre en su camino y recargar su batería interna cuando el nivel de energía esté por debajo de cierto umbral. Por tanto, se proponen los siguientes *Comportamientos* y metas:

Comportamientos:	⟨Buscar-Almacén⟩ ⟨Buscar-Caja⟩ ⟨Recoger-Caja⟩ ⟨Almacenar-Caja⟩ ⟨Evadir-Obstáculos⟩ ⟨Buscar-Recargador-Batería⟩ ⟨Recargar-Batería⟩
Metas Globales:	⟨Almacenar-Todas-Cajas⟩
Metas Específicas:	⟨Mantener-Nivel-energía⟩

Inicialmente, ninguno de los *Comportamientos* define elementos en sus listas de efectos deseados ni de efectos no deseados (i.e., relaciones de excitación o inhibición entre *Comportamientos*), por lo que el propósito de este experimento es que estas relaciones sean aprendidas durante el proceso evolutivo. En cuanto a la meta específica ⟨`Mantener-Nivel-Energía`⟩, es una meta recurrente que es activada cuando el motivador ⟨`Nivel-Energía`⟩ se encuentra en las zonas de infra o sobre-estimulación.

Parámetro	Valor	Parámetro	Valor
Número de Experimentos	30	Prob. de Recombinación en un Punto (p_{rup})	0.3
Número de Épocas Evolutivas	500	Prob. de Recombinación en dos Puntos (p_{rdp})	0.3
Población de RCs	50	Prob. de Transposición de SI (p_{si})	0.1
Núm. de Genes Estructurales	7	Umbral de activación (θ)	100
Longitud cabecera gen	10	Activación promedio (π)	90
Longitud terminación gen	20	Activación desde el estado (ϕ)	90
Longitud Cromosoma	210	Activación desde las metas (γ)	50
Prob. de Mutación (p_m)	0.5	Activación desde las metas protegidas (δ)	90

Tabla 12.25: Configuración Paramétrica EX05

La configuración paramétrica empleada en este experimento se presenta en la tabla 12.25. Nótese que se proporciona una configuración por defecto para los parámetros globales de las RCs puesto que solamente se evaluará el aspecto estructural de estas, y no el aspecto funcional. Obsérvese además que la estructura de una RC se codifica en un cromosoma multigénico de 7 genes, cada uno de una longitud de 30 elementos (10 para el segmento cabecera y 20 para el segmento terminación según la ecuación 9.16). Por último cabe resaltar que los parámetros para el algoritmo PEG (e.g., la probabilidad de mutación, prob. de recombinación, etc.) fueron ajustados empíricamente a partir de la configuración sugerida en (Ferreira, 2006).

Por último, al concluir el experimento se espera que el proceso ontogenético descubra la RC objetivo que es presentada en la figura 12.26, estructura que debe ser identificada en el mínimo número posible de épocas evolutivas.

12.4.1.2. Diseño del Escenario

En la figura 12.27 se presenta el escenario empleado para este experimento.

El procedimiento que se lleva a cabo para el experimento es el siguiente:

1. Se sitúa al robot en un punto aleatorio del escenario y se deja actuar durante 500 épocas evolutivas, cada una de las cuales ejecuta 10 ciclos cognitivos.
2. En cada época es activado el proceso ontogenético que evoluciona la población de RCs. Para este experimento se emplea la función de aptitud FAAE (ecuación 9.25), la cual evalúa únicamente el aspecto estructural de cada RC.

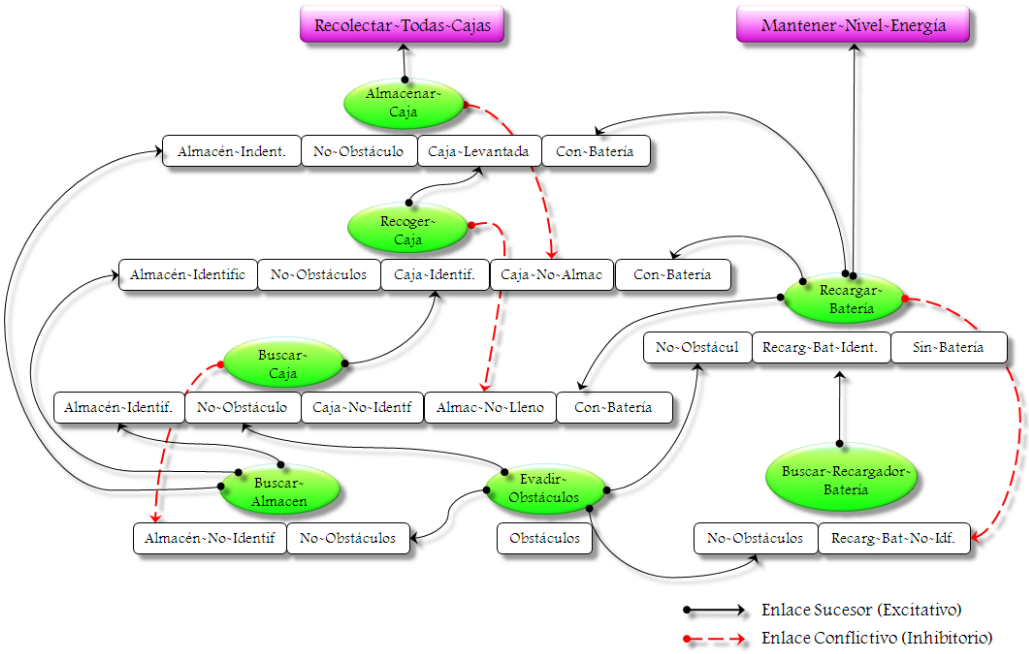


Figura 12.26: Topología de la Red de Comportamientos (RC) objetivo.

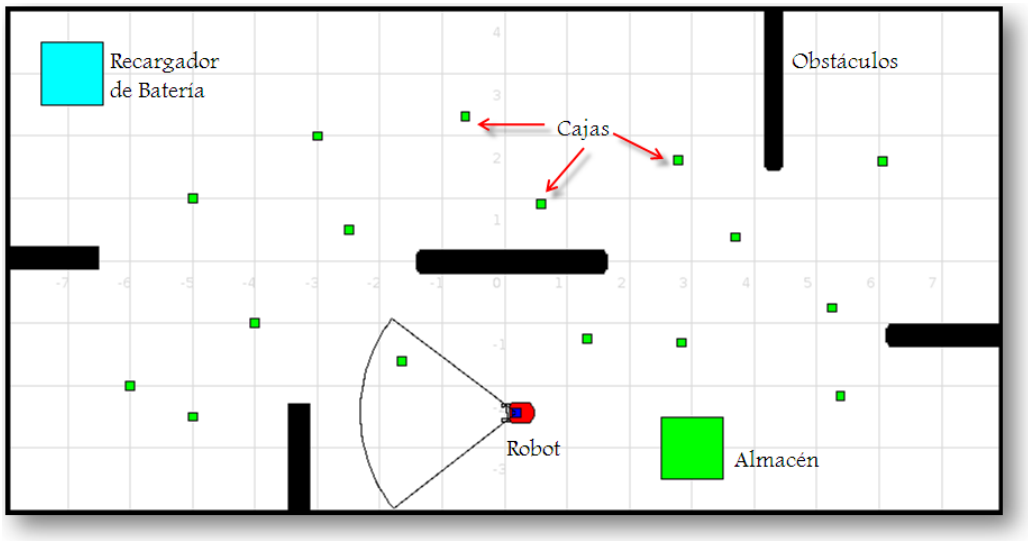


Figura 12.27: Escenario para la meta Recolectar-Todas-Cajas

3. El experimento es ejecutado 30 veces empleando las mismas condiciones iniciales, y se estiman las medias armónicas para los niveles de aptitud de las RCs.

12.4.1.3. Análisis de Resultados

En la gráfica 12.28 se presentan la progresión del nivel de aptitud de la RC mejor adaptada y del nivel de aptitud promedio del repertorio de RCs, durante el transcurso de las épocas evolutivas.

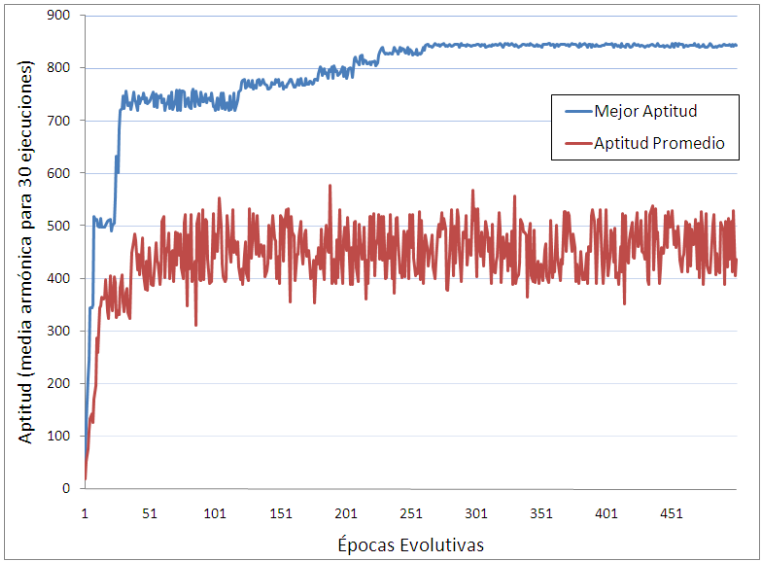


Figura 12.28: Progresión del Mejor Nivel de Aptitud y de la Aptitud Promedio de la Población.

Es importante resaltar que, en cada una de las 30 ejecuciones del experimento, el proceso ontogenético siempre encontró una solución óptima, esto es, una RC con una topología como la presentada en la figura 12.26, para lo cual requirió un mínimo de 259 épocas evolutivas y un máximo de 283 épocas evolutivas. Después de que el sistema alcanza el óptimo global (i.e., después de que el algoritmo halla la RC objetivo), se mantiene en un estado estacionario por el resto de las épocas (como se observa en la progresión del mejor nivel de aptitud).

Adicionalmente, la figura 12.28 sugiere diferentes dinámicas evolutivas para la población de RCs. Las oscilaciones en la progresión del nivel de aptitud promedio, incluso después haber descubierto una solución óptima, resultan ser una característica típica de la Programación Genética en general, y de la PEG en particular. En gran medida, el grado de oscilación se debe al reducido tamaño de la población de RCs (50 individuos), con lo cual las dinámicas de diversificación producen cambios drásticos sobre la población, generando un número variable de individuos poco adaptados, aunque manteniendo siempre la convergencia hacia la solución óptima a través de un número selecto de individuos. Se puede decir por tanto que el algoritmo evolutivo propuesto para el proceso ontogenético posee un equilibrio entre diversidad y convergencia.

La estructura genotípica de la RC hallada por la mejor solución se presenta a continuación (por motivos de simplicidad, solamente es presentado el marco de lectura abierto – MLA – para cada gen):

Gen Estructural 0: C1-D-P1-P2-P8-P2-P1-P1-P4-P1:

Traducción Fenotípica:

Comportamiento: Buscar-Almacén (C1)

Precondiciones:	⟨No-Obstáculos⟩ ⟨Almacén-No-Identificado⟩
Efectos Deseados:	⟨Almacén-Identificado⟩
Efectos No Deseados:	⟨Almacén-No-Identificado⟩

Gen Estructural 1: C2-T-P5-D-D-D-P1-P2-P6-P8-P12-P6-P9:

Traducción Fenotípica:

Comportamiento: Buscar-Caja (C2)

Precondiciones:	⟨Almacén-Identificado⟩ ⟨No-Obstáculos⟩ ⟨Almacén-No-Lleno⟩ ⟨Caja-No-Identificada⟩ ⟨Con-Batería⟩
Efectos Deseados:	⟨Caja-Identificada⟩
Efectos No Deseados:	⟨Caja-No-Identificada⟩

Gen Estructural 2: C3-T-P11-D-P5-D-D-P14-P12-P8-P1-P9-P13:

Traducción Fenotípica:

Comportamiento: Recoger-Caja (C3)

Precondiciones:	⟨Caja-Identificada⟩ ⟨No-Obstáculos⟩ ⟨Almacén-Identificado⟩ ⟨Caja-No-Almacenada⟩ ⟨Con-Batería⟩ ⟨Caja-No-Levantada⟩
Efectos Deseados:	⟨Caja-Levantada⟩
Efectos No Deseados:	⟨Caja-No-Levantada⟩

Gen Estructural 3: C4-D-M1-P13-D-D-P9-P11-P1-P8:

Traducción Fenotípica:

Comportamiento: Almacenar-Caja (C4)

Precondiciones:	⟨Con-Batería⟩ ⟨Caja-Levantada⟩ ⟨Almacén-Identificado⟩ ⟨No-Obstáculos⟩ ⟨Caja-No-Almacenada⟩
Efectos Deseados:	⟨Recolectar-Todas-Cajas⟩
Efectos No Deseados:	⟨Caja-No-Almacenada⟩

Gen Estructural 4: C5-P7-P8-null:

Traducción Fenotípica:

Comportamiento: Evadir-Obstáculos (C5)

Precondiciones:	⟨Obstáculos⟩
Efectos Deseados:	⟨No-Obstáculos⟩
Efectos No Deseados:	null

Gen Estructural 5: C6-D-P3-P4-P4-P8:

Traducción Fenotípica:

Comportamiento: Buscar-Recargador-Batería (C6)

Precondiciones:	⟨Recargador-Batería-No-Identificado⟩ ⟨No-Obstáculos⟩
Efectos Deseados:	⟨Recargador-Batería-Identificado⟩
Efectos No Deseados:	⟨Recargador-Batería-No-Identificado⟩

Gen Estructural 6: C7-D-D-D-P3-D-P9-M2-P4-P10-P8-P10:

Traducción Fenotípica:

Comportamiento: Recargar-Batería (C7)

Precondiciones:	⟨Recargador-Batería-Identificado⟩ ⟨No-Obstáculos⟩ ⟨Sin-Batería⟩
Efectos Deseados:	⟨Con-Batería⟩ ⟨Mantener-Nivel-Energía⟩
Efectos No Deseados:	⟨Sin-Batería⟩

A pesar de hallar una buena solución para la tarea de recolección de cajas, el proceso ontogenético encontró además algunas soluciones alternativas cuyos genes estructura-

les mejoraban la solución objetivo propuesta al inicio de este experimento. Sin embargo, estos genes se hallaron dispersos en cromosomas no óptimos, y por consiguiente, la estructura cromosómica de la RC expuesta anteriormente sigue siendo la mejor solución para el problema. Además, esta configuración estructural de la RC le permitió al sistema mantenerse en un estado estable por casi 250 épocas, lo que significa que durante este tiempo el robot dejó de recibir refuerzos negativos como consecuencia de sus decisiones.

Por ejemplo, se identificó un gen estructural interesante que definía la proposición ⟨Sin-Batería⟩ en la lista de efectos deseados del *Comportamiento* ⟨Recoger-Caja⟩. Esta simple variación causa que el robot verifique si dispone de la suficiente energía para almacenar la caja que tiene levantada sin tener que interrumpir luego la ejecución del *Comportamiento* ⟨Almacenar-Caja⟩. En condiciones normales, si el robot se queda sin energía durante el almacenamiento de una caja, éste deberá soltarla (a fin de ahorrar la energía que supone levantarla) y dirigirse luego hacia el recargador de batería, para luego buscar nuevamente la caja que dejó abandonada en algún lugar del escenario. Sin embargo, con esta nueva proposición, y en el caso de que el robot no tuviese suficiente energía para transportar la caja, entonces se dirigiría primero hasta el recargador de batería (sin soltar la caja) y luego si completaría la tarea de almacenamiento, esta vez sin tener que rehacer la tarea desde el paso de búsqueda de la caja. Esto se debe a que el *Comportamiento* ⟨Recoger-Caja⟩ dispersa cierta cantidad de energía de activación hacia adelante al *Comportamiento* ⟨Recargar-Batería⟩ a través de la proposición ⟨Sin-Batería⟩, haciendo que este último *Comportamiento* sea más competitivo y pueda hacerse con el control del robot después de la ejecución del *Comportamiento* ⟨Recoger-Caja⟩. Es importante notar que esta proposición adicional, así como otras proposiciones definidas por diferentes genes, no se esperaban encontrar en la solución objetivo, con lo cual se pueden considerar como propiedades emergentes resultantes del proceso ontogenético.

Conclusiones del Experimento: a partir de la estructura topológica de la mejor solución encontrada por el proceso ontogenético, se puede señalar lo siguiente:

- Se cumple la condición de que la red converge hacia una solución, esto es, que al menos un *Comportamiento* de la red debe dispersar energía de activación de forma "directa" hacia una meta global. Concretamente, la red define dos *Comportamientos* que dispersan energía directamente hacia dos metas diferentes: ⟨Almacenar--Caja⟩ → ⟨Recolectar-Todas-Cajas⟩, y ⟨Recargar-Batería⟩ → ⟨Mantener-Nivel-Energía⟩.
- Se cumple la condición de que la red no define *Comportamientos* "desconectados", esto es, que existe al menos una proposición que conecta las pre-condiciones de un *Comportamiento* con los efectos deseados o no deseados de algún otro *Comportamiento*.
- Se cumple con la condición de que la red no debe definir ciclos de auto-referencia ni callejones sin salida, esto es, que existe al menos un camino posible (definido por los enlaces de excitación y de inhibición) entre cada uno de los *Comportamientos* de la red y alguna de las metas del agente.

- Se cumple que ninguno de los *Comportamientos* de la RC define proposiciones ambiguas en ninguna de sus listas de proposiciones. Por ejemplo, el *Comportamiento* ⟨Recargar-Batería⟩ define en su lista de pre-condiciones la proposición ⟨Sin-Batería⟩, pero no la proposición contraria ⟨Con-Batería⟩. Esto se cumple para todas las proposiciones de todos los *Comportamientos* de la RC.
- Se cumple que ninguno de los *Comportamientos* de la red define proposiciones contradictorias o auto-inhibitorias, esto es, que ninguna de las proposiciones de la lista de efectos deseados aparece simultáneamente en la lista de efectos no deseados, y viceversa.

12.4.2. Experimento EX06: Análisis del proceso adaptativo para las características de Orientabilidad hacia las metas vs. Orientabilidad Situacional

12.4.2.1. Descripción

Este experimento pretende demostrar cómo el proceso ontogenético ajusta progresivamente los parámetros globales de la Red de Comportamientos con el fin de adaptar las características de orientabilidad hacia las metas y orientabilidad situacional descritas en la sección 9.2.5.

Para este experimento, se emplea la misma configuración paramétrica para la tarea de recolección de cajas descrita en el experimento EX05, salvo que todas las RCs del repertorio son inicialmente configuradas con una estructura topológica como la propuesta en el experimento EX05 y con todos sus parámetros globales inicializados con un valor de 95. Adicionalmente, el robot debe ahora identificar dos tipos diferentes de cajas que se hallan dispersas en su ambiente: tipo A (de color verde) y tipo B (de color naranja); las cuales, una vez identificadas, deben ser llevadas al almacén correspondiente. Por tanto, ahora se requieren *Comportamientos* especializados para cada tipo de caja (e.g., ⟨Buscar-Caja-Tipo-A⟩, etc...).

Debido a que el propósito de este experimento es que el proceso ontogenético descubra los valores de los parámetros globales que mejor se ajustan a los cambios ambientales en el mínimo número de épocas evolutivas, se empleará únicamente la función que evalúa el aspecto funcional de cada RC, esto es, la función FAAF (ecuación 9.35). En este caso, el proceso ontogenético realiza un máximo de 100 épocas de evolutivas por cada ejecución del experimento, donde cada época evolutiva ejecuta un máximo de 100 ciclos cognitivos.

12.4.2.2. Diseño de los Escenarios

Para este experimento se emplearon dos tipos diferentes de escenarios:

- **Escenario 1:** este escenario mide la orientabilidad hacia las metas. Para tal efecto, el robot tiene activa la meta específica ⟨Recolectar-Cajas-Tipo-B⟩, meta que debe cumplir a pesar de que dinámicamente reciba cierta cantidad de energía de

activación relativa al estado observado, que en este caso corresponde a un conjunto de siete cajas de tipo A. Es decir, el robot deberá demostrar una conducta orientada hacia la consecución de la meta especifica inicial que persigue el almacenaje de una caja del tipo B (color naranja), evitando el cambio de atención hacia la recolección de las cajas verdes.

- **Escenario 2:** este escenario mide la orientabilidad situacional. En este caso, el robot percibe al mismo tiempo la caja de tipo B y las siete cajas de tipo A, con lo cual debe tomar la decisión de recolectar primero las cajas de tipo A debido a que recibe mayor activación desde el estado observado.

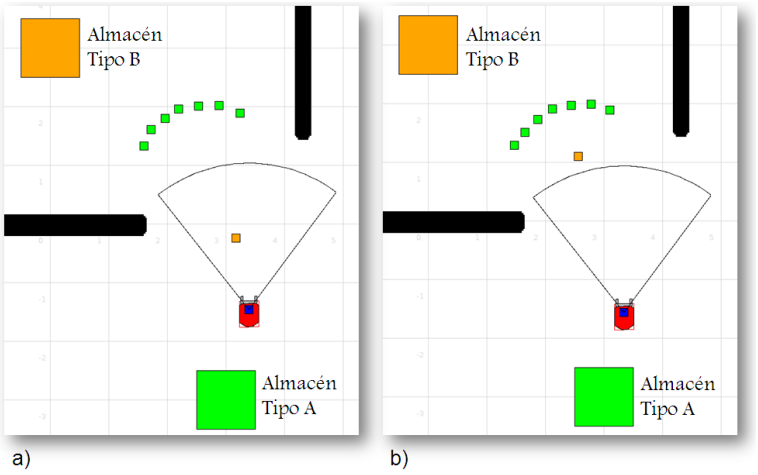


Figura 12.29: Escenarios para el experimento EX06. a) Escenario 1: el robot primero identifica la caja de tipo B y luego percibe las cajas de tipo A. b) Escenario 2: el robot percibe casi inmediatamente la caja de tipo B y las siete cajas de tipo A.

En la figura 12.29 se presentan los escenarios para este experimento. Como nota adicional, vale la pena mencionar que cada uno de los escenarios es presentado de forma intercalada al robot. Es decir, en las épocas evolutivas impares el robot se sitúa en el escenario 1, y en las ápocas impares se sitúa en el escenario 2. Esto con el fin de ajustar paralelamente los parámetros globales para los dos escenarios.

12.4.2.3. Análisis de Resultados

Resultados para el Escenario 1: después de ejecutar las 100 épocas evolutivas, el robot es situado en el Escenario 1 donde exhibe la conducta que se presenta en la figura 12.30. En esta figura se evidencia que el robot se enfoca en alcanzar su meta de almacenar la caja de tipo B sin cambiar el foco de atención cuando percibe las otras siete cajas verdes.

Una vez concluidas las 100 épocas evolutivas del proceso ontogenético, se escoge la RC con la mejor configuración paramétrica, esto es, la RC que ha acumulado el mayor

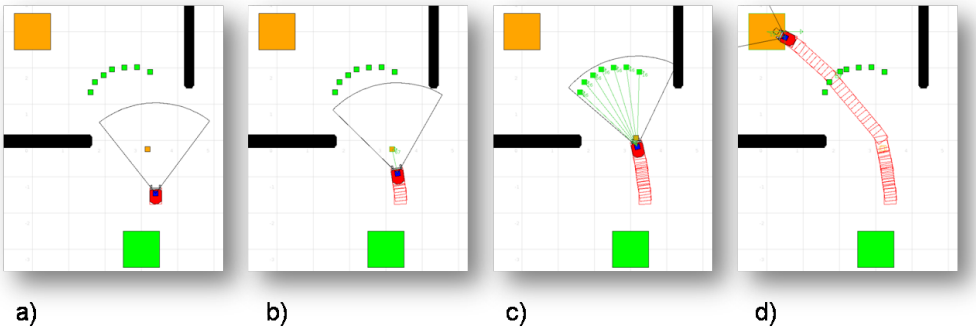


Figura 12.30: Conducta aprendida para el Escenario 1. a) El robot percibe la caja de tipo B. b) El robot se aproxima hasta la caja de tipo B. c) El robot recoge la caja de tipo B y percibe otras siete cajas de tipo A. d) El robot continúa con su meta de almacenar la caja de tipo B sin cambiar el foco de atención.

nivel de aptitud. La gráfica 12.31 presenta la progresión de los niveles de activación para cada uno de los *Comportamientos* de dicha RC en el Escenario 1[14].

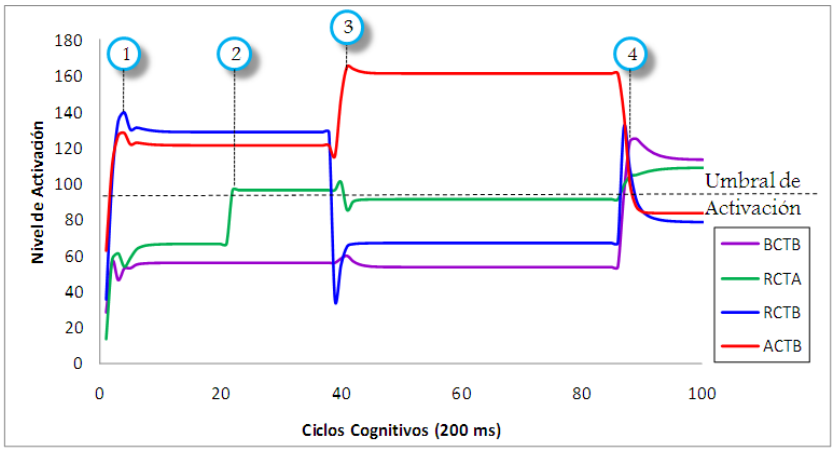


Figura 12.31: Progresión de los niveles de activación para cada *Comportamiento* en el Escenario 1. BCTB: ⟨`Buscar-Caja-Tipo-B`⟩. RCTA: ⟨`Recoger-Caja-Tipo-A`⟩. RCTB: ⟨`Recoger-Caja-Tipo-B`⟩. ACTB: ⟨`Almacenar-Caja-Tipo-B`⟩.

Adicionalmente, en la gráfica 12.31 se han identificado cuatro instantes clave donde se pueden apreciar los cambios de foco de atención del robot:

- Inicialmente, los *Comportamientos* compiten por sobrepasar el umbral de activación. En el instante (1), debido a que la caja de color naranja es la única que se encuentra dentro del campo de percepción del robot, el *Comportamiento* ⟨`Recoger-Caja-Tipo-B`⟩ es quien se convierte en el proceso consciente dominante del

[14]Nótese que por motivos de simplificación del gráfico, sólo son presentados los *Comportamientos* involucrados en la realización de la tarea de recolección de la caja tipo B.

agente. En este mismo instante se dispara la meta ⟨Recolectar-Todas-Cajas--Tipo-B⟩ ya que es un efecto deseado del *Comportamiento* activo.

- En la medida que el robot se aproxima hacia la caja de tipo B para recogerla, empieza a percibir las otras siete cajas de tipo A, produciéndose un incremento en la activación del estado observado, y por ende, aumentando el nivel de activación del *Comportamiento* ⟨Recoger-Caja-Tipo-A⟩, como se observa en (2). No obstante, debido a que el valor del parámetro ϕ definido por el proceso ontogenético no es lo suficientemente alto (como se discutirá más adelante), el *Comportamiento* ⟨Recoger-Caja-Tipo-A⟩ no acumula la cantidad necesaria de energía de activación, y por tanto no puede ejecutarse. Obsérvese además, que debido a que dicho *Comportamiento* no recibe energía de activación desde ningún otro *Comportamiento*, su nivel de activación permanece constante durante el tiempo que las cajas de color verde están dentro del campo de percepción del robot.
- En el instante (3) el robot termina de recoger la caja de color naranja y el *Comportamiento* ⟨Almacenar-Caja-Tipo-B⟩ se hace con el control consciente hasta que el robot deposita la caja en el almacén correspondiente.
- Por último, en el instante (4) se observa que el nivel de activación del *Comportamiento* ⟨Almacenar-Caja-Tipo-B⟩ desciende puesto que el robot ha cumplido la meta y debido a que no existen más *Comportamientos* que dispersen activación hacia éste. Se activa entonces el *Comportamiento* ⟨Buscar-Caja-Tipo--B⟩ quien ha acumulado mayor cantidad de energía de activación.

Resultados para el Escenario 2: el robot es situado en el Escenario 2 donde exhibe la conducta que se presenta en la figura 12.32. En esta figura queda patente que el robot decide cambiar la meta ⟨Recolectar-Todas-Cajas-Tipo-B⟩ (M1) por la meta ⟨Recolectar-Todas-Cajas-Tipo-A⟩ (M2) al recibir una mayor activación desde el estado observado. A diferencia del Escenario 1 donde el robot dispara la meta M1 antes de hallar el resto de cajas verdes, en el Escenario 2 el robot encuentra tanto la caja naranja como las cajas verdes casi en el mismo instante de tiempo, debido a su proximidad, con lo cual decide cambiar el foco de atención hacia las cajas verdes y darle prioridad a la meta M2.

En la gráfica 12.33 se presenta la progresión de los niveles de activación de los *Comportamientos* de la RC con mayor nivel de aptitud, evolucionada por el proceso ontogenético y contextualizada en el Escenario 2.

- Inicialmente, se activa el *Comportamiento* ⟨Buscar-Caja-Tipo-B⟩ que dirige el robot hacia la caja de color naranja. Una vez el robot se ha aproximado lo suficiente a la caja naranja, la energía de activación se concentra en el *Comportamiento* ⟨Recoger-Caja-Tipo-B⟩ quien incrementa progresivamente su nivel de activación, como se observa en el instante (1). En este mismo instante, se dispara la meta ⟨Recolectar-Todas-Cajas-Tipo-B⟩ puesto que es un efecto deseado del *Comportamiento* activo.
- Luego, unos cuantos ciclos cognitivos después el robot percibe las otras cajas verdes, con lo cual los *Comportamientos* ⟨Recoger-Caja-Tipo-B⟩ y ⟨Recoger--Caja-Tipo-A⟩ compiten por el foco de atención hasta que, finalmente, ⟨Reco-

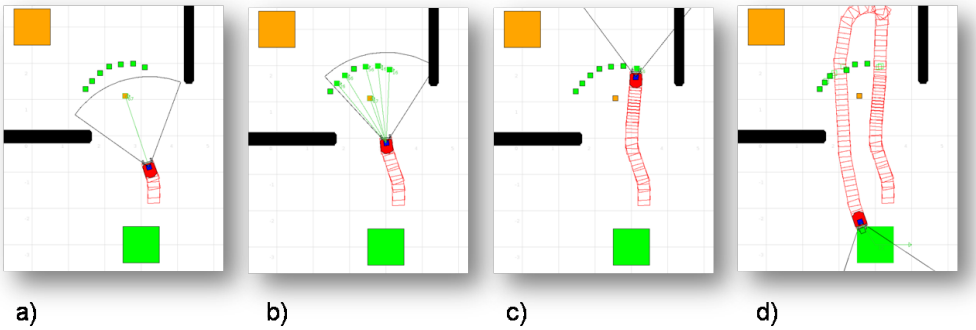


Figura 12.32: Conducta aprendida para el Escenario 2. a) El robot percibe la caja de tipo B y se aproxima hacia ella para recogerla. b) El robot percibe otras cajas de tipo A. c) El robot cambia de meta y se dirige hacia una caja de tipo A. d) El robot almacena la caja de tipo A.

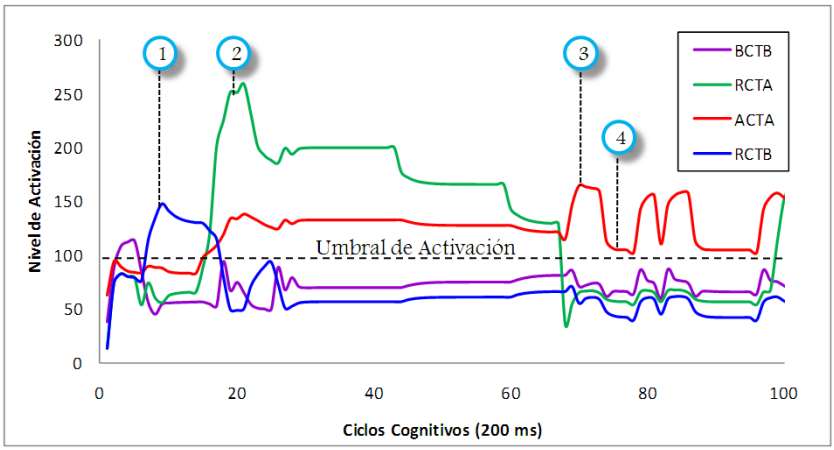


Figura 12.33: Progresión de los niveles de activación para cada *Comportamiento* en el Escenario 2. BCTB: ⟨Buscar-Caja-Tipo-B⟩. RCTB: ⟨Recoger-Caja-Tipo--B⟩. ACTA: ⟨Almacenar-Caja-Tipo-A⟩. RCTA: ⟨Recoger-Caja-Tipo-A⟩.

ger-Caja-Tipo-A⟩ se hace con el control consciente debido a que acumula mayor cantidad de energía en menos tiempo, como se observa en el instante (2). En este mismo instante de tiempo se dispara la meta ⟨Recolectar-Todas--Cajas-Tipo-A⟩, la cual subordina la meta disparada anteriormente. Nótese que en el momento en que el robot decide enfocar su atención sobre alguna de las cajas verdes, el nivel de activación del *Comportamiento* ⟨Recoger-Caja--Tipo-A⟩ empieza a disminuir debido, por un lado, a que deja de recibir energía por el estado percibido en la medida que se aleja de las otras cajas verdes y, por otro, a que el mismo *Comportamiento* produce una señal de auto-inhibición, evitando de esta manera el estar activo indefinidamente.

- En el instante (3) se activa el *Comportamiento* ⟨Almacenar-Caja-Tipo-A⟩ quien venía acumulando energía de activación desde el instante anterior.

- Finalmente, en el instante (4) se observa que el nivel de activación de ⟨Almacenar-Caja-Tipo-A⟩ desciende debido a que el *Comportamiento* ⟨Evadir-Obstáculos⟩ obtiene el control del robot[15]. Luego, el *Comportamiento* ⟨Almacenar-Caja-Tipo-A⟩ recupera nuevamente el control del robot y se mantiene activo hasta que la meta es alcanzada.

A diferencia del Escenario 1, en el Escenario 2 el robot re-orienta el foco de atención y por tanto re-plantea sus metas al encontrarse una mayor cantidad de cajas verdes por recoger. De esta manera, en el Escenario 2, el *Comportamiento* ⟨Recoger-Caja-Tipo-A⟩ se hace dominante debido a que recibe mayor cantidad de energía de activación procedente tanto del estado (las siete cajas percibidas), como de la nueva meta definida en tiempo de ejecución.

Convergencia Evolutiva: en cuanto a la curva evolutiva del proceso ontogenético, se observa en la gráfica 12.34 que la mejor configuración funcional para las RC es encontrada en la época evolutiva no. 12. Después de esta época, el sistema se mantiene en un estado estacionario o estable, es decir, que al evaluar la mejor solución encontrada por medio de la función FAAF (ecuación 9.35), ninguno de sus términos produce refuerzos negativos. Como resultado, la configuración óptima de los parámetros globales para las RCs, encontrada por el proceso ontogenético, es presentada en la tabla 12.26.

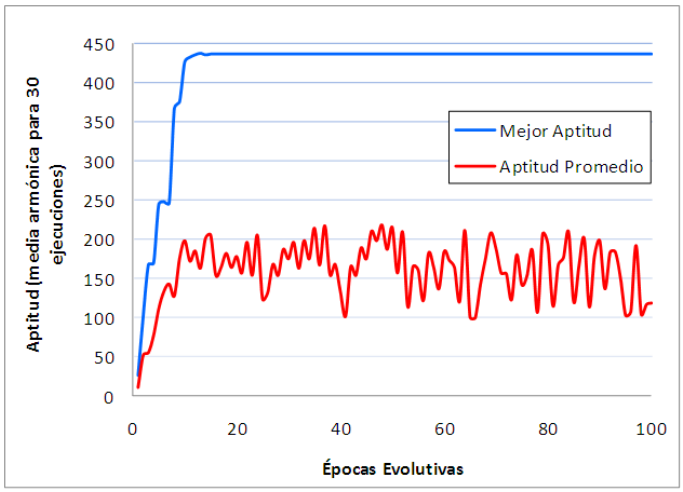


Figura 12.34: Progresión del Mejor Nivel de Aptitud y de la Aptitud Promedio de la Población - EX06.

Interpretación de los Resultados y Conclusiones del Experimento: A partir de los resultados obtenidos puede señalarse lo siguiente:

[15]Por simplicidad de la gráfica, el *Comportamiento* ⟨Evadir-Obstáculos⟩, así como los *Comportamientos* relacionados con la recarga de batería y con la recolección de objetos de tipo A no son presentados en la gráfica

Parámetro	Valor
Umbral de Activación (θ)	95
Nivel Promedio de Activación (π)	92
Activación por el estado observado (ϕ)	23
Activación por metas (γ)	68
Activación por metas protegidas (δ)	85

Tabla 12.26: Configuración de Parámetros Globales - EX06

- Con respecto a los valores iniciales de cada RC, los parámetros globales que sufrieron mayor cambio durante el proceso de adaptación fueron: γ (la cantidad de energía de activación inyectada por una meta a la red) y ϕ (la cantidad de energía de activación inyectada por el estado observado a la red).
- El proceso ontogenético determinó que un valor para γ mayor que el valor para ϕ, mantiene al agente centrado en la realización de sus objetivos. Adicionalmente, el proceso ontogenético descubrió una relación de diferencia apropiada y balanceada entre estos dos parámetros globales ($\gamma = 68$ y $\phi = 23$) puesto que:
  - Un valor muy alto para γ, o un valor muy bajo para ϕ, implicaría que el robot orientase su comportamiento completamente hacia el logro de la meta actual, esto es, que no le daría mucha importancia a los cambios del estado observado y, por consiguiente, el agente difícilmente podría re-dirigir su conducta hacia otras metas y objetivos cuando algo interesante se le presentase. Por ejemplo, en la época evolutiva no. 13 el robot encontró una configuración funcional con valores para $\gamma = 87$ y $\phi = 6$. Esta configuración funcional reflejó una solución parcial puesto que el robot llevó a cabo perfectamente la tarea de recolección de cajas para el Escenario 1, aunque no para el Escenario 2, donde independientemente de la cantidad de cajas verdes observadas, el robot no era capaz de re-dirigir su atención y cambiar de meta como consecuencia.
  - Un valor de ϕ mayor que el de γ provocaría que el robot estuviese continuamente cambiando de meta actual, con lo cual nunca terminaría de recoger las cajas en ninguno de los dos escenarios propuestos. Por ejemplo, las configuración funcionales de las RCs con menor nivel de aptitud demostraban una relación de $\phi > \gamma$ tal que, cuando el robot intentaba recolectar una caja y se encontraba luego con cajas de otro tipo, soltaba la que estaba cargando y se disponía a recoger otra, la cual volvía a soltar después de percibir una caja diferente, y así sucesivamente.
  - A partir de las múltiples ejecuciones del experimento se determina que para mantener un equilibrio apropiado entre la orientabilidad hacia las metas y la orientabilidad situacional de la RC (para los escenarios propuestos), γ debe ser aproximadamente entre un 29 y un 34 % mayor que ϕ.
- Queda demostrada la capacidad del agente para mantener la atención sostenida (*"concentración"*) sobre un estímulo o sobre algún aspecto del mismo, con prefe-

rencia sobre otros estímulos. Este aspecto atencional, determinado en gran parte por la entrada de activación a la red relativa al estado observado (γ), permite que el agente se concentre en la ejecución de una tarea a pesar de la sobrecarga de estímulos percibidos.

12.4.3. Experimento EX07: Análisis del proceso adaptativo para las características de Procesamiento Deliberativo vs. Reactividad

12.4.3.1. Descripción

Este experimento tiene como propósito evaluar la capacidad del proceso ontogenético para evolucionar los parámetros globales del repertorio de RCs, ante situaciones que requieren que el agente actúe reactivamente aunque sin desviar su conducta global de la meta asignada: ⟨`Recolectar-Todas-Cajas`⟩. Asúmase, para este experimento, la misma configuración paramétrica inicial descrita para el experimento EX06.

12.4.3.2. Diseño de los Escenarios

En la figura 12.35 se presenta el escenario empleado durante el experimento. El procedimiento del experimento se detalla a continuación:

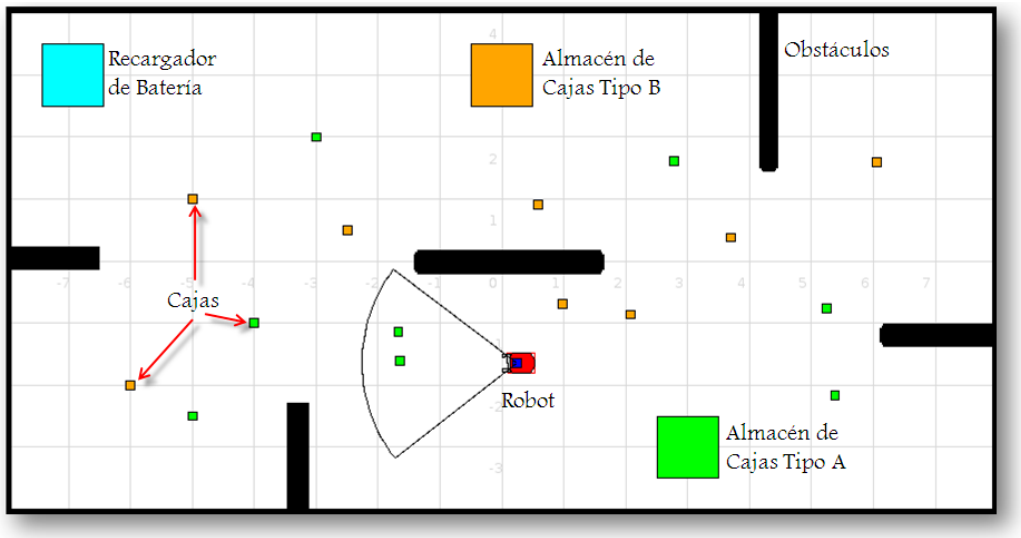


Figura 12.35: Escenario para el experimento EX07.

1. Se asume que el robot ha identificado previamente la ubicación de cada uno de los almacenes.
2. Se sitúa el robot cerca de una caja.
3. Mientras el robot intenta llevar la caja identificada hasta el almacén correspondiente, se desplaza uno de los muros "móviles" y se posiciona justo en frente del robot. Se repite la acción un par de veces más intentando obstaculizar el paso del robot.

4. Como consecuencia, el robot deberá evadir oportunamente el obstáculo que se desplaza dinámicamente frente a él, y deberá retomar la tarea de almacenar la caja que lleva cargada en cuanto la situación se lo permita.
5. El proceso ontogenético emplea 100 épocas evolutivas, cada una conformada por 200 ciclos cognitivos. El experimento se repite 30 veces con el objetivo de calcular las medias armónicas.

12.4.3.3. Análisis de Resultados

Como se puede observar en la figura 12.36, cuando el robot percibe el desplazamiento dinámico de los obstáculos, éste responde con una conducta reactiva y evasiva. No obstante, a pesar de tener que delegar el foco de atención en el *Comportamiento* ⟨`Evadir-Obstáculos`⟩, el robot retoma nuevamente el *Comportamiento* ⟨`Almacenar-Caja-Tipo-A`⟩ una vez se halla libre de obstáculos.

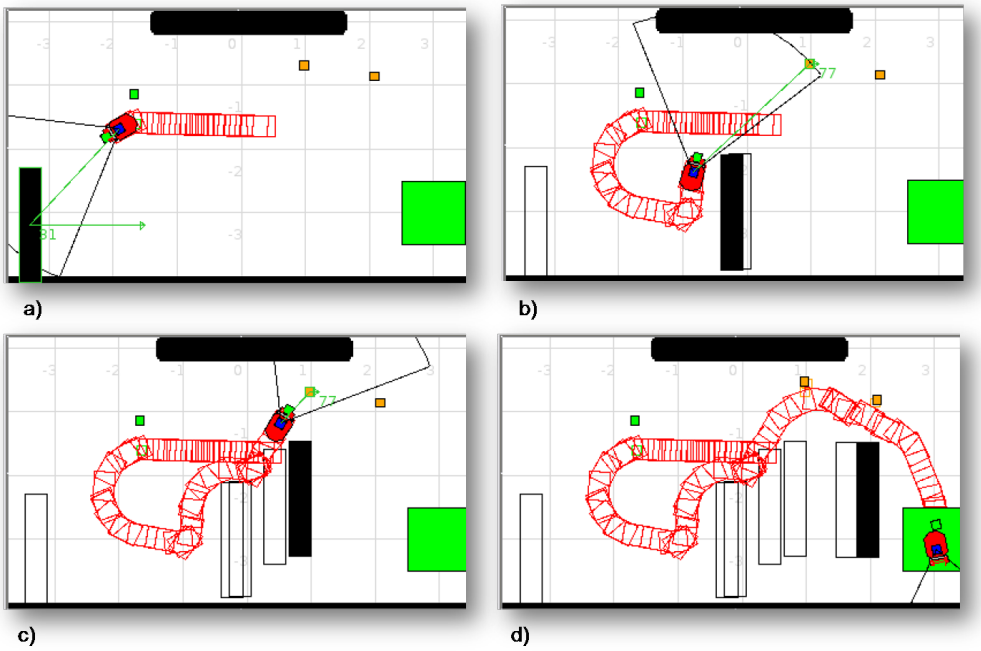


Figura 12.36: Conducta aprendida por el robot. a) El robot recoge una caja de tipo A. b) y c) El robot responde de manera reactiva evadiendo el muro que se desplaza dinámicamente frente a él. d) El robot almacena exitosamente la caja.

La conducta reactiva del agente se observa más claramente en la figura 12.37, de la cual cabe destacar:

- En el instante (1), el robot identifica la caja de color verde y por tanto el *Comportamiento* ⟨`Recoger-Caja-Tipo-A`⟩ recibe mayor energía de activación.
- En el instante (2), después de agarrar y levantar la caja verde, se activa como consecuencia el *Comportamiento* ⟨`Almacenar-Caja-Tipo-A`⟩.

- En el instante (3) se observa que el nivel de activación del *Comportamiento* ⟨`Evadir-Obstáculos`⟩ es incrementado, repentinamente, como una respuesta evasiva a la situación peligrosa que supone el obstáculo móvil justo en frente del robot. Cabe resaltar la magnitud y la velocidad (casi inmediata) con la que es disparado el nivel de activación de ⟨`Evadir-Obstáculos`⟩, en comparación con el resto de *Comportamientos*. Se puede observar además que este patrón de conducta es repetido por el agente cuantas veces halle en su camino el obstáculo móvil.
- En el instante (4) se observa que el agente reanuda nuevamente su meta inicial, la cual intenta alcanzar a través del *Comportamiento* ⟨`Almacenar-Caja-Tipo-A`⟩. Este patrón de conducta lo sigue el robot cada vez que se encuentra libre de obstáculos.

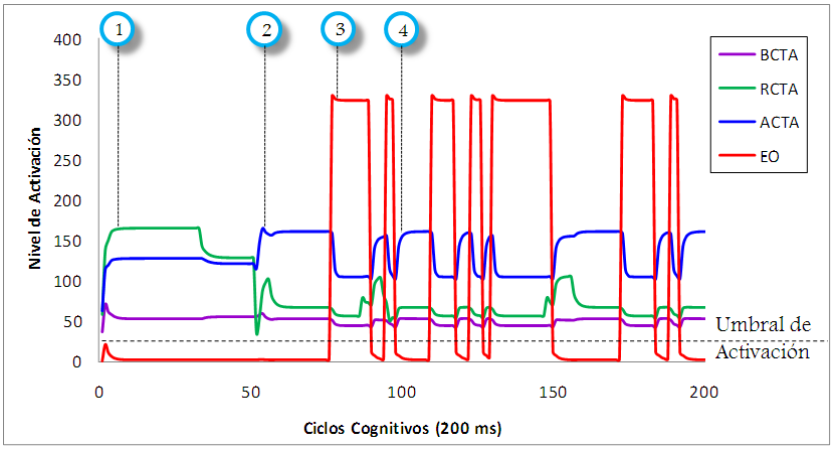


Figura 12.37: Progresión de los niveles de activación para cada *Comportamiento*. BCTA: ⟨`Buscar-Caja-Tipo-A`⟩. RCTA: ⟨`Recoger-Caja-Tipo-A`⟩. ACTA: ⟨`Almacenar-Caja-Tipo-A`⟩. EO: ⟨`Evadir-Obstáculos`⟩.

Convergencia Evolutiva: en la figura 12.38 se observa que el proceso ontogenético converge rápidamente encontrando una configuración funcional óptima en la época no. 9. Concretamente, la mejor solución encontrada define cambios considerables (en comparación con lo valores iniciales) para los parámetros globales θ (el umbral de activación), γ (la cantidad de energía de activación inyectada por una meta a la red), y ϕ (la cantidad de energía de activación inyectada por el estado observado a la red), como se observa en la tabla 12.27.

Interpretación de los Resultados y Conclusiones del Experimento: A partir de los resultados obtenidos puede señalarse lo siguiente:

- El proceso ontogenético ha determinado que una diferencia pequeña entre la cantidad de activación producida por el estado observado ($\phi = 64$) y la cantidad de activación producida por las metas actuales ($\gamma = 57$), permite que el agente mantenga su atención enfocada en las metas, aunque alerta a las situaciones inesperadas

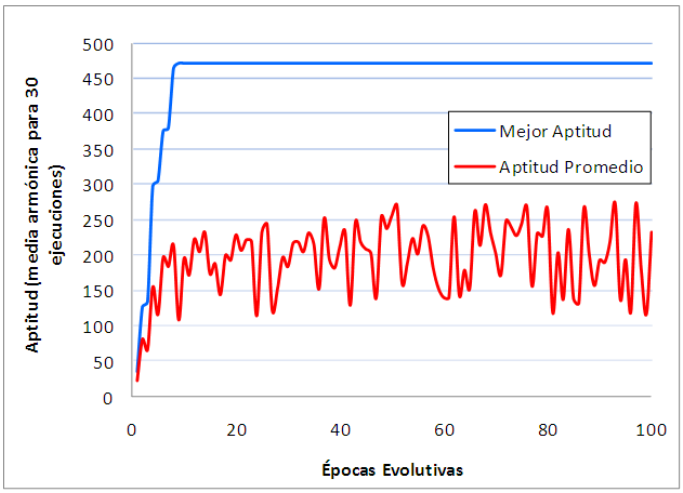


Figura 12.38: Progresión del Mejor Nivel de Aptitud y de la Aptitud Promedio de la Población - EX07.

Parámetro	Valor
Umbral de Activación (θ)	15
Nivel Promedio de Activación (π)	93
Activación por el estado observado (ϕ)	64
Activación por metas (γ)	57
Activación por metas protegidas (δ)	87

Tabla 12.27: Configuración de Parámetros Globales - EX07

y peligrosas que puedan presentarse, ante las cuales actúa de forma reactiva. A partir de las múltiples ejecuciones del experimento se determina que para mantener un equilibrio apropiado entre el procesamiento deliberativo y el procesamiento reactivo de la RC (para los escenarios propuestos), γ debe ser aproximadamente entre un 9 y un 15 % menor que ϕ.

- El proceso ontogenético ha descubierto que al disminuir progresivamente el umbral de activación (θ) en aquellas situaciones donde se requiere una actuación reactiva por parte del agente, se ejecutan menos ciclos correspondientes a las dinámicas de dispersión, con lo cual los *Comportamientos* son activados más rápidamente sin efectuarse una cantidad considerable de procesamiento deliberativo. El valor óptimo encontrado (en promedio) por el proceso ontogenético para el escenario propuesto es $\theta = 15$. Las soluciones que propusieron valores inferiores para θ demostraron una modulación de *Comportamientos* fuertemente reactiva, impidiendo que el robot retomara el *Comportamiento* $\langle$`Almacenar-Caja-Tipo-A`$\rangle$ después de toparse con un obstáculo móvil. Para valores de $\theta > 40$ el robot no tenía suficiente tiempo para reaccionar ante el obstáculo presentado, debido al incremento considerable en el número de ciclos relativos a las dinámicas de dispersión.

- Se demuestra que el agente modula de forma apropiada la activación de *Comportamientos* cuando se presentan distracciones que exigen una re-orientación reactiva del foco de atención. Este factor atencional, denominado oscilación (como se dis-cutió en la sección 9.2.1), es controlado en gran medida por las tres entradas adicionales de energía de activación que se proponen, en esta tesis, al modelo original de Maes: *activación por estímulos novedosos* (ecuación 9.4), *activación por cambios físicos* (ecuación 9.5), y *activación por elementos relevantes* (ecuación 9.6).

12.4.4. Experimento EX08: Análisis del proceso adaptativo para las características de Adaptabilidad vs. Predisposición hacia los Planes en curso

12.4.4.1. Descripción

El propósito del presente experimento es evaluar la configuración de los parámetros globales ajustados por el proceso ontogenético en situaciones donde se requiere un grado de adaptabilidad apropiado que le permita al agente, por un lado, re-orientar del foco de atención de acuerdo con los cambios internos y externos percibidos y, por otro, demostrar una tendencia por reanudar el plan en curso. Asúmase, para este experimento, la misma configuración paramétrica inicial descrita para el experimento EX06.

12.4.4.2. Diseño de los Escenarios

Para llevar a cabo este experimento se proponen tres escenarios diferentes:

- **Escenario 1:** durante la tarea de recolección de cajas, el motivador ⟨`Nivel--Energía`⟩ desciende hasta la zona de infra-estimulación donde es activada la meta ⟨`Mantener-Nivel-Energía`⟩. Idealmente, el robot deberá abandonar la tarea actual, dirigirse hacia el recargador de batería, y una vez incrementado el nivel de energía, reanudar la tarea de recolección de cajas.
- **Escenario 2:** el robot se encuentra llevando una caja hasta el almacén correspondiente, cuya ubicación se asume ha sido aprendida y guardada previamente en la memoria declarativa. Repentinamente, el almacén es desplazado hasta otra ubicación. Idealmente, el robot deberá activar la sub-meta ⟨`Identificar--Almacén`⟩ antes de continuar con el almacenaje de la caja que lleva cargada.
- **Escenario 3:** en este escenario se sitúa el robot frente a una caja verde. Previamente se desconecta de todas las RCs el *Comportamiento* ⟨`Recoger-Cajas--Tipo-B`⟩, con lo cual el agente tendrá que actuar adaptativamente y encontrar una solución alternativa sin emplear dicho *Comportamiento* (tolerancia a fallos).

En cuanto al modelo gráfico de los escenarios, se emplea el mismo diseño propuesto para el experimento EX07 (ver figura 12.35).

12.4.4.3. Análisis de Resultados

Resultados para el Escenario 1: en la figura 12.39 se observa que, después de que las RCs han sido evolucionadas por el proceso ontogenético, el robot exhibe la conducta esperada para el Escenario 1.

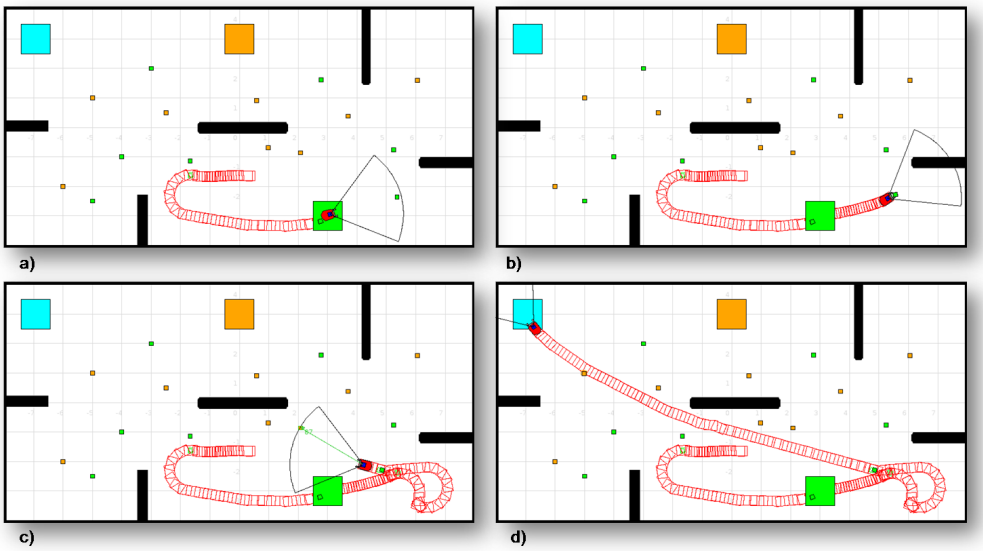


Figura 12.39: Conducta aprendida por el robot para el Escenario 1. a) El robot almacena una primera caja verde. b) El robot recoge una segunda caja verde. c) La energía del robot se agota, por tanto se activa la meta ⟨Mantener-Nivel-Energía⟩ y el robot suelta la caja que lleva cargada. d) El robot se dirige hacia el recargador de batería.

A continuación se destacan algunos aspectos interesantes de las dinámicas de dispersión de energía de la RC mejor adaptada por el proceso ontogenético, en referencia a la gráfica 12.40:

- En el instante (1) el robot percibe una caja de color verde y activa el *Comportamiento* ⟨Recoger-Caja-Tipo-A⟩.
- Una vez el robot ha recogido la caja, se activa el *Comportamiento* ⟨Almacenar--Caja-Tipo-A⟩, como se observa en el instante (2).
- En los instantes (3) y (4), que corresponden al hallazgo de una segunda caja verde, se observa un patrón de activación similar a los instantes (1) y (2).
- En el instante (5) el motivador ⟨Nivel-Energía⟩ alcanza su límite de infra-estimulación, con lo cual el robot activa la meta ⟨Mantener-Nivel-Energía⟩. Esta meta dispersa una gran cantidad de energía de activación hacia el *Comportamiento* ⟨Recargar-Batería⟩, el cual obtiene el control del robot.
- Es importante destacar que mientras está activo ⟨Recargar-Batería⟩, los *Comportamientos* que prometen alcanzar la meta ⟨Recolectar-Todas-Cajas--Tipo-A⟩ (RCTA y ACTA) no re-inicializan a cero su nivel de activación; por

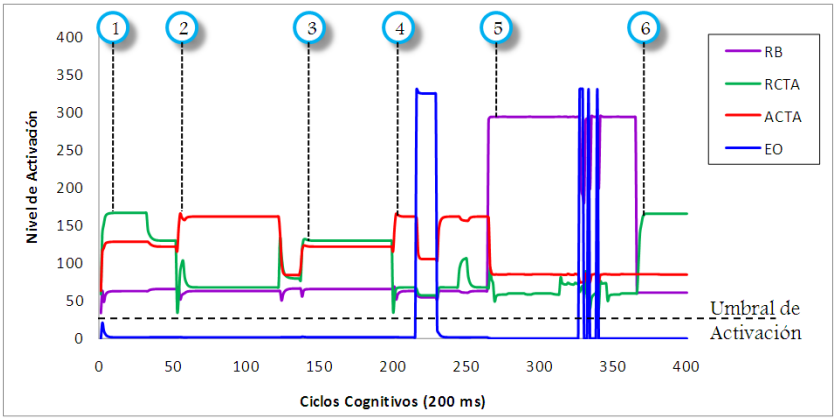


Figura 12.40: Progresión de los niveles de activación para cada *Comportamiento* en el Escenario 1. RB: ⟨`Recargar-Batería`⟩. RCTA: ⟨`Recoger-Cajas-Tipo-A`⟩. ACTA: ⟨`Almacenar-Cajas-Tipo-A`⟩. EO: ⟨`Evadir-Obstáculos`⟩.

consiguiente, el agente después de recargar la batería no necesita re-planificar y puede activar rápidamente el *Comportamiento* ⟨`Recoger-Caja-Tipo-A`⟩, el cual le permite reanudar la tarea de recolección de cajas.

Resultados para el Escenario 2: en la figura 12.41 se observa que después de evolucionadas las RCs, el robot exhibe una conducta como la esperada para el Escenario 2.

A continuación se describen los instantes más relevantes de las dinámicas de dispersión de energía para este escenario, en relación a la gráfica 12.42:

- En el instante (1), el robot percibe una caja de tipo B en frente de él y se dispone a recogerla.
- Una vez recogida la caja, el robot se dirige hacia el almacén respectivo, como se observa en el instante (2).
- En el instante (3), el almacén es desplazado de su ubicación original, con lo cual se cumple la pre-condición ⟨`Almacén-Tipo-B-No-Identificado`⟩. Esta pre-condición, por un lado, impide que el robot continúe con la ejecución de ⟨`Almacenar-Caja-Tipo-B`⟩ (motivo por el cual el robot deja caer la caja) y, por otro, le inyecta una mayor cantidad de energía de activación al *Comportamiento* ⟨`Buscar-Almacén-Tipo-B`⟩, quien obtiene el control del robot.
- Después de que el robot ha evadido un par de obstáculos y ha encontrado finalmente el almacén, se reactiva el *Comportamiento* ⟨`Recoger-Caja-Tipo-B`⟩ quien retoma la meta de recolectar la caja de tipo B que ha dejado caer el robot en el instante anterior. Nótese que, como en el escenario anterior, durante los cambios de foco de atención los niveles de activación de los *Comportamientos* que prometen almacenar la caja de tipo B (RCTB y ACTB) no son re-inicializados a cero, con lo cual se mantienen competitivos y recuperan fácilmente el control del robot, como

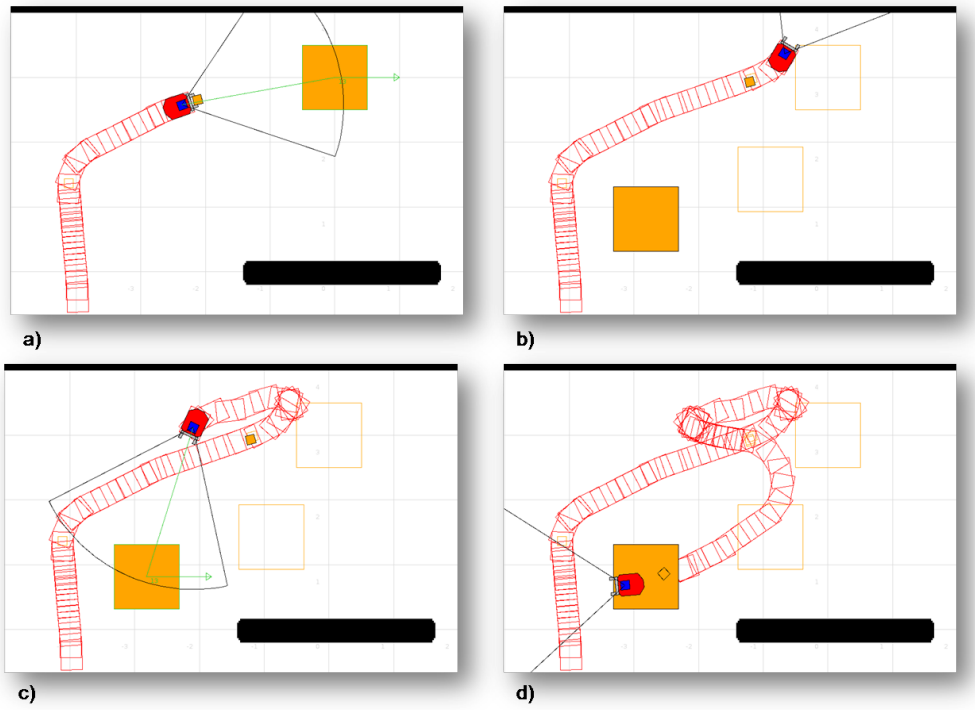


Figura 12.41: Conducta aprendida por el robot para el Escenario 2. a) El robot lleva cargada una caja de tipo B e identifica el almacén correspondiente. b) El almacén es desplazado a otra ubicación, por tanto, el robot re-orienta el foco de atención hacia la búsqueda del almacén y deja caer la caja. c) El robot encuentra la nueva ubicación del almacén. d) El robot reanuda la tarea de recolección de la caja tipo B.

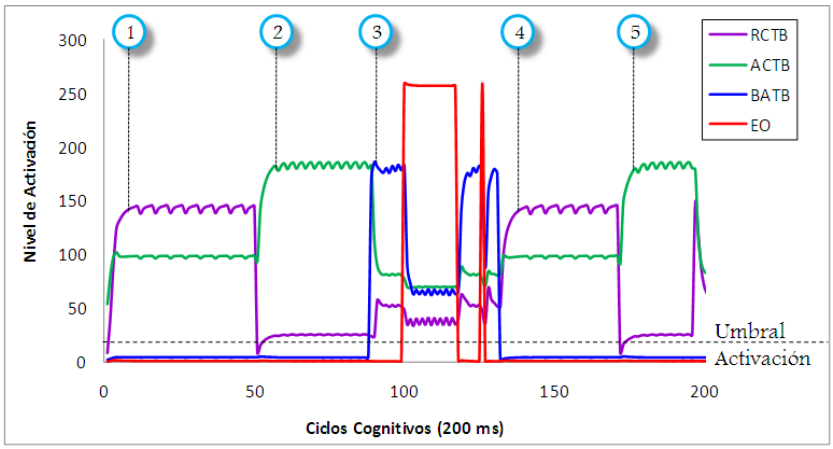


Figura 12.42: Progresión de los niveles de activación para cada *Comportamiento* en el Escenario 2. RCTB: ⟨`Recoger-Cajas-Tipo-B`⟩. ACTB: ⟨`Almacenar-Cajas--Tipo-B`⟩. BATB: ⟨`Buscar-Almacén-Tipo-B`⟩. EO: ⟨`Evadir-Obstáculos`⟩.

se observa en los instantes (4) y (5). Esta propiedad resulta análoga a los procesos de "concentración" y "atención focalizada" del cerebro (como se discutió en la sección 9.2.1).

Resultados para el Escenario 3: como se puede notar en la figura 12.43, el robot exhibe una conducta como la esperada para el Escenario 3, después de que el proceso ontogenético ha evolucionado sus RCs.

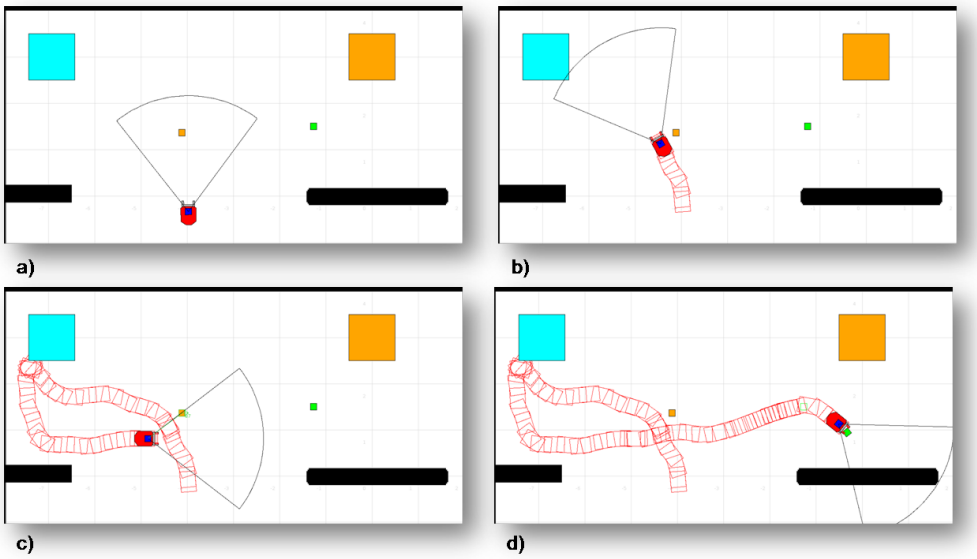


Figura 12.43: Conducta aprendida por el robot para el Escenario 3. a) El robot identifica una caja de tipo B. b) Debido a que no puede recoger la caja de tipo B (ya que el comportamiento correspondiente está desconectado), el robot se dispone a buscar una caja de tipo A. c) El robot continua buscando una caja de tipo A. d) Finalmente el robot encuentra una caja de tipo A y la recoge.

A continuación se hacen algunas observaciones sobre los instantes más relevantes de las dinámicas de dispersión de energía llevadas a cabo por la RC mejor adaptada, las cuales se registran en la gráfica 12.44:

- En el instante (1) el robot percibe una caja de tipo B en frente suyo[16]. Nótese que debido a que el *Comportamiento* ⟨`Recoger-Caja-Tipo-B`⟩ (RCTB) ha sido *"desconectado"* de la RC, el *Comportamiento* ⟨`Buscar-Caja-Tipo-B`⟩ no recibe la suficiente energía de activación hacia atrás desde RCTB, motivo por el cual ⟨`Buscar-Caja-Tipo-A`⟩ es quien se hace con el control consciente del robot, a pesar de que no hay cajas de tipo A en el campo de percepción aunque sí una caja de tipo B.

[16]Es importante recordar la secuencia en que idealmente deberían ser activados los *Comportamientos* en condiciones normales: ⟨`Buscar-Almacén-Tipo-B`⟩ → ⟨`Buscar-Caja-Tipo-B`⟩ → ⟨`Recoger--Caja-Tipo-B`⟩ → ⟨`Almacenar-Caja-Tipo-B`⟩. Para este experimento, asúmase que el robot ya ha identificado previamente la ubicación del almacén y por lo cual no requiere ejecutar el primer *Comportamiento*.

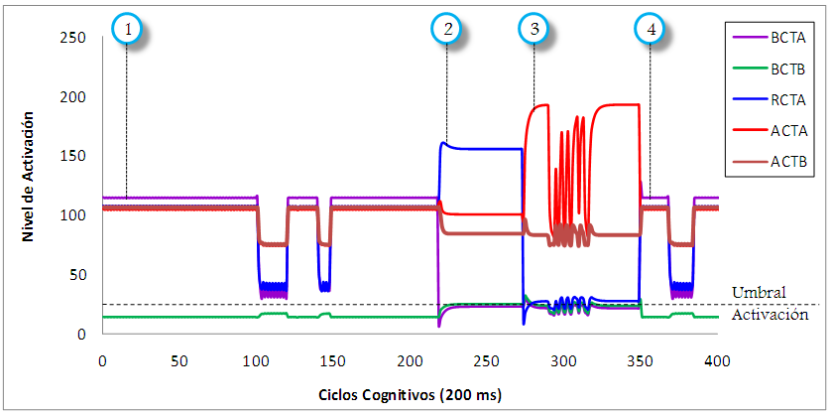


Figura 12.44: Progresión de los niveles de activación para cada *Comportamiento* en el Escenario 3. BCTA: ⟨`Buscar-Cajas-Tipo-A`⟩. BCTB: ⟨`Buscar-Cajas-Tipo-B`⟩. RCTA: ⟨`Recoger-Cajas-Tipo-A`⟩. ACTA: ⟨`Almacenar-Cajas-Tipo-A`⟩. ACTB: ⟨`Almacenar-Cajas-Tipo-B`⟩.

- En el instante (2) el robot encuentra finalmente una caja de tipo A y se dispone a recogerla.
- Una vez el robot ha levantado la caja, procede a su almacenaje, como se observa en el instante (3).
- En el instante (4), el robot vuelve a activar el *Comportamiento* ⟨`Buscar-Caja-Tipo-A`⟩. Adviértase que a pesar de estar desconectado el *Comportamiento* RCTB, la RC encuentra siempre una solución alternativa, en este caso, sin emplear dicho *Comportamiento* ni ninguno de los *Comportamientos* asociados a éste a través de los enlaces excitativos e inhibitorios (e.g., BCTB, ACTB, ...). Se demuestra entonces que la RC evolucionada por el proceso ontogenético es tolerante a fallos. Este tipo de adaptabilidad es una consecuencia de la naturaleza distribuida del algoritmo: puesto que ninguno de los *Comportamientos* es más importante que los otros, las RCs son capaces de re-dirigir el foco de atención aún bajo precondiciones degradadas.

Convergencia Evolutiva: el proceso ontogenético para este experimento fue ejecutado durante 100 épocas evolutivas, en cada una de las cuales se iban intercalando los tres escenarios propuestos. La mejor solución fue hallada por el proceso ontogenético en la época no. 21, como se observa en la gráfica 12.45. A diferencia de los experimentos anteriores, en este experimento el proceso ontogenético tardó una mayor cantidad de épocas evolutivas debido a que tuvo que ajustar tres de los cinco parámetros globales para la RC mejor adaptada (π, γ, y ϕ), como se puede apreciar en la tabla 12.28.

Interpretación de los Resultados y Conclusiones del Experimento: a partir de los resultados obtenidos puede señalarse lo siguiente:

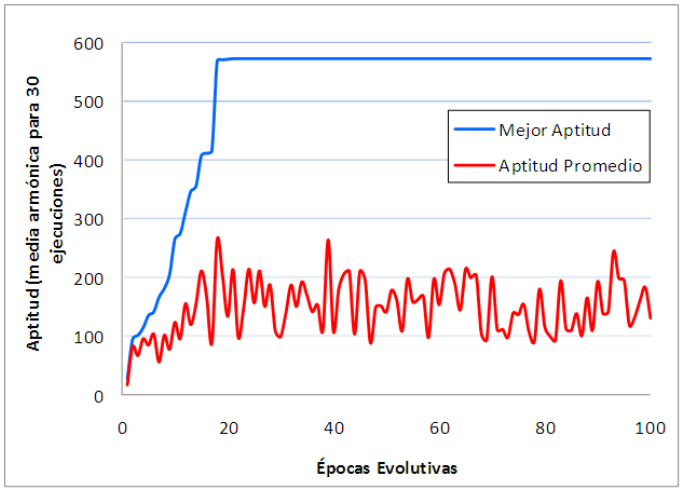


Figura 12.45: Progresión del Mejor Nivel de Aptitud y de la Aptitud Promedio de la Población - EX08.

Parámetro	Valor
Umbral de Activación (θ)	93
Nivel Promedio de Activación (π)	18
Activación por el estado observado (ϕ)	61
Activación por metas (γ)	37
Activación por metas protegidas (δ)	88

Tabla 12.28: Configuración de Parámetros Globales - EX08

- El proceso ontogenético ha determinado que un valor para π (el nivel promedio de activación) "*menor*" que los valores para γ (la cantidad de energía de activación inyectada por una meta a la red) y para ϕ (la cantidad de energía de activación inyectada por el estado observado a la red) garantiza la adaptabilidad de la RC frente a las perturbaciones del entorno y de las metas. Así pues, a partir de las múltiples ejecuciones del experimento se determina que para mantener un equilibrio apropiado entre las características de adaptabilidad y predisposición hacia los planes en curso de la RC (para los escenarios propuestos), π debe ser aproximadamente entre un 46 y un 52 % menor que γ, y entre un 72 y un 77 % menor que ϕ. Si se mantienen estos rangos aproximados el agente no estará continuamente "saltando" entre una meta y otra, y a su vez podrá explotar las oportunidades y adaptarse a las situaciones cambiantes.

- El proceso ontogenético ha determinado los valores óptimos para $\pi = 18$, $\gamma = 37$, y $\phi = 61$. A partir de las soluciones con "menor nivel de aptitud" se deduce que un valor de π considerablemente mayor que los valores de γ y ϕ, hace la RC más adaptativa pero menos orientada hacia los planes en curso, con lo cual, si el agente

percibe perturbaciones en el estado (interno o externo), éste cambiará continuamente las metas globales sin centrarse plenamente en ninguna de ellas.

- A partir de los resultados obtenidos en este experimento, así como en los obtenidos en los experimentos EX06, EX07, y EX08, resulta evidente que la configuración de parámetros globales es un problema "multi-objetivo", con lo cual una única RC no puede definir una configuración ideal para todos los escenarios propuestos. Se requiere entonces de múltiples soluciones compitiendo por sobrevivir en el acervo o repertorio de MATs, de entre las cuales, sólo las mejores serán activadas de acuerdo con la situación observada por el agente.

12.4.5. Experimento EX09: Análisis del proceso adaptativo para la característica de Prevención de Metas en Conflicto

12.4.5.1. Descripción

Este experimento tiene como propósito evaluar la convergencia evolutiva del proceso ontogenético en el descubrimiento de una configuración paramétrica óptima tal que, las RCs puedan prevenir los conflictos entre las múltiples metas que debe alcanzar el agente. Para llevar a cabo este experimento, se propone una tarea donde el robot debe apilar, en un orden específico, un conjunto de cajas.

Para entender mejor el conflicto entre metas que puede generarse, considérese la clásica *Anomalía de Sussman* (Sussman, 1975): el robot debe apilar los bloques que se presentan en el estado inicial de la figura 12.46.a), tal que el estado final sea: el bloque A sobre el bloque B, y el bloque B sobre el bloque C. Sin embargo, el robot solamente puede mover un bloque a la vez. El problema inicia con los bloques A y B sobre la mesa, y el bloque C sobre el bloque A. Se identifican por tanto dos metas: M1 = ⟨`A-sobre-B`⟩ y M2 = ⟨`B-sobre-C`⟩.

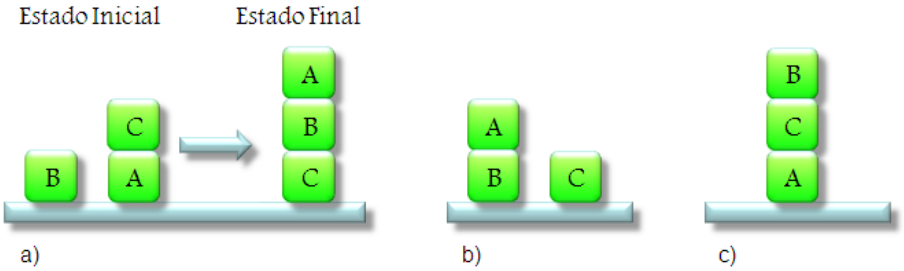


Figura 12.46: Anomalía de Sussman. a) estado inicial y estado final del apilamiento de bloques. b) La meta M1 impide el logro de la meta M2. c) La meta M2 impide el logro de la meta M1.

En el primer caso, supóngase que el agente persigue la meta M1. La solución más directa es quitar el bloque C, y luego poner el bloque A sobre el bloque B. Pero mientras esta secuencia satisface la meta M1, el agente no puede alcanzar la meta M2 sin deshacer la meta M1, con lo cual los dos bloques deben situarse simultáneamente encima del

bloque C, pero esto iría en contra de la restricción de mover sólo un bloque a la vez (véase la figura 12.46.b).

Por otro lado, si el robot decide empezar por la meta M2, la solución más eficiente es poner el bloque B sobre el bloque C, pero de nuevo el robot no puede alcanzar la meta M1 sin deshacer la meta M2. En ambos casos, el robot entraría en un callejón sin salida que no permitiría deshacer la meta previamente alcanzada.

Así pues, el proceso ontogenético deberá descubrir una configuración paramétrica que le permita al robot arbitrar correctamente los *Comportamientos* conflictivos a través de los enlaces inhibitorios.

12.4.5.2. Diseño de los Escenarios

Debido a que este tipo de problema no puede ser simulado con la plataforma *player/stage*, no se definirá un entorno gráfico para este experimento. Por tanto, como resultados del experimento se emplearán únicamente las secuencias de *Comportamientos* activados por la RC mejor adaptada. A continuación se describen los *Comportamientos* empleados:

Comportamiento: Apilar-A-sobre-B	
Precondiciones:	⟨A−libre⟩, ⟨B−libre⟩
Efectos Deseados:	⟨A−sobre−B⟩, ⟨C−libre⟩
Efectos No Deseados:	⟨B−libre⟩, ⟨A−sobre−C⟩

Comportamiento: Apilar-B-sobre-C	
Precondiciones:	⟨C−libre⟩, ⟨B−libre⟩
Efectos Deseados:	⟨B−sobre−C⟩, ⟨A−libre⟩
Efectos No Deseados:	⟨C−libre⟩, ⟨B−sobre−A⟩

Comportamiento: Quitar-C-de-A	
Precondiciones:	⟨C−libre⟩, ⟨C−sobre−A⟩
Efectos Deseados:	⟨A−libre⟩
Efectos No Deseados:	⟨C−sobre−A⟩

La proposición ⟨n−libre⟩ indica que el bloque n no tiene ningún otro bloque encima. Por otro lado, el *Comportamiento* ⟨Quitar−C−de−A⟩ supone que el bloque C está sobre el bloque A, y que es retirado y puesto sobre la mesa. El estado inicial observado por el agente corresponde a: $S(t)$ = ⟨C−libre⟩, ⟨B−libre⟩, ⟨C−sobre−A⟩, y las metas globales son: $G(t)$ = ⟨A−sobre−B⟩, ⟨B−sobre−C⟩.

12.4.5.3. Análisis de Resultados

A partir de las gráficas 12.47 y 12.48, las cuales presentan los resultados obtenidos para la mejor y las dos peores RCs halladas por el proceso ontogenético, se puede señalar lo siguiente:

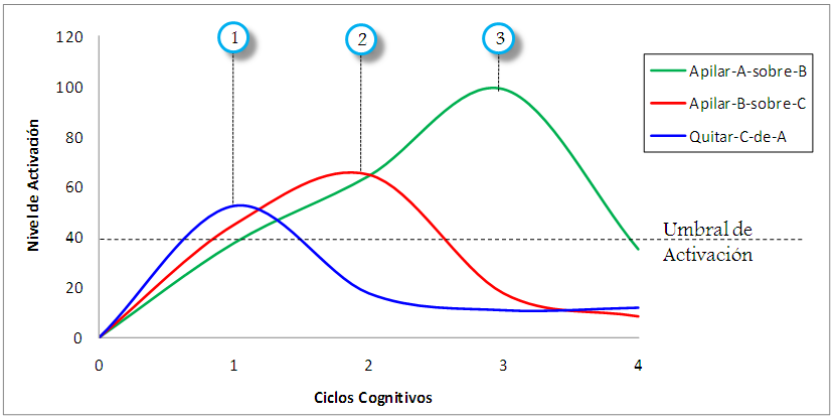


Figura 12.47: Progresión de los niveles de activación para los *Comportamientos* de la RC mejor adaptada. Con $\gamma = 17$, $\delta = 48$, y $\phi = 63$.

- La RC mejor adaptada define un valor para δ (la cantidad de energía de activación que una meta protegida libera de la red) "mayor" que γ (la cantidad de energía de activación que una meta inyecta a la red) y "menor" que ϕ (la cantidad de energía de activación que una proposición del estado observado inyecta a la red). Debido a esta configuración, el *Comportamiento* ⟨`Quitar-C-de-A`⟩ recibe una mayor activación por el estado inicial percibido (⟨`C-libre`⟩ y ⟨`C-sobre-A`⟩), mientras que el *Comportamiento* ⟨`Apilar-A-sobre-B`⟩ es inhibido por la proposición no verdadera ⟨`A-libre`⟩, y el *Comportamiento* ⟨`Apilar-B-sobre-C`⟩ no recibe la cantidad suficiente de activación por la meta ⟨`B-sobre-C`⟩ debido a que el valor de γ es bajo. Adicionalmente, puesto que $\delta > \gamma$, el *Comportamiento* ⟨`Quitar-C-de-A`⟩ substrae una gran cantidad de energía de los otros dos *Comportamientos* a través de sus enlaces de conflicto, con lo cual se hace dominante en el instante (1) y elimina la condición conflictiva ⟨`C-sobre-A`⟩. En el instante (2), se activa el *Comportamiento* ⟨`Apilar-B-sobre-C`⟩ puesto que el valor de δ previene el tener que deshacer una meta protegida o previamente alcanzada, situación que ocurriría si se activase primero ⟨`Apilar-A-sobre-B`⟩. Los resultados se presentan en la gráfica 12.47.
- En el caso de que δ no sea lo suficientemente alto en comparación con el valor de γ, el robot logra la sub-meta de dejar libre el bloque A (quitando el bloque C de encima), como sucedió para la mejor RC adaptada, sin embargo, en el instante (2) se activaría el *Comportamiento* ⟨`Apilar-A-sobre-B`⟩ y la RC entraría en conflicto al no poder luego deshacer la meta alcanzada, como se observa en la gráfica 12.48.a. Por otro lado, si se cumple que $\gamma > \delta$, el *Comportamiento* ⟨`Apilar--A-sobre-B`⟩ se hace dominante en el instante (1) y, por tanto, el robot "nunca"

logra la sub-meta de remover primero el bloque C con el objetivo de dejar libre el bloque A, como se observa en la gráfica 12.48.b.

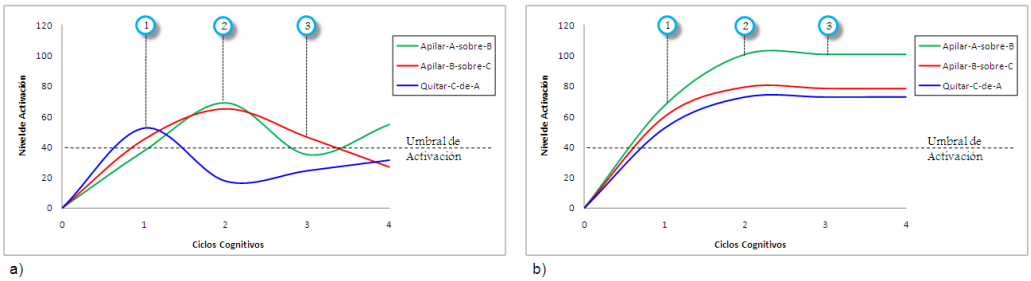


Figura 12.48: Progresión de los niveles de activación para los *Comportamientos* de dos de las RCs peores adaptadas. a) $\gamma = 19$, $\delta = 36$, y $\phi = 21$. b) $\gamma = 54$, $\delta = 22$, $\phi = 16$.

Convergencia Evolutiva: como se puede observar en la gráfica 12.49, la mejor solución fue hallada por el proceso ontogenético en la época no. 27. En este experimento el proceso ontogenético tardó una mayor cantidad de épocas evolutivas debido a que tuvo que ajustar los cinco parámetros globales para la RC mejor adaptada, como se puede apreciar en la tabla 12.29.

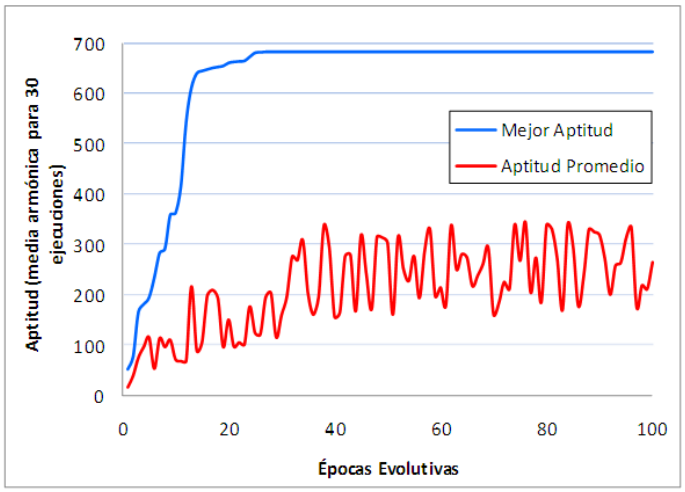


Figura 12.49: Progresión del Mejor Nivel de Aptitud y de la Aptitud Promedio de la Población - EX09.

Interpretación de Resultados y Conclusiones del Experimento: a partir de los resultados obtenidos se puede argumentar lo siguiente:

- El proceso ontogenético ha descubierto que, para que la RC pueda resolver apropiadamente los conflictos entre las metas del escenario propuesto, debe definir:

Parámetro	Valor
Umbral de Activación (θ)	41
Nivel Promedio de Activación (π)	18
Activación por el estado observado (ϕ)	63
Activación por metas (γ)	17
Activación por metas protegidas (δ)	48

Tabla 12.29: Configuración de Parámetros Globales - EX09

- Un valor aproximado para δ entre un 51 y un 65 % mayor que γ. Aquellas soluciones que definían una diferencia entre δ y γ menor al 50 %, no liberaban de la red la suficiente cantidad de energía de activación producida por las proposiciones conflictivas, mientras que aquellas soluciones que definían un valor para γ mayor que δ provocaban que la RC entrara en callejones sin salida al no poder deshacer las metas previamente alcanzadas que presentaban conflicto.
- Un valor aproximado para ϕ entre un 62 y un 73 % mayor que δ. En aquellos casos donde no se cumplía esta condición, la RC no recibía suficiente activación del estado percibido, con lo cual no era capaz de activar primero el *Comportamiento* que lograba la sub-meta de quitar el bloque C encima del bloque A.
- Un valor θ menor que el valor para δ. Por el contrario, en aquellas soluciones donde θ era mayor o igual que δ, la RC era más deliberativa, por tanto, ejecutaba más ciclos cognitivos durante los cuales los *Comportamientos* que prometían alcanzar una meta final de forma directa (i.e., $\langle$`Apilar-A-sobre--B`$\rangle$ y $\langle$`Apilar-B-sobre-C`$\rangle$) tenían la oportunidad de acumular mayor cantidad de energía de activación en comparación con el *Comportamiento* que prometía alcanzar la sub-meta que evitaba el conflicto ($\langle$`Quitar-C--de-A`$\rangle$).

- Así pues, queda demostrada la capacidad del proceso ontogenético para adaptar apropiadamente los parámetros globales de las RCs cuando el agente se enfrenta a situaciones con metas en conflicto.

12.4.6. Experimento EX10: Adaptabilidad del Proceso Ontogenético en la Construcción de Planes

12.4.6.1. Descripción

Este experimento tiene como propósito evaluar el grado de adaptabilidad del proceso ontogenético en la construcción de planes que involucran múltiples tareas. Concretamente, se definen cuatro tareas: dos para la recolección de cada tipo de cajas, y dos para el apilamiento ordenado de las cajas acopiadas en cada almacén. Un plan puede invocar estas

tareas una o varias veces, según se requiera, hasta que se satisfagan todas las metas globales: ⟨Recolectar-Todas-Cajas-Tipo-A⟩, ⟨Recolectar-Todas-Cajas--Tipo-B⟩, ⟨Mantener-Nivel-Energía⟩, ⟨Apilar-Todas-Cajas-Tipo-A⟩, y ⟨Apilar-Todas-Cajas-Tipo-B⟩.

La idea básica del problema es la siguiente: el robot tiene que recolectar los diferentes tipos de cajas (de color verde y naranja) las cuales se encuentra dispersas por todo el entorno, y luego debe almacenarlas en su correspondiente almacén. Las cajas son apiladas una encima de otra en el mismo orden en que son halladas por el robot. Después de que todas las cajas han sido acopiadas en cada almacén, el robot tendrá que apilarlas nuevamente siguiendo un orden determinado.

Así pues, se pretende que el proceso ontogenético descubra un plan que determine una secuencia apropiada de tareas, tal que sea optimizada la función de aptitud $FAFL_{(p)}$ (ecuación 10.9), esto es, que el mejor plan sea quien maximice la cantidad de refuerzo total obtenida por el robot y minimice el coste y la duración del plan. Para tal efecto, se emplea la configuración paramétrica de la tabla 12.30. Cabe notar que los parámetros del algoritmo PEG fueron ajustados empíricamente, tomando como base los criterios sugeridos en (Ferreira, 2006).

Parámetro Algoritmo PEG	**Valor**	**Parámetro Función $FAFL_{(p)}$**	**Valor**
Número de Experimentos	30	Constante $k1$	1
Número de Épocas Evolutivas	500	Constante $k2$	100
Población (repertorio) de Planes	50	Constante $k3$	1000
Número Genes Homeóticos	1	Energía por desplazamiento (Coste)	1 J
Longitud Cabecera Gen Homeótico	10	Energía por levantar caja (Coste)	5 J
Número de FDAs (1 por cada tarea)	4	Energía por apilar caja (Coste)	8 J
Prob. de Mutación ($p_{(m,h)}$)	0.5		
Prob. Recomb. uno y dos puntos ($p_{(rup,h)}$)	0.5		
Prob. de Transposición de SI ($p_{(si,h)}$)	0.3		

Tabla 12.30: Configuración Paramétrica EX10

12.4.6.2. Diseño de los Escenarios

En la figura 12.50 se presenta el escenario empleado para el experimento. Como se puede observar, se han dispuesto cinco cajas del tipo A (*a, b, c, d,* y *e*) y seis cajas del tipo B (*f, g, h, i, j,* y *k*). En la figura 12.50.b, se presenta un ejemplo donde el robot tiene que apilar, en orden alfabético, las cajas que han sido acopiadas inicialmente según el orden de recolección.

Para llevar a cabo este experimento se definen dos tipos de repertorios de MATs: uno para las MATs que se encargan de dirigir las tareas de recolección de cajas, para lo cual se emplea la topología de la RC propuesta en la figura 12.26, salvo que existirán *Comportamientos* individuales para cada tipo de caja; y un repertorio para las MATs que se

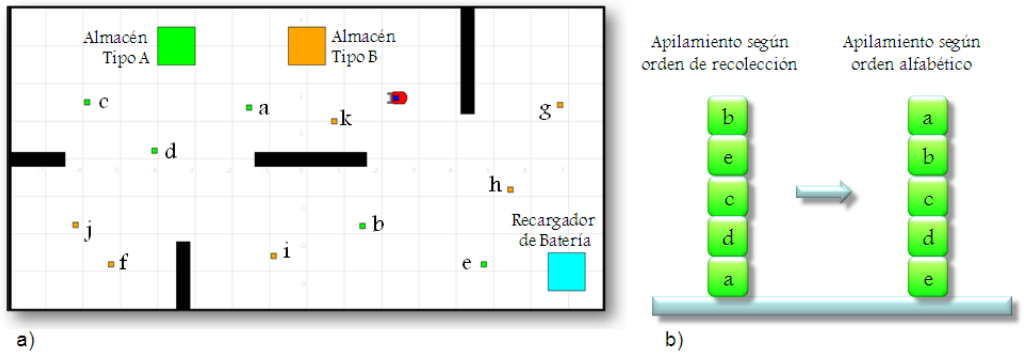


Figura 12.50: Escenario para el experimento EX10. a) Distribución de las cajas y de los almacenes. b) Ejemplo de apilamiento de cajas.

encargarán de coordinar la tarea de apilamiento de cajas, para lo cual se definirá una RC con una topología diferente a la descrita en el experimento EX09. El motivo de esto es que la RC propuesta en EX09 era adecuada, por su simplicidad, para poder apreciar la resolución de las metas en conflicto en dicho escenario concreto, no obstante, la cantidad de *Comportamientos* de la RC aumenta exponencialmente en la medida que se requiere ordenar un mayor número de cajas. Por tanto, se propone una topología de red diferente compuesta de solamente cuatro *Comportamientos*, la cual emplea un enfoque de resolución con tres pilas (similar al problema de las *Torres de Hanói* (Rouse y Macdonald, 1987)): la pila 1 contendrá las cajas amontonadas en desorden, la pila 2 servirá de ubicación temporal, y la pila 3 contendrá finalmente las cajas apiladas de forma correcta. La estructura topológica de la RC descubierta por el proceso ontogenético se presenta a continuación:

Comportamiento: Mover-desde-Pila1-hasta-Pila2	
Precondiciones:	⟨Pila1-no-está-vacía⟩ ⟨Caja-#-está-en-Pila1⟩ ⟨Caja-#-no-está-arriba-de-Pila1⟩
Efectos Deseados:	⟨Caja-#-está-arriba-de-Pila1⟩
Efectos No Deseados:	⟨Caja-#-no-está-arriba-de-Pila1⟩

Comportamiento: Mover-desde-Pila2-hasta-Pila1	
Precondiciones:	⟨Pila2-no-está-vacía⟩ ⟨Caja-#-está-en-Pila2⟩ ⟨Caja-#-no-está-arriba-de-Pila2⟩
Efectos Deseados:	⟨Caja-#-está-arriba-de-Pila2⟩
Efectos No Deseados:	⟨Caja-#-no-está-arriba-de-Pila2⟩

Comportamiento: Mover-desde-Pila1-hasta-Pila3	
Precondiciones:	⟨Pila1-no-está-vacía⟩ ⟨Caja-#-es-la-siguiente⟩ ⟨Caja-#-está-en-Pila1⟩ ⟨Caja-#-no-está-arriba-de-Pila1⟩
Efectos Deseados:	⟨Caja-#-está-arriba-de-Pila3⟩ ⟨Caja-#-sobre-Caja-$⟩
Efectos No Deseados:	⟨Caja-#-no-está-arriba-de-Pila1⟩

Comportamiento: Mover-desde-Pila2-hasta-Pila3	
Precondiciones:	⟨Pila2-no-está-vacía⟩ ⟨Caja-#-es-la-siguiente⟩ ⟨Caja-#-está-en-Pila2⟩ ⟨Caja-#-no-está-arriba-de-Pila2⟩
Efectos Deseados:	⟨Caja-#-está-arriba-de-Pila3⟩ ⟨Caja-#-sobre-Caja-$⟩
Efectos No Deseados:	⟨Caja-#-no-está-arriba-de-Pila2⟩

A diferencia del modelo clásico de Maes (Maes, 1989) que define a priori los valores de todas las proposiciones que emplea la red, la RC propuesta para el apilamiento de cajas hace uso de "variables" (símbolos #, y $), esto es, que los *Comportamientos* pueden hacer referencia a proposiciones que cambian de valor en tiempo de ejecución. Por ejemplo, la proposición ⟨Caja-#-sobre-Caja-$⟩ puede tomar los valores ⟨Caja-A-sobre-Caja-B⟩, ⟨Caja-B-sobre-Caja-C⟩, etc. El valor que tomará "#" dependerá de la siguiente caja que debe ser puesta sobre la pila 3 en orden alfabético descendente. Así pues, el uso de variables permite que los *Comportamientos* puedan ser reutilizados y, de esta manera, la RC puede ordenar una cantidad variable de cajas sin crecer exponencialmente su número de *Comportamientos*.

Debido a que el experimento se centrará en la construcción de planes como secuencias de tareas de alto nivel, las topologías y los parámetros globales de las RCs son evolucionados previamente por el proceso ontogenético. Asimismo, el sistema anticipatorio, los *Comportamientos*, y las memorias declarativas son entrenados y ajustados con anterioridad por el proceso epigenético. La tabla 12.31 resume la configuración paramétrica de las RCs mejor adaptadas para cada caso. Como se puede observar, los parámetros obtenidos caracterizan cada tipo de RC empleada, como se discutió en los experimentos anteriores (e.g., orientabilidad hacia las metas vs. orientabilidad situacional, procesamiento deliberativo vs. reactivo, etc.).

12.4.6.3. Análisis de Resultados

Como se ha mencionado anteriormente, el proceso ontogenético dispondrá de 500 épocas para evolucionar el repertorio de planes (cuyas secuencias de tareas se generan inicialmente de forma aleatoria). En cada época evolutiva, todo el repertorio de planes

Parámetro	Valor (Tarea: Recolectar Cajas)	Valor (Tarea: Apilar Cajas)
Umbral de Activación (θ)	87	39
Nivel Promedio de Activación (π)	25	19
Activación por el estado observado (ϕ)	53	65
Activación por metas (γ)	33	20
Activación por metas protegidas (δ)	92	46

Tabla 12.31: Configuración de Parámetros Globales - EX10

es evaluado empleando la función de aptitud $FAFL_{(p)}$ (ecuación 10.9). Debido a que esta función de aptitud requiere que cada plan sea simulado previamente, y puesto que existirán planes que producirán secuencias de tareas exhaustivas y costosas que probablemente nunca converjan, se define un criterio de paro donde aquellas simulaciones que superen los 1000 ciclos cognitivos serán detenidas. Este criterio se adopta principalmente con el objetivo de garantizar la convergencia de los planes y mejorar el desempeño del proceso de planificación, no obstante, aunque la simulación interna de tales planes será detenida no significa esto que los planes serán descartados del proceso evolutivo, ya que mantendrán una probabilidad baja de ser escogidos, mutados, y recombinados con otros planes en épocas posteriores.

Por simplicidad, y debido a que el experimento se centra únicamente en el proceso de construcción de planes de alto nivel, en la figura 12.51 se presenta solamente el árbol de expresión para el Gen Homeótico de la mejor solución encontrada por el proceso ontogenético. Como se puede observar, los nodos terminales corresponden a las diferentes tareas de recolección y apilamiento de cajas, y los nodos función corresponden a las funciones de conectividad que determinan la secuencia entre dichas tareas. A partir de esta representación genética se puede extraer el plan correspondiente como una secuencia ordenada de tareas, aplicando para ello un recorrido en postorden de los nodos terminales del árbol, como se presenta a continuación:

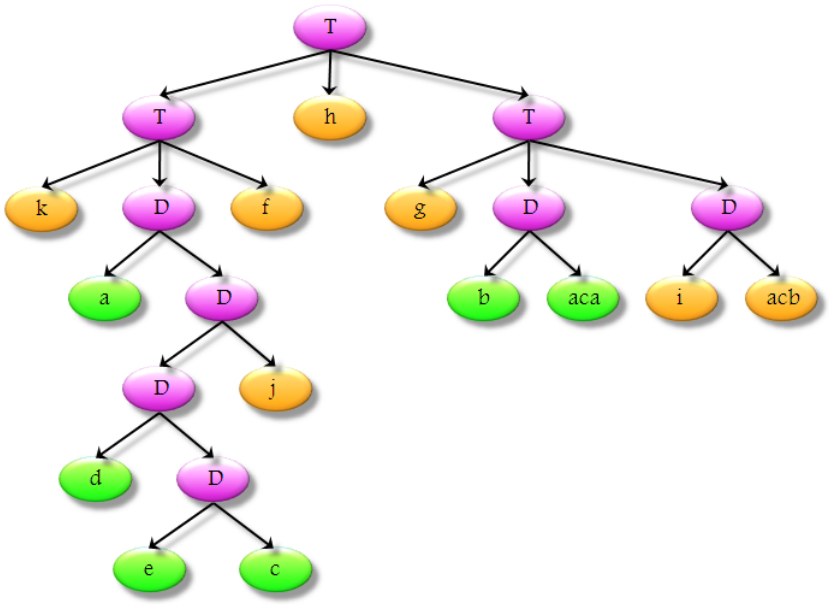


Figura 12.51: Árbol de Expresión para el Gen Homeótico descubierto por el Mejor Plan - EX10. T y D son funciones de conectividad. $\{a, b, c, d, e\}$ corresponden a las tareas de recolección de cajas de tipo A (nodos de color verde). $\{f, g, h, i, j, k\}$ corresponden a las tareas de recolección de cajas de tipo B (nodos de color naranja). "*aca*" corresponde a la tarea de Apilar Cajas del Tipo A. "*acb*" corresponde a la tarea de Apilar Cajas del Tipo B.

Plan:	[k → a → d → e → c → j → f → h → g → b → aca → i → acb]
Traducción:	
Tarea 1:	Recoger caja "k" y almacenarla en el almacén tipo B
Tarea 2:	Recoger caja "a" y almacenarla en el almacén tipo A
Tarea 3:	Recoger caja "d" y almacenarla en el almacén tipo A
Tarea 4:	Recoger caja "e" y almacenarla en el almacén tipo A
Tarea 5:	Recoger caja "c" y almacenarla en el almacén tipo A
Tarea 6:	Recoger caja "j" y almacenarla en el almacén tipo B
Tarea 7:	Recoger caja "f" y almacenarla en el almacén tipo B
Tarea 8:	Recoger caja "h" y almacenarla en el almacén tipo B
Tarea 9:	Recoger caja "g" y almacenarla en el almacén tipo B
Tarea 10:	Recoger caja "b" y almacenarla en el almacén tipo A
Tarea 11:	Apilar ordenadamente las cajas almacenadas de tipo A
Tarea 12:	Recoger caja "i" y almacenarla en el almacén tipo B
Tarea 13:	Apilar ordenadamente las cajas almacenadas de tipo B

Es importante resaltar que esta estructura corresponde a un "plan global", esto es, un plan que determina la secuencia de tareas de alto nivel que el robot debe ejecutar en el entorno, pero que no detalla las conductas y las acciones concretas puesto que éstas son determinadas por cada RC y cada *Comportamiento* activado durante la simulación y/o la ejecución de dicho plan. Por ejemplo, entre las tareas 3 y 4, el robot decide activar el *Comportamiento* ⟨`Recargar-Batería`⟩ (y ejecutar, como consecuencia, todas las acciones respectivas para llevar a acabo este comportamiento) antes de recoger la caja "e", la cual se encuentra muy cerca del recargador de batería (como se observa en la figura 12.52.d). Probablemente, en el caso de que el robot hubiese decidido recargar la batería después de almacenar la caja, habría tenido que recorrer una mayor distancia y, por tanto, ejecutar una mayor cantidad de ciclos cognitivos, un mayor gasto de energía, etc. De la misma forma, el robot decide activar el *Comportamiento* ⟨`Recargar-Batería`⟩ antes de recoger la caja "h".

Nota sobre el proceso de simulación interna: debido a que el proceso de percepción/ actuación llevado a cabo por el agente es continuo, y no discreto (e.g., el entorno "no" se representa como una rejilla discreta de elementos donde el robot, en cada ciclo cognitivo, se desplaza a una única celda de ocupación), pueden existir situaciones (por lo general contiguas en el tiempo) que son percibidas por el agente como idénticas, aunque no lo sean, y como consecuencia, el agente no puede decidir cuál es la mejor acción a ejecutar. Este problema es conocido como *"el problema de solapamiento perceptual (aliasing perceptual)"* (Luca, 2000). La condición por la cual un entorno no contiene ninguna situación como la descrita por el problema de aliasing perceptual, es denominada la *propiedad de Markov*. Los entornos que satisfacen la propiedad de Markov se denominan *entornos markovianos*, y aquellos que no, *entornos no-markovianos* (Sutton y Barto, 1998).

Con el fin de atender el problema de aliasing perceptual provocado por el entorno no-markoviano con el cual interactúa el robot, el Sistema Clasificador Anticipatorio (SA) definirá un mecanismo de aprendizaje basado en *secuencias conductuales* (behavioral sequences) como se describe en (Stolzsmann, 1999; Métivier y Lattaud, 2003). Stolzsmann define una secuencia conductual como una sucesión de acciones. Así pues, el SA definirá en la parte acción de sus clasificadores, secuencias conductuales como conjuntos de acciones que se ejecutarán secuencialmente. En (Stolzsmann, 1999; Stolzsmann y Butz, 2000) se demuestra que este procedimiento, asociado con un mecanismo de planificación de acciones, le permite a un gran número de robots, del tipo Khepera, aprender en entornos no-markovianos.

Convergencia Evolutiva: la gráfica 12.53 presenta dos curvas, que emplean las medias armónicas de los datos obtenidos en 30 ejecuciones del experimento, las cuales demuestran la convergencia del proceso ontogenético. En la parte a), se observa que el proceso ontogenético encuentra el mejor plan en la época no. 417 (en promedio). En la parte b) de la gráfica se observa cómo la duración del plan, en términos de ciclos cognitivos, es optimizada (minimizada) a través de las épocas evolutivas. Así pues, el proceso onto-

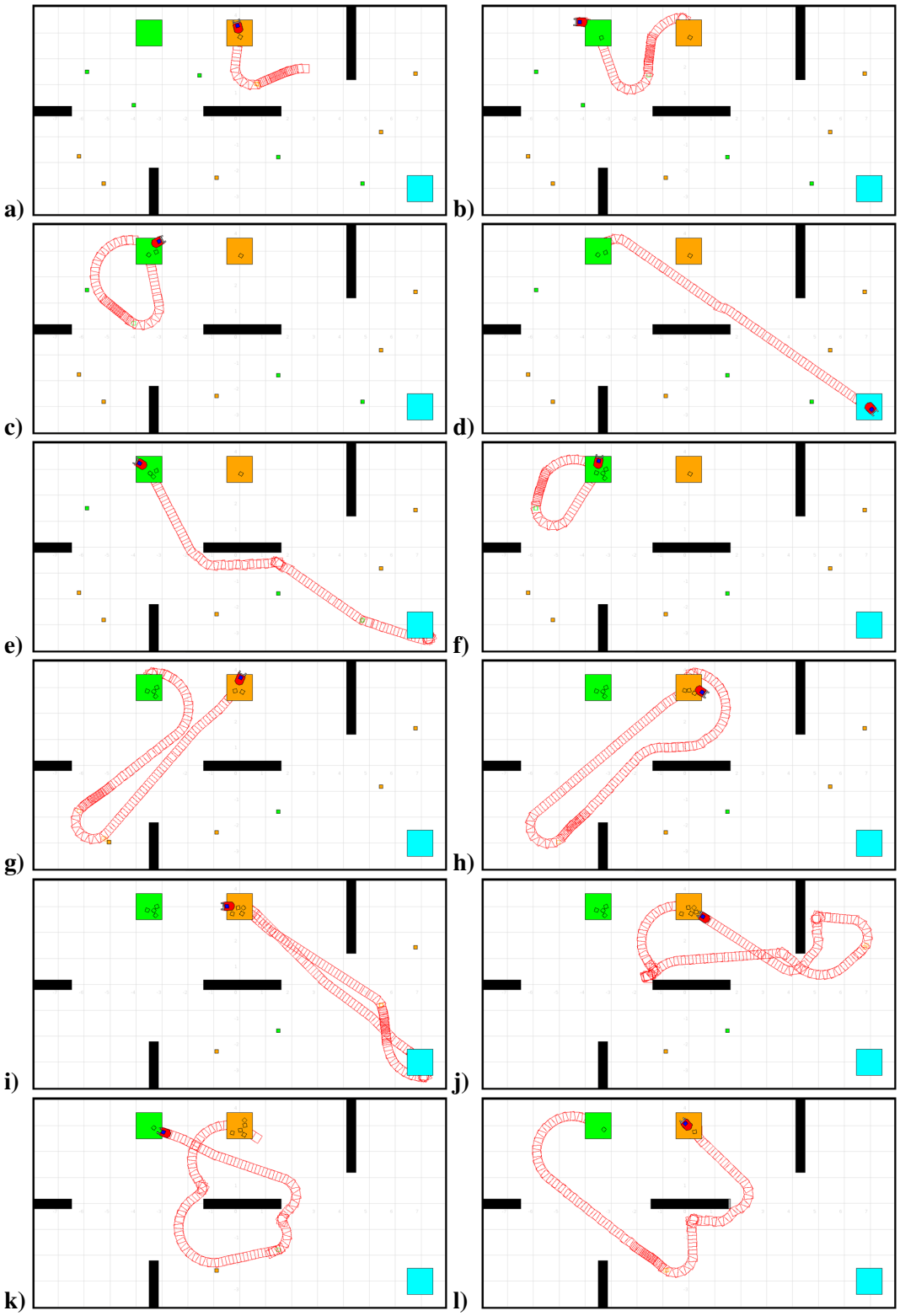
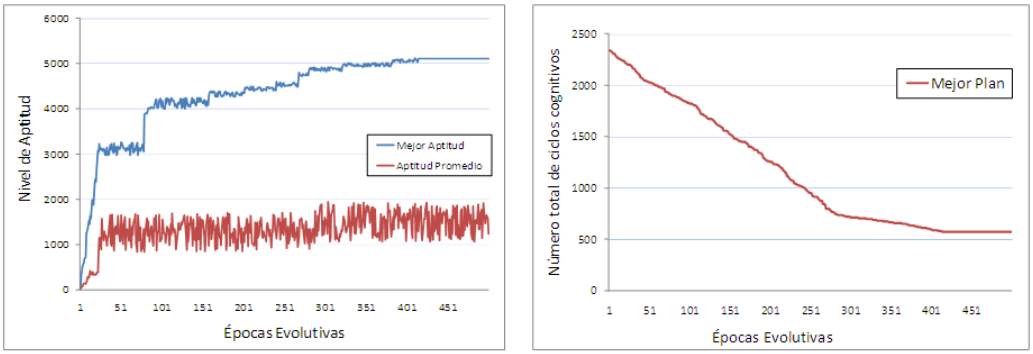


Figura 12.52: Secuencia de tareas del Mejor Plan - EX10. a) recoger caja “j”. b) recoger caja “a”. c) recoger caja “d”. d) recargar batería. e) recoger caja “e”. f) recoger caja “c”. g) recoger caja “j”. h) recoger caja “f”. i) recargar batería y recoger caja “h”. j) recoger caja “g”. k) recoger caja “b” y apilar cajas tipo A. l) recoger caja “i” y apilar cajas tipo B.

genético determina que se requieren como mínimo 576 ciclos cognitivos[17] para llevara cabo el plan.



a) b)

Figura 12.53: Convergencia Evolutiva EX10. a) Progresión del Mejor Nivel de Aptitud y de la Aptitud Promedio de la Población (en base a la función de aptitud $FAFL_p$). b) Convergencia del mejor plan hacia un mínimo de ciclos cognitivos.

Después de que el mejor plan fuese hallado por el proceso ontogenético, el robot lo ejecutó en el escenario propuesto. Es importante mencionar que no se registró una diferencia considerable entre la simulación del mejor plan y su ejecución: mientras que el plan simulado estimó 576 ciclos cognitivos para llevar a cabo todas las tareas, la ejecución del plan tomó 613 ciclos cognitivos. Principalmente, esta diferencia se debió a las acciones de evasión de obstáculos que tuvo que improvisar el robot sobre la marcha.

Conclusiones del Experimento: a partir de los resultados obtenidos se puede señalar lo siguiente:

- Queda demostrada la potencia del proceso ontogenético para producir secuencias optimizadas de tareas donde, por un lado, se maximiza la cantidad de refuerzos positivos obtenidos por los *Comportamientos* y, por otro, se minimiza la duración del plan (en términos de ciclos cognitivos) y el coste asociado a las actuaciones del robot (en términos del consumo de energía).
- Se pueden identificar múltiples niveles de planificación en la arquitectura propuesta: (1), una planificación a corto plazo, realizada por el Sistema Anticipatorio, que predice las consecuencias de las acciones en el estado inmediatamente posterior, lo que le permite al agente reaccionar oportunamente; (2), una planificación a medio plazo, modulada por las dinámicas de dispersión y acumulación de energía en las RCs, donde se persigue un conjunto de expectativas, metas, y sub-metas; y (3), una planificación a largo plazo producida por el proceso ontogenético, donde se traza una línea de actuación en forma de secuencias de tareas de alto nivel, la cual dirige el comportamiento global del agente hacia la consecución de metas más globales. Todo esto, le permite al agente demostrar un comportamiento deliberativo y enfocado hacia las metas, aunque con la capacidad de replanificar y cambiar

[17] Recuérdese que este valor corresponde a la media armónica de los resultados obtenidos.

reactivamente el curso de acción cuando las perturbaciones del estado (interno o externo) lo demandan.

- Resulta evidente la capacidad emergente del proceso ontogenético para confeccionar planes de alto nivel a partir de piezas o pautas cognitivas de más bajo nivel. En comparación con los planificadores clásicos, este proceso de planificación emergente evita el tener que definir a priori las premisas y condiciones que debe seguir el algoritmo de planificación. Es por tanto el propio sistema quien se encarga de producir sus planes, ejecutarlos mentalmente (simularlos), cotejarlos contra su ejecución en el mundo real, y lo más importante de todo, evolucionarlos en la medida que transcurre el tiempo y se recibe retroalimentación interna y del entorno. No obstante, con el fin de potenciar aún más las capacidades de planificación del agente, resultaría interesante explorar en futuras investigaciones, la posibilidad de hibridizar el mecanismo de planificación emergente con los resultados obtenidos por un planificador clásico como STRIPS (véase la discusión de la sección 10.6).

12.4.7. Experimento EX11: Análisis de Tiempos de Respuesta para el Proceso de Consciencia Sintética

12.4.7.1. Descripción

Este experimento tiene como objetivo analizar el desempeño del sistema en términos de los tiempos de respuesta del módulo de Orquestación de la Consciencia. A continuación se describen los pasos que se siguieron para llevar a cabo el experimento:

1. El robot se sitúa en el escenario descrito para el experimento EX10. Sin embargo, se desactiva el mecanismo de planificación puesto que solamente se analizarán los tiempos críticos de respuesta del agente en la escala de la percepción y la actuación.
2. En cada ciclo cognitivo, el robot percibe la información ambiental y construye el estado interno en la MT. Seguidamente, se activará una sola época evolutiva después de la cual se escogerá la MAT más apropiada para dirigir el foco de atención del robot. Luego, se ejecutan las dinámicas de dispersión, y como resultado se activará un *Comportamiento*. Por último, las salidas de los componentes del *Comportamiento* son integradas, y se envía la acción integrada a los actuadores correspondientes.
3. El paso anterior se repite por completo durante 200 ciclos cognitivos, durante los cuales se registran los tiempos de respuesta del sistema. El experimento se repite 30 veces y se estiman las medias armónicas para cada ciclo cognitivo. Se analizan los resultados.

12.4.7.2. Análisis de Resultados

En la gráfica 12.54 se presentan los TRs obtenidos durante el experimento. Como se puede observar, los TRs para el módulo de Orquestación de la Consciencia se mantienen en un rango entre los 186 y los 328 ms aproximadamente. Las fluctuaciones en la gráfica

se deben principalmente a las diversas dinámicas que ejecuta la arquitectura, haciéndose evidente, por un lado, que aquellos ciclos cognitivos más deliberativos requieren de un mayor tiempo de procesamiento y, por otro, que aquellos ciclos cognitivos más reactivos consumen menor cantidad de recursos y tiempo. También cabe destacar que las dinámicas evolutivas llevadas a cabo por los procesos epigenético y ontogenético influyen significativamente, cada cierto intervalo de ciclos cognitivos, sobre el tiempo de procesamiento. En la tabla 12.32 se presenta el resumen de las estadísticas obtenidas.

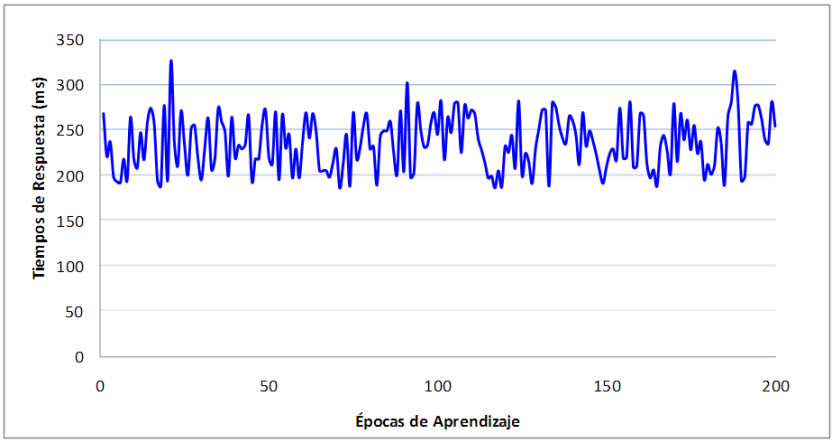


Figura 12.54: Tiempos de Respuesta del módulo de Orquestación de la Consciencia.

Estadística	Valor Obtenido
Mínimo (ms)	186.3
Máximo (ms)	327.8
Media ($\bar{x}$) (ms)	231.36
Desviación Estándar muestral(σ)	29.95
Varianza muestral (σ^2)	897.04

Tabla 12.32: Estadísticas para los Tiempos de Respuesta - EX10

Cabe mencionar que, para este experimento, los TRs obtenidos durante la simulación no pudieron ser cotejados contra los TRs producidos por el robot real debido a ciertas dificultades técnicas (e.g., los robots no estaban equipados con pinzas, entre otras). No obstante, se estima que los resultados obtenidos en un entorno real no estén muy alejados de los resultados de la simulación.

Como conclusión final del experimento, se puede decir que los tiempos de respuesta del agente siguen manteniéndose dentro de un rango aceptable en comparación con los fundamentos cognitivos discutidos en la sección 10.5.

12.5. Experimentos para el Nivel Filogenético

Los experimentos presentados en esta sección se centran en la demostración del incremento en la velocidad de convergencia en el aprendizaje y la adaptación, cuando se integran los procesos EOF y se activa el mecanismo de transmisión de memes (descrito en la sección 10.4).

Cabe mencionar que, al igual que los experimentos para el nivel ontogenético, en esta sección no se llevarán a cabo pruebas no paramétricas para comparar el desempeño de múltiples algoritmos. La razón para esto es que la comparación múltiple implicaría una exhaustiva implementación y adaptación de los algoritmos de prueba en los escenarios propuestos, tarea que no se encuentra dentro del alcance de la presente tesis, aunque sí en el de un trabajo futuro a corto plazo. Principalmente, sería interesante realizar un análisis comparativo del algoritmo memético propuesto contra otras aproximaciones que igualmente hibridizan la *búsqueda local* y la *optimización global*, tales como: el algoritmo de *Recocido Simulado Adaptativo* (Adaptive Simulated Annealing) (Ingber, 1993), el *Algoritmo SALO* (Simulated Annealing and Local Optimization) (Desai y Patil, 1996), la *Búsqueda Tabú Adaptativa* (Hageman et al., 2003), la *Optimización por Colonias de Hormigas* basada en algoritmos meméticos (Duan, 2007), entre otros.

12.5.1. Experimento EX12: Aceleración de la Convergencia en el Aprendizaje de Reglas

12.5.1.1. Descripción

Este experimento tiene como propósito evaluar la aceleración en el aprendizaje de reglas del *Comportamiento* ⟨`Evadir-Obstáculos`⟩, producida por la integración de los procesos epigenéticos, ontogenéticos, y filogenéticos. Para esto, se define el escenario de la figura 12.55, donde están situados cuatro robots.

Los procesos epigenético y ontogenético son activados cada 20 ciclos cognitivos, mientras que el proceso filogenético es activado cada 100 ciclos cognitivos, esto con el fin de que los robots tengan suficiente tiempo para evaluar las reglas adaptadas antes de que se produzca una nueva época evolutiva (filogenética). La configuración paramétrica utilizada para el proceso filogenético se presenta en la tabla 12.33.

12.5.1.2. Análisis de Resultados

En la gráfica 12.56 se presentan las curvas de aprendizaje de reglas para diferentes configuraciones del experimento. Como se puede observar, en la medida que es incrementado el número de robots que interactúan concurrentemente en la simulación, el proceso de aprendizaje converge más rápidamente debido a la transmisión de conocimiento entre los mismos.

En la tabla 12.34 se presentan las estadísticas obtenidas para cada una de las configuraciones del experimento. Cabe recordar que, al igual que para el experimento EX03

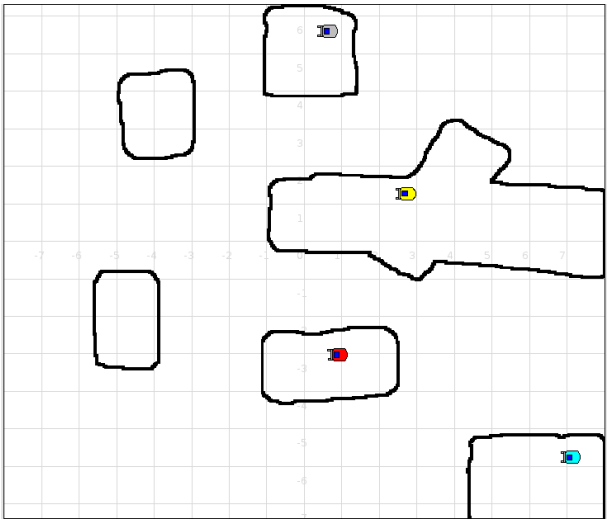


Figura 12.55: Escenario para el experimento EX12.

Parámetro	Valor
Número de Experimentos	30
Número de Épocas Evolutivas	50
Número de ciclos cognitivos por época	100
Probabilidad de Mutación	0.5
Prob. de Recombinación en uno y dos puntos	0.3
Prob. de Recombinación en uno y dos puntos	0.3
Prob. de Recombinación Genética	0.4
Prob. de Transposición de SI	0.35
Prob. de Transposición Genética	0.5

Tabla 12.33: Configuración Paramétrica del Proceso Filogenético - EX12

(sección 12.3.3), se considera que el aprendizaje converge cuando se alcanza un error mínimo del 5 %, esto es, cuando por cada época de 100 ciclos cognitivos la retroalimentación obtenida por el *Comportamiento* no supera un máximo de 5 refuerzos negativos.

Interpretación de Resultados y Conclusiones del Experimento: a partir de los resultados obtenidos se puede argumentar que:

- Queda demostrado que la integración de los procesos epigenéticos, ontogenéticos, y filogenéticos acelera la convergencia del aprendizaje de memes del tipo regla en función del número de robots (agentes) empleados en la simulación. Concretamente, se evidencia una mejora en la convergencia de hasta casi el 77 % cuando se emplean cuatro robots.

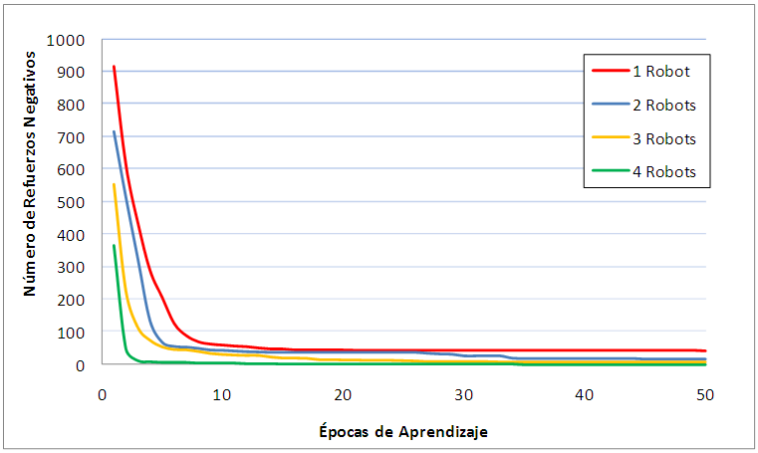


Figura 12.56: Curvas de aprendizaje empleando 1, 2, 3, y 4 robots.

Estadística	1 Robot (A)	2 Robots (B1)	3 Robots (B2)	4 Robots (B3)
Época de Convergencia (EC)	13	8	6	3
Tasa de Convergencia $[1-(EC_{B_i}/EC_A)]\,\%$	–	38.46 %	53.84 %	76.92 %
Mínimo de Refuerzos Negativos	42	15	10	0
Error Cuadrático Medio (ECM)	18.06	15.46	12.79	10.56
Tasa de ECM $[1-(ECM_{B_i}/ECM_A)]\,\%$	–	14.40 %	29.18 %	41.53 %

Tabla 12.34: Estadísticas de las Curvas de Aprendizaje - EX12

- Los resultados sugieren que, bajo un enfoque multi-agente, el proceso filogenético no sólo acelera la convergencia del aprendizaje, sino que garantiza además dicha convergencia hacia óptimos globales. Por ejemplo, en el escenario donde se emplean cuatro robots se logra un mínimo de cero refuerzos negativos (0 % de error) después de la época evolutiva no. 35 (en promedio).
- Como consecuencia de lo anterior, el error cuadrático medio respecto a la cantidad de refuerzos negativos obtenidos por cada agente, es reducido significativamente (en un 41.53 %) en las simulaciones que emplean un enfoque multi-robot.

12.5.2. Experimento EX13: Aceleración de la Convergencia en el Diseño de Redes de Comportamientos

12.5.2.1. Descripción

Este experimento tiene como propósito evaluar la aceleración en la convergencia adaptativa de las Redes de Comportamientos, producida por la integración de los procesos epigenéticos, ontogenéticos, y filogenéticos. Para esto, se define un escenario de recolección de cajas similar al descrito en el experimento EX05, con la diferencia de que esta vez se activa el mecanismo de transmisión de memes del tipo RC para 2, 3, y 4 robots. En

cuanto al proceso filogenético, se emplea la misma configuración paramétrica propuesta para el experimento EX12.

El procedimiento llevado a cabo para el experimento se describe a continuación:

- En cada ciclo cognitivo, se suman los resultados de evaluar la función de aptitud para el aspecto estructural – FAAE – y la función de aptitud para el aspecto funcional – FAAF – (ecuaciones 9.25 y 9.35) de la mejor RC seleccionada por cada agente. Se registra el promedio de los resultados obtenidos por cada uno de los robots activos.
- Al finalizar cada época filogenética (conformada por 100 ciclos cognitivos), se estima la media armónica de los resultados obtenidos.
- El procedimiento continúa durante 50 épocas evolutivas. Luego, se representan gráficamente las curvas de tendencia.
- Adicionalmente, en cada época evolutiva, se registra el error absoluto como la diferencia entre el nivel de aptitud de la RC activa y el máximo nivel de aptitud alcanzado cuando el sistema converge.

12.5.2.2. Análisis de Resultados

En la gráfica 12.57 se observan cuatro robots llevando a cabo la tarea de recolección de cajas. Cabe destacar que cada robot actúa de manera autónoma persiguiendo sus propias metas en consonancia con la información sensorial percibida y con su estado interno (expectativas, planes, motivadores, etc.). No obstante, y a pesar de la transmisión de conocimiento efectuada entre los agentes, éstos no coordinan ni delegan la ejecución de tareas, motivo por el cual ocurren situaciones donde los robots persiguen la misma meta simultáneamente (como se observa en la circunferencia de la figura). Sin embargo, vale la pena aclarar que este aspecto está fuera del alcance de la presente tesis, aunque es un tema que se pretende abordar en trabajos futuros.

En la gráfica 12.58 se puede apreciar la convergencia del proceso filogenético en los distintos escenarios con 1, 2, 3, y 4 robots. Obsérvese que el proceso filogenético acelera la convergencia hacia un óptimo global en la medida que se emplea una mayor cantidad de robots. En la tabla 12.35 se presentan las estadísticas obtenidas para cada una de las configuraciones del experimento.

Estadística	1 Robot (A)	2 Robots (B1)	3 Robots (B2)	4 Robots (B3)
Época de Convergencia (EC)	27	14	10	7
Tasa de Convergencia $[1 - (EC_{B_i}/EC_A)]\,\%$	–	48.14 %	62.96 %	74.07 %
Máximo Nivel de Aptitud	683	692	694	702
Error Cuadrático Medio (ECM)	215.71	169.33	122.43	104.73
Tasa de ECM $[1 - (ECM_{B_i}/ECM_A)]\,\%$	–	21.50 %	43.24 %	54.44 %

Tabla 12.35: Estadísticas de las Curvas de Aprendizaje - EX13

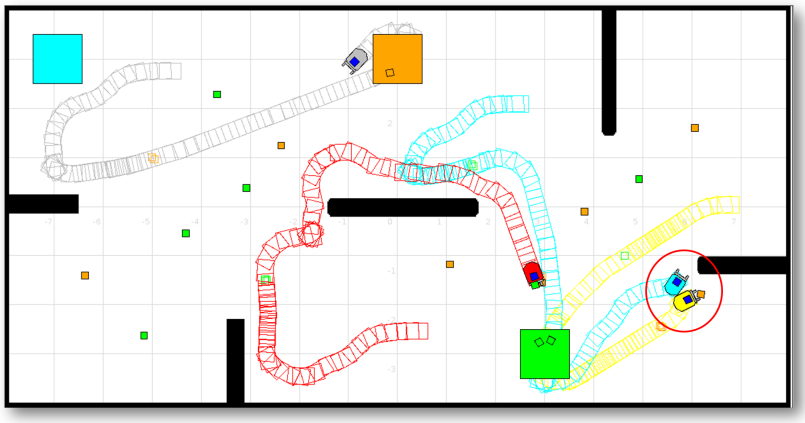


Figura 12.57: Escenario Multi-robot EX13. Cada robot ejecuta concurrentemente la tarea de recolección de cajas, pero debido a que no existe una coordinación global de tareas, los robots pueden intentar recoger la misma caja simultáneamente.

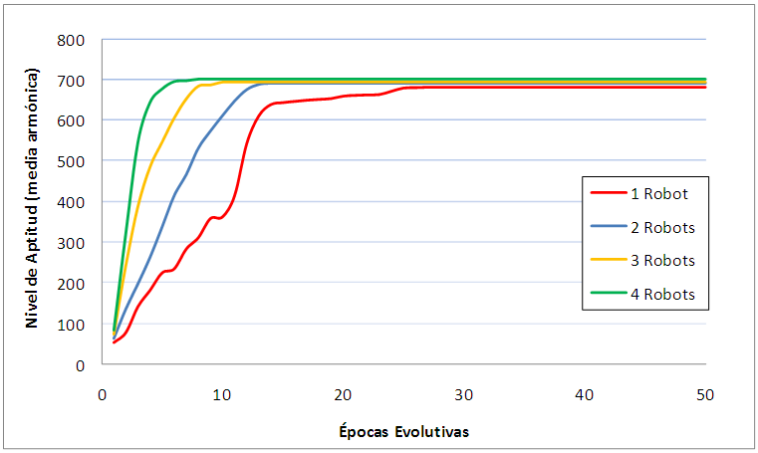


Figura 12.58: Curvas de Tendencia para la Optimización de RCs empleando 1, 2, 3, y 4 robots.

Interpretación de Resultados y Conclusiones del Experimento: a partir de los resultados obtenidos se puede argumentar que:

- Queda demostrado que la integración de los procesos epigenético, ontogenético, y filogenético acelera la convergencia en el diseño adaptativo de memes del tipo RC, en función del número de robots (agentes) empleados en la simulación. Concretamente, se evidencia una mejora en la convergencia de hasta el 74 % cuando se emplean cuatro robots.
- Los resultados sugieren que el proceso filogenético evita que el sistema caiga en óptimos locales. Por ejemplo, cuando se emplea un solo robot, el máximo nivel de aptitud obtenido es de 683 unidades de refuerzo positivo, mientras que cuando se emplean cuatro robots, el máximo nivel de aptitud asciende a 702 unidades.

- Por último, el error cuadrático medio es reducido significativamente (hasta un 54.44 %) en las simulaciones que emplean el mecanismo de transmisión de memes.

12.5.3. Experimento EX14: Aceleración de la Convergencia en la Confección de Planes

12.5.3.1. Descripción

Este experimento tiene como propósito evaluar la aceleración en la convergencia adaptativa de los memes del tipo Plan, producida por la integración de los procesos epigenéticos, ontogenéticos, y filogenéticos. Para esto, se define un escenario donde los robots tendrán que recoger y apilar cajas, similar al descrito en el experimento EX10 (sección 12.4.6), con la diferencia de que esta vez se activa el mecanismo de transmisión de memes del tipo RC para 2, 3, y 4 robots. En cuanto al proceso filogenético, se emplea la misma configuración paramétrica propuesta para el experimento EX12 (sección 12.5.1).

El procedimiento llevado a cabo para el experimento se describe a continuación:

- En cada ciclo cognitivo, se evalúa la función de aptitud FAFL (ecuación 10.9) del mejor plan encontrado por cada agente. Se registra el promedio de los resultados obtenidos por cada uno de los robots activos.
- Al finalizar cada época filogenética (conformada por 100 ciclos cognitivos), se estima la media armónica de los resultados registrados.
- El procedimiento continúa durante 50 épocas evolutivas. Luego, se representan gráficamente las curvas de tendencia.
- Adicionalmente, en cada época evolutiva, se registra el error absoluto como la diferencia entre el nivel de aptitud del mejor plan y el máximo nivel de aptitud alcanzado cuando el sistema converge.

12.5.3.2. Análisis de Resultados

En la gráfica 12.59 se presentan las curvas de tendencia para la optimización de planes. Una vez más se puede observar que el proceso filogenético acelera la convergencia hacia un óptimo global cuando está activo el mecanismo de transmisión de memes. En la tabla 12.36 se presentan las estadísticas obtenidas para cada una de las configuraciones del experimento.

Interpretación de Resultados y Conclusiones del Experimento: a partir de los resultados obtenidos se puede argumentar que:

- Queda demostrado que la integración de los procesos epigenético, ontogenético, y filogenético acelera la convergencia en el diseño adaptativo de memes del tipo Plan, en función del número de robots (agentes) empleados en la simulación.

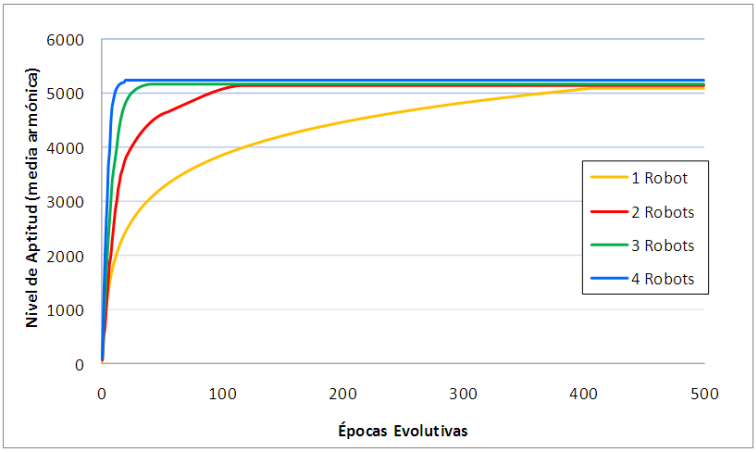


Figura 12.59: Curvas de Tendencia para la Optimización de Planes empleando 1, 2, 3, y 4 robots.

Estadística	1 Robot (A)	2 Robots (B1)	3 Robots (B2)	4 Robots (B3)
Época de Convergencia (EC)	407	116	42	21
Tasa de Convergencia $[1-(EC_{B_i}/EC_A)]\,\%$	–	71,49 %	89,68 %	94,84 %
Máximo Nivel de Aptitud	5.080	5.134	5.171	5.247
Duración Mínima del mejor Plan (ciclos cog.)	576	568	563	554
Error Cuadrático Medio (ECM)	1.098,72	649,70	489,09	380,54
Tasa de ECM $[1-(ECM_{B_i}/ECM_A)]\,\%$	–	40,86 %	55,48 %	65,36 %

Tabla 12.36: Estadísticas de las Curvas de Aprendizaje - EX14

Concretamente, se evidencia una mejora en la convergencia de hasta casi un 95 % cuando se emplean cuatro robots.

- Los resultados sugieren que el proceso filogenético encuentra óptimos globales (o soluciones cercanas) cuando se encuentra activo el mecanismo de transmisión de memes. Por ejemplo, en el escenario donde se emplearon cuatro robots, el mejor plan registró el máximo nivel de aptitud y la mínima duración (en términos de ciclos cognitivos) en comparación con los escenarios con 1, 2, y 3 robots.
- Por último, la integración de los procesos EOF reduce significativamente el error cuadrático medio de las observaciones (hasta un 65.36 %).

Capítulo 13

Resultados y Conclusiones

"La vida es el arte de sacar conclusiones suficientes a partir de datos insuficientes".
– Samuel Butler.

Este capítulo se divide en tres secciones principales. La primera de ellas presenta los resultados más significativos obtenidos a partir del estudio comparativo (benchmarking) donde se evaluaron las características funcionales de METÁFORA contra las de otras 30 arquitecturas cognitivas, según los criterios expuestos en el anexo A. En la segunda sección, se describe el grado con que METÁFORA satisface los requisitos que una arquitectura cognitiva debería atender desde la perspectiva de aplicabilidad en el desarrollo de sistemas de control complejo cognitivo. Por último, en la tercera sección, se presentan las contribuciones más significativas de la investigación.

13.1. Resultados del Estudio Comparativo (Benchmarking)

A continuación, se sintetizan los resultados más relevantes obtenidos a partir del estudio comparativo del anexo A:

En relación a las Características Arquitectónicas Generales:

- METÁFORA, en comparación con el resto de arquitecturas en general, demuestra un diseño funcional y estructural con un enfoque fuertemente *Adaptativo*. A pesar de que algunas de las arquitecturas analizadas del paradigma emergente e híbrido contemplan mecanismos adaptativos que les permite ajustar y refinar sus estructuras cognitivas, éstos actúan solamente sobre un único nivel de organización (e.g., las unidades de conocimiento procedimental, o los parámetros del sistema, etc.). Por el contrario, los mecanismos adaptativos de METÁFORA operan simultáneamente sobre múltiples niveles de organización y en diferentes escalas temporales.

- Comparativamente, el alto nivel de *adaptabilidad* de METÁFORA no sólo está relacionado con la inclusión de las máquinas darwinianas (que por definición, son algoritmos evolutivos), sino que, además, se debe a otros factores importantes tales como: (1) el *aprendizaje*, que a través de la modulación de las señales de retroalimentación favorece y acelera la *adaptación* (e.g., el aprendizaje procedimental implícito y explícito, el aprendizaje declarativo, el aprendizaje anticipatorio, etc.); (2) las dinámicas de dispersión en las RCs, las cuales orientan *adaptativamente* el foco de atención del agente mediante el balanceo de la energía de activación entre los *Comportamientos*; (3) la acumulación y/o disminución de los niveles de activación de ciertos componentes de la arquitectura (e.g., la activación de nivel base – ANB – de los motivadores, las metas, los chunks semánticos y episódicos, etc.), los cuales varían *adaptativamente* en función del estado percibido (interno y externo); y (4) el proceso de transmisión y evolución de estructuras cognitivas (memes) encargado de acelerar el proceso *adaptativo* en toda la población de agentes.
- Como consecuencia de los factores antes mencionados, METÁFORA exhibe un comportamiento adaptativo al que debe enteramente su *autonomía*, o en otras palabras, la autonomía de METÁFORA es autentica debido a que la conducta del agente no obedece a una configuración predeterminada de los componentes, sino que es el propio agente quien se encarga de diseñar adaptativamente dicha configuración durante el transcurso de su vida[1].

En relación al Tipo de Arquitectura:

- En comparación con las aproximaciones emergentes e híbridas analizadas, METÁFORA es una arquitectura con un diseño fuertemente orientado hacia la característica de *Planificar-Luego-Compilar*. Esto se evidencia a partir de lo siguiente:

  - La planificación en METÁFORA es un proceso emergente y acumulativo: los planes se crean a partir del ensamblaje de estructuras cognitivas más simples, las cuales se depuran y perfeccionan mediante un proceso co-evolutivo. Así pues, el éxito de un plan "no" radica en su capacidad para descomponer las tareas de alto nivel y realizar una búsqueda exhaustiva de las acciones concretas mediante el método de prueba y error (lo que conduce a un problema de explosión combinatorial), ni tampoco en su capacidad de inferir una secuencia de acciones a partir de un conjunto *"predeterminado"* de semánticas formales (e.g., operadores, restricciones, etc.), que es lo que suelen hacer los planificadores clásicos; sino que, por el contrario, el éxito de un plan en METÁFORA depende de la construcción en paralelo de múltiples pautas cognitivas con niveles incrementales de abstracción (e.g., una regla $\rightarrow$ un *Comportamiento* $\rightarrow$ una RC $\rightarrow$ un Plan ...), donde por cada nivel de abstracción, el proceso evolutivo (mediante la selección y la replicación) garantiza la supervivencia de aquellas pautas mejor adaptadas.
  - A diferencia de las aproximaciones puramente emergentes, METÁFORA define un mecanismo de traducción que convierte las reglas sub-simbólicas en

[1] Tomando en cuenta el ciclo de vida de un agente descrito en la sección 10.4.3

reglas simbólicas, lo que permite por un lado, aprovechar toda la plasticidad y robustez del nivel emergente y, por otro, obtener unidades de conocimiento explícitas a partir de las cuales se pueden generar planes comunicables y explicables a otros agentes, e incluso a humanos.

- En METÁFORA, la construcción inteligente de planes no sólo es el resultado de un proceso evolutivo a partir del cual emergen secuencias complejas de actuaciones, sino que además es un proceso que se refina y ajusta continuamente mediante la simulación interna o la ejecución real del plan. Así pues, la precisión de cada plan puede mejorar en la medida que: (1) avanza el procesos evolutivo de las estructuras cognitivas involucradas (en todos los niveles de organización), (2) aumenta la cantidad de conocimiento predictivo que es depurado mediante el aprendizaje anticipatorio; y (3) incrementa la cantidad de información histórica fiable almacenada en las memorias episódica, semántica, y procedimental.
- A diferencia de la gran mayoría de arquitecturas clásicas de enfoque cognitivista, en METÁFORA, un plan no determina una secuencia específica de acciones que deben ejecutarse rigurosamente, sino que propone una sucesión tentativa de pasos, cuya eficiencia es ensayada previamente mediante la simulación interna, y la cual es perfeccionada y corregida sobre la marcha, es decir, que el agente tiene la capacidad para improvisar inteligentemente durante la ejecución del plan en el mundo real. Una vez se ha cotejado la eficiencia del plan en el mundo real, los planes son almacenados como "*rutinas reactivas*" que pueden ser reutilizadas por el agente en situaciones futuras y que no requieren nuevamente de un procesamiento costoso.

- METÁFORA, a diferencia de la mayoría de arquitecturas analizadas, posee la característica de ser *Interrumpible* y *Reanudable*. Esto quiere decir que METÁFORA es capaz de procesar continuamente los eventos que se producen en el entorno, determinar su relevancia, interrumpir los procesos cognitivos actuales cuando los eventos lo ameritan, y reanudar luego las tareas y metas que se estaban llevando a cabo antes de la interrupción. Estas características son el resultado de la orques-tación de los procesos conscientes y subconscientes realizada por el agente en tiempo de ejecución.
- Por último, en comparación principalmente con las aproximaciones cognitivistas y emergentes, METÁFORA define un sistema de control del tipo *Pizarra*, a través del cual son accesibles por todos los módulos de la arquitectura, los contenidos conscientes y subconscientes. Esta pizarra mental o área de trabajo global corresponde a la Memoria de Trabajo.

En relación a las Propiedades del Ambiente Percibido:

- METÁFORA demuestra una capacidad idónea para procesar no sólo información discreta sino también información de naturaleza continua, con lo cual pueden construirse concreciones contextuales tanto para ambientes *estáticos* como *dinámicos*.
- METÁFORA posee cierta versatilidad para operar tanto en entornos *reales* como *simulados*, debido a que dispone de mecanismos que filtran la inmensa cantidad de

información percibida y centran la atención del agente. Esto implica que, en entornos altamente realistas, el agente cognitivo podrá: (1) filtrar la cantidad de información a la que se puede atender simultáneamente de forma variable, dependiendo de la complejidad de la tarea y de la organización de la información percibida; (2) enfocar selectivamente su atención sólo sobre aquellos estímulos importantes e ignorar el resto de información irrelevante; y (3) re-orientar la atención de forma apropiada hacia otro subconjunto de estímulos cuando la situación lo requiera.

- En cuanto a los mecanismos para tratar el *Conocimiento Imperfecto* sobre el ambiente, los principales aportes realizados son:
  - La arquitectura propuesta re-evalúa constantemente el conocimiento previamente adquirido, mediante: (1) las dinámicas auto-organizativas llevadas a cabo en el nivel emergente; y (2) los procesos de comprobación y verificación efectuados en el nivel cognitivista (e.g., la utilidad de las reglas y los chunks, la generalización y especialización de reglas procedimentales, el ajuste de los niveles de positividad y negatividad, etc.)
  - METÁFORA mantiene la consistencia y validez de los modelos internos a través de la modulación de las señales de retroalimentación.
  - En el módulo Comportamental, los mecanismos de integración de salidas compensan la acción errónea (o conocimiento imperfecto) que pueda haber construido un determinado componente, con la "resonancia" de las acciones correctas formadas por el resto de componentes.
- En comparación con la mayoría de aproximaciones híbridas, METÁFORA distingue claramente los diferentes modos conductuales que puede exhibir el agente en relación a los *Niveles de Importancia* correspondientes a los cambios ambientales percibidos. Esto significa que METÁFORA tiene la capacidad de producir respuestas que van desde lo puramente reactivo hasta soluciones que involucran la re-planificación parcial o total de las acciones futuras.
- Respecto a las *Limitaciones Computacionales*, se puede argumentar lo siguiente:
  - *Memoria*: METÁFORA define repertorios o espacios de memoria con tamaños fijos, donde las estructuras cognitivas se auto-organizan internamente mediante procesos de *aprendizaje* y *olvido*. De esta manera, el agente puede almacenar nuevas experiencias y liberar el espacio ocupado por conocimiento obsoleto.
  - *Módulos Desconectables*: debido a que METÁFORA no define una arquitectura fuertemente acoplada, es posible habilitar o deshabilitar algunos de sus módulos con el fin de reducir el consumo de recursos. Por ejemplo, si se define a priori que el propósito de un *Comportamiento* es puramente reactivo, es posible deshabilitar los componentes del nivel cognitivista que suponen un mayor gasto de tiempo de procesamiento. De la misma forma, si no se requiere que el agente construya planes deliberativos a largo plazo, puede desconectarse el mecanismo de planificación del módulo Intencional, etc.
  - *Paralelismo*: el tiempo de aprendizaje de las estructuras cognitivas es reducido significativamente debido al enfoque multi-agente empleado. Es decir, cada agente no tiene que aprender por sí solo todas las conductas esperadas,

sino que, por el contrario, cada agente realiza solamente un aprendizaje local y parcial, que luego es compartido entre todos los agentes activos. Por tanto, se combinan las fortalezas de la *Búsqueda Local* con las fortalezas de la *Optimización Global*, función que es desempeñada por el Algoritmo Memético.

- Por último, debido a la cualidad de *Interrumpible* de METÁFORA, es posible suspender temporalmente la ejecución de los procesos cognitivos para dar prioridad al procesamiento de nuevos eventos ambientales, motivo por el cual el proceso llevado a cabo en cada ciclo cognitivo se enmarca en tiempos de respuesta aceptables.

En relación a las Capacidades del Agente para producir Comportamientos Inteligentes:

- Pese a que la aproximación arquitectónica propuesta define un modelo simplificado para el procesamiento perceptual, el cual omite aspectos considerados en investigaciones más especializadas (e.g., la representación de información espacial, el procesamiento de imágenes, la memoria sensorio/perceptual, la calibración de los sensores, la fusión de datos sensoriales, etc.), se puede argumentar a su favor que integra múltiples características como la *percepción entusiasta*, la *monitorización de acciones*, y la *percepción múltiple*.

  Estas características le permiten al agente supervisar su actuación, y actualizar el modelo interno del mundo como consecuencia, conservando siempre la eficiencia en el desempeño del procesamiento perceptual. Esto se logra principalmente, como resultado del "desacoplamiento" del módulo de Percepción del resto de módulos de procesamiento cognitivo (Comportamental, Intencional, Declarativo, etc.), es decir, que el módulo de Percepción puede definir una cantidad variable de procesadores independientes, que operan en paralelo y asíncronamente (e.g., cada uno de los dispositivos del robot: láser, sonar, fiducial, etc.), los cuales "no" envían sus datos de forma directa a cada componente de procesamiento cognitivo (lo cual ocasionaría una dependencia entre el componente y el dispositivo de percepción correspondiente); sino que, en su lugar, se propone un módulo intermedio, la Memoria de Trabajo, donde son construidos los perceptos, quienes tienen un tiempo de validez asociado con la cantidad de activación acumulada recibida desde los dispositivos perceptuales.

  Así pues, si en un momento determinado un dispositivo tarda en capturar la información del ambiente, los procesos cognitivos internos no se detienen hasta que esta información esté disponible, sino que utilizan en su lugar la información remanente de ciclos cognitivos anteriores, esto es, los perceptos que aún están vigentes en la Memoria de Trabajo (similar a como funciona nuestro cerebro). Se puede decir entonces que, debido a esto, los módulos cognitivos de METÁFORA procesan la información ambiental de forma continua en vez de procesarla discretamente cada cierto intervalo de tiempo o cada vez que se sincronicen los dispositivos perceptuales.

- Al igual que para el proceso de percepción, METÁFORA propone un modelo simplificado para el procesamiento de la actuación del agente. No obstante, se puede

argumentar que METÁFORA cumple con las características de *reactividad*, *reacción de emergencia*, *acciones múltiples*, y *tratamiento de la incertidumbre en la actuación*. Principalmente, se puede destacar la capacidad de METÁFORA para procesar simultáneamente múltiples acciones de diferente naturaleza (e.g., acciones reactivas, actos reflejos, acciones deliberativas, etc.). Esta capacidad se logra gracias a que METÁFORA puede procesar múltiples *Comportamientos* concurrentemente (de manera consciente o subconsciente), y que cada *Comportamiento* a su vez puede controlar un sub-conjunto de actuadores diferentes. La única restricción de este mecanismo es que cuando existe un conflicto entre *Comportamientos* activos que empleen los mismos actuadores, tendrá prioridad de ejecución el *Comportamiento* que opera de forma consciente, invalidando el resto de señales conflictivas que envían a los actuadores los *Comportamientos* subconscientes.

- Respecto a la capacidad de planificación, los aportes principales que realiza METÁFORA son:

  - *Algoritmo en Cualquier Momento*: debido a que la generación de planes está determinada por un proceso darwiniano (ontogenético), es posible que en algunos casos la obtención de un plan optimizado requiera de una cantidad considerable de tiempo. Sin embargo, para mitigar los efectos de una planificación costosa, METÁFORA puede definir un intervalo de tiempo o un número de épocas evolutivas específico después del cual será interrumpida la planificación a fin de dar paso a otros procesos cognitivos. Este proceso continua hasta que se satisfaga una condición de paro o se encuentre el plan deseado. La principal ventaja de esto es que se evitan las sobrecargas en el procesamiento, sin perjudicar la eficiencia de la planificación, puesto que des-pués de cada interrupción se obtiene un plan sub-óptimo, esto es, la mejor aproximación al plan óptimo. En la medida que pasan los intervalos de planificación, el sistema es capaz de re-planificar y mejorar cada vez más el plan encontrado previamente.
  - *Re-planificación*: a pesar de que el proceso ontogenético genera una secuencia específica de tareas de alto nivel, el sistema es capaz de re-planificar en tiempo de ejecución gracias a las dinámicas de re-orientación del foco de atención que modulan las Redes de Comportamientos.
  - *Simulación*: el proceso de simulación permite que la construcción de planes sea más realista (puesto que considera los tiempos de respuesta que podría tomar la ejecución del plan en el entorno real) y preventiva respecto a los posibles cambios ambientales.

- En relación al Aprendizaje, se puede argumentar lo siguiente:

  - *Problema de Utilidad*: METÁFORA atiende el problema de utilidad mediante la definición de *mecanismos de selección de reglas* (procedimentales, semánticas, episódicas, etc.). Estos mecanismos mejoran la eficiencia del sistema puesto que acotan el espacio de búsqueda en cada ciclo cognitivo. Por otro lado, los *mecanismos de generalización de conocimiento* son otra forma de minimizar el problema de utilidad, por ejemplo, una regla generalizada abstrae las características comunes de un sub-conjunto de reglas, las cuales pue-

den delegar su responsabilidad en dicha regla generalizada y luego ser desactivadas (o eliminadas).

- *Soporte a la Inconsistencia del Conocimiento*: la arquitectura propuesta puede aprender conocimiento que contradice lo previamente aprendido. Esto implica que las unidades de conocimiento (en todos los niveles organizativos de la arquitectura) no son definitivas, sino que deben estar continuamente compitiendo contra otras unidades a fin de sobrevivir y mantenerse persistentes.
- *Aprendizaje Deliberativo:* METÁFORA define meta-estrategias que determinan cuándo, cómo, y qué se debe aprender u olvidar. En el nivel emergente, los procesos auto-organizativos, mediante la modulación de las señales de retroalimentación, se encargan de esta función (e.g., el reajuste de pesos sinápticos en las redes neuronales, o las dinámicas de recorte y maduración clonal del sistema inmune, etc.). Por otra parte, los mecanismos de aprendizaje vertical (ascendente y descendente) se encargan de determinar qué conocimiento debe ser asimilado (en el nivel emergente) y consolidado (en el nivel cognitivista), y cuál debe ser olvidado.
- *Soporte a la Degradación del Aprendizaje Múltiple*: este característica se evidencia principalmente en el módulo Comportamental donde se emplean diferentes algoritmos de aprendizaje (supervisado, por refuerzo, por descubrimiento, etc.). Concretamente, el problema de la degradación del aprendizaje es solucionado a través de los mecanismos propuestos para la integración de acciones. Por ejemplo, el mecanismo de combinación de pesos, a través de la retroalimentación y del aumento o disminución de los niveles de positividad y negatividad, se encarga de determinar (mediante la asignación de un peso) si el aprendizaje llevado a cabo por un componente del *Comportamiento* ha sido más efectivo en relación con los demás. Estos pesos varían adaptativamente en función del desempeño histórico que tiene cada componente.

- En cuanto a la capacidad de *Cooperación*, METÁFORA propone un mecanismo de trasmisión de conocimiento (i.e., el *algoritmo memético*) que mejora sustancialmente el desempeño del sistema, acelerando la convergencia en el aprendizaje y en la adaptación.
- Respecto a la capacidad de *Meta-razonamiento*, METÁFORA propone un módulo de meta-cognición encargado de administrar los mecanismos que "evolucionan" las diferentes estructuras cognitivas del agente, tanto a nivel individual como social. Por otro lado, este módulo administra los mecanismos para la regulación homeostática y la modulación de la retroalimentación, ambos influyentes sobre los procesos de aprendizaje y planificación.

En relación a las Propiedades Cognitivas del Agente:

- En cuanto a la propiedad de *Representación del Conocimiento*, queda patente la cualidad híbrida de la arquitectura propuesta: METÁFORA almacena unidades de conocimiento tanto *procedimentales* como *declarativas*, soportadas por representaciones en ambos niveles de procesamiento (conocimiento simbólico en el nivel

cognitivista y conocimiento sub-simbólico en el nivel emergente). En el nivel cognitivista se conserva una representación *uniforme* de las unidades de conocimiento, empleando para ello reglas y chunks que pueden ser fácilmente comunicables e intercambiables entre los módulos, independientemente de su naturaleza (procedimental, semántica, o episódica). Por otro lado, pese a que la representación del conocimiento no es uniforme entre los niveles emergente y cognitivista, la arquitectura define un conjunto de mecanismos que restablecen la uniformidad entre estos dos niveles mediante la articulación apropiada de mecanismos para la *interpretación*, *transformación*, y *sincronización* vertical de las unidades de conocimiento.

- Respecto a la propiedad de *Razonamiento en Tiempo Real*, se puede argumentar que, por un lado, debido a que METÁFORA dedica un nivel transversal a los módulos de la arquitectura donde la información es procesada de forma reactiva e implícita, los tiempos de respuesta son reducidos significativamente y, por otro, gracias a que METÁFORA define unidades atómicas de conocimiento (e.g., las reglas y los chunks), el proceso de razonamiento puede ser interrumpido temporalmente y luego reanudado a fin de dar paso a otros procesos cognitivos o al procesamiento de nueva información.

- Por último, en relación con las propiedades del *Aprendizaje* merece la pena resaltar que METÁFORA no define un aprendizaje monótono, con lo cual puede adquirir nuevo conocimiento que revalide y reemplace el obsoleto, aumentando así el grado de aplicabilidad de la arquitectura en dominios reales. Por otra parte, METÁFORA define múltiples técnicas de aprendizaje (e.g., aprendizaje inductivo, supervisado, por refuerzo, por descubrimiento, etc.), lo que le permite compensar las debilidades de cada técnica en particular, e incrementar la robustez y la capacidad de tolerancia a fallos del sistema.

13.2. Evaluación de los requisitos para el diseño de Sistemas Inteligentes

En esta sección se evalúa el diseño de METÁFORA de acuerdo con la taxonomía de requisitos para el diseño de sistemas inteligentes propuesta por (Anderson y Lebiere, 2003). Principalmente, se emplean los aspectos funcionales de su análisis, en lugar de considerar aquellos requisitos no-funcionales específicos a la cognición humana. No obstante, los criterios analizados reflejan un intento por capturar y medir gran parte de los componentes necesarios para producir un comportamiento inteligente, en vez de presentar únicamente los criterios de desempeño que se atienden con frecuencia en la evaluación de sistemas inteligentes en el área de la IA. Adicionalmente, se consideran dos criterios propuestos por (Wray y Lebiere, 2007) para la evaluación de arquitecturas cognitivas: la capacidad de extensión incremental (incrementallity) y la adaptabilidad, los cuales han demostrado ser útiles para capturar y expresar el valor acumulativo de las soluciones basadas en arquitecturas cognitivas para un dominio multi-tarea, y a su vez, para una gran diversidad de dominios.

Relacionados con la Capacidad de Extensión Incremental (incrementallity): este criterio mide la capacidad de extender la funcionalidad de una arquitectura cognitiva desde un conjunto de tareas a otro conjunto, el cual puede ser un superconjunto del conjunto original (como un ejemplo de *generalización*) o reflejar un conjunto diferente de requisitos (como un ejemplo de la *robustez* de la arquitectura). Típicamente, la capacidad de extensión incremental se mide como el grado de solapamiento entre las soluciones de dos conjuntos de problemas. Una métrica comúnmente utilizada para medir la capacidad de extensión incremental son las líneas de código (SLOC), tal que: *Incrementallity = SLOCs originales / SLOCs totales.*

En cuanto al caso de generalización (e.g., en la transición incremental entre las tareas de evadir obstáculos, identificar objetos, y recoger cajas), se comprobó que METÁFORA sólo aumentó entre un 4 y un 6 % en la cantidad de SLOCs para las soluciones más complejas. Por otro lado, en el caso de robustez (e.g., en la tarea de apilamiento de cajas, cuya resolución era independiente de la ejecución de las otras tareas), se evidenció que METÁFORA aumentó un 9 % en el número de SLOCs respecto a solución original (i.e., la concreción funcional). Cabe destacar que el bajo porcentaje de líneas de código necesarias para la solución de tareas más complejas se debe principalmente a que METÁFORA no requiere, en gran medida, de intervención humana para la re-programación de las bases de conocimiento y las estructuras cognitivas más complejas, ya que éstas se autogestionan a través de los procesos auto-organizativos multinivel de la propia arquitectura. Esto es un factor diferencial importante respecto a otras arquitecturas cognitivas típicas (e.g., ACT-R, SOAR, Clarion, etc.) donde por cada nueva concreción contextual de la arquitectura se requiere definir en gran parte del conocimiento sobre el dominio (e.g., el conjunto de reglas de producción simbólicas). En METÁFORA, en todo caso, el definir este tipo de conocimiento simbólico es opcional, ya que el sistema es capaz de obtener autómamente dicho conocimiento a través de los componentes del nivel emergente de la arquitectura, para luego transmitirlo a los componentes del nivel cognitivista.

Relacionados con los Tiempos de Respuesta: para que un sistema inteligente se comporte, valga la redundancia, "inteligentemente", éste debe ser capaz de reconocer una situación que sea de su interés, determinar una respuesta apropiada, y luego actuar coherentemente. Sin embargo, el mundo en el que el agente opera puede cambiar continuamente, con lo cual éste debe desempeñar con rapidez sus procesos internos, en relación a la velocidad de cambio del ambiente o, de lo contrario, sus oportunidades de sobrevivir o ser exitoso disminuirán en el largo plazo. En las diferentes tareas que se probó METÁFORA, desde las más reactivas hasta las más deliberativas, y a pesar de los continuos cambios ambientales que tuvo que afrontar el agente cognitivo, la actuación de éste mantuvo siempre sus tiempos de respuesta dentro de rangos aceptables (como los propuestos en la sección 10.5). Esta eficiencia en el tiempo de respuesta se evidenció tanto para las simulaciones como para los experimentos con robots reales (en aquellos casos donde fue posible realizar este tipo de verificación).

Cabe destacar, además, que los tiempos de respuesta de cada agente fueron incrementalmente reducidos en la medida que éste automatizaba o hacía subconsciente la respuesta a ciertas situaciones (i.e., en la medida que el agente generaba modelos mentales implícitos y sub-simbólicos del mundo) requiriendo así de menos tiempo de procesamiento para

responder a dichas situaciones. Adicionalmente, estos tiempos de respuesta (o tiempos de aprendizaje) fueron reducidos como consecuencia del proceso mismo de evolución memética por el cual, después de cada generación evolutiva, las estructuras cognitivas eran asimiladas como *respuestas innatas* (i.e., respuestas que requieren de menos tiempo de procesamiento).

Relacionados con la Adaptabilidad: METÁFORA evidencia dos tipos de adaptabilidad: una a corto plazo donde el agente exhibe un cambio de comportamiento frente a las condiciones variables del ambiente (e.g., el aprendizaje), y una a largo plazo donde el agente adquiere conocimiento más general sobre el mundo a fin de mejorar su desempeño en situaciones futuras (e.g., la adaptación por evolución). A diferencia de la *robustez* (característica también importante y que será discutida más adelante) que garantiza un desempeño aceptable del sistema cognitivo en situaciones imprevistas, la *adaptabilidad*, por otro lado, provee un proceso eficiente por el cual se extienden las capacidades cognitivas iniciales especificadas por un diseñador, haciendo el sistema más apto para adaptarse a las dinámicas cambiantes de un dominio. Esta adaptabilidad se reflejó en los experimentos, principalmente, en la convergencia del aprendizaje y de los procesos evolutivos con el transcurso del tiempo. Gracias a esta adaptabilidad, METÁFORA es una arquitectura cognitiva no dependiente del contexto que puede operar en una cantidad diversa de dominios sin requerirse una re-programación significativa.

Relacionados con la Escalabilidad: una arquitectura cognitiva es escalable si, sin requerirse un cambio directo de sus mecanismos subyacentes, ésta puede tratar con problemas que demandan una gran cantidad de conocimiento y que aumentan incrementalmente en complejidad. Desde esta perspectiva, METÁFORA cumple con el requisito de *escalabilidad* a través de tres aspectos: la extensibilidad en el número de *Comportamientos*, el proceso de adaptación multi-nivel, y la interacción social. METÁFORA puede agregar o eliminar el conjunto inicial de *Comportamientos* (y los componentes asociados a estos, e.g., las memorias declarativas, los sistemas de anticipación, entre otros), en tiempo de ejecución, a fin de extender la conducta del agente sin perjudicar significativamente la eficiencia del sistema. No obstante, un aumento considerable en el número de *Comportamientos* podría afectar el desempeño y los tiempos de respuesta del sistema, lo cual conduciría al planteamiento de una solución que emplee paralelismo real, como se propuso en las secciones 9.5 y 10.6.

Por otro lado, el proceso de adaptación multi-nivel permite que las diversas estructuras cognitivas que emplea METÁFORA se auto-organicen en la medida que nuevos *Comportamientos* son agregados, evitando que el desempeño del sistema se vea perjudicado mientras que la complejidad de la arquitectura es incrementada. Adicionalmente, la capacidad de escalabilidad de METÁFORA se beneficia del diseño multi-agente de la arquitectura. La interacción social entre múltiples agentes y la adquisición cooperativa de conocimiento demostraron una considerable aceleración de la convergencia en el aprendizaje de nuevas estructuras cognitivas, en proporción con el coste del desempeño global (el cual puede ser también mejorado empleando una solución de paralelismo real). Así pues, METÁFORA puede ser escalable fácilmente a tareas de mayor complejidad si se

definen grupos de agentes que aprenden sub-dominios del problema, conocimiento que posteriormente es compartido y asimilado por el resto de agentes.

Relacionados con la Capacidad de Almacenar Conocimiento: este criterio intenta definir la cantidad de conocimiento que puede representar un agente basado en la arquitectura cognitiva. En el caso de METÁFORA, esto puede ser relativamente sencillo de medir en el nivel cognitivista (e.g., a través del número de reglas simbólicas, chunks, reglas asociativas, etc.), mientras que no lo es en el nivel emergente (ya que la información en este nivel se encuentra distribuida en redes de elementos). Un aumento excesivo de las unidades de conocimiento puede mejorar la capacidad del agente para resolver problemas, aunque simultáneamente degrada el desempeño del sistema (e.g., en el caso de definirse un millón de reglas de producción en el componente RP). Por su parte, METÁFORA define un método elegante para maximizar la capacidad de almacenamiento de conocimiento mientras que se minimiza la cantidad de memoria que se requiere para esto. La capacidad de almacenar conocimiento incrementa exponencialmente en la medida que METÁFORA incorpora mecanismos de generalización del conocimiento (e.g., los mecanismos de generalización definidos tanto para el nivel cognitivista como para el emergente de los *Comportamientos*, las memorias declarativas, y los sistemas de anticipación), mientras que la cantidad de memoria requerida es optimizada mediante los procesos continuos de *aprendizaje* y *olvido*, recreados por los procesos de *cooperación* y *competencia* que dirigen las dinámicas auto-organizativas.

Relacionados con la Robustez: la *robustez* es la capacidad del agente para resistir de forma exitosa, autónoma, y dinámica ante las perturbaciones en eventos y tareas. No existen métricas generales para medir la *robustez*, aunque se han propuesto métricas específicas del dominio (Wray y Lebiere, 2007). Una posibilidad consiste en medir la *robustez* como la proporción aritmética del grado de éxito vs. el grado de perturbación. Se considera entonces que la arquitectura cognitiva es robusta cuando el agente mantiene su grado de éxito dentro de una región de estabilidad a pesar de las condiciones degradadas del sistema. Así pues, la *robustez* de METÁFORA se evidenció en aquellos experimentos donde fue evaluada la capacidad de tolerancia a fallos del sistema: en el experimento 12.3.3 se demostró que los diferentes mecanismos de integración de acciones permitían a los *Comportamientos* ser más robustos, compensando las acciones erróneas de un componente con las acciones correctas de los demás; en el experimento 12.4.4 se demostró que las redes de comportamientos son capaces de encontrar una solución alternativa a pesar de haberse desconectado alguno de sus *Comportamientos*; y en los tres niveles organizativos (epigenético, ontogenético, y filogenético) quedó de manifiesto que las estructuras cognitivas poco aptas podían convertirse gradualmente (en tiempo de ejecución) en soluciones potencialmente exitosas a través de las dinámicas de los procesos evolutivos.

Relacionados con la Validez Neuro-psicológica: este criterio pretende responder la pregunta: ¿la arquitectura hace algún intento por modelar los aspectos del comportamiento humano tomando en cuenta la base neuro-psicológica? El diseño de METÁFORA, a pesar de ser un modelo cognitivo que no intenta reflejar fielmente el funcionamiento del

cerebro humano, sí que está inspirado en ciertas teorías cognitivas que hacen que éste sea un modelo neuro-psicológicamente plausible. En la tabla 13.1 se resume la correlación de los módulos funcionales de METÁFORA con las teorías neuro-psicológicas.

Componente de METÁFORA	**Teoría Neuro-psicológica**
Los *Comportamientos*	Son los *actores* en la metáfora psicológica del teatro de la consciencia (Baars, 1997)
	Reflejan la teoría de aprendizaje procedimental de reglas (Hikosaka et al., 2001)
	En analogía con la función de las Micro-columnas Corticales (Bonin, 1971), los Ganglios Basales (Amos, 2000), el Cuerpo Estriado (Graybiel y Kimura, 1995), el Tálamo, y el Pallidum (Middleton y Strick, 2000).
	Los componentes simbólicos y sub-simbólicos representan la dicotomía entre aprendizaje consciente vs. aprendizaje subconsciente (Willingham et al., 1989; Lewicki et al., 1992)
	Modelan un conjunto de redes neuronales del tipo perceptrón multicapa, como mecanismos de procesamiento cognitivo (Rumelhart y McClelland, 1986; Sun et al., 2001)
	Emplean un conjunto de sistemas inmunes artificiales como mecanismos auto-organizativos de micro-cognición (N. de Castro y Timmis, 2002; Maturana y Varela, 1980; Varela y Coutinho, 1991)
Módulo de Percepción	Representa las memorias sensorial (neo-corteza prefrontal-temporal) y perceptual (corteza perirrinal) (Davachi et al., 2003)
Memoria de Trabajo	Representa el área de trabajo global (corteza prefrontal dorsolateral (Baddeley, 1993; Anderson y Lebiere, 1998)
	Es el *escenario* en la metáfora psicológica del teatro de la consciencia (Baars, 1997)
	Es una memoria de corto plazo, de capacidad limitada, que provee un acceso global a los contenidos subconscientes (Miller, 1956)
Memorias Declarativas	En analogía con la memoria de largo plazo declarativa (lóbulo temporal medial, la región hipocampal, y las cortezas perirrinal, entorrinal y parahipocampal adyacentes) (Squire et al., 2004)
	Son la *audiencia* en la metáfora psicológica del teatro de la consciencia (Baars, 1997)
Módulo Motivacional	En analogía con el concepto psicológico de emociones primarias (Ekman, 1992), o motivadores (drives) (Frijda, 1994; Dennett, 1987)
	Se inspira en la regulación homeostática de emociones primarias observada en los animales y el hombre (McFarland y Bosser, 1993; Breazeal, 2000)
Generador de Metas	Corresponde a las *personas tras bambalinas* en la metáfora del teatro de la consciencia (Baars, 1997)
	En analogía con las funciones ejecutivas de la corteza frontal del cerebro (Fuster et al., 2000).
Sistema de Anticipación y Prospección	Refleja los principios de la teoría cognitiva del control del comportamiento anticipado (Hoffmann, 1993)
	Como mecanismo de simulación interna, es responsable del comportamiento implícito y la imaginería mental (Hesslow, 2002).
Módulo de Orquestación de la Consciencia	Se fundamenta en la teoría neurológica sore la consciencia primaria (compuesta por el troncoencéfalo, el sistema límbico, y el sistema tálamo-cortical), que a su vez se apoya en la teoría darwinista (Edelman, 1987)

Componente de METÁFORA	Teoría Neuro-psicológica
	Se inspira en la metáfora psicológica del teatro de la consciencia (Baars, 1997).
	Define una función adaptativa para el mecanismo atencional de acuerdo con (Broadbent, 1958; Posner, 1978; Shallice, 1972): la selección del conjunto de contenidos subconscientes y la selección de los procesos y respuestas que se van a ejecutar.
Módulo de Meta-cognición	Refleja las variables psicológicas propuestas por (Brown, 1987) acerca de *la persona*, *la tarea*, y *la estrategia*.
	Los mecanismos EOF se inspiran en los procesos darwinianos encargados de evolucionar las pautas espaciotemporales a diferentes niveles de organización neuronal (Calvin, 2001).

Tabla 13.1: Correlación entre los componentes de METÁFORA y las teorías neuro-psicológicas

13.3. Contribuciones originales relativas a la propuesta arquitectónica

Pese a que la definición estructural de METÁFORA resulta similar a la distribución de componentes de una arquitectura cognitiva típica, lo que resulta realmente innovador es la caracterización funcional de cada uno de los módulos y mecanismos que la confirman. A continuación, se enuncian las contribuciones más generales de la presente tesis que merecen la pena ser destacadas por su originalidad y significancia:

- Se ha propuesto un modelo cognitivo híbrido que se enfoca en el ensamblaje de un conjunto de sub-sistemas que intentan modelar ciertas facultades cognitivas, a través de mecanismos computacionales aparentemente inconexos, pero cuya integración en un sistema cognitivo complejo resulta ser más que la simple suma de sus partes. Esta integración es una acomodación de partes discretas, partes que en su gran mayoría actúan concurrentemente y en cooperación, en un ajuste fuertemente sinérgico y robusto. Es posible entonces argumentar que METÁFORA es una arquitectura holística.
- METÁFORA continuamente se re-inventa: a pesar de definir una estructura subyacente rígida (en el sentido que describe un conjunto finito de módulos y componentes cuya funcionalidad es específica), exhibe, por otro lado, una gran capacidad de flexibilidad y plasticidad a través de los procesos de adaptación, el re-diseño funcional y evolutivo de ciertos componentes, y la auto-organización constante de sus estructuras cognitivas.
- Se propone una arquitectura de múltiples niveles jerárquicos de auto-organización, a través de los cuales son generadas estructuras cognitivas de complejidad incremental (desde reglas simples hasta complejos planes deliberativos) que se auto-ensamblan de forma dinámica a fin de crear modelos mentales aproximados de la

realidad. En cada nivel organizativo se definen procesos de competencia y cooperación entre los elementos constitutivos, donde la competencia permite el refinamiento de los mismos, mientras que la cooperación posibilita la emergencia de un nuevo nivel de organización más complejo.

- Se proponen tres procesos de inspiración biológica (i.e., la epigénesis, la ontogénesis, y la filogénesis), los cuales se apoyan en procesos darwinianos que operan en diferentes escalas de tiempo del pensamiento y la actuación, donde determinadas pautas cognitivas sobreviven y se reproducen mejor que las pautas que compiten contra ellas. El proceso global resultante bien podría llamarse "evolución cognitiva", término que resumiría la evolución inter-agente (determinada por los procesos de epigénesis y ontogénesis) y la evolución intra-agente (determinada por el proceso de filogénesis).
- Como sistema de micro-cognición, se ha propuesto un modelo híbrido de un Sistema Inmune Artificial que integra: los principios biológicos de la teoría inmunológica y la autopoiesis, el modelo de asignación de recompensas de los sistemas clasificadores, un mecanismo de aprendizaje por contraejemplos, y un algoritmo genético especializado. Como resultado, este innovador sistema demostró ser más robusto, adaptativo, y eficiente en comparación con otros sistemas de aprendizaje automático basados en poblaciones de individuos.
- Se ha propuesto una extensión evolutiva, basada en Programación de Expresiones Genéticas, al modelo original de Redes de Comportamientos (RC) propuesto por (Maes, 1989). Esta extensión evita la necesidad de una pre-programación de la topología de cada RC (bajo el supuesto que un agente requiere de múltiples RCs especializadas por tarea), y permite a cada una el definir su diseño estructural y funcional (en tiempo de ejecución) como resultado de una adaptación a las necesidades del agente con respecto a su entorno. No obstante, persiste aún la cuestión de cómo lograr que el proceso evolutivo (más precisamente, el proceso ontogenético) sea capaz de crear autonomamente y desde cero, el conjunto de *Comportamientos* que componen la red. Se espera en todo caso poder dar solución a esta cuestión en un trabajo futuro.
- En la presente tesis se ha propuesto un algoritmo memético complejo que, a diferencia de uno típico, construye estructuras cognitivas complejas (los memeplexes) a partir del ensamblaje evolutivo de otras más simples (los memes). En los resultados de la experimentación quedó demostrado que la evolución memética, así como la transmisión cultural de memes entre una población de agentes, acelera exponencialmente la convergencia en el aprendizaje y la adaptación a nuevos entornos.
- Se ha intentado construir agentes con capacidad de imaginación, no sólo entidades que crean representaciones internas pasivas del mundo exterior (lo cual ha sido logrado ya por la IA en cierto grado), sino entidades que crean representaciones que inevitablemente conducen a la producción de planes y estrategias, y finalmente, a la habilidad de dirigir la actuación de forma similar a como lo hacemos cuando somos conscientes de nuestra conducta.
- El modelo que subyace al diseño computacional de METÁFORA se compone de un conjunto de bloques de construcción cibernéticos que permiten: (1) la definición de los flujos de información de cada componente con independencia de su imple-

mentación computacional (e.g., los *integradores* son implementados de diferentes formas: como una función de activación en las redes neuronales, como un mecanismo de integración de acciones en los *Comportamientos*, como una función de acumulación de energía de activación en las RC, etc.); (2) la construcción de diversos sub-sistemas a partir de la reutilización y ensamblaje característico de dichos bloques; y (3) el entendimiento de los flujos de causa y efecto que pueden existir en un sub-sistema a través de los lazos directos y de retroalimentación que conectan los diferentes bloques.

Reflexiones Generales:

- A diferencia de un proceso natural de evolución cognitiva, un proceso artificial como el propuesto, donde se hereda el conocimiento adquirido a las próximas generaciones, acelera el proceso de aprendizaje y adaptación a nuevos entornos. Supóngase que se construye un conjunto de agentes que interactúan con un entorno nuevo, y que cíclicamente, en cada generación evolutiva, son elegidos los mejores para su clonación. Como las copias incluirían todos los recuerdos hasta la fecha (esa es una de las ventajas de la inteligencia artificial: se pueden incluir en la clonación capacidades adquiridas), la experiencia sería acumulativa y auténticamente lamarckiana: los agentes descendientes no tendrían que repetir los errores de los agentes padres.
- Aún estamos muy lejos de poder crear una mente humana artificial que supere, o por lo menos asemeje, la natural, y METÁFORA no es una excepción de esta realidad. No obstante, la presente aproximación pone de manifiesto que la solución no recaerá sólo en los esfuerzos que se hagan por modelar, cada vez con mayor precisión, los fenómenos cognitivos de orden superior, sino que además, la solución apuntará hacia un enfoque verdaderamente darwiniano donde los agentes serán capaces de evolucionar su propio diseño, incluso, de manera no prevista por el diseñador humano.
- Debemos asegurarnos de que la receta para el fenómeno emergente de la cognición artificial no es completamente pre-programada, sino que puede ser pasada de generación en generación, y modificarse así misma con el propósito de persistir ante los cambios ambientales y en escalas de tiempo más largas. Cada agente podrá alcanzar niveles superiores de inteligencia si su futuro está en sus manos, y para poder otorgar esta autonomía debemos renunciar a tener un control directo y absoluto sobre su diseño.

Conclusión Final: a la vista de los resultados obtenidos, se puede concluir que el desarrollo de la presente tesis ha atendido satisfactoriamente los objetivos inicialmente planteados, en la construcción de un modelo cognitivo híbrido que exhibe propiedades emergentes, adaptativas y auto-organizativas, y contribuyendo significativamente en el área del diseño de arquitecturas para agentes inteligentes.

Capítulo 14

Trabajo Futuro

"El mundo del futuro será una lucha cada vez más esforzada contra las limitaciones de nuestra inteligencia, no una confortable hamaca en la que podamos tumbarnos mientras nos atienden nuestros esclavos robots".
– Norbert Wiener.

Durante el extenso desarrollo de la investigación surgieron muchas ideas, algunas de ellas demasiado ambiciosas, otras quizás más aterrizadas... ideas que, aunque no todas tuvieron cabida en el alcance de la presente tesis, son ideas que podrían guiar futuros enfoques de investigación y desarrollo que extenderían y mejorarían la funcionalidad que exhibe actualmente METÁFORA. A continuación se sintetizan las principales líneas (algunas de ellas debatidas en la sección de discusión, al final de cada capítulo) en las que podría profundizarse la investigación y el desarrollo en un futuro:

Respecto a los módulos de Percepción y Efector

- A fin de hacer más robusto el procesamiento perceptual de METÁFORA, sería interesante incluir un mecanismo que se encargase de fusionar la información sensorial capturada por cada uno de los diferentes sensores del agente (término conocido como *Sensor Fusion*). El propósito de este mecanismo consistiría en hacer más confiable, precisa, y completa la información percibida en cada instante. Adicionalmente, sería interesante incluir métodos estadísticos, como el filtrado de partículas (*particle filtering* en inglés), a fin de estimar el cambio de estado del sistema a lo largo del tiempo. La utilidad de este método radica en que permite predecir, con mayor precisión, aquella información que bien por el ruido en los datos o por la incertidumbre, no es correctamente percibida y analizada por el agente.
- Se requiere de un Sistema de Lógica Difusa que realice dos tareas: (1) convertir los valores reales capturados por los sensores del agente en etiquetas lingüísticas interpretables por el resto de módulos de la arquitectura (e.g., una estatura de 1.87cm corresponde a una etiqueta lingüística de "muy alto"); y (2) convertir los valores

lingüísticos de las acciones que procesa el módulo Efector en valores reales (e.g., la acción con valores lingüísticos "rotar la cabeza ligeramente hacia la derecha" podría convertirse en la acción con valores reales "rotar la cabeza -20º"). (Véase la sección 7.9).

- Se requiere del diseño de dos memorias diferenciadas: una memoria sensorial que registra los estímulos sensoriales entrantes e integra la información miltimodal captada por los sentidos, y una memoria perceptual para el reconocimiento, categorización e interpretación de los objetos percibidos y sus relaciones. (Véase la sección 8.8).

Respecto al módulo Conexionista de los *Comportamientos*:

- Actualmente, el módulo Conexionista de METÁFORA emplea un tipo particular de red neuronal, el perceptrón multicapa, que define dos tipos de aprendizaje: el aprendizaje supervisado (que emplea el algoritmo de retropropagación), y el aprendizaje por refuerzo (que emplea el algoritmo Q-learning). La auto-organización en este tipo de redes se fundamenta en su plasticidad sináptica, esto es, en la capacidad de ajustar adaptativamente los pesos entre las neuronas como resultado de un aprendizaje. No obstante, existe otro tipo de red neuronal, las redes de Kohonen (o mapas auto-organizados), que efectúan un aprendizaje "no supervisado", mediante el cual descubren la estructura subyacente de los datos introducidos en la red. Resultaría interesante incluir este tipo de red dentro del módulo Conexionista, la cual sería empleada cuando no se dispone de un conjunto de patrones de entrenamiento (en el caso del aprendizaje supervisado) o cuando no se dispone de un refuerzo ambiental continuo (en el caso del aprendizaje por refuerzo). (Véase la sección 7.9)
- Adicionalmente, la plasticidad sináptica del modelo conexionista actual podría ser mejorada a través de un mecanismo adaptativo que modificase, en tiempo de ejecución, no sólo los pesos entre las neuronas de la red sino también su estructura topológica (e.g., el número de capas, el número de neuronas por capa, el tipo de propagación de la activación, etc.). Se presume que esta adaptabilidad puede ser lograda mediante un proceso evolutivo artificial, en especial, a través de la programación de expresiones genéticas. (Véase la sección 7.9)

Respecto al módulo de Producciones de los *Comportamientos*:

- Actualmente, METÁFORA emplea como proceso de aprendizaje, el mecanismo de extracción de reglas sub-simbólicas y la posterior especialización y generalización de dichas reglas. No obstante, resultaría interesante incorporar un mecanismo de *aprendizaje por inducción* que permitiese al módulo de Producciones adquirir nuevo conocimiento con independencia del nivel emergente del *Comportamiento*. Este tipo de aprendizaje extraería y generalizaría colecciones de reglas o patrones simbólicos a partir de un conjunto de observaciones e información no estructurada suministrada en forma de ejemplos. A fin de lograr esta funcionalidad, podría evaluarse el desempeño de alguno de los siguientes algoritmos de aprendizaje: la

generación de reglas a partir de árboles de decisión (Quinlan, 1987), los espacios de versiones (Dubois y Mohamed, 2002), la poda incremental del error reducido (algoritmo IREP) (Furnkranz, 1994), la poda incremental repetida para reducir el error (algoritmo RIPPER) (Cohen, 1995), el aprendizaje de reglas empleando taxonomías (algoritmo TRIPPER) (Vasile et al., 2004), entre otros. (Véase la sección 7.9)

- Con el propósito de potenciar la funcionalidad del módulo de Producciones, podría incluirse un Sistema Basado en Conocimiento que definiese un motor de inferencia con múltiples mecanismos de razonamiento (e.g., el encadenamiento hacia atrás y hacia adelante, el razonamiento bajo incertidumbre, la representación de las reglas como máquinas de estados –RETE– etc.) el cual permitiese cotejar los resultados inferidos. Una opción interesante, debido a su compatibilidad con el lenguaje de programación Java, además de que cumple con las características antes mencionadas, es el motor de reglas JESS[1]. (Véase la sección 7.9)

Respecto a los *Comportamientos* en general:

- El proceso ontogenético propuesto tiene la capacidad de establecer, dinámicamente, las relaciones de inhibición y excitación entre los *Comportamientos* que conforman una RC, así como de modificar el conjunto inicial (previamente especificado) de premisas que sirven de pre-condiciones para la activación de los mismos. No obstante, el proceso ontogenético no posee la capacidad (al menos por ahora) de caracterizar, por completo y desde cero, un *Comportamiento*, como resultado de una exigencia nueva del ambiente (o de un estado interno) que debe ser suplida por el agente a través de una conducta.

  Se identifica por tanto la necesidad de proponer un mecanismo por el cual sean identificadas nuevas oportunidades para crear, de forma autónoma y dinámica, los *Comportamientos* que emplea el agente. Este proceso de creación de *Comportamientos* debería accionarse en respuesta a una meta o sub-meta (y sus respectivos estados motivacionales que la activaron) que no puede ser alcanzada por ninguno de los *Comportamientos* que dispone el agente en ese momento. Así pues, el mecanismo tendrá que definir por cada *Comportamiento* nuevo: el conjunto de pre-condiciones que lo activarían, sus listas correspondientes de efectos deseados y no deseados, el conjunto de perceptos que procesarían todos sus componentes (i.e., RP, RES, SIA, y PMR), y el tipo de salidas o acciones que serían enviadas al módulo Efector.

  Adicionalmente, el mecanismo debería realizar los siguientes procesos: (1) la generalización de aquellos *Comportamientos* creados que tienen características comunes que pueden ser factorizadas, caso en el que se eliminarían los *Comportamientos* redundantes, extrayendo primero las reglas más relevantes de cada uno e integrándolas en un sólo *Comportamiento* más general; (2) la especialización de aquellos *Comportamientos* creados que no logran realizar la tarea asignada por sí solos, requiriéndose posiblemente el dividir la tarea en unidades más pequeñas o

[1] `http://www.jessrules.com/jess/`

en sub-metas (i.e., en *Comportamientos* más específicos); y (3) la evolución estructural de las conexiones entre los *Comportamientos* creados. En la medida que se crean nuevos *Comportamientos*, estos son asociados con los demás *Comportamientos* existente a través de las RCs, estructuras que requieren ser optimizadas (e.g., reducir los ciclos de auto-referencia, evitar los caminos sin salida, etc.).

Respecto a las memorias declarativas:

- A fin de optimizar el almacenamiento y recuperación de vastas cantidades de conocimiento en las memorias declarativas, se propone emplear el sistema *slipnet* (Hofstadter y Mitchell, 1994) para la memoria semántica, y una red Kanerva (Kanerva, 1988) para la memoria episódica. Estos tipos de sistemas requerirían un mecanismo que permitiese la comunicación efectiva de unidades de conocimiento entre sí, y entre el módulo Comportamental. Se presume que esta funcionalidad podría ser desempeñada por un sistema de creencias basado en ontologías. (Véase la sección 8.8).
- Resulta necesario incluir enfoques adicionales de razonamiento semántico que produzcan un estado inferido más enriquecido y aproximado a la realidad. Se propone la inclusión del razonamiento basado en similitudes (similarity-based reasoning) (Hullermeier, 2001), el razonamiento causal (Kemp et al., 2006), o una combinación de los dos. (Véase la sección 8.8).

Respecto al módulo Motivacional:

- Se requiere de un modelo extendido donde se tengan en cuenta los *Estados Afectivos y de Ánimo*, los *Estados Físicos*, los *Rasgos de la Personalidad*, las *Actitudes*, y las *Emociones* del agente, con el propósito de hacer más "creíble" la actuación de éste en relación a la comunicación con otros agentes artificiales o con humanos. (Véase la sección 8.8).
- En el modelo actual de regulación homeostática de los motivadores, se emplea un método discreto para determinar el valor o estado del motivador en relación a la zona de estimulación (cada una limitada por umbrales específicos) en la que se encuentre su intensidad (e.g., el motivador ⟨`hambre`⟩ tiene un valor de ⟨`hambriento`⟩ en la zona de infra-estimulación, ⟨`con-hambre-normal`⟩ en la zona homeostática, y ⟨`sin-hambre`⟩ en la zona de sobre-estimulación). Este método permite valorar, de forma sencilla, el estado de cada motivador, no obstante, produce incertidumbre cuando la intensidad del motivador se encuentra en la frontera de cada zona de estimulación, es decir, cuando la intensidad es equivalente al valor del umbral.

  Resultaría más realista el considerar el límite de cada zona como una frontera borrosa y no como un valor discreto. Además, este inconveniente surgiría también al tratar otro tipo de elementos motivacionales, como los enunciados en el punto anterior. De manera similar a como se ha tratado previamente en trabajos relacionados con *emoción artificial* (e.g., (Seif El-Nasr et al., 2000b; Bui et al., 2001;

Imbert, 2005)), la solución a este problema recaería nuevamente en el diseño de un Sistema de Lógica Difusa que definiese reglas borrosas para la valoración de las emociones (primarias y secundarias).

- En extensiones futuras de METÁFORA resultaría interesante definir un mecanismo de *intuición* a través del cual el agente llegase a la solución de un problema *sin* razonar con respecto a él, en especial, cuando no existe suficiente información disponible para razonar apropiadamente. Para lograr este objetivo, podría simularse la manera en que los *marcadores somáticos*[2] influyen en la toma de decisiones.

  El funcionamiento básico del mecanismo sería el siguiente: los marcadores somáticos sintéticos operarían de forma subconsciente (en forma de bucles «como si») generando, para los resultados negativos de las acciones del agente, la inhibición de la tendencia a actuar de dicha manera, reduciendo así las probabilidades de tomar una decisión potencialmente negativa. En el peor de los casos, el agente obtendría una ganancia de tiempo, durante el cual el proceso de deliberación consciente podría incrementar las probabilidades de tomar una decisión apropiada (incluso, la más apropiada). Adicionalmente, sería posible descartar por completo una opción negativa, o hacer que una positiva se haga más probable mediante la intensificación del impulso para actuar.

  La idea de reducir las consecuencias negativas y aumentar las positivas no es nueva en el diseño de METÁFORA: es el principio que subyace a todos los componentes que emplean algún tipo de aprendizaje por refuerzo. No obstante, este aprendizaje se hace de forma directa, sin concebir la intervención de las emociones durante el mismo. Es por esto que la inclusión de los "marcadores somáticos" permite que la reacción emotiva pase de ser una simple consecuencia, por ejemplo de alguna decisión negativa, a influir en la misma toma de decisiones, posibilitando la anticipación de las consecuencias y guiando el proceso de resolución final.

Respecto a la Planificación:

- A fin de mejorar la precisión de los planes generados por el proceso ontogenético, se podría incluir un planificador clásico (e.g., SHOP2) cuyos resultados serían cotejados con los del primero, generando una solución integrada o sirviendo de guía durante el desarrollo evolutivo de los planes. (Véase la sección 10.6).
- Con el propósito de extender la estructura secuencial de los planes (e.g., incluyendo ramificaciones condicionales, bucles, etc.), se podría aumentar el conjunto de

[2]Los marcadores somáticos, propuestos por Damasio (Damasio, 1994), son un caso especial de sentimientos generados a partir de emociones secundarias. Estas emociones y sentimientos han sido conectados, mediante aprendizaje, a resultados futuros predecibles de determinados supuestos. Cuando un marcador somático negativo se yuxtapone a un determinado resultado futuro, la combinación funciona como un timbre de alarma. En cambio, cuando lo que se superpone es un marcador somático positivo, se convierte en una guía de incentivo.

funciones que emplea el proceso ontogenético mediante la incorporación de primitivas sintácticas como las que propone el lenguaje TAEMS para la construcción de Redes de Tareas Jerárquicas. (Véase la sección 10.6).

- Se presume que el desempeño del sistema, cuando se lleva a cabo la simulación concurrente de múltiples planes, puede ser mejorado si se emplea una solución de paralelismo real. (Véase la sección 10.6).

- La codificación genética propuesta para un plan se basa en un enfoque unicelular, el cual define un plan como una célula que contiene múltiples genes que especifican la estructura y funcionalidad de las redes de comportamientos asociadas, y un gen homeótico que determina la manera en que interactúan dichas redes de comportamientos a fin de lograr una meta global. A partir de esto, se propone la idea de emplear un enfoque "multicelular" que defina múltiples genes homeóticos, donde cada uno represente condiciones diferentes de activación del plan (e.g., si ocurre *"x"* ejecútese el plan *A*, sino, si ocurre *"y"* ejecútese el plan *B*...), o diferentes micro-planes que conforman un macro-plan. Se estaría hablando entonces de un nuevo nivel de organización donde se construyen estrategias basadas en micro-planes de resolución de problemas.

- Resultaría interesante incluir un componente de *aprendizaje por imitación* en los *Comportamientos*, a través del cual se reconstruiría mentalmente la conducta de un experto (humano o artificial). Por ejemplo, el agente observaría primero la actuación de un experto ejecutando una tarea especifica y, luego, generaría internamente múltiples secuencias de *Comportamientos* y acciones concretas que intentarían replicar la actuación del experto. Estas secuencias competirían a través de un proceso evolutivo, apoyándose para esto en la simulación interna y, finalmente, la secuencia que produjera los resultados más cercanos a la actuación del experto, sería la secuencia ganadora que sería luego asimilada por los *Comportamientos* implicados.

  Supóngase que el agente observa un estado actual (t) antes de que el experto ejecute alguna acción, y luego observa el estado siguiente ($t + 1$) donde el entorno ha cambiado como consecuencia de ejecutar una regla. Si la acción ejecutada por el experto es comunicada explícitamente, se puede construir una tupla del tipo `<estado-inicial><acción><estado-final>`, pero en el caso de que no se provea dicha acción, que resulta ser el caso más interesante, el agente tendría que inferir la acción o el conjunto de acciones que ejecutó el experto.

  Cabe notar que la imitación tendría más relevancia si no se hiciese sobre estados discretos (e.g. $t \rightarrow t + 1$), sino en escalas de tiempo más amplias, esto es, en términos de planes completos. Por ejemplo, supóngase el caso que el agente que está aprendiendo intentase imitar la manera en que el agente experto realiza una secuencia de movimientos hasta coger una manzana. El estado inicial sería antes de iniciar el movimiento del brazo, y el estado final cuando el experto ha cogido ya la manzana. El agente simularía internamente, y en paralelo, multiples secuencias de comportamientos y acciones concretas que intentarían reproducir, con cierto nivel de fidelidad, el movimiento realizado por el experto.

Respecto al módulo de Meta-cognición:

- Resultaría atractivo el incluir un nivel auto-reflexivo en la arquitectura, que le permitiese al agente no sólo evaluar las consecuencias de sus acciones (proceso que en cierta medida es logrado a través de la retroalimentación que reciben gran parte de los componentes de METÁFORA), sino además, que le permitiese ejercer un entendimiento sobre su propia conducta, su personalidad, sus preferencias, sus metas a largo plazo, su función como individuo de una sociedad, su objetivo de "ser" o de "existir"... en otras palabras, un nivel más sofisticado de consciencia que lo motive a *"ser un mejor agente"* en armonía con el ambiente y con otros agentes.
- Es posible que el mecanismo descubridor de reglas, redes de comportamientos, y planes (i.e., la máquina darwiniana) sea capaz también de soportar los procesos de *imaginación* y *creatividad* del agente. Como se ha podido apreciar en el desarrollo de la presente tesis, los procesos darwinianos tienen el poder de "crear" estructuras cognitivas, desde las más simples a las más complejas, a partir de la replicación y la combinación aleatoria de partes que no necesariamente estaban relacionadas entre sí. No obstante, en el marco de un proceso de imaginación en el que las partes son extraídas de innumerables dominios, las combinaciones realizadas de esta forma serían infinitas en número y la mayoría de ellas no producirían soluciones de mayor interés.

  Considérese el siguiente pasaje del matemático Henri Poincaré (Poincare, 1955): *crear consiste precisamente en no realizar combinaciones inútiles y en efectuar aquellas que son útiles y que sólo son una reducida minoría. La invención es discernimiento y elección. Entre las combinaciones elegidas, las más fértiles serán con frecuencia las formadas por elementos procedentes de dominios que están muy apartados. No es que quiera decir que para la invención baste unir objetos tan dispares como sea posible; la mayoría de combinaciones así formadas serían completamente estériles. Pero algunas de ellas, muy pocas, son las más fructíferas de todas.*

  Tomando en cuenta el anterior pasaje, y considerando la evidencia al respecto en el campo de las neuro-ciencias (Salk, 1985; Damasio, 1994), es posible argumentar que los procesos creativos del cerebro humano no sólo se basan en el pensamiento lógico y la capacidad analítica, sino que además se fundamentan en procesos que ocurren a nivel subconsciente, esto es, la creatividad reside en una "combinación entre *intuición* y *razón*". Así pues, el proceso ontogenético, actuando como mecanismo descubridor, se encargaría de producir nuevas combinaciones de elementos de diversos contextos; luego, el proceso de intuición[3] se ocuparía de filtrar las combinaciones más fértiles; y, finalmente, la simulación interna sería el laboratorio donde las nuevas combinaciones serían puestas a prueba, es decir, donde se validarían los *"qué pasaría si..."*. El proceso de creatividad emergería entonces de la interacción continua de estos mecanismos.
- Un objetivo ambicioso que se podría buscar en un largo plazo, es que METÁFORA estuviese en la capacidad de crear sus propias "metáforas", esto es, expresiones

[3] Mencionado anteriormente en las líneas de investigación relacionadas con el módulo Motivacional.

y conceptos que son trasladados de un contexto específico a otro que no denota literalmente, a fin de sugerir una comparación y facilitar su comprensión. Como propone Wilson (Wilson, 1998): "las metáforas, consecuencia de la activación expansiva del cerebro durante el aprendizaje, son los bloques básicos con los que se construye el pensamiento creativo. Conectan y refuerzan de manera sinérgica las diferentes esferas de la memoria".

Se presume que el proceso computacional de creación de metáforas implicaría, grosso modo, tres pasos básicos: (1) la abstracción de un conjunto de conceptos que forman una idea, mediante la identificación semántica de la función de los elementos que intervienen y la relación entre dichos elementos (e.g., el sistema solar podría ser descrito como un elemento central –el sol– que a través de una fuerza física mantiene otros elementos –los planetas– orbitando alrededor suyo); (2) el traslado de la idea abstraída en otro dominio del conocimiento donde se cumplen ciertas características de similitud entre los elementos y las relaciones semánticas identificadas en el paso anterior, en otras palabras, un proceso de contextualización (e.g., el átomo posee también un elemento central que mantiene orbitando alrededor suyo otros elementos); y (3) la validación de la idea contextualizada (e.g., ¿tiene sentido decir que un átomo es un sistema solar en miniatura?). No obstante, la creación de metáforas es un proceso muy complejo, que requeriría de un potente mecanismo de meta-cognición capaz de desensamblar, trasladar, y reensamblar ideas en diferentes contextos; procesos que a su vez necesitarían de otros más específicos, y donde sería imprescindible la interacción de todos los módulos internos dedicados al procesamiento cognitivo de METÁFORA.

> *"... Y lo mejor de todo es que un proceso darwiniano proporciona un mecanismo para la metáfora: se pueden codificar relaciones entre relaciones y darles forma hasta obtener algo de calidad".*
> — W. Calvin, How brains think, 2001.

Respecto a otros módulos:

- Durante el desarrollo de la tesis se evidenció una fuerte necesidad de un subsistema que permitiese al agente comunicar, en un lenguaje entendible por los humanos, sus procesos internos de razonamiento, estados de ánimo, intenciones, motivaciones, expectativas, creencias, dudas, etc. Esta funcionalidad podría ser llevada a cabo por un *sistema de procesamiento computacional del lenguaje natural*.
- En la arquitectura propuesta se considera la trasmisión de memes como un proceso de "interacción social" por el cual cooperan *implícitamente* los agentes. Sin embargo, resulta necesario definir en METÁFORA, un módulo (o un nivel cognitivo completo) dedicado al comportamiento social de los agentes. Este módulo definiría la interacción social entre agentes a través de una comunicación expresa de las intenciones de cada quien, a partir de las cuales se coordinaría la ejecución de tareas, la transmisión *explícita* de conocimiento (proceso de enseñanza-aprendizaje), la creación de alianzas sociales (cooperación), la cultura, etc...

Respecto al diseño de Sistemas Tutores Inteligentes: desde hace un par de años, a parte de los sistemas cognitivos artificiales, mis intereses han estado además enfocados hacia el desarrollo de Sistemas Tutores Inteligentes[4]. He aquí algunas ideas de cómo convergen estos dos campos de interés:

- Muy probablemente, en un futuro no muy lejano, los cerebros artificiales desplazarán incluso a algunos de los trabajadores más preparados. Sin embargo, podrían existir algunos beneficios importantes para los humanos: por ejemplo, considérese una máquina docente (o agente pedagógico) que actuase como un ayudante del profesor, una máquina capaz de mantener conversaciones con los alumnos, que nunca se aburriese con los ejercicios, que siempre procurase aportar la variedad necesaria para mantener interesados a los estudiantes, que pudiese adaptar y personalizar las estrategias pedagógicas a las necesidades de un estudiante concreto, y que pudiese captar señales de trastornos del desarrollo cognitivo y del aprendizaje, como la dislexia o la incapacidad para centrar la atención.
- Estos agentes pedagógicos, tendrían también la capacidad de ser autodidactas: no requerirían la expresa intervención de un humano para aprender acerca de su propio entorno. El conocimiento adquirido por cada agente podría ser enseñado (implícita y explícitamente) a la siguiente generación de agentes, facilitando la evolución de otros más listos, por simple variación y selección. Al fin y al cabo, se podría clonar al mejor alumno. A partir de ahí, a cada clon se le educaría de un modo algo diferente. Con una experiencia suficientemente variada, algunos podrían adquirir características deseables: valores como sociabilidad o interés por el bienestar humano. Y una vez más se seleccionaría al mejor alumno para clonarlo. Es probable que durante esta evolución artificial, que en términos temporales puede ser más rápida que la evolución cognitiva natural, el conjunto de agentes artificiales descubra la manera de alcanzar nuevos niveles de organización (e.g., meta-metáforas), con lo cual, podrían indicarnos la manera de "educar" a los humanos para que dieran el mismo paso. Hablaríamos, quizás, de una co-evolución cognitiva hombre-máquina, donde el avance evolutivo en un tipo de individuo (humano o agente artificial) sirviese de peldaño para el siguiente salto evolutivo cognitivo en el otro tipo de individuo.

Como se puede observar, aún queda mucho por hacer. Esta tesis es apenas un boceto, un borrador de lo que pienso que podría ser un modelo cognitivo auto-organizado, no obstante, es necesario continuar refinando más el modelo... aunque dudo mucho que se llegue a un estado donde no haya nada más por mejorar (al menos en el campo de la cognición artificial).

¡Y colorín colorado, este cuento se ha acabado!!!

[4]Como se puede comprobar en las publicaciones (Romero et al., 2002; Romero, 2002; Romero, 2003).

(1)

Parte IV

Apéndices

Apéndice A

Análisis Comparativo de las Arquitecturas Cognitivas Estudiadas

El presente estudio comparativo fue llevado a cabo en dos fases distintas de la investigación, como se expone a continuación:

1. En una fase inicial, el análisis comparativo permitió la identificación de las principales fortalezas y debilidades de las 30 Arquitecturas Cognitivas descritas en la sección 3.2, sirviendo como fundamento para la definición del problema y su respectiva solución.
2. En una fase final de la investigación, el análisis comparativo permitió contrastar las características diferenciadoras más relevantes que aporta la arquitectura propuesta (METÁFORA) en comparación con las otras 30 Arquitecturas Cognitivas.

Así pues, los resultados obtenidos en las dos fases del análisis comparativo se presentan de forma sintetizada en las secciones siguientes. Por último, es importante resaltar que la comparación entre las arquitecturas bajo estudio se llevará a cabo desde cinco perspectivas específicas[1]:

1. Características Arquitectónicas Generales
2. Tipo de Arquitectura
3. Propiedades del Ambiente Percibido
4. Capacidades del Agente para producir Comportamientos Inteligentes
5. Propiedades Cognitivas del Agente

Arquitectura	Paradigma	Personificación	Percepción	Actuación	Anticipación	Adaptación	Motivación	Autonomía
SOAR	C				○			●
EPIC	C		○	○	○			○
ACT-R	C		○	○	○			●
ICARUS	C		○	○	○			○
ADAPT	C	●	●	●	○			○
BDI	C		○	○	○		○	●
ATLANTIS	C	●	●	●	○			○
COGNITIVA	C	○	○	○	○	○		○
DCA	C		●	●				○
ERE	C		●	●	●			○
HOMER	C		○	○	●			○
MAX	C		○	○				○
PRODIGY	C		○	○	●	○		○
TETON	C				○			○
Theo-Agent	C		●	○				○
RALPH-MEA	C		○	○	●		○	○
AAR-Subsumción	E	●	●	●			○	●
GWM	E	○	○	○	●		●	●
SASE	E	●	●	●	○	●	●	●
DARWIN	E	●	●	○		●	●	●
HUMANOID	H	●	●	●	●		○	○
CEREBUS	H	●	●	●	○			○
COG	H	●	●	●	○	○		○
KISMET	H	●	●	●			●	●
CLARION	H		○	○				●
LIDA	H		●	○	●		●	●
POLYSCHEME	H	○	○	○	○			●
4CAPS	H		○	○	○			●
NOVAMENTE	H		○	○	○			●
DUAL	H	○	●	○			●	
METAFORA	H		○	○	●	●	○	●

Tabla A.1: Comparativo de las Características Arquitectónicas Generales

A.1. Estudio de las Características Arquitectónicas Generales

En la tabla 16.1 se resume, para cada arquitectura bajo estudio, el grado de presencia de las características deseables en un Sistema Cognitivo Artificial (SCA) discutidas en la

[1]En su mayoría, los criterios para clasificar una Arquitectura Cognitiva según el tipo de perspectiva han sido propuestos por el laboratorio de Inteligencia Artificial, Departamento de Ingeniería Eléctrica e Informática (EECS), de la Universidad de Michigan: `http://ai.eecs.umich.edu/cogarch2/`

sección 3.1. Se han omitido las primeras cinco características (Operación Computacional, Esquema Representacional, Bagaje Semántico, Restricciones Temporales, y Epistemología Inter-Agente) debido a que éstas pueden ser inferidas directamente a partir del paradigma al cual pertenece la arquitectura: Cognitivista, Emergente o Híbrido, denotado por "C", "E", o "H" respectivamente. Adicionalmente, se empleará el signo "•" para advertir que la característica analizada se encuentra presente con mucha relevancia, el signo "○" para indicar que la característica está débilmente presente, y un espacio en blanco para señalar que la característica no se encuentra presente. En cuanto a la característica de "Adaptación" solamente se marcará un signo "•" si el sistema tiene la capacidad de evolucionar (en el sentido de crear nuevos esquemas representacionales o modelos a través de un proceso evolutivo) en lugar de simplemente efectuar un aprendizaje (en el sentido de estimar los parámetros del modelo), como se propone en (Weng, 2004).

Característica	**Paradigma**					
	Cognitivista		**Emergente**		**Híbrido**	
	Frec.	**% Total**	**Frec.**	**% Total**	**Frec.**	**% Total**
Personificación	3	18,75 %	4	100,00 %	6	54,54 %
Percepción	14	87,50 %	4	100,00 %	11	100,00 %
Actuación	14	87,50 %	4	100,00 %	11	100,00 %
Anticipación	13	81,25 %	2	50,00 %	8	72,72 %
Adaptación	0	0,00 %	2	50,00 %	3	27,27 %
Motivación	4	25,00 %	4	100,00 %	5	45,45 %
Autonomía	16	100,00 %	4	100,00 %	11	100,00 %

Tabla A.2: Frecuencia con que son contempladas las Características Arquitectónicas Generales en cada uno de los Paradigmas Cognitivos, y Porcentajes del total de muestras analizadas.

En la tabla 16.2 se presenta el análisis de frecuencias para cada una de las características arquitectónicas generales observadas. A partir de los resultados, se puede concluir lo siguiente:

- Las arquitecturas del paradigma Cognitivista evidencian una mayor tendencia hacia el diseño de sistemas deliberativos cuyo proceso de toma de decisiones está fuertemente acoplado a los mecanismos de percepción y actuación.
- Las arquitecturas del paradigma emergente demuestran un claro interés hacia el diseño de agentes personificados (embodied agents), cuyos procesos cognitivos se modifican continuamente como consecuencia de la percepción/actuación activa, los estados internos, y las dinámicas internas. Por otro lado, a diferencia del paradigma Cognitivista, la característica de *Autonomía* no es consecuencia de una pre-programación hábil, sino que surge como resultado de los procesos autoorganizativos del agente.

- En las arquitecturas de enfoque híbrido destacan, además de la percepción/actuación activa del agente, la característica de *Autonomía*. Así pues, resulta evidente el propósito de las arquitecturas híbridas por conservar las fortalezas heredadas de los paradigmas cognitivista y emergente.
- Como nota final, obsérvese que la característica de *Adaptación*, desde el punto de vista del refinamiento de los esquemas representacionales como consecuencia de un proceso evolutivo, no es una propiedad que se haya abordado con la suficiente relevancia en la mayoría de las arquitecturas analizadas.

A.2. Análisis por Tipo de Arquitectura

Tipos de Arquitecturas Analizados: a continuación se presentan diferentes puntos de vista sobre cómo una arquitectura cognitiva podría ser integrada y organizada, y bajo qué principios podría fundamentarse su diseño. Cabe aclarar que estas categorías no incluyen todos los posibles esquemas de organización, así como tampoco es factible unificar todas las arquitecturas en una sola categoría. Por tal motivo, se exponen a continuación solamente los tipos de arquitecturas subyacentes a los diseños de las arquitecturas descritas en la sección 3.1.

- **Arquitecturas Multi-capa**: este tipo de diseño busca construir agentes simples, capaces de operar en ambientes dinámicos. Los comportamientos más complejos se construyen jerárquica e incrementalmente en forma de capas.
- **Arquitecturas Basadas en Marcos**: este tipo de arquitectura organiza y representa todo el conocimiento en Marcos (frame-based definitions) (Martin, 1998). Su importancia se basa en que permite una *representación unificada del conocimiento*. Las arquitecturas basadas en marcos mantienen el conocimiento acerca del mundo, incorporan niveles de meta-conocimiento, y están limitadas a crear inferencias solamente a partir del conocimiento disponible pre-programado.
- **Arquitecturas Modulares**: la filosofía de diseño que sustenta una arquitectura modular es que un sistema inteligente puede ser creado a través de la integración de componentes independientes funcionalmente, como por ejemplo un componente de Planificación y un componente de Aprendizaje. El diseño modular no sólo enfrenta el problema de definir qué capacidades son necesarias para la inteligencia sino también cuál es la descomposición funcional apropiada, y qué niveles de comunicación deben existir entre los módulos.
- **Arquitecturas con Mecanismo Central Solucionador de Problemas**: en este tipo de arquitecturas, el conocimiento es representado como secuencias de acciones (planes), las cuales conducen hacia el logro de una o varias metas. La arquitectura dispone de un *conocimiento de control central* que le permite acotar el espacio de búsqueda e incrementar el desempeño.
- **Arquitecturas con Componentes Asíncronos**: este tipo de diseño es usualmente empleado para reducir la complejidad en el tratamiento de la información percibida por el agente. Su filosofía se basa en enfocar, de manera asíncrona, cada componente de la arquitectura hacia un sub-conjunto específico o nivel de importancia.

- **Arquitecturas del tipo Planificar-luego-Compilar**: una gran cantidad de arquitecturas cognitivas poseen cierta habilidad para planificar, sin embargo, sólo algunas tienen la capacidad de convertir los planes elaborados en *operaciones compiladas*, es decir, en secuencias de acciones concretas. Esta propiedad le permite al agente pasar de una ejecución puramente planificadora a una ejecución reactiva.
- **Arquitecturas Multi-Componente**: las arquitecturas de este tipo están diseñadas como un conjunto de componentes distribuidos físicamente. Estos componentes pueden corresponder a divisiones funcionales, aunque no necesariamente. La mayoría de las arquitecturas descomponen el sistema en componentes fundamentales con el propósito de reducir la complejidad total que tienen que enfrentar (e.g., componentes para la percepción, la actuación, el diálogo, etc.)
- **Arquitecturas del tipo Pizarra**: este tipo de arquitectura coordina la comunicación entre los módulos y genera una estructura explícita del conocimiento. Incluye además un sistema de control, denominado "pizarra", donde publica globalmente y determina el orden de procesamiento de la información.
- **Arquitecturas con Ciclos de Decisión**: este tipo de arquitecturas están usualmente relacionadas con restricciones de ejecución en tiempo-real. Normalmente, estas arquitecturas ejecutan una operación simple (e.g. un paso de un plan) constreñida a una unidad específica de tiempo.
- **Arquitecturas Interrumpibles**: este tipo de arquitectura interrumpe el proceso de razonamiento del agente con el fin de procesar los eventos ocurridos en el ambiente. Esto significa que el agente realiza continua y asíncronamente los siguientes procesos: percibe su entorno y está alerta a los eventos novedosos que acontecen en el mismo, incorpora dichos eventos dentro de su proceso de razonamiento actual, y luego reanuda los procesos de razonamiento que estaba procesando antes de producirse la interrupción.

Resultados del Análisis: en la tabla 16.3 se presenta la clasificación por Tipo de Arquitectura. Los resultados de las frecuencias y porcentajes según el tipo de arquitectura se exponen en la tabla 16.4.

Como se puede observar en la tabla 16.4, los tipos de arquitectura empleados con mayor preferencia por los sistemas cognitivistas son: *Interrumpible*, *Planificar-luego-Compilar*, *Modular*, y *Solucionador Central*. Esto demuestra que los sistemas cognitivistas mantienen una fuerte tendencia hacia el diseño de arquitecturas compuestas por módulos con funciones pre-definidas y especializadas, delegando con frecuencia la resolución de problemas en un módulo central. Gracias a su capacidad de representar el conocimiento como unidades articulables de información (e.g., reglas de producción, chunks, etc.), estos sistemas pueden crear (por lo general) planes deliberativos en forma de secuencias de acciones concretas, los cuales pueden ser almacenados y ejecutados luego como rutinas o conductas reactivas.

Por otro lado, los sistemas emergentes presentan cierta tendencia hacia el diseño arquitectónico de tipo *Multi-Componente*, *Interrumpible*, y de *Componentes Asíncronos*, quedando clara la ausencia de un control centralizado que es reemplazado en su lugar, por

Arquitectura	Multi-capa	Basada en Marcos	Modular	Solucionador Central	Componentes Asíncronos	Planificar-luego-Compilar	Multi-Componente	Pizarra	Ciclo de Decisión	Interrumpible
SOAR				●		○			●	○
EPIC					●		●		●	○
ACT-R			●		●				●	○
ICARUS			●	○	●	○	●			○
ADAPT			●		●		○			○
BDI			○		●	○	○			○
ATLANTIS	●				●		●			●
COGNITIVA	●		○		●	●				○
DCA				○	●	○	●	●	●	○
ERE			●	○		○	○			●
HOMER		●	●	○		○				○
MAX		●		○		○			●	○
PRODIGY		○	●	●		○			●	
TETON				●		○			●	○
Theo-Agent		●				●			●	○
RALPH-MEA	●		●			○				
AAR-Subsumción	●						●			●
GWM					●		○		○	○
SASE					●		○			○
DARWIN			●		●		○			○
HUMANOID	●		●		●		○			○
CEREBUS					●		●			○
COG			●		●		○			○
KISMET			●		●					○
CLARION	●		●		○	○	●	○	○	○
LIDA			●		●	○	●	●	○	●
POLYSCHEME	●	●	●		●	○	●			○
4CAPS	●		●		○		●		○	○
NOVAMENTE			●		●		●	●		○
DUAL		●	●		●		●	○	○	○
METAFORA	●		●		●	●	●	●	○	●

Tabla A.3: Análisis Comparativo por Tipo de Arquitectura

Tipo de Arquitectura	Paradigma					
	Cognitivista		Emergente		Híbrido	
	Frec.	% Total	Frec.	% Total	Frec.	% Total
Multi-capa	3	18,75 %	1	25,0 %	5	45,45 %
Basada en Marcos	4	25,00 %	0	0,0 %	2	18,18 %
Modular	9	56,25 %	1	25,0 %	10	90,90 %
Solucionador Central	8	50,00 %	0	0,0 %	0	0,00 %
Componentes Asíncronos	8	50,00 %	3	75,0 %	11	100,00 %
Planificar-luego-Compilar	12	75,00 %	0	0,0 %	4	36,36 %
Multi-Componente	7	43,75 %	4	100,0 %	10	90,90 %
Pizarra	1	6,25 %	0	0,0 %	5	45,45 %
Ciclo de Decisión	8	50,00 %	1	25,0 %	5	45,45 %
Interrumpible	14	87,50 %	4	100,0 %	11	100,00 %

Tabla A.4: Frecuencia con que son contemplados los Tipos de Arquitectura en cada uno de los Paradigmas Cognitivos y Porcentajes del total de muestras analizadas.

la integración de múltiples componentes simples, a partir de cuya interacción emergen comportamientos más complejos en el agente.

En cuanto al paradigma híbrido se evidencia cierta inclinación hacia el diseño de arquitecturas cognitivas de tipo *Componentes Asíncronos*, *Interrumpible*, *Modular*, y *Multi-Componente*, demostrando en términos generales el intento por integrar los tipos de arquitecturas más relevantes según el paradigma cognitivo (cognitivista o emergente). Es importante notar que, al igual que en el paradigma emergente, las aproximaciones híbridas evitan incluir un solucionador de problemas central.

Por último, cabe destacar que aunque el tipo de arquitectura *Planificar-luego-Compilar* parece estar estrechamente relacionado con las aproximaciones cognitivistas, se evidencia un débil intento de las aproximaciones híbridas por definir mecanismos que integren la planificación deliberativa, el razonamiento reactivo, y el aprendizaje de conocimiento, con el fin de mejorar la reactividad del sistema mientras se mantiene la flexibilidad del mismo.

A.3. Análisis por Propiedades del Ambiente Percibido

Propiedades Analizadas: en esta sección se exponen las diferentes propiedades ambientales que pueden soportadas por las diferentes aproximaciones arquitectónicas bajo estudio. Estas propiedades pueden resumirse en 4 categorías, como se propone en (Newell et al., 1993).

- **Ambiente Dinámico vs. Estático**: los ambientes dinámicos resultan ser muy susceptibles a los cambios, especialmente a los cambios que no están bajo el control

del agente, mientras que los ambientes estáticos incluyen variables restringidas y acotadas que están bajo el control del agente. Un signo "•" indica que la arquitectura soporta los ambientes dinámicos, mientras que un espacio en blanco indica que sólo soporta ambientes estáticos.

- **Ambiente Real vs. Simulado**: un ambiente real es simplemente, y valga la redundancia, el propio mundo real (el cual es dinámico y posee una cantidad infinita de información). Así pues, esta propiedad evalúa la capacidad del agente para tratar la explosión combinatorial de posibilidades que se presentan cuando se enfrenta a demasiada información ambiental, así como su capacidad para responder en un tiempo limitado. Por otro lado, los ambientes simulados son una representación virtual del mundo con un número restringido de variables, donde los tiempos de respuesta pueden o no ser críticos. Un signo "•" indica que la arquitectura puede interactuar con ambientes reales, mientras que un espacio en blanco indica que sólo interactúa con ambientes simulados.

- **Conocimiento Imperfecto sobre el Ambiente**: cuando el conocimiento del agente sobre su entorno no es completamente preciso ni completo, se dice que es imperfecto. En los ambientes dinámicos, este tipo de conocimiento es adquirido por el agente cuando algo cambia sin estar bajo su control, invalidando parte del conocimiento previamente almacenado. Otra causa común del conocimiento imperfecto se debe a la imprecisión de los sistemas sensoriomotrices del agente, lo que conduce a la construcción de percepciones incorrectas y a la incertidumbre sobre si las acciones han sido ejecutadas correctamente. Para solucionar este tipo de problemas, los sistemas cognitivos suelen definir un amplio rango de estrategias tales como: no generar ningún modelo del ambiente, mantener la consistencia del modelo del mundo a través de los sensores y los efectores, asignar probabilidades a cada unidad de conocimiento (métodos de tratamiento de la incertidumbre), asumir que se tiene conocimiento imperfecto y tratar de corregirlo cuando afecte al sistema (conocimiento sobre los fallos), entre otras.

- **Niveles de Importancia**: en los ambientes reales y dinámicos, los eventos ocurridos se relacionan con el agente de varias formas. Algunos de estos eventos pueden ser más importantes que otros, y por tanto, resulta conveniente que el agente reaccione a dichos eventos de forma diferente de acuerdo con la importancia que tienen para sí mismo. Esta propiedad evalúa por tanto si el agente es solamente reactivo, o si tiene la capacidad de hacer una re-planificación dinámica teniendo en cuenta los niveles de importancia y prioridad de los eventos.

- **Limitaciones Computacionales**: cuando un agente interactúa con un *entorno real* sus capacidades computacionales pueden limitarse de acuerdo con varios criterios, tales como: no disponer de recursos suficientes para evaluar un número infinito (o extremadamente alto) de decisiones en paralelo, estar constreñido a tiempos de respuesta específicos, disponer de un tamaño limitado de memorias (lo cual se resuelve filtrado del conocimiento adquirido a través de meta-estrategias), entre otros. Algunas arquitecturas consideran aspectos de desempeño que tratan de algún modo las limitaciones computacionales del agente, mientras otras simplemente no realizan ningún tratamiento de tales limitaciones.

Resultados del Análisis: la tabla 16.5 presenta la clasificación de las arquitecturas según las propiedades del ambiente. En la tabla 16.6 se resumen las frecuencias con que son percibidas las distintas propiedades del ambiente en cada arquitectura analizada, según el tipo de paradigma.

A partir de la tabla 16.6, se puede observar que el paradigma cognitivista presenta una mayor tendencia hacia el diseño de arquitecturas que producen modelos internos del mundo como consecuencia de situar a los agentes en ambientes *Dinámicos*. Asimismo, se puede observar que por lo general este tipo de arquitecturas toman en cuenta las *Limitaciones Computacionales* del sistema, definiendo para ello meta-estrategias que filtran la información percibida, y mecanismos que generalizan el conocimiento adquirido (ahorrando la cantidad de memoria utilizada).

Por otra parte, el paradigma emergente demuestra una fuerte inclinación hacia el diseño de arquitecturas que permiten situar al agente en ambientes *Dinámicos* y *Reales*, escenarios idóneos para observar como emergen los comportamientos complejos del sistema ante las situaciones cambiantes del entorno. Se observa además que las aproximaciones emergentes tratan los *Niveles de Importancia* de acuerdo con el nivel de adaptabilidad definido por la arquitectura, es decir, la capacidad del agente para reaccionar de forma diferente dependiendo de la relevancia de los cambios ambientales, está relacionada directamente con su capacidad de adaptación a tales cambios. Por último, se observa que el tratamiento de las *Limitaciones Computacionales* es una consecuencia de la capacidad de auto-organización definida por la arquitectura, así por ejemplo, las limitaciones relativas a la capacidad de memoria pueden ser solucionadas como resultado de las dinámicas de creación, modificación y destrucción de elementos definidas por la propia arquitectura.

Por último, se observa que las arquitecturas de enfoque híbrido atienden generalmente todas las propiedades relacionadas con el ambiente, aunque cabe destacar que las implementaciones en entornos reales son pocas o los resultados obtenidos en las mismas son poco concluyentes (como se argumenta también en (Duch et al., 2002)). No obstante, merece la pena resaltar que existe una tendencia generalizada en los tres paradigmas hacia el diseño de agentes cognitivos inmersos en ambientes *Dinámicos* y *Reales*, lo que permite observar comportamientos más adaptativos y creíbles en lugar de estar sujetos a las limitaciones que supone la interacción con entornos virtuales.

A.4. Análisis de las Capacidades del Agente para producir Comportamientos Inteligentes

Capacidades Analizadas: el comportamiento inteligente en un agente cognitivo está soportado por una gran variedad de *capacidades*, las cuales están caracterizadas por la arquitectura y el conocimiento subyacentes. En esta sección se presentan siete capacidades inherentes en el modelado de la inteligencia general (humana), las cuales han sido también estudiadas por el área de la cognición artificial (Newell et al., 1993).

- **Percepción**: el sistema de percepción representa la interfaz dinámica con el mundo exterior, cuya responsabilidad es ser el único canal de entrada que recibe datos

Arquitectura	Dinámico vs. Estático	Real vs. Simulado	Conocimiento Imperfecto	Niveles de Importancia	Limitaciones Computacionales
SOAR	●	●	●	●	○
EPIC	●	●		○	●
ACT-R	●	●	●	●	●
ICARUS	●	●	●	●	●
ADAPT	●	●	○	●	
BDI	●	●			○
ATLANTIS	●	●	●	●	
COGNITIVA	●		●	●	●
DCA	●		●	●	●
ERE	●	●	●		○
HOMER	●	●	○		●
MAX	●	●	●	●	○
PRODIGY			●		●
TETON	●		●		○
Theo-Agent	●	●	●	●	●
RALPH-MEA	●	●	●	●	○
AAR-Subsumpción	●	●		●	
GWM	●	●	●	●	○
SASE	●	●		●	●
DARWIN	●	●	●		●
HUMANOID	●	●	●	●	
CEREBUS	●			●	●
COG	●	●	●		
KISMET	●	●	●	●	●
CLARION			○		●
LIDA	●	●	●	●	●
POLYSCHEME	●	●	○	●	○
4CAPS	●	●	●	○	●
NOVAMENTE			○	○	○
DUAL	●		○	○	○
METAFORA	●	●	●	●	●

Tabla A.5: Análisis Comparativo por Propiedades del Ambiente

nuevos capturados desde el entorno, los cuales serán luego procesados por el siste-

Característica	Paradigma					
	Cognitivista		Emergente		Híbrido	
	Frec.	% Total	Frec.	% Total	Frec.	% Total
Ambiente Dinámico	15	93,75 %	4	100,00 %	9	81,81 %
Ambiente Real	12	75,00 %	4	100,00 %	7	63,63 %
Conocimiento Imperfecto	14	87,50 %	2	50,00 %	10	90,90 %
Niveles de Importancia	11	68,75 %	3	75,00 %	9	81,81 %
Limitaciones Computacionales	14	87,50 %	3	75,00 %	9	81,81 %

Tabla A.6: Frecuencia con que son tratadas las diferentes propiedades del ambiente en cada Arquitectura analizada

ma interno (modelo cognitivo). Algunos aspectos a considerarse en la percepción son:

- **Percepción Entusiasta (PE)**: este tipo de percepción está continuamente capturando información del medio y mantiene actualizado el modelo del mundo. Supone que el coste de la percepción es bajo.
- **Percepción Lenta (PL)**: este tipo de percepción supone un coste de procesamiento alto, por tanto se ejecuta cada cierto intervalo de tiempo.
- **Monitorización de Acciones (MA)**: este tipo de percepción monitoriza constantemente las acciones del agente comparando el estado actual con el estado deseado. Este mecanismo es idóneo cuando el ambiente es incierto y cambia durante el proceso deliberativo.
- **Percepción Múltiple (PM)**: en este tipo de percepción se ejecutan de forma asíncrona y en paralelo, múltiples percepciones sobre el ambiente, contribuyendo significativamente con la eficiencia en el desempeño.

- **Actuación**: el sistema de actuación es el único canal de salida a través del cual el agente influye sobre el mundo externo como resultado de una deliberación efectuada internamente. El tiempo que tarda un agente en encontrar una secuencia de acciones que prometa alcanzar una meta es un aspecto crítico, particularmente en el dominio de problemas que requieren de una ejecución en tiempo real. Así pues, algunos aspectos a considerarse al respecto son:

  - **Reactividad (R)**: hace referencia a las respuestas reactivas con restricciones en el tiempo de procesamiento, propias de los sistemas en tiempo real.
  - **Reacción de Emergencia (RE)**: en este tipo de respuesta, la actuación normal del agente es interrumpida para dar paso a otras acciones que tienen una mayor prioridad (e.g., los actos reflejos), cuando se presentan situaciones de emergencia.
  - **Acciones Múltiples (AM)**: son acciones que se ejecutan de manera simultánea, tanto reactivas como deliberativas.
  - T**ratamiento de la Incertidumbre en la Actuación (IA)**: este tipo de actuación se emplea cuando los datos procesados por el agente poseen niveles

altos de incertidumbre y ruido, o cuando el agente interactúa con un entorno altamente cambiante provocando que las acciones en curso sean obsoletas. Usualmente, la incertidumbre es reducida mediante mecanismos perceptuales que monitorizan las acciones y los cambios ambientales cada cierto intervalo de tiempo.

- **Planificación**: se refiere al proceso de computar varios pasos de un procedimiento para la solución de un problema, antes de que se ejecute realmente. Si el agente no puede "deshacer" en su entorno los pasos de la solución propuesta, entonces la planificación se convierte en un proceso extremadamente importante. Aunque en el mundo real los pasos de un plan son irrevocables, en una simulación por ordenador no lo son. De esta manera se pueden eludir las restricciones del mundo real mediante la búsqueda de una solución completa en un mundo simulado donde se pueden revertir las acciones. Luego de encontrar una solución, se puede ejecutar ésta en el mundo real. Algunos aspectos a considerarse en la planificación se enuncian a continuación:

  - **Funciones de Utilidad (FU)**: estas funciones son empleadas para determinar la mejor decisión que pueda tomar el agente en una situación dada. Un método conveniente para lograr esto es emplear la teoría probabilística mediante la cual se pueden estimar los valores de las variables aleatorias involucradas una situación específica. El uso de las funciones de utilidad en la planificación contribuye significativamente a la eficiencia del desempeño global del sistema (Bender, 1996), puesto que determinan los operadores disponibles que puede aplicar el agente en cada paso del plan.
  - **Algoritmo en Cualquier Momento (ACM)**: este algoritmo de planificación en tiempo real, genera la mejor respuesta (por demanda) dentro del alcance de la información disponible que ha sido explorada en un tiempo permitido. La respuesta generada por el algoritmo es una aproximación a la respuesta correcta, y cuanto más tiempo se permita a la planificación, más precisa será la respuesta (Bender, 1996; Zilberstein y Russell, 1995).
  - **Re-planificación (RP)**: la evaluación de un plan generado es un proceso necesario para la re-optimización (re-planificación) del plan, mediante la cual es posible enfrentar los cambios dinámicos del ambiente y la incertidumbre.
  - **Simulación antes de la ejecución real (SE)**: la simulación ejerce una importancia significativa sobre el desempeño de la planificación, particularmente cuando se modelan entornos complejos donde la salida de las operaciones internas son acciones físicas que influyen sobre el mundo exterior, debido a que las acciones físicas toman relativamente mucho más tiempo.

- **Aprendizaje**: esta capacidad le permite al agente ejecutar cambios adaptativos en su modelo interno del mundo, de modo que la próxima vez que realice una tarea lo hará de manera más eficiente y efectiva. El aprendizaje cubre un amplio rango de fenómenos, que van desde el *refinamiento de habilidades* (e.g., mejorar la manera de ejecutar las tareas a través de la práctica) hasta la *adquisición de conocimiento*. Algunos aspectos a considerar respecto al aprendizaje se enuncian a continuación:

- **Problema de Utilidad (PU)**: mientras que las reglas aprendidas por el agente pueden reducir el tiempo de solución de un problema dirigiendo la búsqueda más cuidadosamente, por otro lado, pueden incrementar el tiempo de solución del problema forzando a los mecanismos de razonamiento a que las consideren. A fin de minimizar el número total de nodos de expansión en el espacio de búsqueda, cuantas más *reglas de control* (o reglas de meta-conocimiento) se aprendan mucho mejor. Sin embargo, a fin de minimizar el tiempo total de procesamiento que requiere solucionar un problema, se debe buscar un equilibrio entre el conocimiento adquirido y las reglas de control.
- **Soporte a la Inconsistencia del Conocimiento (SIC)**: las arquitecturas que sólo soportan *conocimiento consistente* tratan únicamente con conocimiento exacto y no contradictorio, esto es, que las unidades de conocimiento corresponden sólo a enunciados afirmativos o negativos los cuales "no" tienen asociadas medidas de incertidumbre. Por otro lado, una arquitectura que tolera la inconsistencia en el conocimiento trata generalmente sus unidades de conocimiento como un conjunto de hipótesis que compiten entre sí por ser elegidas, o como un conjunto de afirmaciones que poseen valores de confidencia variables (creencias parciales – partial beliefs). Por lo general se emplea una medida numérica como factor de credibilidad, la cual es una técnica usada en el conocimiento de control donde las reglas pueden ser clasificadas de acuerdo con su buen desempeño. Las ventajas de tolerar la inconsistencia en el conocimiento son, por un lado, que se incrementa la flexibilidad en la representación y, por otro, que se incrementa la flexibilidad en el aprendizaje y el razonamiento.
- **Aprendizaje Deliberativo (AD)**: ocurre cuando la arquitectura define un mecanismo para decidir si lo que aprende el agente posee valor o no. Esta decisión depende de varios aspectos: ¿cuándo debe realizarse el aprendizaje?, ¿qué debe ser aprendido y qué debe ser ignorado?, ¿por cuánto tiempo deben estar almacenadas las reglas aprendidas antes de ser olvidadas?, ¿cómo se realiza el aprendizaje?, etc. Algunas de las ventajas de este tipo de aprendizaje son: el ahorro de memoria como consecuencia de almacenar sólo las reglas más efectivas, el ahorro de tiempo en los procesos de inferencia y razonamiento, y la reducción en el esfuerzo que el diseñador tiene que dedicar para asegurarse que el agente aprenda sólo lo que se supone debe aprender.
- **Aprendizaje Involuntario (AI)**: esta propiedad indica que el aprendizaje se lleva a cabo de manera "automática", e.g., el agente no considera los posibles costes asociados con el aprendizaje de una determinada pieza de conocimiento. El tratamiento de todo este "extra" conocimiento puede ralentizar el desempeño del sistema debido a que todo el conocimiento debe ser explorado exhaustivamente cada vez que el agente necesite recuperar una pieza de conocimiento. Para compensar esto, las arquitecturas que emplean aprendizaje involuntario emplean mecanismos eficientes para la selección y activación de las unidades de conocimiento, de manera que el conocimiento irrelevante no es procesado y, como consecuencia, se evite la degradación del desempeño. Algunas de las ventajas de este tipo de aprendizaje son: se simplifica el proceso de aprendizaje, se minimiza el uso de recursos (e.g., tiempo y memoria)

al no almacenarse estadísticas ni categorías de las unidades de conocimiento, etc.

- **Soporte a la Degradación del Aprendizaje Múltiple (DAM)**: este problema ocurre cuando se emplean múltiples mecanismos de aprendizaje en la misma arquitectura, cuyos procesos pueden interferir unos con otros degradando el desempeño global del sistema. Para evitar estas indeseables interacciones, se define generalmente algún mecanismo mediador que selecciona el mejor componente de aprendizaje que debe activarse de acuerdo con la situación.

- **Lenguaje Natural (LN)**: las capacidades de comprensión y generación del lenguaje natural permiten a los agentes el poder comunicarse entre sí, y particularmente, con humanos. La comprensión del lenguaje natural corresponde a la interpretación de las palabras que provienen del mundo percibido, mientras que la generación de lenguaje natural corresponde al envío de palabras al mundo externo, las cuales son compiladas internamente como resultado de los procesos deliberativos del agente.
- **Cooperación y Coordinación (CC)**: algunas arquitecturas definen un nivel superior al nivel racional o deliberativo, denominado el "nivel social". El propósito principal de este nivel es mejorar el desempeño en la realización de tareas a través de la cooperación efectiva entre múltiples agentes. E particular, si los agentes cooperan con humanos, entonces necesitan las capacidades para comprender y generar lenguaje natural que sea comprensible por humanos. En cuanto a la comunicación con otros agentes, usualmente se envían datos básicos, los cuales son interpretados y tratados como cualquier otro tipo de dato que pueda percibir el agente.
- **Meta-razonamiento (MR)**: el meta-razonamiento contribuye al incremento de la racionalidad del agente en términos de una mejora en la calidad del desempeño o eficiencia del proceso de razonamiento. Como su nombre lo indica, el meta-razonamiento puede consistir de varias meta-estrategias que combinan otras capacidades tales como el Aprendizaje y la Planificación. Por ejemplo, los mecanismos de aprendizaje/planificación requieren deliberación acerca de qué aprender/ planificar, cuándo aprender/planificar, por qué aprender/planificar algo, etc., con el fin de lograr la combinación óptima empleando recursos limitados, y sorteando los múltiples factores de conflicto que influyen sobre la situación actual.

Resultados del Análisis: en la tabla 16.7 se resumen las capacidades para producir comportamientos inteligentes analizadas anteriormente, para cada una de las arquitecturas bajo estudio. El signo • indica que la capacidad correspondiente está presente en la arquitectura, y un espacio en blanco advierte que no lo está. Por otro lado, en la tabla 16.8 se presentan las frecuencias de las capacidades presentes en cada arquitectura analizada, según el tipo de paradigma.

A partir de la tabla 16.8 se puede deducir que las aproximaciones del paradigma cognitivista emplean con mayor frecuencia una *Percepción Activa o Entusiasta*, ya que ésta supone que el coste del procesamiento perceptual no es alto (esto debido, por lo general, a que el agente interactúa con entornos simulados donde la complejidad de la información percibida puede reducirse considerablemente). Adicionalmente, se observa cierta tendencia hacia el diseño de *Mecanismos Reactivos* que responden con prontitud a los estímulos percibidos.

Arquitectura	Percepción				Actuación			Planificación				Aprendizaje					LN	CC	MR
	PE	PL	MA	PM	R	RE	AM	IA	FU	ACM	RP	SE	PU	SIC	AD	AI	DAM			
SOAR			•	•	•	•	•	•		•			•			•			•	
EPIC		•			•					•	•									
ACT-R		•	•	•			•			•	•			•	•					
ICARUS	•				•				•	•					•	•				
ADAPT		•		•		•					•									
BDI	•							•			•			•						
ATLANTIS	•			•							•			•		•				
COGNITIVA	•		•		•		•				•							•		
DCA					•				•						•					
ERE	•				•				•							•				
HOMER	•			•							•						•			
MAX			•			•		•	•	•	•	•		•						•
PRODIGY													•	•	•			•	•	•
TETON								•			•					•				
Theo-Agent		•			•											•	•			•
RALPH-MEA	•				•	•			•	•	•	•								
AAR-Subsumpción	•	•	•	•	•	•	•													
GWM	•			•								•			•					
SASE	•			•			•					•			•					
DARWIN		•		•			•							•	•					
HUMANOID	•		•	•	•						•			•	•		•			•
CEREBUS	•		•	•			•							•	•					
COG	•			•			•							•		•		•	•	
KISMET	•			•	•															
CLARION	•				•		•		•				•		•		•		•	•
LIDA	•	•		•	•		•			•	•	•	•	•	•			•	•	•
POLYSCHEME			•	•	•		•				•	•		•	•	•			•	•
4CAPS	•				•	•							•			•	•			
NOVAMENTE		•			•	•							•	•				•	•	
DUAL	•		•	•	•		•	•	•				•				•	•	•	•
METAFORA	•		•	•		•	•	•	•	•	•	•	•		•		•	•	•	•

Tabla A.7: Análisis Comparativo por Capacidades del Agente

Característica		Paradigma					
		Cognitivista		Emergente		Híbrido	
		Frec.	% Total	Frec.	% Total	Frec.	% Total
Percepción	PE	7	43,75 %	3	75,0 %	9	81,81 %
	PL	4	25,00 %	1	25,0 %	2	18,18 %
	MA	4	25,00 %	1	25,0 %	5	45,45 %
	PM	5	31,25 %	4	100,0 %	8	72,72 %
Actuación	R	9	56,25 %	1	25,0 %	8	72,72 %
	RE	4	25,00 %	1	25,0 %	3	27,27 %
	AM	3	18,75 %	3	75,0 %	7	63,63 %
	IA	4	25,00 %	0	0,0 %	2	18,18 %
Planificación	FU	5	31,25 %	0	0,0 %	3	27,27 %
	ACM	6	37,50 %	0	0,0 %	2	18,18 %
	RP	10	62,50 %	0	0,0 %	4	36,36 %
	SE	2	12,50 %	2	50,0 %	3	27,27 %
Aprendizaje	PU	2	12,50 %	0	0,0 %	6	54,54 %
	SIC	5	31,25 %	1	25,0 %	7	63,63 %
	AD	6	37,50 %	3	75,0 %	7	63,63 %
	AI	6	37,50 %	0	0,0 %	3	27,27 %
	DAM	2	12,50 %	0	0,0 %	4	36,36 %
Lenguaje Natural	LN	2	12,5 %	0	0,0 %	3	27,27 %
Cooperación	CC	3	18,75 %	0	0,0 %	5	45,45 %
Meta-razonamiento	MR	3	18,75 %	0	0,0 %	6	54,54 %

Tabla A.8: Frecuencia con que son consideradas las diferentes Capacidades del Agente

Una característica importante que cabe resaltar es que las arquitecturas del paradigma cognitivista demuestran, por lo general, un empleo sistemático de estrategias de *Planificación*, característica que se observa con menos frecuencia en los sistemas de inspiración emergente e incluso en los de inspiración híbrida. Por tanto, se evidencia en los sistemas cognitivistas en general, una preferencia marcada hacia la generación de planes estructurados y orientados hacia la satisfacción de los estados internos del agente (e.g., metas, intenciones, emociones, expectativas, motivadores, deseos, etc.).

En cuanto al aprendizaje, los sistemas cognitivistas intentan mantener la *Consistencia en el Conocimiento* a través de mecanismos de representación fuertemente estructurados, los cuales evitan la ambigüedad o la contradicción en las unidades de conocimiento. En cuanto a los mecanismos de representación suelen emplearse los sistemas basados en marcos, las redes semánticas, los guiones, las reglas de producción, las ontologías, etc.

Por otro lado, en cuanto a las características de *Lenguaje Natural*, *Cooperación y Coordinación*, y *Meta-razonamiento*, en términos generales, son pocas las contribuciones que se han desarrollado al respecto en las arquitecturas analizadas de inspiración cogni-

tivista. Aunque se han realizado algunos aportes de forma aislada, no se han incorporado como capacidades intrínsecas del paradigma.

Si se observan ahora las arquitecturas del paradigma emergente, es posible determinar que existe una mayor tendencia hacia la inclusión de las capacidades de *Percepción Entusiasta* y *Percepción Múltiple* (en cuanto al tipo de Percepción), lo cual se corresponde con las capacidades de *Reacción Emergente* y *Actuación Múltiple* (en cuanto al tipo de Actuación). Los sistemas emergentes continuamente perciben el mundo exterior empleando múltiples mecanismos de percepción en paralelo, algunos de ellos generan respuestas reactivas, mientras que otros son la base para gestar procesos cognitivos más complejos.

En cuanto a la capacidad de aprendizaje, las aproximaciones emergentes suelen emplear un *Aprendizaje Deliberativo* basado en técnicas de aprendizaje por refuerzo y aprendizaje no supervisado. En cuanto a las características de *Lenguaje Natural*, *Cooperación y Coordinación*, y *Meta-razonamiento*, ninguno de los sistemas emergentes analizados demuestra haber realizado un aporte significativo al respecto.

Por otro lado, respecto a las arquitecturas híbridas se pueden observar múltiples mecanismos de percepción: *Percepción Entusiasta*, *Percepción Múltiple*, y *Monitorización de Acciones*. En cuanto a la actuación, la tendencia de los diseños arquitectónicos apunta hacia la inclusión de las capacidades de *Reactividad* y *Actuación Múltiple*. Por otro lado, respecto a la capacidad de planificación se identificaron algunos aportes relacionados con la *Re-Planificación* de tareas, característica que se ha heredado principalmente de los sistemas cognitivistas. En cuanto a la capacidad de Aprendizaje se puede observar una marcada combinación de enfoques, en especial para las características de *Consistencia del Conocimiento* y de *Aprendizaje Deliberativo*: comúnmente, se combinan técnicas de aprendizaje supervisado y no supervisado, aprendizaje por ejemplos, aprendizaje por imitación, aprendizaje por descubrimiento, aprendizaje por refuerzo, entre otros.

Por último, cabe resaltar que las arquitecturas híbridas han demostrado un gran esfuerzo por incorporar características tales como: procesamiento del *Lenguaje Natural* (implementado a través de módulos de diálogo interactivos), mecanismos de *Cooperación y Coordinación* (implementados mediante enfoques multi-agente, técnicas de negociación, inteligencia de enjambre – *swarm intelligence* –, entre otras), y procesos de *Meta-razonamiento* (implementados a través de reglas de control, meta-estrategias, etc.). Estos aportes realizados por el paradigma híbrido generan un valor agregado respecto a los paradigmas cognitivista y emergente típicos.

A.5. Análisis de las Propiedades Cognitivas del Agente

Propiedades Analizadas: las propiedades del agente determinan qué tan competente puede ser éste en el desempeño de distintos tipos de tareas cognitivas. Como se verá más adelante, las *propiedades* distan de las capacidades del agente, siendo estas primeras, características mucho más detalladas que describen la arquitectura del agente. Esto quiere decir, por ejemplo, que diferentes arquitecturas pueden exhibir una capacidad para "aprender", aunque cada una lo haga en instantes diferentes, con tipos de datos diferen-

tes, y sobre cuestiones diferentes; estas sutiles distinciones permiten describir las propiedades cognitivas del agente. A continuación se presentan cuatro tipos de propiedades principales propuestas por (Newell et al., 1993): las propiedades relacionadas con el conocimiento, el razonamiento, el aprendizaje, y la organización.

- **Propiedades relacionadas con la Representación del Conocimiento:**
  - **Declarativo vs. Procedimental**: esta propiedad indica el tipo de conocimiento que es almacenado. El conocimiento procedimental se refiere al conocimiento sobre cómo hacer las cosas y está relacionado con el aprendizaje de destrezas. Por otro lado, el conocimiento declarativo se refiere al conocimiento que especifica los objetos, sus propiedades y sus relaciones con otros objetos.
  - **Representación Global vs. Específica**: el conocimiento global es necesario para representar una visión unificada del mundo, aunque el mantenimiento de esta visión puede resultar muy complicada. Por otro lado, el conocimiento específico mantiene una representación parcial (y a veces insuficiente) del mundo, con lo cual el agente debe hacer inferencias de aquello que no conoce.
  - **Simbolismo vs. Sub-simbolismo**: una arquitectura simbólica manipula símbolos para representar su conocimiento. Una de las principales ventajas de las representaciones simbólicas es que gran parte del conocimiento humano es simbólico, por tanto, el modelarlo dentro de la base de conocimiento de la arquitectura resulta mucho más fácil. Además, los procesos artificiales de razonamiento simbólico resultan análogos a los realizado por los humanos, con lo cual el agente puede explicar y comunicar más fácilmente sus decisiones. Por otra parte, las arquitecturas sub-simbólicas no emplean símbolos para representar su conocimiento, sino datos de más bajo nivel (e.g., bits). Como ventajas de las arquitecturas sub-simbólicas se puede mencionar, por un lado, que son mucho más rápidas, lo cual es importante en ambientes dinámicos, y por otro, que suelen ser representaciones mucho menos costosas.
  - **Conocimiento de Control de Caja Negra vs. Caja Blanca**: en el esquema de Caja Blanca la mayoría de las reglas de control pueden inspeccionar y decidir sobre otras reglas, permitiendo al agente razonar sobre su propio comportamiento mediante la observación de la cadena de inferencia aplicada (esta es una característica propia del meta-razonamiento). Por otra parte, el esquema de Caja Negra es más resistente a cambiar aquello que ya conoce el agente; las reglas deben ser producidas de forma atómica, y por tanto facilitar su comprensión, sin embargo, este esquema no ofrece una interpretación sobre las propias actuaciones del agente, impidiendo así la posibilidad de incluir un nivel reflexivo dentro de la arquitectura.
  - **Representación Uniforme vs. Representación No Uniforme del Conocimiento**: las arquitecturas con representación uniforme almacenan el conocimiento en una base da datos central empleando para ello un determinado formato. En contraposición, los métodos no uniformes son usados con frecuencia en arquitecturas débilmente acopladas que almacenan conocimiento específico, y no de uso general, mediante diversos mecanismos de representación del conocimiento.

- **Propiedades relacionadas con el Razonamiento**
  - **Razonamiento en Tiempo Real vs. Razonamiento Sin Restricciones Temporales**: el razonamiento en tiempo real es la capacidad que tiene el agente para percibir el ambiente y generar una respuesta adecuada en un tiempo crítico. Generalmente, en este tipo de razonamiento los sensores deben estar capacitados para recibir información incompleta de forma dinámica. Una aproximación convencional en las arquitecturas de tiempo real es el empleo de módulos dedicados a la reactividad, o el empleo de un mecanismo de razonamiento cuya base de conocimiento esté conformada por piezas lo suficientemente pequeñas que permitan la interrupción en cualquier paso del proceso de inferencia. Por tal motivo, cuanto menos tiempo se requiera para razonar, mejor será la eficiencia del sistema.
- **Propiedades relacionadas con el Aprendizaje:**
  - **Aprendizaje Deliberativo vs. Involuntario**: el aprendizaje deliberativo define cuándo el agente debe o no aprender algo. Como ventajas del aprendizaje deliberativo se puede mencionar: dispone el uso de la memoria sólo para almacenar las unidades de conocimiento más efectivas, economiza el tiempo de procesamiento y de inferencia puesto que las decisiones se toman sobre una base de conocimiento refinado, y disminuye la cantidad de código empleado para validar que el agente aprenda sólo lo que se supone debe aprender. En el aprendizaje involuntario la arquitectura almacena automáticamente todas las reglas que crea. Algunas de sus ventajas son: emplea mecanismos de aprendizaje mucho más simples, y economiza tiempo y memoria al no almacenar ningún tipo de información sobre las jerarquías y categorías de las unidades de conocimiento almacenadas.
  - **Aprendizaje Monótono vs. No Monótono**: el aprendizaje monótono se da cuando un agente no puede aprender ningún conocimiento que contradiga o revalide lo que ya sabe, y su ventaja principal es que mantiene la consistencia del conocimiento adquirido. En el aprendizaje no monótono, el agente puede adquirir conocimiento que contradiga lo que ya sabe, para lo cual emplea mecanismos de reemplazamiento del conocimiento obsoleto y de validación de nuevas creencias. Su ventaja principal es que incrementa la aplicabilidad de la arquitectura en dominios reales.
- **Propiedades relacionadas con la Organización**
  - **Centralidad vs. Modularidad**: las arquitecturas centralizadas son aquellas en las cuales una unidad (módulo central) realiza la mayor parte de la función cognitiva del agente. Algunas ventajas de este tipo de arquitectura son: desaparecen los problemas de comunicación entre componentes, no hay necesidad de duplicar la funcionalidad a través de los módulos, y puede lograrse relativamente una mayor eficiencia del sistema. Por otro lado, las arquitecturas modulares asignan diferentes tareas a cada una de las unidades (módulos) que posee el agente y, por tanto, el control de la arquitectura se encuentra distribuido en lugar de estar centralizado. En las arquitecturas modulares, el logro de las metas de alto nivel se realiza mediante la descomposición del

problema en sub-metas, las cuales son luego asignadas al módulo más apropiado para su resolución y, finalmente, se emplea un mecanismo que integra los resultados de cada módulo. Algunas ventajas de la modularidad son: las arquitecturas son fáciles de modificar y mantener, y pueden ser además analizadas con mayor comprensiblidad.

Resultados del Análisis: en la tabla 16.9 se resumen las propiedades del agente analizadas anteriormente, para cada una de las arquitecturas bajo estudio. El signo • indica que se cumple la primera parte de la dicotomía de la propiedad, mientras que el signo ◦ indica que se cumple la parte contrapuesta, por ejemplo, para la propiedad de representación del conocimiento declarativo vs. procedimental, un signo • indica que el conocimiento es representado de forma declarativa, mientras que un sign ◦ indica que el conocimiento es representado de forma procedimental. Si los dos valores de la propiedad están presentes en la arquitectura, se presentarán entonces los dos signos simultáneamente. Por otro lado, si se muestra un espacio en blanco es porque la arquitectura no soporta dicha propiedad o porque la información al respecto es insuficiente y no es posible determinar si la propiedad es empleada por la arquitectura o no. Por otra parte, en la tabla 16.10 se presentan las frecuencias de las propiedades del agente presentes en cada arquitectura analizada analizada, según el tipo de paradigma.

A partir de los resultados presentados en la tabla 16.10, se puede observar que el paradigma cognitivista emplea con frecuencia representaciones *procedimentales* y *uniformes* del conocimiento, las cuales almacenan por lo general, información de tipo *simbólica*. Esto deriva de la inclinación del paradigma hacia el diseño de sistemas basados en reglas de producción, donde es garantizada la homogeneidad de la base de conocimiento. En cuanto a la propiedad del Razonamiento, se observa una clara tendencia hacia el diseño de arquitecturas que interactúan en *Tiempo Real* con el ambiente. Respecto a la propiedad de Aprendizaje, las arquitecturas cognitivistas analizadas no demuestran una preferencia clara hacia el diseño de sistemas que empleen meta-conocimiento para determinar qué debe aprender el agente (aprendizaje *deliberativo* vs. *involuntario*), o acerca de la monotonicidad del aprendizaje (aprendizaje *monótono* vs. *no monótono*). Por último, el enfoque cognitivista demuestra una cierta tendencia hacia el desarrollo de arquitecturas *modulares* que no poseen un control centralizado de sus funciones cognitivas.

En cuanto al paradigma emergente, se observa una preferencia patente hacia la representación *sub-simbólica* de conocimiento *procedimental* y *específico* (e.g., los modelos conexionistas de redes neuronales). Como consecuencia de ello, el agente difícilmente puede realizar inferencias sobre las unidades atómicas de conocimiento (e.g., una sola neurona artificial), con lo cual el conocimiento de control de estos sistemas es categorizado como un esquema de *caja negra*. Se observa además, que la representación del conocimiento suele ser *no uniforme*, debido a que se emplean con frecuencia diferentes técnicas y mecanismos para almacenar el conocimiento adquirido. Desde el punto de vista del Razonamiento, al igual que en el paradigma cognitivista, se observa una fuerte tendencia hacia el desarrollo de arquitecturas que interactúan con el ambiente en *tiempo real*. En cuanto a la propiedad del Aprendizaje, se evidencia cierta preferencia por métodos de aprendizaje *involuntarios* que almacenan automáticamente la información percibida por el agente, y luego, mediante los procesos auto-organizativos, el sistema de-

Arquitectura	Representación del Conocimiento						Razonamiento	Aprendizaje		Organización
	Declarativo vs. Procedimental	Global vs. Específico	Simbólico vs. Sub-simbólico	Caja Negra vs. Caja Blanca	Consistencia vs. Inconsistencia	Uniforme vs. No uniforme	Tiempo Real vs. Sin restricciones temporales	Deliberativo vs. Involuntario	Monótono vs. No Monótono	Centralidad vs. Modularidad
SOAR	○	●	●	●		●	●	○	○	●
EPIC	○	○	●	●		●	●			●
ACT-R	●○	●	●○	●	●	●	●	●	○	○
ICARUS	○	●	●	○		●	●	●	●	○
ADAPT	●○	○	●	○	●	●	○	●		○
BDI	●○	○	●	●	●	●	○			○
ATLANTIS	●○	○	●	●	●	○	●	○	●	○
COGNITIVA	●○	●○	●	●	●	●	○			●
DCA	○	●	●	○		●	●	●	○	○
ERE	○	○	●	●		○	●	○	●	○
HOMER	○	●	●	○		○	●	○	●	○
MAX	○	●	●	○	○	●	●		○	
PRODIGY	○	●	●	○	●	●	○	●	●	○
TETON	●○	●	●	○		●	●		○	●
Theo-Agent	○	●	●	○		●	●	●	○	●
RALPH-MEA	●○	●	●	○		●	●			○
AAR-Subsumpción	○	○	○	●		○	●			○
GWM	○	○	○	●	●	○	●	○		○
SASE	○	○	○	●	○	○	●	●	○	●
DARWIN	○	○	○	●	○	●	○	○	○	○
HUMANOID	●	●	●○			●		●	●	○
CEREBUS	●○	○	●○	○	○	○	●	○	○	○
COG	●	○	●○	●	●	●	○			○
KISMET	●	●	●○	●	○		○	●	●	○
CLARION	●○	●○	●○	○	●	●	○	●	○	○
LIDA	●○	●○	●○	○	○	○	●	●	○	○
POLYSCHEME	●○	○	●○	●	○	○	○	○	○	○
4CAPS	○	●○	●○	○	●	●	●	○	○	○
NOVAMENTE	○	○	●○	●	●	○	○	○	●	○
DUAL	●○	●○	●○	○	○	○	○	●	●	○
METAFORA	●○	●○	●○	○	○	○	●	●	○	○

Tabla A.9: Análisis comparativo por Propiedades del Agente

cide qué se debe conservar y qué se debe olvidar. Por último, en cuanto a la propiedad de Organización, es predominante la característica *Modular*, donde cada componente realiza funciones asíncronas e independientes y no existe un mecanismo central de control, sino que, por el contrario, el control se encuentra distribuido en los múltiples componentes del sistema.

Característica		Paradigma					
		Cognitivista		Emergente		Híbrido	
		Frec.	% Total	Frec.	% Total	Frec.	% Total
Representación del Conocimiento	Declarativo	7	43,75 %	0	0,0 %	9	81,81 %
	Procedimental	16	100,00 %	4	100,0 %	8	72,72 %
	Global	11	68,75 %	0	0,0 %	7	63,63 %
	Específico	6	37,50 %	4	100,0 %	9	81,81 %
	Simbólico	16	100,00 %	0	0,0 %	11	100,00 %
	Sub-simbólico	1	6,25 %	4	100,0 %	11	100,00 %
	Caja Negra	7	43,75 %	4	100,0 %	4	36,36 %
	Caja Blanca	9	56,25 %	0	0,0 %	6	54,54 %
	Consistente	6	37,50 %	1	25,0 %	4	36,36 %
	Inconsistente	1	6,25 %	2	50,0 %	6	54,54 %
	Uniforme	13	81,25 %	1	25,0 %	4	36,36 %
	No Uniforme	3	18,75 %	3	75,0 %	6	54,54 %
Razonamiento	Tiempo Real	12	75,00 %	3	75,0 %	4	36,36 %
	No Tiempo Real	4	25,00 %	1	25,0 %	6	54,54 %
Aprendizaje	Deliberativo	6	37,50 %	0	0,0 %	6	54,54 %
	Involuntario	4	25,00 %	3	75,0 %	4	36,36 %
	Monótono	7	43,75 %	1	25,0 %	4	36,36 %
	No Monótono	5	31,25 %	1	25,0 %	6	54,54 %
Organización	Centralidad	5	31,25 %	1	25,0 %	0	0,00 %
	Modularidad	10	62,50 %	3	75,0 %	11	100,00 %

Tabla A.10: Frecuencia con que son consideradas las diferentes Propiedades del Agente.

Respecto al paradigma híbrido, las aproximaciones analizadas presentan una gran variedad de propiedades. En relación a la propiedad de Representación del Conocimiento se observa claramente una tendencia a hibridizar las diferentes técnicas: por ejemplo, el almacenamiento de unidades de conocimiento *declarativas* y *procedimentales*, soportado por representaciones *simbólicas* y *sub-simbólicas*, etc. A diferencia de los dos paradigmas anteriores, el híbrido demuestra una mayor preferencia por el diseño de arquitecturas cuyos mecanismos de razonamiento no están constreñidos por restricciones temporales. Esto se debe a que los sistemas híbridos deben emplear intervalos de tiempo más largos a fin de poder integrar y sincronizar (en cada ciclo cognitivo) las diferentes representaciones simbólicas y sub-simbólicas. Se observa además, que respecto a las propiedades del Aprendizaje existe una sutil inclinación hacia el diseño de mecanismos de aprendizaje *deliberativo* y *no monótono*. Finalmente, la propiedad de Organización en las aproximaciones híbridas se enfoca hacia el diseño de sistemas *modulares*, donde son integrados múltiples componentes de propósito general cuya función es determinada en tiempo de ejecución como consecuencia de los procesos adaptativos y auto-organizativos del sistema.

A.6. Resultados Generales del Análisis Comparativo

A continuación se presentan de manera concisa algunas observaciones sobre los resultados del estudio comparativo.

Las arquitecturas de enfoque cognitivista simulan directamente la funcionalidad de las características inteligentes que se pretenden conseguir (e.g., el aprendizaje, la percepción, el lenguaje natural, etc.). Estas características son diseñadas como módulos especializados que se comunican con un solucionador central, el cual se encarga de coordinar todos los procesos cognitivos. Los sistemas cognitivistas manipulan unidades de información simbólica, a partir de las cuales representan el conocimiento, infieren nuevos hechos (razonamiento), y crean planes a futuro. La principal fortaleza de estos sistemas es su capacidad de explicar, de forma comprensible para un humano, la cadena inferencial producida que subyace cada decisión tomada por el agente. Por otro lado, su principal debilidad es la eminente necesidad de pre-definir (o pre-programar) las habilidades cognitivas del agente, lo cual se evidencia por ejemplo, en sus sistemas representacionales rígidos y estereotipados, o en el empleo de técnicas de aprendizaje monótono.

Por otro lado, las arquitecturas emergentes intentan no sólo simular sino emular los procesos cognitivos, inspirándose para ello en el funcionamiento de bajo nivel del cerebro (e.g., las neuronas). Así pues, las arquitecturas bajo este enfoque suelen procesar información sub-simbólica que es procesada de forma distribuida por múltiples componentes simples, de cuya interacción se espera que emerja espontáneamente el comportamiento inteligente. Usualmente, estos sistemas establecen un fuerte acoplamiento con el ambiente a través de la personificación del agente, con lo cual se incrementa la aplicabilidad de la arquitectura en dominios reales y dinámicos. La principal fortaleza de este tipo de arquitecturas es la autonomía que otorgan al agente, la cual ya no deriva directamente de una perspicaz programación sino de los propios procesos auto-organizativos del sistema. Por otro lado, la principal desventaja es la considerable cantidad de tiempo y/o recursos computacionales que puede requerir en muchos casos, la emergencia del comportamiento inteligente en el agente.

Por último, las arquitecturas híbridas, como era de esperarse, combinan lo mejor de ambas perspectivas. Así pues, se evidencia una clara intención por fusionar niveles de conocimiento de diferente naturaleza (simbólica y sub-simbólica), diversos mecanismos de representación del conocimiento, variadas técnicas de aprendizaje, mecanismos de meta-cognición, entre otros, con el objetivo de construir modelos más fiables y robustos para la resolución de problemas. Por tanto, una de las razones por las que los Sistemas Inteligentes Híbridos han llegado a ser tan populares en la comunidad científica en las últimas dos décadas, se debe a su capacidad de tratar con problemas complejos del mundo real de forma más adaptativa y convergente de lo que usualmente lo hacen los sistemas puramente cognitivistas y emergentes.

Apéndice B

Histogramas de Frecuencia y Gráficos Q-Q

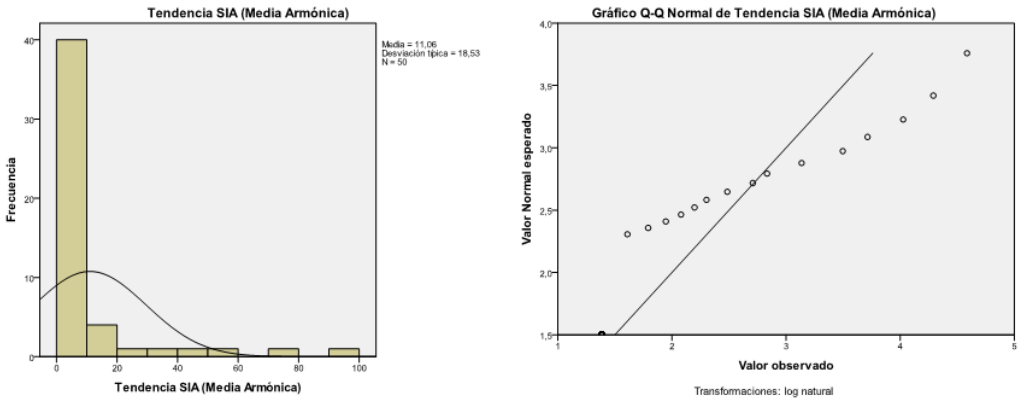


Figura B.1: Escenario 1 - SIA: Histograma y Gráfico Q-Q

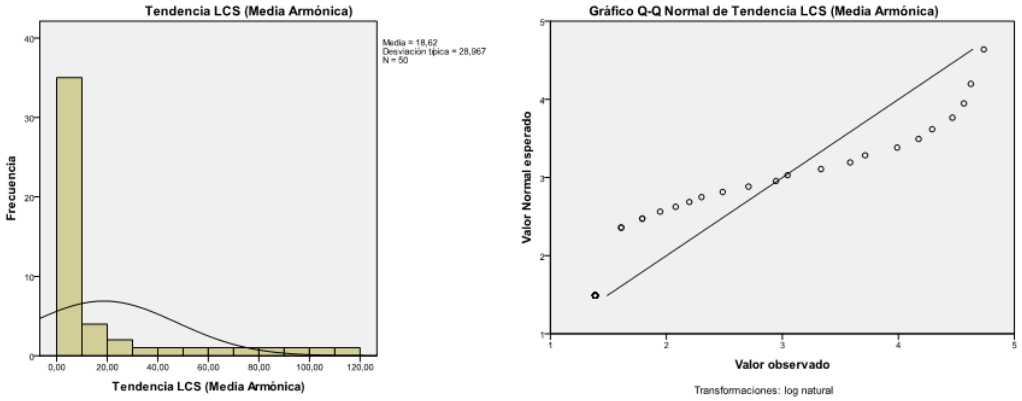


Figura B.2: Escenario 1 - LCS: Histograma y Gráfico Q-Q

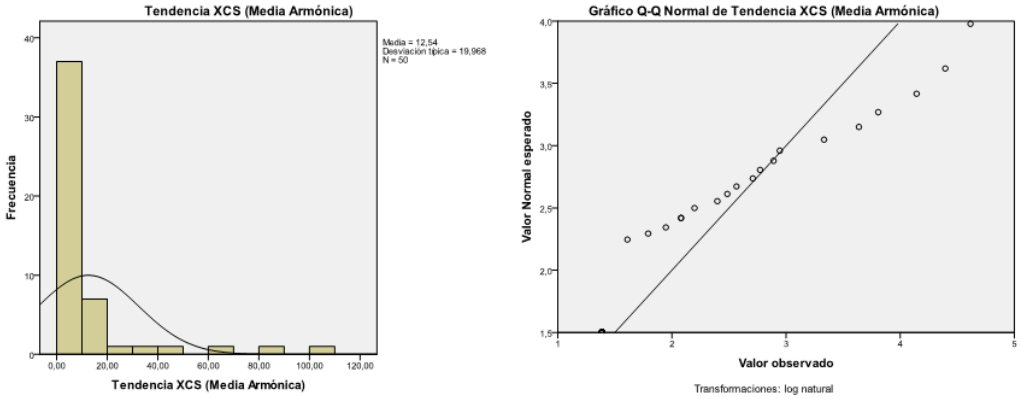


Figura B.3: Escenario 1 - XCS: Histograma y Gráfico Q-Q

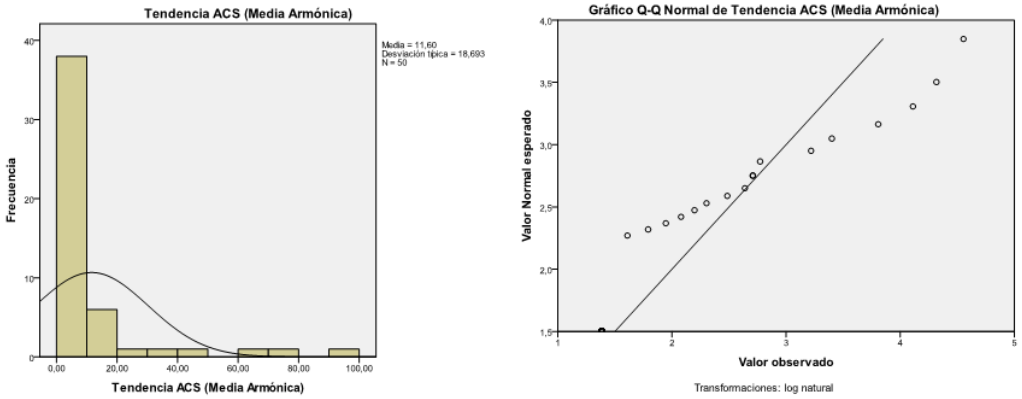


Figura B.4: Escenario 1 - ACS: Histograma y Gráfico Q-Q

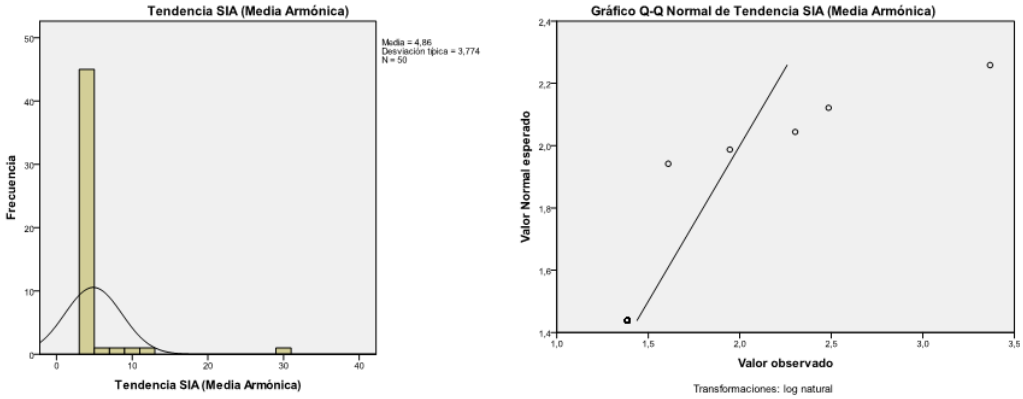


Figura B.5: Escenario 2 - SIA: Histograma y Gráfico Q-Q

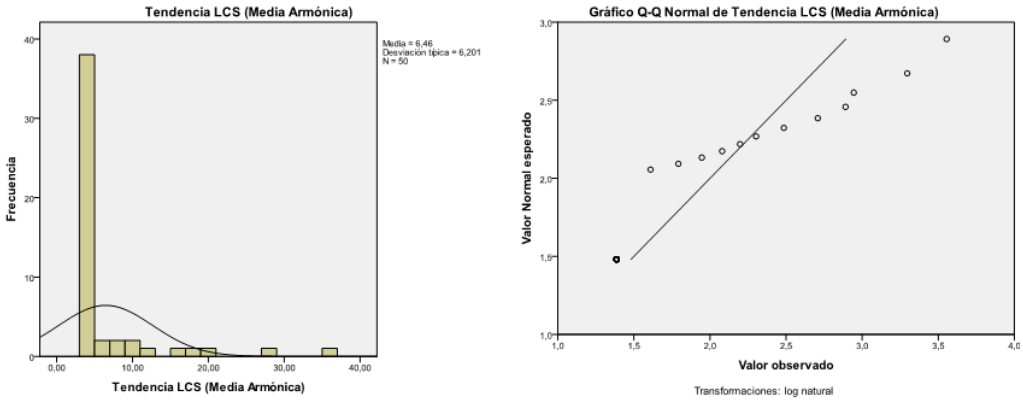


Figura B.6: Escenario 2 - LCS: Histograma y Gráfico Q-Q

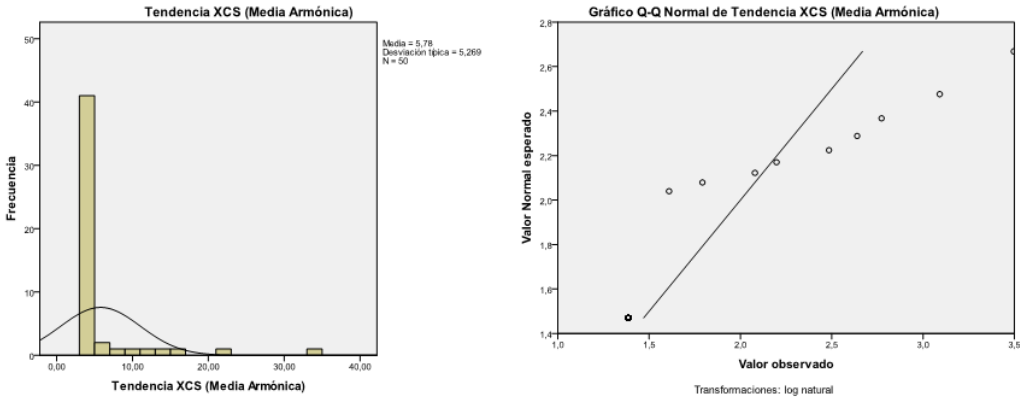


Figura B.7: Escenario 2 - XCS: Histograma y Gráfico Q-Q

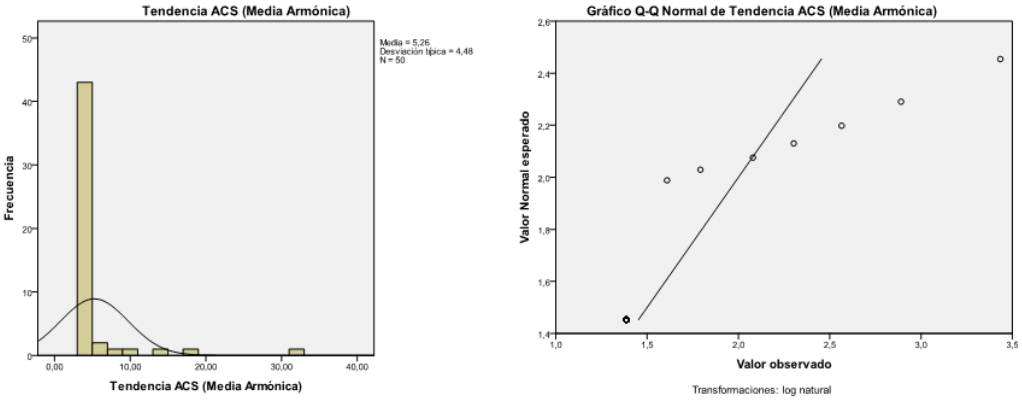


Figura B.8: Escenario 2 - ACS: Histograma y Gráfico Q-Q

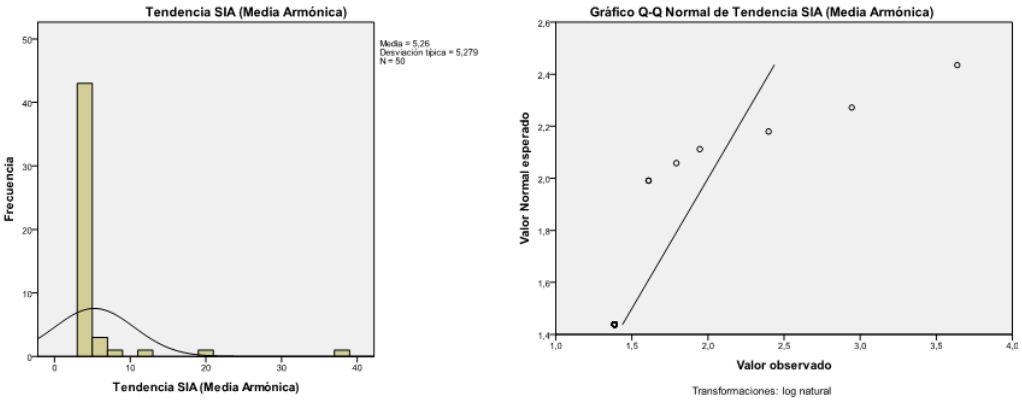


Figura B.9: Escenario 3 - SIA: Histograma y Gráfico Q-Q

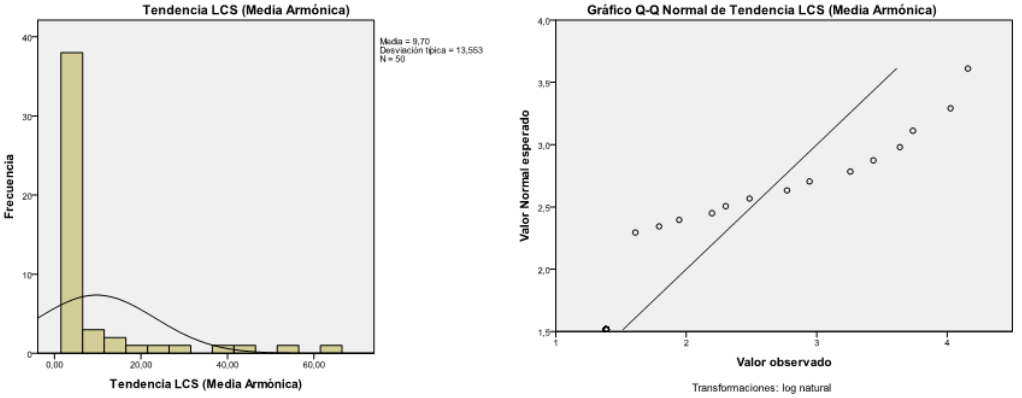


Figura B.10: Escenario 3 - LCS: Histograma y Gráfico Q-Q

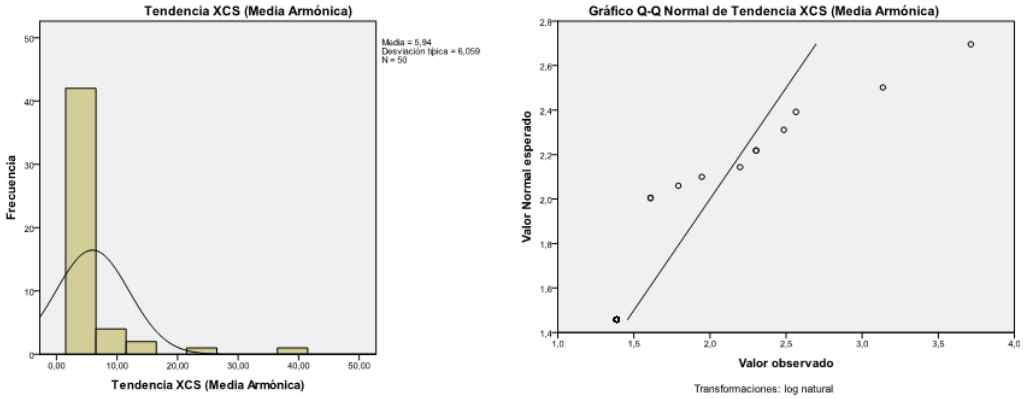


Figura B.11: Escenario 3 - XCS: Histograma y Gráfico Q-Q

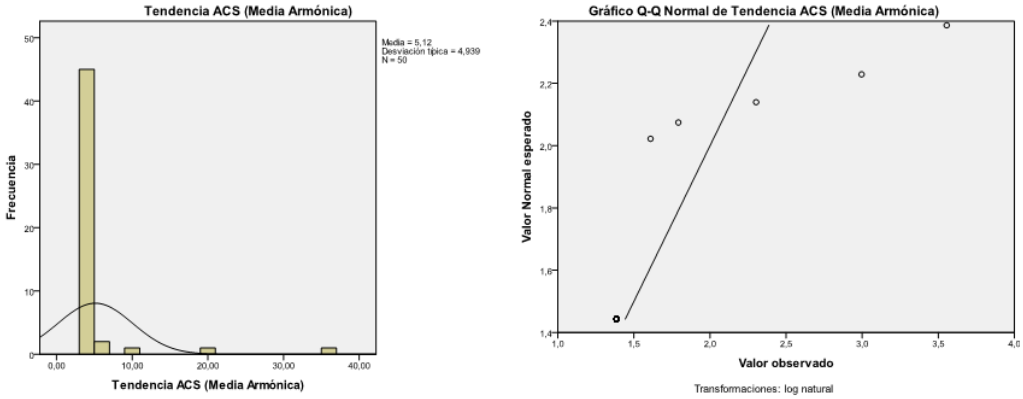


Figura B.12: Escenario 3 - ACS: Histograma y Gráfico Q-Q

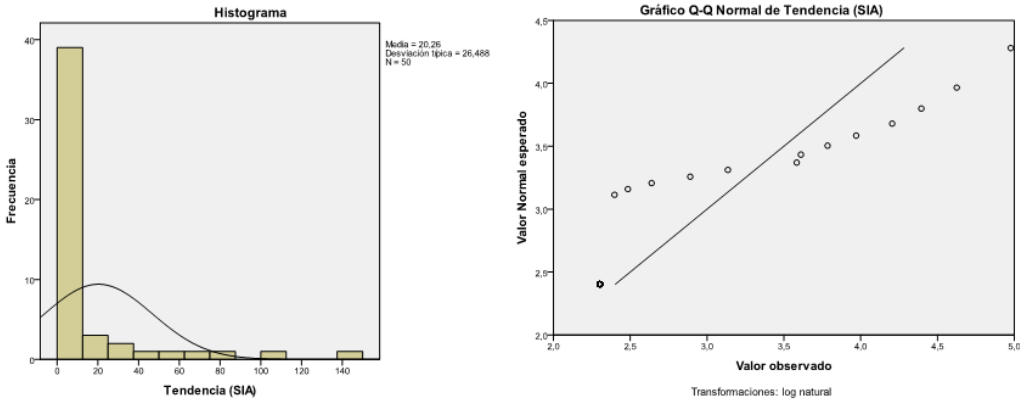


Figura B.13: Escenario 4 - SIA: Histograma y Gráfico Q-Q

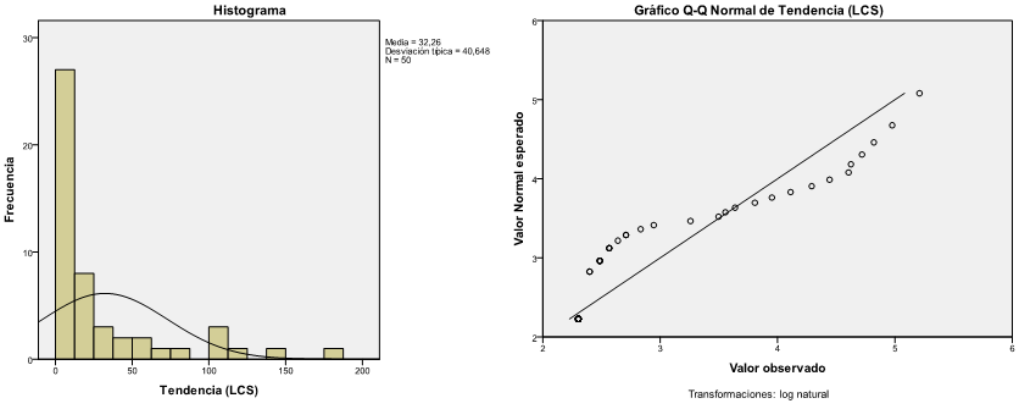


Figura B.14: Escenario 4 - LCS: Histograma y Gráfico Q-Q

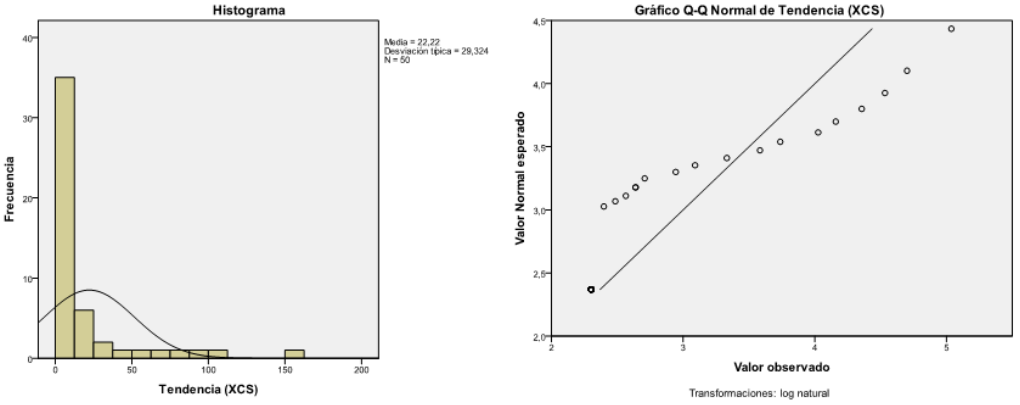


Figura B.15: Escenario 4 - XCS: Histograma y Gráfico Q-Q

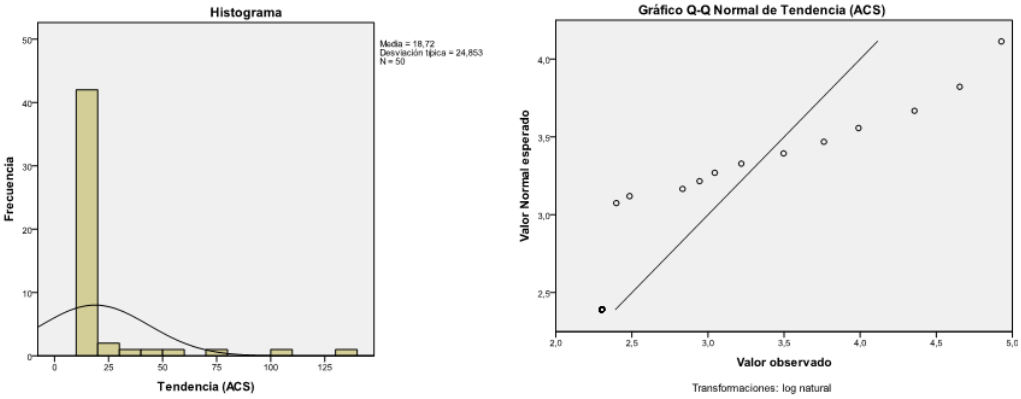


Figura B.16: Escenario 4 - ACS: Histograma y Gráfico Q-Q

Parte V

Bibliografía

Bibliografía

Ackley, D. y Littman, M. (1992) Interactions between learning and evolution. *Artificial Life II*, págs. 487–507.

Al-Maqaleh, B. y Bharadwaj, K. (2005) Genetic programming approach to hierarchical production rule discovery. *Transactions on Engineering, Computing and Technology*, 6: págs. 271–274.

Alain, C., Woods, D. y Knight, R. (1998) A distributed cortical network for auditory sensory memory in humans. *Brain Res*, 812: págs. 23–37.

Aleksander, I. (1990) Neural systems engineering: Towards a unified design discipline? *Comput. Control Eng. J.*, 1(6): págs. 259–265.

Alexander, P., Braubach, L. y Lamersdorf, W. (2005) Jadex: A BDI Reasoning Engine. *Multi-Agent Programming*.

Alpaydin, E. (2005) *Introduction to Machine Learning–Adaptive Computation and Machine Learning*. MIT Press, New York.

Alsmadi, M., Omar, K., Noah, S. y Almarashdah, I. (2009) Performance Comparison of Multi-layer Perceptron (Back Propagation, Delta Rule and Perceptron) algorithms in Neural Networks. *IEEE International Advance Computing Conference*, págs. 296–299.

Amos, A. (2000) A computational model of information processing in the frontal cortex and basal ganglia. *Journal of Cognitive Neuroscience*, 12: págs. 505–519.

Amunts, K., Kedo, O., Kindler, M., Pieperhoff, P., Mohlberg, H., Shah, N., Habel, U., Schneider, F. y Zilles, K. (2005) Cytoarchitectonic mapping of the human amygdala, hippocampal region and entorhinal cortex: intersubject variability and probability maps. *Anat Embryol*, 210(5–6): págs. 343–352.

Anderson, J. (1993a) *Neurocomputing. Foundations of Research*. MIT Press, New York.

Anderson, J. (1993b) *Rules of the Mind*. Lawrence Erlbaum Associates, NJ.

Anderson, J. y Lebiere, C. (1998) *The atomic components of thought*. Lawrence Erlbaum Associates, NJ.

Anderson, J. y Lebiere, C. (2003) Newell test for a theory of cognition. *Behavioral and Brain Science*, 26: págs. 287–637.

Anderson, J. y Qin, Y. (2004) An integrated theory of the mind. *Psychology Review.*, 111(4): págs. 1036–1060.

Anderson, J. R. (1996) ACT: A simple theory of complex cognition. *Amer. Psychol.*, 51: págs. 355–365.

Andrade, E. (2001) Las interrelaciones entre Genotipo/Fenotipo/Ambiente: Una aproximación semiótica al debate evolución-desarrollo. *Revista Colombiana de Filosofía de la Ciencia*, 6(12–13): págs. 109–142.

Anwar, A. y Franklin, S. (2003) Sparse distributed memory for conscious software agents. *Cognitive Systems Research*, 4(4): págs. 339–354.

Arbib, M. (1980) *From Schema Theory to Language*. Oxford University Press, New York.

Arbib, M. (1998) Schema theory. *Encyclopedia of Artificial Intelligence. 2nd Edition.*, 2: págs. 1427–1443.

Arteaga, G. y Pimienta, H. (2004) Sobre la organización columnar de la corteza cerebral. *Revista colombiana de psiquiatría*, 33: págs. 76–101.

Arthur, W. B. (1997) *The Economy as an Evolving Complex System II*. Addison–Wesley, California.

Arvelakis, A., Reczko, M., Stamatakis, A., Symeonidis, A. y Tollis, I. (2005) Using treemaps to visualize phylogenetic trees. *Proc. 6st International Symposium on Biological and Medical Data Analysis, (ISBMDA)*, págs. 283–293.

Ashby, F. G. y Waldron, E. M. (2000) The neuropsychological bases of category learning. *Current Directions in Psychological Science*, 9: págs. 10–14.

Ashby, W. R. (1956) *An introduction to cybernetics*. Chapman and Hall, London.

Ba-Alwi, F. y Bharadwaj, K. (2005) Discovery of production rules with fuzzy hierarchy. *Second World Enformatica Congress (WEC'05)*, págs. 77–80.

Baars, B. J. (1988) *A Cognitive Theory of Consciousness*. Cambridge University.

Baars, B. J. (1997) *In the theater of consciousness: The workspace of the mind*. Oxford University Press, New York Oxford.

Baars, B. J. (2002) The conscious assess hypothesis: Origins and recent evidence. *Trends in Cognitive Science*, 6(1): págs. 47–52.

Baddeley, A. D. (1991) *Working Memory*. Churchill Livingstone, Oxford.

Baddeley, A. D. (1993) *Working memory and conscious awareness*. In Theories of memory, ed. A. Collins, Howe: Erlbaum.

Bain, A. (1855) *Los sentidos y la inteligencia*. Springer, New York.

Baldwin, J. M. (1896) A new factor in evolution. *Amer. Naturalist*, 30(354): págs. 441–451.

Bapi, R. S., Pammi, V. y Miyapuram, K. P. (2005) Investigation of sequence learning: A cognitive and computational neuroscience perspective. *Current Science*, 89: págs. 1690–1698.

Baron-Cohen, S. (1995) *Mindblindness*. MA: MIT Press, Cambridge.

Bartlett, P. y Downs, T. (1992) Using Random Weights to Train Multilayer Networks of Hard-Limiting Units. *IEEE Transactions on Neural Networks*, 3(2): págs. 202–210.

Barto, A., Bradtke, S. y Singh, S. (1995) Learning to Act using Real-Time Dynamic Programming. *In Artificial Intelligence*, 72: págs. 81–138.

Bartz-Beielstein, T. (2006) Experimental Research in Evolutionary Computation: The New Experimentalism. *Natural Computing Series*.

Beer, R. (2004) Autopoiesis and Cognition in the Game of Life. *ARTIFICIAL LIFE*, 10: págs. 10–309.

Belew, R. (1989) When both individuals and populations search: Adding simple learning to the genetic algorithm. *In 3th Intern. Conf. on Genetic Algorithms*, págs. 15–22.

Bender, E. (1996) *Mathematical Methods In Artificial Intelligence*. IEEE Computer Society Pres.

Benjamin, D., Lyons, D. y Lonsdale, D. (2004) Adapt: A cognitive architecture for robotics. *Cognitive Modeling*.

Benjamin, D., Lonsdale, D., Lyons, D. y Patel, S. (2008) Using Cognitive Semantics to Integrate Perception and Motion in a Behavior-Based Robot. *Learning and Adaptive Behaviors for Robotic Systems, ECSIS Symposium*, 0: págs. 77–82.

Benson, S. y Nilsson, N. (1995) Reacting planning and learning in an autonomous agent. *En Machine Intelligence*. Clarendom Press.

Bergfelt, N. y Linaker, F. (2002) Self-organized modulation of a neural robot controller. *Proc. International Joint Conference on Neural Networks - IJCNN*, págs. 495–500.

Berretta, R., Cotta, C. y Moscato, P. (2003) *Enhancing the performance of memetic algorithms by using a matching-based recombination algorithm*. M. Resende and J. Pinho de Sousa, Boston MA.

Berry, D. y Broadbent, B. (1988) Interactive tasks and the implicit-explicit distinction. *Psychology*, 79: págs. 251–272.

Berry, D. y Dienes, D. (1993) *Implicit learning: theoretical and empirical issues*. Hove, L. Erlbaum Associates.

Bersini, H. (2002) Self-Assertion versus self-recognition: a tribute to Francisco Varela. *En ICARIS*, págs. 107–112.

Bertsekas, D. y Tsitsiklis, J. (1996) *Neuro-Dynamic Programming*. Athena Scientific.

Bickhard, M. (2000) Autonomy, function and representation. *Artificial Intelligence, Special Issue on Communication and Cognition*, 17(3–4): págs. 111–131.

Blackmore, S. (2003) The evolution of meme machines. *Ontopsychology and Memetics*, págs. 233–240.

Blank, D., Lewis, J. y Marshall, J. (2005) The Multiple Roles of Anticipation in Developmental Robotics.

Bohl, E. y Lancaster, P. (2006) Implementation of a markov model for phylogenetic trees. *Journal of Theoretical Biology*, 239: págs. 324–333.

Bongard, J. (2002) Evolving Modular Genetic Regulatory Networks. *Proc. IEEE Congress of Evolutionary Computation, (CEC)*, págs. 1872–1877.

Bonin, M. W., v. (1971) On columnar arrangement of nerve cells in cerebral cortex. *Brain Res*, 27(1): págs. 1–9.

Booker, L., Goldberg, D. y Holland, J. (1989) Classifier Systems and Genetic Algorithms. *Artificial Intelligence*.

Bourgine, P. y Stewart, J. (2004) Autopoiesis and Cognition. *Artificial Life*, 10: págs. 327–345.

Bozarth, M. (1994) Pleasure systems in the brain. *En Pleasure: The politics and the reality*. D.M. Warburton (ed.), New York.

Brachman, R. J. (2002) Systems that know what they're doing. *IEEE Intelligent Systems*, 17: págs. 67–71.

Bratman, M. E., Israel, D. y Pollack, M. (1988) Plans and Resource–Bounded Practical Reasoning. *Computational Intelligence*, págs. 349–355.

Breazeal, C. (2000) *Sociable machines: Expressive social exchange between humans and robots*. Tesis Doctoral, MIT, Cambridge.

Breazeal, C. (2003) Emotion and sociable humanoid robots. *Int. J. Human–Computer Studies*, págs. 119–155.

Broadbent, D. (1958) *Perception and Communication*. Pergamon Press, Lóndres.

Brooks, R. y Stein, L. (1994) Building Brains for Bodies. *Autonomous Robotics*, 1: págs. 7–25.

Brooks, R. A. (1986) A robust layered control system for a mobile robot. *IEEE J. Robot. Autom.*, RA–1(1): págs. 14–23.

Brooks, R. A. (1991) How to build complete creatures rather than isolated cognitive simulators. *Architectures for Intelligence*, págs. 225–239.

Brown, A. (1987) Metacognition, executive control, self-regulation and other more mysterious mechanisms. *Metacognition, motivation and understanding*, págs. 65–116.

Browne, A. y Sun, R. (2001) Connectionist inference models Neural Networks. *Psychological Review*, 14: págs. 1331–1335.

Bruner, J., Goodnow, J. y Austin, J. (1956) *A study of thinking*. Wiley, New York.

Bui, T., Heylen, D., Poel, M. y Nijholt, A. (2001) Generation of Facial Expressions from Emotion Using a Fuzzy Rule Based System. *En Proceedings of the 14th Australian Joint Conference on Artificial Intelligence*.

Burghart, C., Mikut, R., Stiefelhagen, R., Asfour, T., Holzapfel, H., Steinhaus, P. y Dillman, R. (2005) A cognitive architecture for a humanoid robot: A first approach. *in Proc. IEEE–RAS Int. Conf. Humanoid Robots*, págs. 357–362.

Burghart, C., Mikut, R. y Holzapfel, H. (2007) Cognition-oriented Building Blocks of Future Benchmark Scenarios for Humanoid Home Robots. *En AAAI Workshop Evaluating Architectures for Intelligence*.

Burnet, F. M. (1978) Clonal Selection and After. *In Theoretical Immunology*, págs. 63–85.

Butz, A., Goldberg, D. y Stolzmann, C. (2002a) The anticipatory classifier system and genetic generalization. *Natural Computing: an international journal*, 1(4): págs. 427–467.

Butz, M. y Wilson, S. (2000) An Algorithmic Description of XCS. *En Conference on Evolutionary Computation*, págs. 253–272. Springer-Verlag.

Butz, M., Sigaud, O. y Gerard, P. (2002b) Internal models and anticipations in adaptive learning systems. *In Proceedings of the Workshop on Adaptive Behavior in Anticipatory Learning Systems*, págs. 1–23.

Butz, M., Sigaud, O., Pezzulo, G. y Baldassarre, G. (eds.) (2007) *Anticipatory Behavior in Adaptive Learning Systems: From Brains to Individual and Social Behavior*. Springer-Verlag.

Byrne, M. D. (2003) Cognitive architecture. *in The Human–Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, págs. 97–117.

Byrne, M. D. y Anderson, J. R. (1998) Perception and action. *The atomic components of thought*, págs. 167–200.

Cañamero, D. (1997) Modeling Motivations and Emotions as a Basis for Intelligent Behavior. *Proceedings of the First International Symposium on Autonomous Agents (Agents'97)*, págs. 148–155.

Calvin, W. (1996) *The cerebral code: Thinking a thought in the mosaics of the mind*. MIT Press.

Calvin, W. (2001) *How brains think. Evolving Intelligence, Then and Now*. Science Masters Series, Harvard.

Carbonell, J., Knoblock, C. y Minton, S. (1991) Prodigy: An integrated architecture for planning and learning. *Architectures for intelligence*, págs. 241–278.

Cassimatis, N. (2007) Adaptive Algorithmic Hybrids for Human-Level Artificial Intelligence. *Advances in Artificial General Intelligence*.

Cassimatis, N. L. (2002) *A cognitive arhitecture for integrating multiple representation and inference schemes*. Tesis Doctoral, Massachusetts Institute of Technology.

Castillo, P., Prieto, M., Castellano, J. y Merelo, J. (2001) Diseño de Redes Neuronales Artificiales mediante Algoritmos Evolutivos. *Revista Iberoamericana de Inteligencia Artificial*, 5(14): págs. 36–44.

N. de Castro, L. y Timmis, J. (2002) *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer, Massachusetts.

Catucci, G., Abbattista, F., Gadaleta, R., Guaccero, D. y Semeraro, G. (2004) Empathy: A Computational Framework for Emotion Generation. *En WSC*, págs. 265–277.

Cavalli-Sforza, L. (1967) Phylogenetic analysis: Models and estimation procedures. *Evolution*, 21(3): págs. 550–570.

Chalup, S. (2001) Issues of neurodevelopment in biological and artificial neural networks. *Proc. Fifth Biannual Conference on Artificial Neural Networks and Expert Systems*, págs. 40–45.

Charniak, E. y McDermott, D. (1985) *Introduction to Artificial Intelligence*. Adison–Wesley, Massachusetts.

Chechik, G., Meilijson, I. y Ruppin, E. (1999) Neuronal regulation: A mechanism for synaptic pruning during brain maturation. *Neural Computation*, 11(8): págs. 2061–2080.

Chengkai, L., Chen-chuan, K., Chang, I. y Ilyas, F. (2006) Supporting Ad-hoc Ranking Aggregates. *En In SIGMOD*, págs. 61–72. ACM Press.

Christensen, W. D. y Hooker, C. A. (2000) An interactivist–constructivist approach to intelligence: Self–directed anticipative learning. *Philosophical Psychology.*, 13(1): págs. 5–45.

Clark, A. y Grush, R. (1999) Towards a cognitive robotics. *Adaptive Behavior*, 7(1): págs. 5–16.

Clark, A. y Karmiloff-Smith, A. (1993) The cognizer's innards: a psychological and philosophical perspective on the development of thought. *Mind and language*, 8(4): págs. 487–519.

Clark, H. (1935) The conflicting psychologies of learning: a way out. *Psychological Review*, 42: págs. 491–516.

Clark, H. H. (1994) Managing problems in speaking. *Speech Communication*, 15: págs. 243–250.

Cleeremans, A., Destrebecqz, A. y Boyer, M. (1998) Implicit learning: news from the front. *Trends in Cognitive Science*, 2(10): págs. 406–416.

Cohen, W. W. (1995) Fast effective rule induction. *En Proceedings of the Twelfth International Conference on Machine Learning*, págs. 321–336.

Conway, M. A. (2001) *Sensory-perceptual episodic memory and its context: autobiographical memory*. Oxford University Press.

Cramer, N. L. (1985) A representation for the adaptive generation of simple sequential programs. *En Proceedings of the First International Conference on Genetic Algorithms and Their Applications*. J. J. Grefenstette.

Crina, G. y Ajith, A. (2007) *Hybrid Evolutionary Algorithms*. Studies in Computational Intelligence, Boston MA.

Cybenko, G. (1989) Approximation by superpositions of a sigmoidal function. *Math control, signals and systems*, 2(4).

Czarn, A., MacNish, C., Vijatan, K., Turlach, B. y Gupta, R. (2004) Statistical exploratory analysis of genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(4): págs. 405–421.

Damasio, A. (1994) *Descartes' Error. Emotion, Reason, and the Human Brain*. Putnam.

Dasgupta, D. (2002) *Artificial Immune Systems and Their Applications*. Springer Verlag, New York.

Davachi, L., Mitchell, J. y Wagner, A. (2003) Multiple routes to memory: Distinct medial temporal lobe processes build item and source memories. *Proc. Natl. Acad. Sci.*, 100: págs. 2157–2162.

Davidor, Y. (1991) Epistasis variance: A view-point on GA-hardness. *Foundations of Genetic Algorithms*, págs. 23–35.

Dawkins, R. (1992) *The Selfish Gene*. Oxford University Press, New York.

DeJong, K. (1992) Are genetic algorithms function optimizers? *Proc. of the 2nd Int. Conf. on Parallel Problem Solving from Nature, (PPSN)*, págs. 3–14.

DeJong, K. A. (1975) *An Analysis of the Behavior a Class of Genetic Adaptive Systems*. Tesis Doctoral, University of Michigan.

Delgado, J., Ferrús, A., Mora, F. y Rubia, F. (1998) *Manual de neurociencia*. Síntesis, Madrid.

Delius, J. (1991) The Nature of Culture. *En The Timbergen Legacy*, págs. 71–99. Chapman and Hall, London.

Demazeau, Y. y Tavares da Silva, J. (2002) Vowels Co-ordination Model. *En International Joint Conference on Autonomous Agents and Multiagent Systems*, págs. 1129–1136.

Dennett, D. (1991) *Consciousness Explained*. Little, Brown, Boston.

Dennett, D. C. (1987) *The Intentional Stance*. MA: MIT Press, Cambridge.

Desai, R. y Patil, R. (1996) SALO: Combining Simulated Annealing and Local Optimization for Efficient Global Optimization. *En Proceedings of the 9th Florida AI Research Symposium FLAIRS*, págs. 233–237.

Dietterich, T. (1998) Approximate Statistical Test For Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10(7): págs. 1895–1923.

Dill, K. y Perkowski, M. (2001) Baldwinian learning utilizing genetic and heuristic algorithms for logic synthesis and minimization of incompletely specified data with Generalized Reed–Muller forms. *Journal of Systems Architecture*, 47(6): págs. 477–489.

Dodd, W. (2005) *The Design of Procedural, Semantic, and Episodic Memory Systems for a Cognitive Robot*. Tesis Doctoral, Vanderbilt University.

Dooley, K. (2002) Organizational Complexity. *International Encyclopedia of Business and Management*, págs. 5013–5022.

Dorer, K. (2005) Extended Behavior Networks for Behavior Selection in Dynamic and Continuous Domains. *Cognitive Science*, págs. 40–45.

Drummond, M., J., B. y Kedar, S. (1991) The entropy reduction engine: integrating planning, scheduling and control. *SIGART Bull.*, 2(4): págs. 61–65.

Duan, X., H. Yu (2007) Hybrid Ant Colony Optimization using Memetic Algorithm for Traveling Salesman Problem. *En Proceedings of the IEEE symposium on Approximate Dynamic Programming and Reinforcement Learning*, págs. 92–95.

Duarte, O. (1997) *UNFUZZY: software para el diseño, simulación e implementación de sistemas de lógica difusa*. Tesis Doctoral, Universidad Nacional de Colombia.

Dubois, V. y Mohamed, Q. (2002) Concept learning with approximation: Rough version spaces. *Rough Sets and Current Trends in Computing: Proceedings of the Third International Conference, RSCTC 2002*, págs. 239–246.

Duch, W., Adamczak, R. y Grabczewski, K. (2001) A new methodology of extraction, optimization and application of crisp and fuzzy logical rules. *IEEE Transactions on Neural Networks*, 12: págs. 277–306.

Duch, W., Oentaryo, R. y Pasquier, M. (2002) Cognitive Architectures: Where Do We Go from Here?

Eccles, J. (1966) Long–loop reflexes from the spinal cord to the brain stem and cerebellum. *Atti della accademia medica lombarda*, 21(3): págs. 870–878.

Edelman, G. M. (1987) *Neural Darwinism: The Theory of Neuronal Group Selection*. Basic Books, New York.

Eggenberger, P. (2003) Genome-physics interaction as a new concept to reduce the number of genetic parameters in artificial evolution. *Proc. IEEE 2004 Congress of Evolu-tionary Computation - (CEC)*, págs. 191–198.

Eiben, A. y Smith, J. (2003) *Introduction to Evolutionary Computing*. Springer.

Ekman, P. (1992) Are there basic emotions? *Psychological Review*, 99(3): págs. 550–553.

Ekman, P. y Oster, H. (1982) Are there basic emotions? *Emotion in the Human Face*, págs. 147–174.

Erikson, E. (1973) *Childhood and Society*. W.W. Norton, New York.

Erol, K., Hendler, J. y Nau, D. (1994) HTN Planning: Complexity and Expressivity. *En In Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94*, págs. 1123–1128. AAAI Press.

Erp, M. y Schomaker, L. (2000) Variants Of The Borda Count Method For Combining Ranked Classifier Hypotheses. *En in the Seventh International Workshop on Frontiers in Handwriting Recognition*, págs. 443–452.

Eshelman, L. (1991) The CHC Adaptive Search Algorithm: how to Have Safe Search When Engaging in Nontraditional Genetic Recombination. *Foundations of GENETIC ALGORITHMS*, págs. 265–283.

Faulkner, A. y Wei, J. (2000) Performance of a real-time multi-layer perceptron algorithm for the extraction of the voicing and voice fundamental frequency information. *Speech, Hearing and Language*, págs. 296–299.

Ferber, J. (1999) *Multi-Agent System: An Introduction to Distributed Artificial Intelligence*. Addison Wesley Longman.

Ferreira, C. (2001) Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. *Complex Systems*, 13(2): págs. 87–129.

Ferreira, C. (2004) Designing Neural Networks Using Gene Expression Programming. *En 9th Online World Conference on Soft Computing in Industrial Applications*.

Ferreira, C. (2006) *Gene Expression programming: mathematical modeling by an artificial intelligence*. Springer-Verlag.

Firby, R. J. (1989) *Adaptive Execution in Dynamic Domains*. Tesis Doctoral, Yale University, Department of Computer Science.

Flavell, J. (1976) *Metacognitive aspects of problem solving*. Resnick, L. (Ed.).

Floreano, D. y Urzelai, J. (2000) Evolutionary robots with on-line self-organization and behavioral fitness. *Neural Networks*, 13: págs. 431–443.

Frank, M. J., Loughry, B. y O'Reilly, R. C. (2000) Interactions between frontal cortex and basal ganglia in working memory: A computational model.

Franklin, S. (2006) The LIDA architecture: Adding new modes of learning to an intelligent, autonomous, software agent. *En Int. Conf. on Integrated Design and Process Technology*.

Franklin, S. y Graesser, A. (1996) Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. *En International Workshop on Agent Theories, Architectures, and Languages*.

Franklin, S., Baars, B., Ramamurthy, U. y Ventura, M. (2005) The Role of Consciousness in Memory. *Brains, Minds and Media*, 1: págs. 1224–1245.

Freeman, W. J. (2003) Evidence from human scalp electroencephalograms of global chaotic itinerancy. *Chaos*, 13: págs. 1067–1077.

Freeman, W. J. y Núñez, R. (1999) Reclaming Cognition: the primacy of action, intention and emotion. *Consciousness Studies*, 6: págs. 9–19.

Freisleben, B. y Merz, P. (1996) A Genetic Local Search Algorithm for Solving Symmetric and Asymmetric Traveling Salesman Problems. *En In Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, págs. 616–621. IEEE Press.

Frijda, N. (1994) Emotions require cognitions, even if simple ones. *The Nature of Emotion*, págs. 197–202.

Furnkranz, J. (1994) Pruning Methods for Rule Learning Algorithms. *En Proceedings of the 4th International Workshop on Inductive Logic Programming*, págs. 321–336.

Fuster, J. M., Bodner, M. y Kroger, J. K. (2000) Cross-modal and cross-temporal association in neurons of frontal cortex. *Nature*, 405: págs. 347–351.

Futuyma, D. J. (1986) *Evolutionary Biology*. Sinuaer Asociates, Massachusetts.

G., T. y F., B. (1999) Review: Technology of agents of software. *Ci. Inf.*, 28(3): págs. 302–309.

García, S. y Herrera, F. (2008) Design of Experiments in Computational Intelligence: On the Use of Statistical Inference. *En 3rd International Workshop on Hybrid Artificial Intelligent Systems(HAIS08)*, tomo 5271 de *Lecture Notes on Computer Science*, págs. 4–14. Springer-Verlag.

García, S., Molina, D., Lozano, M. y Herrera, F. (2007) Un estudio experimental sobre el uso de test no paramétricos para analizar el comportamiento de los algoritmos evolutivos en problemas de optimización. *En Proceedings Congreso Español sobre Metaheurísticas and Algoritmos Evolutivos y Bioinspirados*, págs. 275–285.

García, Y. (2002) *Procesos markovianos de decisión en el modelamiento de agentes racionales*. Universidad Nacional de Colombia, Bogotá, Colombia.

García Sevilla, J. (1997) *Psicología de la Atención*. Síntesis, Madrid.

Gat, E. (1997) On Three–Layer Architectures. *Artificial Intelligence and Mobile Robots*, págs. 195–210.

Georgeff, M., Pell, B., Pollack, M., Tambe, M. y Wooldridge, M. (1999) The Belief–Desire–Intention Model of Agency. *Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures and Languages (ATAL–98)*, págs. 1–10.

Gobet, F., Lane, P., Croker, S., Cheng, P., Jones, G., Oliver, I. y Pine, J. (2001) Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, 5: págs. 236–243.

Goertzel, B. (1997) *From Complexity to Creativity*. NY: Plenum Press.

Goertzel, B. (2006) *The Hidden Pattern*. BrownWalker Press.

Gold, K., Doniec, M., Crick, C. y Scassellati, B. (2009) Robotic Vocabulary Building Using Extension Inference and Implicit Contrast. *Artificial Intelligence Journal*, 173(1): págs. 145–166.

Goldberg, D. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, Massachusetts.

Goldberg, D. y Deb, K. (1991) A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*, 1: págs. 69–93.

Goldspink, C. y Kay, R. (2003) Organizations as self–organizing and sustaining systems: a complex and autopoietic systems perspective. *International Journal of General Systems*, 32(5): págs. 459–474.

Gorges-Schleuter, M. (1997) Asparagos96 and the traveling salesman problem. *In Proceedings of 1997 IEEE International Conference on Evolutionary Computation*, págs. 171–174.

Grand, S. (2000) *Creation. Life and how to make it*. Weindenfeld and Nicolson, London, UK.

Graybiel, A. y Kimura, M. (1995) Adaptive neural networks in the basal ganglia. *In J. C. Houk, J. L. Davis, and D. G. Beiser (Eds.), Models of information processing in the basal ganglia*, págs. 103–116.

Guerra, A., Fallah, A. y Soldano, H. (2004) Learning in BDI Multi-agent Systems. *En Computational Logic in Multi-Agent Systems*.

Guthrie, E. (1930) Conditioning as a principle of learning. *Psychological Review*, 37: págs. 412–428.

Haeckel, E. (1883) *The History of Creation*. Kegan Paul Eds., London.

Hageman, J., Wehrens, R., Sprang, H. y Buydens, L. (2003) A hybrid Genetic Algorithm-Tabu Search approach for optimising multilayer optical coatings. *En Anal. Chim. Acta (special CAC 2002 issue)*, tomo 490, págs. 211–222.

Halgren, E., Boujon, J., C. ad Clarke, Wang, C. y Chauvel, P. (2002) Rapid distributed fronto-parieto-occipital processing stages during working memory in humans. *Cerebral Cortex*, 12: págs. 710–728.

Harvey, I. (1893) The puzzle of the persistent question marks: a case study of genetic drift. *In 5th International Conference on Genetic Algorithms*, págs. 15–22.

Hayes, B., Collinot, A., Dabija, V. y Drakopoulos, J. (1995) An Architecture for Adaptive Intelligent Systems.

Hayes-Roth, B., Washington, R., Hewett, R., Hewett, M. y Seiver, A. (1989) Intelligent real–time monitoring and control. *International Joint Conference on Artificial Intelligence*.

Haygood, R. y Bourne, L. (1970) Attribute and rule learning aspects of conceptual behavior. *Psychological Review*, 28(3): págs. 249–276.

Haykin, S. (1999) *Neural Networks, a comprehensive foundation*. Prentice Hall, second edition, New York.

Hebb, D. O. (1949) *The Organization of Behavior: A neuropsychological theory*. Wiley, New York.

Heidegger, M. (1996) *Being and Time*. trans. by Joan Stambaugh, Albany: State University of New York Press.

de la Herrán, M. (2005) *Arena Sensible*. REDcientífica.

Hesslow, G. (1994) Will neuroscience explain consciousness? *J. Theoret. Biol.*, 171: págs. 29–39.

Hesslow, G. (2002) Conscious thought as simulation of behaviour and perception. *Trends Cognitive Sci.*, 6(6): págs. 242–247.

Hikosaka, O., Nakahara, H., Rand, M., Sakai, K., Lu, Z. y Nakamura, K. (2001) Parallel neural networks for learning sequential procedures. *Trends in Neuroscience*, 22: págs. 464–471.

Hinton, G. y Nowlan, S. (1997) How learning can guide evolution. *Complex Systems*, 1: págs. 492–505.

Hochberg, Y. (1998) A sharper Bonferroni procedure for multiple tests of significance. *Biometrika*, 75: págs. 811–818.

Hodges, A. (1983) *Alan Turing: The enigma*. Burnett Books, New York.

Hoffmann, J. (1993) *Anticipation and cognition: The function of anticipations in human behavioral control and perception*. Goettingen, Germany: Hogrefe.

Hofstadter, D. R. y Mitchell, M. (1994) The Copycat Project: A model of mental fluidity and analogy-making. *In Advances in connectionist and neural computation theory*, 2: págs. 1067–1077.

von Hofsten, C. (2003) On the development of perception and action. *Handbook of Developmental Psychology*, 1: págs. 114–140.

Holland, J. (1998) *Emergence, from chaos to order*. Oxford university press.

Holland, J., Holyoak, K., Nisbett, R. y Thagard, P. (1989) *Induction, Processes of Inference, Learning and Discovery*. MIT Press, Massachusetts.

Holley, J. (2004) First investigations of dream-like cognitive processing using the anticipatory classifier system. Inf. téc., Somerset, UK.

Hollnagel, E. y Woods, D. D. (1999) Cognitive systems engineering: New wind in new bottles. *Human–Computer Studies*, 51: págs. 339–356.

Horling, B., Lesser, V., Vincent, R., Wagner, T., Raja, A., Zhang, S., Decker, K. y Garvey, A. (1999) *The Taems White Paper*. University of Massachusetts.

Hornik, K., Stinchcombe, M. y White, H. (1989) Multilayer feedforwardnetworks are universal approximators. *Neural networks*, 2(5).

Horswill, I. (2001) Tagged behavior–based systems: Integrating cognition with embodied activity. *IEEE Intell. Syst.*, págs. 30–38.

Horswill, I. (2007) Psychopathology, narrative, and cognitive architecture (or: why NPCs should be just as screwed up as we are). *En AAAI Fall Symposium on Intelligent Narrative Technologies*.

Houk, J. C. y Wise, S. P. (1995) Distributed modular architectures linking basal ganglia, cerebellum, and cerebral cortex: Their role in planning and controlling action. *Cerebral Cortex*, 2: págs. 95–110.

Huber, M. (2007) Agent Autonomy: Social Integrity and Social Independence. *En Proceedings of the Fourth Information Technology: New Generations*, págs. 282–288.

Hull, J. J., Commike, A. A. y Ho, T. K. (1990) Multiple algorithms for handwritten character recognition. *In Proc. 1st Int. Workshop on Frontiers in Handwriting Recognition*, págs. 117–124.

Hullermeier, E. (2001) Similarity-based inference as evidential reasoning. *International journal of approximate reasoning*, 26(2): págs. 67–100.

Imbert, R. (2005) *Una arquitectura cognitiva multinivel para agentes con comportamiento influido por características individuales y emociones, propias y de otros agentes*. Tesis Doctoral, Universidad Politécnica de Madrid.

Ingber, L. (1993) Simulated Annealing: Practice versus Theory. *En Mathematical Computer Modelling*, tomo 11, págs. 29–57.

Isa, I., Omar, S., Saad, Z. y Osman, M. (2010) Performance Comparison of Different Multilayer Perceptron Network Activation Functions in Automated Weather Classification. *En 2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*, págs. 4–14. Springer-Verlag.

Izard, C. (1998) Cognition is one of four types of emotion activating systems. *The Nature of Emotion*, págs. 203–208.

J. Timmis, J. y Neal, M. (2001) A Resource Limited Artificial Immune System. *Knowledge Based Systems*, 14(3): págs. 121–130.

Jacobson, I., Booch, G. y Rumbaugh, J. (2000) *El Proceso Unificado de Desarrollo de Software*. Pearson Addisson-Wesley, Barcelona.

Jaenisch, R. y Bird, A. (2003) Epigenetic regulation of gene expression: how the genome integrates intrinsic and environmental signals. *Nature*, 33: págs. 245–254.

Jakobi, N., Husbands, P. y Harvey, I. (1995) Noise and the reality gap: The use of simulation in evolutionary robotics. *Lecture Notes in Computer Science*, 929.

James, W. (1890) *The Principles of Psychology*. Springer, New York.

Jansen, D. (1980) When is it coevolution? *Evolution*, 34: págs. 611–612.

Jeannerod, M. (1994) The representing brain: neural correlates of motor intention and imagery. *Behav. Brain Sci.*, 17: págs. 187–245.

Jensen, J. y Pedersen, A.-M. (2000) Probabilistic models of dna sequence evolution with context dependent rates of substitution. *Advances in Applied Propability*, 32(2): págs. 499–517.

Jerison, H. (1990) Brain evolution: new light on old Princeton. *Science*, 170: págs. 1224–1225.

Jerne, N. (1973) The Immune System. *Scientific American*, 1(229): págs. 52–60.

Jiang, J. y Kamel, M. S. (2006) Aggregation of reinforcement Learning Algorithms. *IEEE Congress on Evolutionary Computation*, págs. 72–88.

Johnson, S. (2004) *Sistemas Emergentes. O qué tienen en común hormigas, neuronas, ciudades y software*. Turner/Fondo de Cultura Económica, Madrid.

Jordan, M. (1995) Why the logistic function? A tutorial discussion on probabilities and neural networks. Inf. téc., Massachusetts Institute of Technology.

Just, M. y Varma, S. (2007) The organization of thinking: What functional brain imaging reveals about the neuroarchitecture of complex cognition. *Cognitive, Affective, and Behavioral Neuroscience*, 7: págs. 153–191.

K., C. y Kleeman, C. (1997) Accurate Odometry and Error Modelling for a Mobile Robot. *En In IEEE International Conference on Robotics and Automation*, págs. 2783–2788.

Kahneman, D. (1973) *Attention and Effort*. Engledwood Cliffts, New Jersey.

Kandel, E., Schwartz, J. y Jessell, T. (1991) *Principles of Neural Science*. Prentice-Hall International Inc. (3rd edition).

Kandel, E., Schwartz, J. y Jessell, T. (2000) *Principles of Neural Science*. McGraw-Hill, New York.

Kanerva, P. (1988) *Sparse Distributed Memory*. The MIT Press, Cambridge MA.

Kasabov, N. y Benuskova, L. (2004) Computational neurogenetics. *Journal of Computational and Theoretical Nanoscience*, 1(1): págs. 1–15.

Kauffman, S. (1995) *At home in the universe: the search for laws of complexity*. Viking, London.

Kelso, J. (1995) *Dynamic Patterns – The Self–Organization of Brain and Behaviour*. MIT Press.

Kemp, C., Shafto, P., Berke, A. y Tenenbaum, J. (2006) Combining causal and similarity-based reasoning. NIPS.

Kenneth, S. y Miikkulainen, R. (2002) Evolving Neural Networks Through Augmenting Topologies. *Evolutionary Computation*, 10(2): págs. 99–127.

Kieras, D. y Meyer, D. (1998) The role of cognitive task analysis in the application of predictive models of human performance. *EPIC Tech. Rep.*, 11.

Kittler, J. y Hatef, M. (1996) Improving recognition rates by classifier combination. *In Proc. 5th Int. Workshop on Frontiers of Handwriting Recognition*, págs. 81–102.

Klahr, D., Langley, P. y Neches, R. (1987) *Production System Models of Learning and Development*. The MIT Press., Cambridge, Mass.

Klein, S., Cosmides, L., Tooby, J. y Chance, S. (2002) Decisions and the evolution of memory, multiple systems, multiple functions. *Psychological Review*, 109: págs. 306–329.

Kohonen, T. (2001) *Self-Organizing Maps*. Springer-Verlag.

Kosiol, C., Bofkin, L. y Whelan, S. (2006) Phylogenetics by lokelihood: Evolutionary modeling as a tool for understanding the genome. *Journal of Biomedical Informatics*.

Kosslyn, S., Behrmann, M. y Jeannerod, M. (1995) The cognitive neuroscience of mental imagery. *Neurophysiologia*, 33(11): págs. 1335–1344.

Koza, J. (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press.

Koza, J. R. y Rice, J. (1991) Genetic generation of both the weights and architecture for a neural network. *En Proceedings of the International Joint Conference on Neural Networks*.

Krichmar, J. L. y Edelman, G. M. (2006) Principles underlying the construction of brain–based devices. *Adaptation in Artificial and Biological Systems*, 2: págs. 37–42.

Kunde, W. (2001) Response–effect compatibility in manual choice reaction tasks. *Journal of Experimental Psychology: Human Perception and Performance*, 27: págs. 387–394.

Kuokka, D. R. (1991) MAX: A meta–reasoning architecture for 'X'. *SUGART Bulletin*, 2: págs. 93–97.

Kuzmin, V. (2002) Connectionist Q-learning in Robot Control Task. *Proceedings of Riga Technical University*, págs. 112–121.

Lamarck, J. (1809) *Philosophie zoologique*. Flammarion.

Langley, P. (2005) An adaptive architecture for physical agents. *IEEE/WIC/ACM Int. Conf. Intell. Agent Technol.*, págs. 18–25.

Lave, J. (1988) *Cognition in practice*. Cambridge, UK, Cambridge University Press.

Lavrac, N. y Dzeroski, S. (1994) *Inductive Logic Programming*. Ellis Horword, New York.

Lehman, J. F., Laird, J. y Rosenbloom, P. (1997) An overview of the epic architecture for cognition and performance with application to human–computer interaction. *Human–Computer Interaction*, 12: págs. 391–438.

Lehman, J. F., Laird, J. y Rosenbloom, P. (2006) A gentle introduction to SOAR, an architecture for human cognition: 2006 update. *Sternberg and Scarborough*.

Lehmann, D., Strik, W. K., Henggeler, B., Koenig, T. y Koukkou, M. (1998) Brain electric microstates and momentary conscious mind states as building blocks of spontaneous thinking: I. Visual imagery and abstract thoughts. *Int. J. Psychophysiol*, 29: págs. 1–11.

Leslie, A. M. (1994) Tomm, toby and agency: Core architecture and domain specificity. *in Mapping the Mind: Specificity in Cognition and Culture*, págs. 119–148.

Levene, H. (1974) Robust tests for equality of variances. *Journal of the American Statistical Association*, 69(346): págs. 364–367.

Levy, S. (1992) *Artificial life: The quest for a new creation*. Phanteon, London.

Lewicki, P., Hill, T. y Czyzewska, M. (1992) Nonconscious acquisition of information. *American Psychologist*, 47: págs. 796–801.

Libet, B. (1985) Unconscious cerebral initiative and the role of conscious will in voluntary action. *Behavioral and Brain Sciences*, 8: págs. 529–566.

Lin, L. y Mitchell, T. M. (1992) *Memory approaches to reinforcement learning in non-Markovian domains*. CMU-CS-92-138, Carnegie Mellon University Pittsburg.

Looks, M., Goertzel, B. y Pennachin, C. (2004) Novamente: An integrative architecture for general intelligence. *En AAAI Fall Symposium, Achieving Human-level intelligence*.

Lorenz, K. (1973) *Foundations of Ethology*. Springer-Verlag, New York.

Luca, P. (2000) Adaptive Agents with Reinforcement Learning and Internal Memory. *En In International conference on the Simulation of Adaptive Behavior*, págs. 333–342. MIT Press.

Luce, R. (1959) *Individual Choice Behavior: A Theoretical Analysis*. Wiley, New York.

Luce, R. (2000) *The utility of gains and losses*. Lawrence Erlbaum Associates, NJ.

Maes, P. (1989) How to do the right thing. *Connection Science Journal*, 1: págs. 291–323.

Maes, P. (1992) Learning Behavior Networks from Experience. *Proceedings of the First European Conference on Artificial Life*, págs. 48–57.

Maes, P. (1994) Modeling Adaptative Autonomous Agents. *Artificial Life Journal*, 1(2): págs. 135–162.

Mainzer, K. (2007) Complex Systems and the Evolution of Mind–Brain. *Thinking in Complexity*, 5: págs. 123–178.

Maniadakis, M. (2006) *Design and integration of agent-based partial brain models for robotics systems by means of hierarchical cooperative coevolution*. Tesis Doctoral, University of Crete.

Maniezzo, V. (1994) Genetic evolution of the topology and weight distribution of neural networks. *IEEE Transactions on Neural Networks*, págs. 39–53.

Mao, Z. (2005) *Modeling the role of the Basal Ganglia in the motor control and motor programming*. Tesis Doctoral, Massachusetts Institute of Thechnology.

Marcel, A. (1983) Conscious and unconscious perception: Experiments on visual masking and word recognition. *Cognitive Psychology*, 15: págs. 197–237.

Martin, W. (1998) Frames as definition models for terms. *En Proceedings of the International Converence on Professional Communication and Knowledge Transfer*, tomo II, págs. 189–220.

Martinelli, A., Tomatis, N., Tapus, A. y Siegwart, R. (2003) Simultaneous Localization and Odometry Calibration for Mobile Robot. *Intl. Conference on Intelligent Robots and Systems*, págs. 1499–1504.

Mataric, M. (1992) Behavior–based control: Main properties and implications. *Proceedings of the IEEE International Conference on Robotics and Autonomation*, págs. 2–8.

Mataric, M. (2001) Learning in behavior-based multi-robot systems: policies, models and new behaviors. *Cognitive Systems Research. Special issue on multi-agent learning*, 2: págs. 2–8.

Mathias, K. y Whitley, L. (1992) Genetic operators, the fitness landscape and the traveling salesman problem. *Parallel Problem Solving From Nature II*, págs. 221–230.

Maturana, H. y Varela, F. (1987) *The Tree of Knowledge – The Biological Roots of Human Understanding*. New Science Library, London, U.K.

Maturana, H. R. y Varela, F. J. (1980) *Autopoiesis and Cognition – The Realization of the Living Boston Studies on the Philosophy of Science*. Holland: D. Reidel Publishing Company.

Matzinger, P. (2002) The danger model: a renewed sense of self. *Science*, 296: págs. 301–305.

McClelland, J. L. (1995) Constructive Memory and Memory Distortions: A Parallel-Distributed Processing Approach. *Memory Distortion*, págs. 59–90.

McCullagh, P. y Nelder, J. (1989) *Generalized Linear Models*. Chapman and Hall, London.

McFarland, D. y Bosser, T. (1993) *Intelligent Behavior in Animals and Robots*. MIT Press, Cambridge, MA.

McLeod, P., Plunkett, K. y Rolls, E. (1998) *Introduction to Connectionist Modelling of Cognitive Processes*. Oxford University Press, New York.

Medler, D. A. (1998) A brief history of connectionism. *Neural Computing Surveys*, 1: págs. 61–101.

Merz, P. y Freisleben, B. (1997) Genetic local search for the TSP: New results. *In Proceedings of 1997 IEEE International Conference on Evolutionary Computation*, págs. 159–163.

Miall, R. (1997) Sequences of sensory predictions. *Behav. Brain Sci.*, 20: págs. 258–259.

Miall, R. y Wolpert, D. (1996) Forward models for physiological motor control. *Neural Networks*, 9(8): págs. 1265–1279.

Miñarro, J. y Rodríguez, M. (1996) Sistema Dopaminérgico y Refuerzo: acción de los antagonitas dopaminérgicos. *Psicol. Gral. y Aplic.*, 49(2): págs. 211–220.

Middleton, F. y Strick, P. L. (2000) Basal ganglia and cerebellar loops: Motor and cognitive circuits. *Brain Research Reviews*, 31: págs. 236–250.

Middleton, F. y Strick, P. L. (2009) Reverse Engineering the Vertebrate Brain: Methodological Principles for a Biologically Grounded Programme of Cognitive Modelling. *Cognitive Computation*, 1: págs. 29–41.

Miller, G. A. (1956) The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2): págs. 81–97.

Milsum, J. y Bertalanffy, L. (1978) *Tendencias en la teoría general de sistemas*. Alianza Universidad, Madrid.

Minsky, M. (1988) *The Society of Mind*. Simon and Schuster, New York.

Mitchell, T., Allen, J., Chalasani, P., Etzioni, O., Ringuette, M. y Schlimmer, J. (1991) Theo: A framework for self–improving systems. *Architectures for Intelligence*, págs. 323–355.

Mori, N., Takeda, M. y Matsumoto, A., K. (2005) Statistical comparison study between Genetic Algorithms and Bayesian Optimization Algorithms. *Progress of Theoretical Physics Supplement*, 157: págs. 353–356.

Moscato, P. y Cotta, C. (2003) Una Introducción a los Algoritmos Meméticos. *Inteligencia Artificial, Revista Iberoamericana de IA*, 7(19): págs. 131–148.

Mountcastle, V. (2003) The columnar organization of the neocortex. *Brain*, 120: págs. 701–722.

Métivier, M. y Lattaud, C. (2003) Anticipatory System Using Behavioral Sequences in Non-Markov Environments. *Lecture Notes in Computer Science*, págs. 143–162.

N. De Castro, L. y Von Zuben, F. J. (2001) Learning and Optimization Using the Clonal Selection Principle. *IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems*, 6(3): págs. 239–151.

Nanas, N. y Roeck, A. (2009) Autopoiesis, the immune system, and adaptive information filtering. *Natural Computing: an international journal*, 8(2): págs. 387–427.

Nau, D., Au, T., Ilghami, O., Kuter, U., Munoz, H., William, J., Wu, D. y Yaman, F. (2005) Applications of SHOP and SHOP2. *IEEE Intelligent Systems*, 20: págs. 34–41.

Nebel, B. y Babovich-Lierler, Y. (2004) When Are Behaviour Networks Well-Behaved? *En ECAI*, págs. 672–676.

Negatu, A., D'Mello, S. y Franklin, S. (2007) Cognitively Inspired Anticipatory Adaptation and Associated Learning Mechanisms for Autonomous Agents. *Anticipatory Behavior in Adaptive Learning Systems: From Brains to Individual and Social Behavior*, págs. 108–127.

Nestor, A. y Kokinov, B. (2004) Towards Active Vision in the DUAL Cognitive Architecture. *Information Theories and Applications*, 11: págs. 9–15.

Newell, A. (1990) *Unified Theories of Cognition*. MA:Harvard University, Cambridge.

Newell, A. y Simon, H. (1995) GPS, a program that simulates human thought. *Computers and thought*, págs. 279–293.

Newell, A. y Simon, H. A. (1976) Computer science as empirical inquiry: Symbols and search. *Communications of the Association for Computing Machinery*, 19: págs. 113–126.

Newell, A., Rosenbloom, P. y Laird, J. (1993) *The Soar Papers: Research on integrated intelligence*. MA:MIT Press, Cambridge.

Nilson, N. J. (2006) *Introduction to Machine Learning*. Standford University, Standford, CA.

Nisbett, R. y Wilson, T. (1977) Telling more than we can know: Verbal reports on mental processes. *Psychological Review*, v84 i3.

Nokes, T., VanLehn, K. y Belenky, D. M. (2008) Coordinating principles and examples through analogy and explanation. *En Conference of the Cognitive Science Society*, pág. 2365.

Nosofsky, R., Palmeri, T. y McKinley, S. (1994) Rule-plus-exception model of classification learning. *Psychological Review*, 101(1): págs. 53–79.

Ogasawara, G. y Russell, S. (1994) Real–time, decision–theoretic control of an autonomous underwater vehicle. *SUGART Bulletin*, 4.

Ogden, B., Dautenhahn, K. y Stribling, P. (2002) Interactional structure applied to the identification and generation of visual interactive behaviour: Robots that (usually) follow the rules. *Gesture and Sign Languages in Human–Computer Interaction*, LNAI 2298: págs. 254–268.

Oguz, C. y Ercan, M. (2005) A genetic algorithm for hybrid flow-shop scheduling with multi-processor tasks. *Scheduling 8*, págs. 323–351.

Ortony, A., Clore, G. y Collins, A. (1988) *Cognitive Structure of Emotions*. Cambridge University Press, Cambridge.

Pielke, R. A. (1984) *Mesoscale Meteorological Modeling*. Academic Press.

Plutchik, R. (1991) *The Emotions*. University Press of America, Lanham, MD.

Poincare, H. (1955) *The creative process*. Mentor Books, UCLA, Los Angeles.

Poli, C., Langdon, W. y McPhee, N. (2008) *A field guide to genetic programming*. MA: MIT Press.

Pollard, D. (2003) *A user's guide to measure theoretic probability*. Cambridge University Press.

Portellano, J. A. (2005) *Introducción a la neuropsicología*. McGraw-Hill, Madrid.

Posner, M. (1978) *Chronometic explorations of the mind*. Hillsdale, New Jersey.

Posner, M. (1980) Orienting of attention. *Quarterly Journal of Experimental Psychology*, 32: págs. 3–25.

Potter, M. y DeJong, K. (2000) Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evol. Computation*, 8: págs. 1–29.

Qin, Y., Bothell, D. y Anderson, J. R. (2007) ACT-R meets fMRI. *En LNAI 4845*, págs. 205–222.

Quick, T., Nehaniv, C., Dautenhahn, K. y Roberts, G. (2003) Evolving embodied genetic regulatory network-driven control systems. *Proc. 7th European conference on Artificial Life, (ECAL)*, págs. 266–277.

Quinlan, J. R. (1987) Generating production rules from decision trees. *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, págs. 304–307.

Qureshi, F., Terzopoulos, D. y Gillett, R. (2004) The cognitive controller: a hybrid, deliberative/reactive control architecture for autonomous robots. *En IEA/AIE'2004*, págs. 1102–1111.

Radclife, N. y Surry, P. (1994) Fitness Variance of Formae and Performance Prediction. *Proceedings of the 3rd Workshop on Foundations of Genetic Algorithms*, págs. 51–72.

Radcliffe, N. (1994) The algebra of genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 10: págs. 339–384.

Rahman, A. F. y Fairhurst, M. C. (1997a) Exploiting second order information to design a novel multiple expert decision combination platformfor pattern classification. *Electronics Letters*, 33(6): págs. 476–487.

Rahman, A. F. y Fairhurst, M. C. (1997b) A generalised approach to the recognition of structurally similar handwritten characters. *IEE Proc. on Vision, Image and Signal Processing*, 144(1): págs. 15–22.

Rahman, A. F. y Fairhurst, M. C. (1998) A multipleexpert decision combination strategy for handwritten character recognition. *In Proc. Int. Conf. on Computational Linguistics, Speech and Document Processing*, págs. A23–A28.

Rakic, P. (2002) Pre- and post- developmental neurogenesis in primates. *Clinical Neuroscience Research*, 2: págs. 29–39.

Rao, A. S. y Georgeff, M. P. (1991) Modeling Rational Agents within a BDI–Architecture. *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR91)*, págs. 473–484.

Ratha, N. y Jain, A. (2001) High Performance Multilayer Perceptron on a Custom Computing Machine.

Reber, A. (1989) Implicit learning and tacit knowledge. *Experimental Psychology*, 118(3): págs. 219–235.

Reátegui, N. y Sattler, C. (1999) *Metacognición: estrategias para la construcción del conocimiento*. CEDEHUM, Lima.

Reynolds, R. G. (1994) An Introduction to Cultural Algorithms. *in Proceedings of the 3rd Annual Conference on Evolutionary Programming*, págs. 131–139.

Roggen, D., Floreano, D. y Mattiussi (2003) A morphogenetic evolutionary system: Phylogenesis of the poetic circuit. *Proc. of the Fifth International Conference on Evolvable Systems, (ICES)*, págs. 153–164.

Romero, D. y Niño, L. (2006) An Immune-based Multilayered Cognitive Model for Autonomous Navigation. *CEC*, págs. 1115–1122.

Romero, O. (2002) Sistema Tutor Inteligente adaptativo para la enseñanza de matemáticas básicas. *Revista Científica*, 4: págs. 77–83.

Romero, O. (2003) La Inteligencia Artificial en la Educación. *Revista Magisterio, Educación y Pedagogía*, 5: págs. 51–58.

Romero, O. (2010) Evolving the way of doing the right thing. *Soft Computing Journal*. In press.

Romero, O. y De Antonio, A. (2007a) Analysis of Emergent Properties in a Hybrid Bio-inspired Architecture for Cognitive Agents. *International Workshop on Hybrid Artificial Intelligence Systems HAIS*, págs. 115–122.

Romero, O. y De Antonio, A. (2007b) Multilayered Evolutionary Architecture for Behaviour Arbitration in Cognitive Agents. *IAENG Journals*, 15(2): págs. 193–202.

Romero, O. y De Antonio, A. (2008a) Bio-inspired Cognitive Architecture for Adaptive Agents based on an Evolutionary Approach. *Autonomous Agents and Multiagent Systems AAMAS*, págs. 89–96.

Romero, O. y De Antonio, A. (2008b) Hybrid Behaviour Orchestration in a Multilayered Cognitive Architecture using an Evolutionary Approach. *IEEE World Congress on Computational Intelligence WCCI*, págs. 89–96.

Romero, O. y De Antonio, A. (2009a) Hybridization of cognitive models using evolutionary strategies. *IEEE Congress on Evolutionary Computation CEC 09*, págs. 89–96.

Romero, O. y De Antonio, A. (2009b) Modulation of multi-level evolutionary strategies for artificial cognition. *Genetic and Evolutionary Computation Conference GECCO*, págs. 109–116.

Romero, O. y Ramirez, J. (2002) Animales artificiales con Sistemas Clasificadores y Algoritmos Genéticos. *Revista Ingeniería*, 7(1): págs. 51–58.

Romero, O., Ramirez, J. y Rojas, S. (2002) Intelligent Tutoring System for Mathematics using an evolutionary student model. *Artificial Intelligence And Applications*, págs. 327–333.

Rosbe, J., Chong, R. y Kieras, D. (2001) Modeling with Perceptual and Memory Constraints: An EPIC-Soar Model of a Simplified Enroute Air Traffic Control Task. *SOAR Technology Inc. Report*.

Rosen, R. (1985) *Anticipatory Systems*. Pergamon Press.

Rosenblatt, F. (1961) *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books.

Rosselló-Mir, J. (1999) Selección para la acción, selección para la percepción. *Atención y percepción*, págs. 99–150.

Rouse, W. y Macdonald, H. (1987) *Mathematical recreations and essays*. Dover Publications.

Rovithakis, G., Maniadakis, M. y Zervakis, M. (2004) A hybrid neural network/genetic algorithm approach to optimizing feature extraction for signal classification. *IEEE Transactions on Systems, Man and Cybernetics*, 34(1): págs. 695–703.

Roy, D. (2005) Semiotic schemas: a framework for grounding language in action and perception. *Artificial Intelligence*, 167(1–2): págs. 170–205.

Rubenstein, J. (1982) *Principios de Psicología General*. Grijalbo, México D.F.

Rubinstein, A. (2005) Instinctive and Cognitive Reasoning: Response Times Study. *En Features of Experimentally Observed Bounded Rationality, European Economic Review, XLII*, págs. 413–436. Harvard University Press.

Ruiz Vargas, J. y Botella, J. (1987) *Esquizofrenia: Un enfoque cognitivo*. Alianza, Madrid.

Rumelhart, D. y McClelland, J. (1986) *Parallel Distributed Processing: Explorations in the microstructures of cognition*. MIT Press, Cambridge.

Rumelhart, D., Geoffrey, E. y Williams, R. (1986) *Learning Internal Representations by Error Propagation*. MIT Press.

Russell, S., y Norvig, J. (2003) *Artificial Intelligence: A Modern Approach (Second Edition)*. Prentice Hall.

Sacerdoti, E. D. (1977) *A Structure for Plans and Behavior*. Elsevier-North Holland.

Saint-Cyr, J. A., Taylor, A. E. y Lang, A. E. (1988) Procedural learning and neostriatal dysfunction in man. *Brain*, 111: págs. 941–959.

Salk, J. (1985) *The anatomy of reality*. Praeger, New York.

Sapena, O. (2005) *Planificación independiente del dominio en entornos dinámicos de tiempo restringido.*. Tesis Doctoral, Universidad de Valencia.

Sarmiento, A. (2009) Autopoiesis, bucles, emergencia, variedades topológicas y una conjetura sobre la consciencia humana. *Ciencia, Pensamiento y Cultura*, págs. 870–878.

Scassellati, B. (2002) Theory of mind for a humanoid robot. *Autonomous Robots*, págs. 13–24.

Schöner, G. (2006) *Development as change of dynamic systems: Stability, instability and emergence*. Oxford University Press, NewYork.

Schubotz, R. y von Cramon, D. (2001) Functional organization of the lateral premotor cortex. *Cognitive Brain Research*, 11: págs. 97–112.

Seger, C. (1994) Implicit learning. *Psychological Bulletin*, 115(2): págs. 163–196.

Seif El-Nasr, M., Yen, J. y Ioerger, T. (2000a) FLAME – A Fuzzy Logic Adaptive Model of Emotions. *Autonomous Agents and Multi–Agent Systems*, 3(3): págs. 219–257.

Seif El-Nasr, M., Yen, J. y Ioerger, T. R. (2000b) FLAME — A Fuzzy Logic Adaptive Model of Emotions. *Autonomous Agents and Multi-Agent Systems*, 3(3): págs. 219–257.

Shallice, T. (1972) Dual functions of consciousness. *Psychological Review*, 79(5): págs. 383–393.

Shanahan, M. P. (2006) A cognitive architecture that combines internal simulation with a global workspace. *Consciousness and Cognition*, 15: págs. 433–449.

Shapiro, D. y Schmidt, R. (1982) The schema theory: Recent evidence and developmental implications. *En The development of movement control and coordination*. Clark, J. K. . J., New York.

Shapiro, S. y Wilk, M. B. (1965) An analysis of variance test for normality (complete samples). *Biometrika*, 52(3–4): págs. 591–611.

Sheskin, D. (2003) *Handbook of Parametric and Nonparametric Statistical Procedures*. CRC Press.

Shirley, P. (1989) An introduction to ultrasonic sensing. *Sensors*, 6(11): págs. 167–200.

Skinner, B. (1950) Are theories of learning necessary? *Psychological Review*, 57(4): págs. 193–216.

Smolensky, P. (1988) On the proper treatment of connectionism. *Behavioral and Brain Sciences*, 11(1): págs. 1–74.

Smolensky, P. (1996) *Computational, dynamical and statistical perspectives on the processing and learning problems in neural network theory*. in Mathematical Perspectives on Neural Networks, NJ: Erlbaum.

Squire, L., C., S. y Clark.R. (2004) El lóbulo temporal medial. *Annual Review of Neuroscience*, 27: págs. 279–306.

Srihari, S. N., Ho, T. K. y Hull, J. J. (1994) Decision combination in multiple classifier systems. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(1): págs. 66–75.

Stanley, W., Mathews, R., Bus, R. y Kotler-cope, S. (1989) Insight without awareness: On the interaction of verbalization, instruction and practice in a simulated process control task. *Quarterly Journal of Experimental Psychology*, 22: págs. 553–377.

Stewart, J. (1993) Cognition without neurons: adaptation, learning and memory in the Immune System. *Cognitiva*, 5(2): págs. 187–202.

Stolzmann, C. (1998) Anticipatory Classifier System. *Genetic Programming*, págs. 658–664.

Stolzsmann, W. (1999) Latent learning in Khepera robots with Anticipatory Classifier Systems. *En In Genetic and Evolutionary Computation Conference Workshop Program*, págs. 290–297.

Stolzsmann, W. y Butz, M. (2000) Latent learning and Action Planning in robots with Anticipatory Classifier Systems. *En LCS*, págs. 301–317.

Stroop, J. (1935) Studies on interference in serial verbal reactions. *Journal of Experimental Psychology*, 18: págs. 622–643.

Sun, R. (1994) *Integrating rules to and Connectionism for robust commonsense reasoning*. John Wiley and Sons, New York.

Sun, R. (1995) Robust Reasoning: integrating rule-based and similarity-based reasoning. *Artificial Intelligence*, 75(2): págs. 241–296.

Sun, R. y Alexandre, F. (1997) *Connectionist symbolic integration*. Hillsdale.

Sun, R. y Zhang, X. (2003) Accessibility versus Action-Centeredness in the Representation of Cognitive Skills. *En Proceedings of the Fifth International Conference on Cognitive Modeling*.

Sun, R. y Zhang, X. (2004) Top-Down versus Bottom-Up Learning in Skill Acquisition. *En in Proceedings of the 24th Annual Conference of the Cognitive Science Society*, págs. 63–89.

Sun, R., Merrill, E. y Peterson, T. (1998) A bottom-up model of skill learning. *En Proceedings of the 20th Cognitive Science Society Conference*, págs. 1037–1042. Lawrence Erlbaum Associates.

Sun, R., Merril, E. y Peterson, T. (2001) From implicit skills to explicit knowledge: A bottom-up model of skill learning. *Cognitive Science*, 25(2): págs. 203–244.

Surmann, H. y Maniadakis, M. (2001) Learning feed-forward and recurrent fuzzy systems: a genetic approach. *Journal of Systems Architecture*, 47(7).

Sussman, G. (1975) *A Computer Model of Skill Acquisition*. Tesis Doctoral, Elsevier Science Inc.

Sutton, R. y Barto, A. (1998) *Reinforcement Learning: An introduction*. MIT Press, Cambridge.

Symeonidis, A. y Tollis, I. (2004) Visualization of biological information with circular drawings. *Proc. 5st International Symposium on Biological and Medical Data Analysis, (ISBMDA)*, págs. 468–478.

Taatgen, N. y Anderson, J. (2008) ACT-R. *Constraints in Cognitive Architectures*, págs. 170–185.

Tamee, K., Bull, L. y Pinngern, O. (2008) Towards Clustering with Learning Classifier Systems. *En Learning Classifier Systems in Data Mining*, tomo 125, págs. 191–204. Springer Berlin / Heidelberg.

Tanaka, Y. y Sagi, D. (1998) A perceptual memory for low-contrast visual signals. *Proc. Natl. Acad*, 95: págs. 12.729–12.733.

Tanembaum, A. (2001) *Computer Networks*. Prentice Hall.

Tate, A. (1990) Generating Project Networks. *In Proc. IJCAI-77*, págs. 888–893.

Thach, W. (1998) What is the role of the cerebellum in motor learning and cognition? *Trends Cognitive Sci.*, 2(9): págs. 331–337.

Thelen, E. (1995) *Time–scale dynamics and the development of embodied cognition*. Port and T. van Gelder editors, Massachusetts.

Thelen, E. y Smith, L. B. (1994) *A Dynamic Systems Approach to the Development of Cognition and Action*. MIT Press, MA: Cambridge.

Thompson, E. y Varela, F. (2001) Radical Embodiment: neural dynamics and consciousness. *TRENDS in Cognitive Sciences*, 5(10): págs. 418–425.

Thorndike, E. (1932) *The Fundamentals of Learning*. Teachers College Press, New York.

Toan, P., Winikoff, M. y Padgham, L. (2005) Learning Within the BDI Framework: An Empirical Analysis. *En Knowledge-Based Intelligent Information and Engineering Systems*.

Treisman, A. y Schmidt, H. (1982) Illusory conjunctions in the perception of objects. *Cognitive Psychology*, 14: págs. 107–141.

Tulving, E. (2002) Episodic Memory: From Mind to Brain. *Annual Review of Psychology*, 53: págs. 1–25.

VanLenh, K. (1995) Cognitive Skill Adquisition. *Annual review of Psychology*, 47: págs. 147–188.

VanLenh, K. y Ball, W. (1991) Goal Reconstruction: How Teton Blends Situated Action and Planned Action. *Architectures for Intelligence*, págs. 147–188.

VanRullen, R. y Koch, C. (2003) Is perception discrete or continuous? *Trends in Cognitive Science*, 7: págs. 207–213.

Varela, F. (1979) *Principles of Biological Autonomy*. Elsevier North Holland, New York.

Varela, F. y Coutinho, A. (1991) Second generation immune network. *Immunol Today*, 12(5): págs. 159–166.

Varela, F. y Maturana, H. (1973) *De Máquinas y Seres Vivos: Una teoría sobre la organización biológica*. Editorial Universitaria, Santiago Chile.

Varela, F. J. (1992a) Autopoiesis and a Biology of Intentionality. *En Autopoiesis and Perception: A Workshop with ESPRIT BRA 3352*, págs. 4–14.

Varela, F. J. (1992b) *Whence perceptual meaning? A cartography of current ideas, in Understanding Origins – Contemporary Views on the Origin of Life, Mind and Society*. Eds. Norwell, MA: Kluwer, Boston Studies in the Philosophy of Science.

Vasile, F., Silvescu, A., Kang, D. y Honavar, V. (2004) TRIPPER: Rule learning using taxonomies.

Vaughan, R. y Zuluaga, M. (2006) Use Your Illusion: Sensorimotor Self-simulation Allows Complex Agents to Plan with Incomplete Self-knowledge. *From Animals to Animats*, 9: págs. 298–309.

Veloso, M., Carbonell, J., Perez, A., Borrajo, A., Fink, E. y Blythe, J. (1996) Integrating Planning and Learning: The PRODIGY Architecture. *Journal of Theoretical and Experimental Artificial Intelligence*, 7(1).

Velásquez, J. (1996) *Cathexis: a computational model for the generation of emotions and their influence in the behavior of autonomous agents*. Tesis Doctoral, MIT.

Vere, S. (1991) Organization of the Basic Agent. *SIGART Bulletin*, 2: págs. 151–155.

Vere, S. y Bickmore, T. (1990) A basic agent. *Computational Intelligence*, 6: págs. 41–60.

Vernon, D. (2006) The space of cognitive vision. *Cognitive Vision Systems: Sampling the Spectrum of Approaches*, 6: págs. 7–26.

Vernon, D., Metta, G. y Sandini, G. (2007) A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents. *IEEE Transactions on Evolutionary Computation*, 11: págs. 151–180.

Waldrop, M. (1992) *Complexity: The Emerging Science at the Edge of Order and Chaos*. Simon and Schuster, New York.

Watkins, C. (1989) *Learning with delayed rewards*. Tesis Doctoral, Cambridge University, UK.

Watkins, C. y Dayan, P. (1992) Q-Learning. *Machine Learning 8*, págs. 279–292.

Weaver, W. (2007) Science and Complexity. *American Scientist*, 36(536): págs. 11–21.

Weng, J. (2004) A theory of developmental architecture. *in Proc. 3rd Int. Conf. Development and Learning*.

Whiting, B. y Barton, R. (2003) The evolution of the cortico-cerebellar complex in primates: anatomical connections predict patterns of correlated evolution. *Journal of Human Evolution*, 44: págs. 3–10.

Widrow, B. y Hoff, M. (1960) Adapting switching circuits. *Western Electronic Show and Convention*, 4: págs. 96–104.

Wiener, N., Rosenblueth, A. y Bigelow, J. (1943) Behavior, Purpose and Teleology. *Philosophy of Science*, 10: págs. 18–24.

Wilkins, D. (1988) *Practical Planning: Extending the classical AI planning paradigm*. Morgan-Kaufmann.

Willingham, D., Nissen, M. y Bullemer, P. (1989) On the development of procedural knowledge. *Eperimental psychology: learning, memory and cognition*, 15: págs. 1047–1060.

Wilson, E. O. (1998) *Consilience : the unity of knowledge*. Little, Brown.

Wilson, S. (1995) Classifier fitness based on accuracy. *Evolutionary Computation*, 3(2): págs. 149–175.

Wilson, S. (2000) State of XCS Classifier System Research. *Lecture Notes in Computer Science*, 1813: págs. 63–81.

Winikoff, M. (2005) JACK intelligent agents: An industrial strength platform. *Multi-Agent Programming*, 0.

Winograd, T. y Flores, F. (1986) *Understanding Computers and Cognition – A New Foundation for Design*. Addison–Wesley Publishing Company, Massachusetts.

Wise, S. P., Murray, E. A. y Gerfen, C. R. (1996) The frontal cortex basal ganglia system in primates. *Critical Reviews in Neurobiology*, 10: págs. 317–356.

Wolpert, D. y Macready, W. (1997) No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1): págs. 67–82.

Wolpert, D., Miall, R. y Kawato, M. (1998) Internal models in the cerebellum. *Trends Cognitive Sci.*, 2(9): págs. 338–347.

Wooldridge, M. (2002) *An Introduction to MultiAgent Systems*. John Wiley and Sons, Chichester, England.

Wooldridge, M. y Jennings, N. (1994) Agent Theories, Architectures, and Languages: A Survey. *En Intelligent Agents - Theories, Architectures, and Languages*, págs. 1–32. Lecture Notes in Artificial Intelligence, Springer-Verlag.

Wooldridge, M., Jennings, M. y Kinny, D. (2000) The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems*, 3: págs. 285–312.

Wray, R. y Lebiere, C. (2007) Metrics for cognitive architecture evaluation. *En In Proceedings of the AAAI-07 Workshop on Evaluating Architectures for Intelligence*.

Wuhong, H., Haifeng, D., Licheng, J. y Jing, L. (2005) Lamarckian Clonal Selection Algorithm with Application. *Lecture Notes in Computer Science*, 3696: págs. 317–322.

Zadeh, L. A. (1983) A Computational Apprach to Fuzzy Quantifiers in Natural Languages. *Computing and Mathematics with Applications*, págs. 149–184.

Zar, J. (1999) *Biostatistical Analysis*. Prentice Hall.

Ziemke, T. y Thieme, M. (2002) Neuromodulation of reactive sensorimotor mappings as a short-term mechanism in delayed response tasks. *Adaptive Behavior*, 10(3–4): págs. 185–199.

Zilberstein, S. y Russell, S. (1995) Approximate Reasoning Using Anytime Algorithms.

Zunino, G. y Christensen, H. (2002) Navigation in realistic environment. *Centre for Autonomous Systems*, págs. 200–209.