Analysis of Emergent Properties in a Hybrid Bio-inspired Architecture for Cognitive Agents

Oscar J. Romero and Angélica de Antonio

Politécnica de Madrid University, LSIIS Department,
Boadilla del monte, 28660, Madrid, Spain
ojrlopez@hotmail.com, angelica@fi.upm.es

Abstract. In this work, a hybrid, self-configurable, multilayered and evolutionary architecture for cognitive agents is developed. Each layer of the subsumption architecture is modeled by one different Machine Learning System MLS based on bio-inspired techniques. In this research an evolutionary mechanism supported on Gene Expression Programming to self-configure the behaviour arbitration between layers is suggested. In addition, a co-evolutionary mechanism to evolve behaviours in an independent and parallel fashion is used. The proposed approach was tested in an animat environment using a multi-agent platform and it exhibited several learning capabilities and emergent properties for self-configuring internal agent's architecture.

Keywords: Subsumption Architecture, Hybrid Behaviour Co-evolution, Gene Expression Programming, Artificial Immune Systems, Extended Classifier Systems, Neuro Connectionist Q-Learning Systems.

1 Introduction

In the last decades, Cognitive Architectures have been an area of study that collects disciplines as artificial intelligence, human cognition, psychology and more, to determine necessary, sufficient and optimal distribution of resources for the development of agents exhibiting emergent intelligence. One of the most referenced is the Subsumption Architecture proposed by Brooks [1].

According to Brooks [1], the Subsumption Architecture is built in layers. Each layer gives the system a set of pre-wired behaviours, where the higher levels build upon the lower levels to create more complex behaviours: The behaviour of the system as a whole is the result of many interacting simple behaviours. Another characteristic is its lack of a world model, which means that its responses are always and only reflexive as proposed by Brooks.

However, Subsumption Architecture results in a tight coupling of perception and action, producing high reactivity, poor adaptability, no learning of new environments, no internal representation and the need that all patterns of behaviours are pre-wired.

The present research focuses on developing a Hybrid Multilayered Architecture for Cognitive Agents based on Subsumption theory. Additionally this work proposes an Evolutionary Model which allows the Agent to self-configure and evolve its processing layers through the definition of inhibitory and suppressor processes, between behaviour layers instead of having a pre-configured structure of them. On the

E. Corchado et al. (Eds.): Innovations in Hybrid Intelligent Systems, ASC 44, pp. 1–8, 2007.
springerlink.com

other hand, instead of using an Augmented Finite Machine System as Subsumption theory states [1] where no internal representation is done, in this paper we propose that each behaviour layer is driven by a different bio-inspired machine learning system (chosen from a repertoire where behaviour co-evolution occurs) which learns from the environment and generates an internal world-model by means of an unsupervised and reinforced learning.

The remainder of the paper is organized as follows. The evolutionary approach for self-configurable cognitive agents is detailed in Section 2. Section 3 discusses the experimental results and emergent properties obtained. Finally concluding remarks are shown in Section 4.

2 Proposed Hybrid, Self-configurable and Evolutionary Approach for Cognitive Agents

In order to design a hybrid, self-configurable, scalable, adaptable, and evolutionary architecture for cognitive systems which exhibits emergent behaviours and learning capabilities, the proposed work is explained as follows.

Consider a virtual environment where there exists several agents interacting with objects, food, each others, etc.; it arises some mayor questions and constraints:

- Change of environmental conditions, i.e. about objects: quantity, type of object, location, size, etc., about other agents: intentions and desires, goals, etc.
- There are a variable number of desired behaviours: avoiding-obstacles, wandering, feeding, hunting, escaping-from-depredators, etc.
- How many behaviours can be integrated into a single agent? And how can agents arbitrate these behaviours?
- When does a single agent know if it has to inhibit or suppress a behaviour without using a pre-established applicability predicate or rule?
- How can a behaviour, that drives one of the layers in a single multilayered agent, generate a model of the world, couple with the environment via the agent's sensors and actuators, learn from its own interaction with the environment and receive a reinforcement of its actions, so the internal state of the behaviour evolve?

These questions address the following proposed approach of a hybrid, self-configurable and bio-inspired architecture for cognitive agents. The Fig. 1 shows a hybrid architecture from which all the questions mentioned above can be solved. An internal architecture based on subsumption principles but with few variations can be observed in every agent:

- Each processing layer is connected randomly with one different bio-inspired learning machine system (Extended Classifier System XCS [2], Artificial Immune System AIS [3], Neuro Connectionist Q-Learnig System NQL [4], Learning Classifier System LCS [5] and scalable to others) which replaces the typical Augmented Finite State Machines proposed by Brook's architecture [1].
- After being trained in the agent, every behaviour is sent to a behaviour repertoire according to its category, where a co-evolutionary mechanism based on [6] is applied so that every behaviour not only will learn in a local way inside of each

agent but also will evolve in a global way, and afterwards it will be selected by another agent in the next generation.

- There is an evolutionary process driven by a Gene Expression Programming Algorithm GEP [7], which is in charge of self-configuring the agent's behaviour arbitration: defining the number of layers, behaviours, connections and hierarchies between them (inhibition, suppression, aggregation, etc.). GEP Algorithm generates a set of applicability predicates per agent where it is determined which behaviour will be activated at a certain situation and its time of activation.

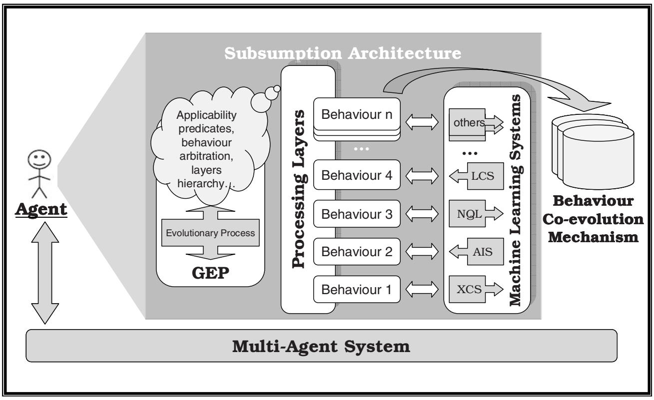

Fig. 1. Hybrid and Evolutionary Architecture for Cognitive Agents

2.1 Hybrid Architecture: Behaviours Driven by Machine Learning Systems

Every behaviour layer in the multilayered architecture is associated to a Machine Learning System MLS, that allows the architecture being hybrid and not only reactive since each behaviour will be able to exert deliberative processes using the acquired knowledge. Besides, this mechanism gives plasticity to the architecture because every behaviour "learns" in an unsupervised, independent and parallel way through its interaction with the environment, generating internal representations, rules and both specific and generalized knowledge. This mechanism is favored by the MLSs characteristics: robustness, fault tolerance, use of bio-inspired techniques, adaptability and it does not require a previous definition of knowledge (unsupervised learning). There are two principles formulated by Stone [8] that have motivated this proposed layered learning approach:

- "Layered learning is designed for domains that are too complex for learning a mapping directly from an agent's sensory inputs to its actuator outputs. Instead the layered learning approach consists of breaking a problem down into several behavioral layers and using MLSs at each level. Layered learning uses a bottom up incremental approach to hierarchical task decomposition."
- "MLS is used as a central part of layered learning to exploit data in order to train and or adapt the overall system. MLS is useful for training behaviours that are difficult to fine-tune manually."

To make the hybridization process consistent, a common interface for all MLSs (XCS [2], AIS [3], NQL [4], LCS [5], etc.) is proposed so, although each MLS has a different internal process, they all have a similar structure that it lets the system to be scalable introducing new MLSs if is required and connecting them in an easy way with each behaviour layer in the agent's multilayered architecture.

2.2 Hybrid Behaviour Co-evolution: Evolving Globally

A co-evolutionary mechanism [6] is proposed to evolve each type of behaviour separately in its own genetic pool. Most evolutionary approaches use a single population where evolution is performed; instead, the behaviours are discriminated in categories and make them evolve in separate behaviour pools without any interaction. First, each agent defines a specific set of behaviours that builds its own multilayered structure. For each required agent's behaviour, a behaviour instance is chosen from the pool (this instance is connected with one MLS). Subsequently each agent will interact with the environment and each agent's behaviour will learn a set of rules and generate an own knowledge base. After certain period of time a co-evolutionary mechanism is activated. For each behaviour pool is applied a probabilistic selection method of behaviours where those behaviours that had the best performance (fitness) will have more probability to reproduce. Then, a crossover genetic operator is applied between each pair of selected behaviours: a portion of knowledge acquired by each agent's behaviour (through its MLS) is interchanged with the other one. Finally, new random rules are generated until complete the maximum size of rules that behaviours can have in their own knowledge base, so a new pair of behaviours is created and left in the corresponding behaviour pool to be selected by an agent in the next generation.

2.3 Self-configurable Architecture: Behaviour Arbitration

If each agent has an arbitrary set of behaviours, how to determine: the interaction between them, the hierarchy levels, the Subsumption process (inhibition and suppression) and the necessary layers to do an adequate processing? These questions are solved next. The internal multilayered structure of each agent is decomposed in atomic components which can be estimated and used to find the optimal organization of behaviours during the agent's lifetime [8]. The main goal is that the agent in an automatic way self-configures its own behaviours structure. The GEP algorithm proposed by Ferreira [7] is used to evolve internal structures of each agent and generate a valid arbitration of behaviours. GEP uses two sets: a function set and a terminal set [7]. In our work, the proposed function set is: {AND, OR, NOT, IFMATCH, INHIBIT, SUPRESS}. The AND, OR and NOT functions are logic operators used to group and exclude subsets of objects, behaviours, etc. The conditional function IFMATCH is a typical applicability predicate that matches with a specific problem situation. This function has four arguments; the first three arguments belong to the rule's antecedent: the first indicates what object is sensed, second one is the activated sensor and the third argument is the current behaviour running on the agent. If the first three arguments are applicable then the fourth argument, the rule's consequent, is executed. The fourth argument should be a INHIBIT or SUPPRESS function, or maybe and AND/OR function if more elements are necessary. The

INHIBIT/SUPPRESS functions have two arguments (behaviourA, behaviourB) and indicate that behaviourA inhibits/suppresses behaviourB. On the other hand, the terminal set is composed by the behaviour set, the environmental element set (objects, agents, food, etc.) and an agent's sensor set. Additionally "wildcard" elements are included so whichever sensor, behaviour or object can be referenced.

Each agent has a chromosome with information about its self structure, i.e. the agent A can have the following chromosome (applicability predicate):

IFMATCH: There is a *wall* AND is Activated *looking-for-food* behaviour AND Reading by *sensor1*

THEN: *Avoiding-obstacle* INHIBIT *wandering* AND *looking-for-food* behaviours, Analyzing this rule we can infer that the agent has three behaviour layers: avoiding-obstacle, wandering and looking-for-food, and the two last ones are inhibited by the first one when sensor1 identifies a wall in front of the agent.

However, these chromosomes don't have always a valid syntax, so the GEP mechanism is used to evolve the chromosome until it becomes in a valid syntactic rule. Each gene is become to a tree representation and then a genetic operator set is applied between genes of the same agent and genes of other agents [7]: selection, mutation, root transposition, gene transposition, two-point recombination and gene recombination. After certain number of evolutionary generations, valid and better adapted agent's chromosomes are generated.

2.4 Emergent Properties of the Architecture

Brooks postulates in his paper [1] the possibility that intelligence can emerge out of a set of simple, loosely coupled behaviours, and emergent properties arise (if at all) due to the complex dynamics of interactions among the simple behaviours and that this emergence is to a large extent accidental. According to that, the proposed architecture articulates a behaviour set that learns about environmental conditions in an independent and parallel manner, and on the other hand evolve inside a categorized pool. Each simple behaviour can be applied to a subset of specific situations but not to the whole problem space, however the individual level interaction between behaviours (inside each agent) allows covering a wider range of problem states and some characteristics are generated: robustness, knowledge redundancy, fault tolerance and a big plasticity level, so emergent properties in the individual and inside of the society (Multi-agent systems) appear. Then, the emergent properties arise from three points of view in a bottom-up approach:

- Atomic: in each behaviour of the multilayered architecture, when the associated MLS learns from the environment, in an automate way.
- Individual: the agent self-configures its internal structure (chromosome), hierarchy and arbitration of behaviours through an evolutionary process driven by GEP.
- Social: a hybrid behaviour co-evolution mechanism is applied to all agent's behaviours, so behaviours learn not only themselves via the MLS associated but also cooperating and sharing the acquired knowledge between them.

It is important to notice that emergence in different levels, from atomic to social point of view, provokes an overall emergence of the system, where some kind of intelligence we hope to arise. The experimentation focused on discovering some characteristics of identity in the animats, i.e. we expected to see some animat agents behaving like depredators and others behaving like preys after the system has evolved several generations. Depredators should include some behaviours like avoiding-obstacles, looking-for-water, persecuting-preys, rounding-up, hunting-preys, etc. and Preys should include some behaviours like avoiding-obstacles, looking-for-food, looking-for-water, hiding, escaping, etc. Nevertheless, expected emergent properties can vary according to the environment and the behaviour set.

3 Experimentation

An artificial life environment called Animat (animal + robot) [5] is proposed to test the experiments. The environment simulates virtual agents competing for getting food and water, avoiding obstacles, hunting, escaping from depredators, etc. This animat environment was selected because is more friendly to see emergent behaviours but it is not the only environment where the proposed architecture could be applicable.

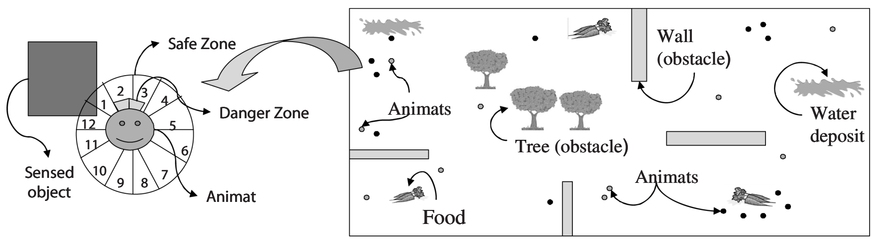

Fig. 2. Simulated Animat Environment and sensor animat distribution

Each animat controlled by an agent disposes a set of 14 proximity sensors (see Fig. 2) simulating a limited sight sense. 12 sensors read a safe zone and 2 sensors read a danger zone (to avoid collisions), as proposed by D. Romero [9]. Additionally, an environment with objects, food, water deposits, animats, obstacles, traps, etc. is simulated. Several performance experiments of all algorithms (XCS, AIS; NQL, LCS and GEP), analysis of learning convergence, fitness progression and analysis of syntactically well-formed genes in GEP, were done in our research, however, we only going to present the relevant results of analyzed emergent properties next.

Analysis of evolved architectures: after the whole system has evolved during a specific number of generations, we have analyzed the final structures of the best adapted agents where emergent properties arose.

Fig. 3 shows the genotype (Expression Trees - ET in GEP) and phenotype respectively of an initial architecture of a random agent without any evolutionary phase; in contrast, fig. 4 shows the genotype and phenotype of the evolved architecture of the same selected agent. In fig. 3.b the chromosome represents four behaviours: looking-for-water, looking-for-food, avoiding-obstacles and hiding, where l-f-w inhibits l-f-f and hiding and l-f-w suppresses a-o, but there is a contradictory process when l-f-f tries to suppress l-f-w and l-f-f has been already inhibited by l-f-w (see fig. 3.b). This is solved with the evolved architecture in fig. 4.b, which proposes a new structure adding escaping-from-depredators behaviour and excluding hiding behaviour.

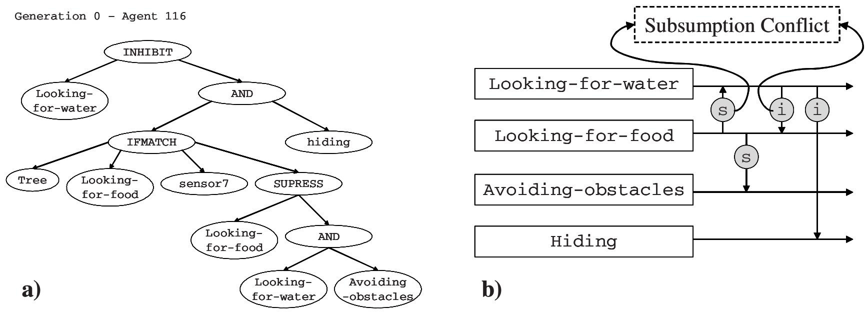

Fig. 3. Genotype and Phenotype of an initial Agent's Architecture

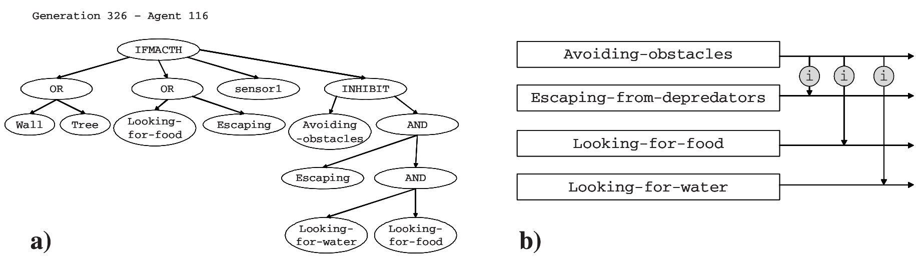

Fig. 4. Genotype and Phenotype of the Agent's Architecture after 326 evolutionary generations

As depicted in fig. 4.b, the initial contradictory inhibitory/suppressor processes in the agent's architecture (see fig. 3.b) are solved, and only hierarchical inhibitory processes are proposed by the evolved architecture. Furthermore, we can deduce too that evolved architecture has collected a set of specific behaviours becoming the agent to an animat with a identity of prey. It is important to notice in evolved architecture that *escaping-from-depredators* behaviour inhibits *looking-for-food* and *looking-for-water* behaviours but if the animat is escaping and its *sensor7* reads a *wall* or a *tree*, then *escaping-from-depredators* behaviour is inhibited by *avoiding-obstacles* behaviour until the obstacle is not in front of the animat anymore, and after that the animat continues its getaway, so we can say that emergent behaviour arises.

Finally, the experimentation demonstrate that specific parameter configurations in MLSs, GEP and Co-evolutionary mechanism are required to reach certain robustness, learning and adaptation capacities in the overall system. Nevertheless, emergent properties didn't arise always or in a quick way, in several experiments animats died quickly and they couldn't learn to survive.

4 Conclusions

The integration of multiple Machine Learning Systems in controlling the behaviours layers of a hybrid Subsumption Architecture approach, instead of using the typical Augmented Finite State Machines, have demonstrated important advantages in learning about the world of the agent, making internal knowledge representations and adapting to environmental changes.

The evolutionary and learning mechanisms used in this work, provided a plasticity feature allowing the agent to self-configure its own multilayered behaviour-based architecture; thus it can avoid creating exhaustive and extensive knowledge bases, pre-wired behaviour-based structures and pre-constrained environments.

In our future work we expect to continue working on designing more adaptive and self-configurable architectures, using fuzzy techniques in the MLSs to improve the sensors readings. In the future, one concrete application of this research will be the development of a Cognitive Module for Emotive Pedagogical Agents where the agent will be able to self-learn about its own perspectives, believes, desires, intentions, emotions, skills and perceptions.

Acknowledgments. Supported by the Programme Alban, the European Union Programme of High Level Scholarships for Latin America, scholarship No. E05D056455CO”.

References

1. R.A. Brooks, A Robust Layered Control System For A Mobile Robot, IEEE Journal Of Robotics And Automation, RA-2 (1986), 14-23.
2. S.W. Wilson, State of {XCS} Classifier System Research, Lecture Notes in Computer Science, 1813 (2000), 63-81.
3. L. N. de Castro, J. Timmis, Artificial Immune Systems: A New Computational Intelligence Approach, Ed. Springer (2002)
4. V. Kuzmin, Connectionist Q-learning in Robot Control Task, Proceedings of Riga Technical University (2002), 112-121.
5. J.H. Holland, Induction, Processes of Inference, Learning and Discovery, Mich:Addison-Wesley (1953).
6. Farahmand, Hybrid Behavior Co-evolution and Structure Learning in Behavior-based Systems, IEEE Congress on EC, Vancouver (2006), 979-986.
7. Ferreira, Gene Expression Programming: A new adaptive algorithm for solving problems, Complex Systems, forthcoming, (2001).
8. P.Stone, Layered Learning in Multiagent Systems, Thesis CS-98-187 (1998)
9. D. Romero, L. Niño, An Immune-based Multilayered Cognitive Model for Autonomous Navigation, IEEE Congress on EC, Vancouver (2006), 1115-1122.