Multiagent Infrastructure for Agent Interoperability in Open Computational Environments

Katia P. Sycara
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA. 15213
http://www.cs.cmu.edu/~softagents
Talk Outline

- Motivations for Interoperability
- Multiagent Infrastructure
- Agent Discovery: ANS and Middle Agents
- Semantic Markup and DAML-S
- RETSINA Infrastructure
- RETSINA Applications
- Conclusions
Motivations for Dynamic Interoperability

• Challenges
  – Dynamic Environment
  – Transitory Execution Environment for Agent applications
  – Undependable Infrastructure
    • Systems
    • Networks/Connectivity
    • Services

• Goals
  – Automatic / Self Configuring Agents and Infrastructure
  – Fault Tolerance and Dynamic Reconfiguration
  – Scalability in Quantity, Loading and Dispersing into Groups or over WANS
    (while still allowing systems to “find” each other)
The Many Faces of Interoperability

• Syntactic and Static via *known* Registries
  – At the component Level
  – At the connector Level
  – At the Software Architecture Level

• Semantic and Dynamic (Agents)
  – Agent/component Discovery over LANs and WANs (Discovery Mechanisms)
  – Ontological Understanding of Agent Behaviors (Ontologies, Languages, e.g. DAML-S)
  – Interacting in Meaningful Ways (Conversational Policies)
### MAS Infrastructure Requirements

<table>
<thead>
<tr>
<th>MAS Infrastructure</th>
<th>Individual Agent Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MAS Interoperation</strong></td>
<td><strong>Interoperation</strong></td>
</tr>
<tr>
<td>Translation Services Interoperator Services</td>
<td>Interoperation Modules</td>
</tr>
<tr>
<td><strong>Capability to Agent Mapping</strong></td>
<td><strong>Capability to Agent Mapping</strong></td>
</tr>
<tr>
<td>Middle Agents, Ontologies</td>
<td>Middle Agent Components</td>
</tr>
<tr>
<td><strong>Name to Location Mapping</strong></td>
<td><strong>Name to Location Mapping</strong></td>
</tr>
<tr>
<td>Agent Name Service</td>
<td>ANS Component</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td><strong>Security</strong></td>
</tr>
<tr>
<td>Certificate Authority Cryptographic Service</td>
<td>Security Module Private/Public Keys</td>
</tr>
<tr>
<td><strong>Performance Services</strong></td>
<td><strong>Performance Services</strong></td>
</tr>
<tr>
<td>MAS Monitoring Reputation Services</td>
<td>Performance Service Modules</td>
</tr>
<tr>
<td><strong>Multi-Agent Management Services</strong></td>
<td><strong>Management Services</strong></td>
</tr>
<tr>
<td>Logging Activity Visualization Launching</td>
<td>Logging and Visualization Components</td>
</tr>
<tr>
<td><strong>ACL Infrastructure</strong></td>
<td><strong>ACL Infrastructure</strong></td>
</tr>
<tr>
<td><strong>Communications Infrastructure</strong></td>
<td><strong>Communication Modules</strong></td>
</tr>
<tr>
<td>Discovery Message Transfer</td>
<td>Discovery Message Transfer Modules</td>
</tr>
</tbody>
</table>

### Operating Environment
Machines, OS, Network, Multicast Transport Layer, TCP/IP, Wireless, Infrared, SSL
Agents with Middle/Infrastructure Agents

Intelligent Software Agent

- Work Flow Manager
- Planner
- Scheduler
- Execution Monitor
- Resource Discovery
- Resource Management
- Protocols Vocabulary Dialog
- Communications

Agent-Support Infrastructure

- Dynamic Discovery of Available Infrastructure Services
- Name Services / White Pages Agent Name Services
- Point-to-Point Communications And Interactions
- Capability Lookup Svc Yellow Pages Matchmaker
- Peer-to-Peer Communications And Interactions
- Capability Mediation Broker Services InterOperator Services
- Community Discovery and Aggregation
- Coalition Services
- Natural Lang Proc
- Voice/Speech/Vision

Copyright Katia Sycara
## MAS Discovery

### MAS Infrastructure

<table>
<thead>
<tr>
<th>MAS Interoperation</th>
<th>Interoperation Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation Services</td>
<td>Interoperation Modules</td>
</tr>
<tr>
<td>Interoperator Services</td>
<td>Interoperation Modules</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Capability to Agent Mapping</th>
<th>Capability to Agent Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middle Agents, Ontologies</td>
<td>Middle Agent Components</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name to Location Mapping Mapping</th>
<th>Name to Location Mapping Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent Name Service</td>
<td>ANS Component</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Security</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate Authority</td>
<td>Private/Public Keys</td>
</tr>
<tr>
<td>Cryptographic Service</td>
<td>Performance Service Modules</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance Services</th>
<th>Performance Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAS Monitoring</td>
<td>Performance Service Modules</td>
</tr>
<tr>
<td>Reputation Services</td>
<td>Management Services</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multi-Agent Management Services</th>
<th>Management Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logging</td>
<td>Logging and Visualization Components</td>
</tr>
<tr>
<td>Activity Visualization</td>
<td>Activity Visualization Components</td>
</tr>
<tr>
<td>Launching</td>
<td>Launching</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACL Infrastructure</th>
<th>ACL Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public Ontology</td>
<td>Communication Modules</td>
</tr>
<tr>
<td>Protocol Servers</td>
<td>Discovery</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Communications Infrastructure</th>
<th>Communication Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery</td>
<td>Discovery</td>
</tr>
<tr>
<td>Message Transfer</td>
<td>Message Transfer Modules</td>
</tr>
</tbody>
</table>

### Individual Agent Infrastructure

<table>
<thead>
<tr>
<th>Interoperation</th>
<th>Interoperation Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interoperation Modules</td>
<td>Interoperation Modules</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Capability to Agent Mapping</th>
<th>Capability to Agent Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middle Agent Components</td>
<td>Middle Agent Components</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name to Location Mapping Mapping</th>
<th>Name to Location Mapping Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent Name Service</td>
<td>ANS Component</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Security</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Module</td>
<td>Private/Public Keys</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Performance Services</th>
<th>Performance Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Service Modules</td>
<td>Performance Service Modules</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Management Services</th>
<th>Management Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logging and Visualization Components</td>
<td>Management Services</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACL Infrastructure</th>
<th>ACL Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parser, Private Ontology, Protocol Engine</td>
<td>ACL Infrastructure</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operating Environment</th>
<th>Operating Environment</th>
</tr>
</thead>
</table>
Discovery Mechanisms

• Agent Discovery:
  – Static: a priori knowledge of agent’s IP address
  – Dynamic Local: based on RMI or SSDP
  – Dynamic Widearea: Retsina A2A
  – Informed: Facilitated through Infrastructure/Middle Agents (e.g. ANS, JINI, Matchmakers, Brokers)

• Discovery of Middle Agent Infrastructure
  – Static: a priori knowledge of Middle Agent’s address (e.g. UDDI)
  – Dynamic Local:
    • RMI-based (e.g. JINI/GRID)
    • SSDP-based: Retsina ANS, Retsina Matchmakers and Brokers
  – Dynamic Widearea: Retsina A2A
Agent Name Service v.2 (ANS)

The ANS is a name registration and lookup service to facilitate communication among agent applications.
Agent Communications

Agent 1: I want to talk on the network to other agents.

Agent 1: Now, I need to find other agents, and let them find me.
ANS Client Initialization
search via SSDP discovery process

Agent 1

• apple
• lemon

ANS Client

Communicator

who's Out There?

ANS Server
“apple”

ANS Server
“pear”

ANS Server
“banana”
ANS Client Initialization
active servers reply that they are available

Agent 1
• apple
• lemon
• pear

ANS Server
“apple”
I’m here

ANS Server
“pear”
I’m here

ANS Server
“banana”

Copyright Katia Sycara
ANS Client Operation

new servers announce that they are alive

Agent 1

• apple
• lemon
• pear
• banana

Hey Everyone I just came online

ANS Server “apple”

ANS Server “pear”

ANS Server “banana”
ANS Client Operation

entries are “pruned” if they are unreachable

Agent 1

• apple
• lemon
• pear
• banana

To:
apple

To:
lemon

ANS Server
“apple”

To:
pear

ANS Server
“pear”

To:
banana

ANS Server
“banana”

Copyright Katia Sycara
ANS Client/Server Operation

Agent Registration with Server

“Push”

To: apple

Push to banana

Agent 1

• apple
• lemon
• pear
• banana

ANS Server
“apple”

ANS Server
“banana”

ANS Server
“pear”

Push to pear
ANS Client/Server Operation
Agent Lookup with Server Forwarding

Agent 1

Lookup “Susan”

To:
apple

ANS Server
“apple”

1. Lookup in Local Cache
2. Check with Group Partners
3. Check Hierarchy Servers
Branching Nature of Extended ANS Lookup

- Agent
  - Primary ANS
    - Primary’s discovered partners
    - Discovered partners of other servers in hierarchy

- Discovery Group-Partners provide scaling and fault tolerance
- Hierarchy Partners allow linking different organizational groups
## DAML-S: Interoperability Language

### MAS Infrastructure

<table>
<thead>
<tr>
<th>MAS Interoperation</th>
<th>Interoperation Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation Services Interoperator Services</td>
<td>Capability to Agent Mapping Middle Agent Components</td>
</tr>
<tr>
<td>Name to Location Mapping Agent Name Service</td>
<td>Name to Location Mapping ANS Component</td>
</tr>
<tr>
<td>Security Certificate Authority Cryptographic Service</td>
<td>Security Security Module Private/Public Keys</td>
</tr>
<tr>
<td>Performance Services MAS Monitoring Reputation Services</td>
<td>Performance Services Performance Service Modules</td>
</tr>
<tr>
<td>Multi-Agent Management Services Logging Activity Visualization Launching</td>
<td>Management Services Logging and Visualization Components</td>
</tr>
<tr>
<td>ACL Infrastructure Public Ontology Protocol Servers</td>
<td>ACL Infrastructure Parser, Private Ontology, Protocol Engine</td>
</tr>
<tr>
<td>Communications Infrastructure Discovery Message Transfer</td>
<td>Communication Modules Discovery Message Transfer Modules</td>
</tr>
</tbody>
</table>

### Individual Agent Infrastructure

| Capability to Agent Mapping Middle Agents, Ontologies |
| Name to Location Mapping Agent Name Service |
| Security Certificate Authority Cryptographic Service |
| Performance Services MAS Monitoring Reputation Services |
| Multi-Agent Management Services Logging Activity Visualization Launching |
| ACL Infrastructure Public Ontology Protocol Servers |
| Communications Infrastructure Discovery Message Transfer |

### Operating Environment

Machines, OS, Network, Multicast Transport Layer, TCP/IP, Wireless, Infrared, SSL
Agent Discovery

By capability advertisement and Request through Middle Agents
Service Transaction

- Providers and requesters interact with each other directly
  - a negotiation phase to find out service parameters and preferences (if not taken into account in the locating phase)
  - delegation of service
- Providers and requesters interact through middle agents
  - middle agent finds provider and delegates
  - hybrid protocols
- Reasons for interacting through middle agents
  - privacy issues (anonymization of requesters and providers)
  - trust issues (enforcement of honesty; not necessarily keep anonymity of principals); e.g. NetBill
Matchmaker

- Requester
  - Request for service
  - Unssorted full description of \((P_1, P_2, \ldots, P_k)\)
  - Delegation of service
  - Results of service request

- Matchmaker
  - Advertisement of capabilities +para.

- Provider 1
- Provider n

Copyright Katia Sycara
Classified Ads

Requester 1

Classified Ads

Requester n

(R₁, R₂, ..., Rₖ) contact info.

Offer of service

Service results

Provider 1

Advertisement of capabilities

Provider selects requester

Copyright Katia Sycara
Facilitator
Combines Agent Location and Transaction Phases

Requester

Facilitator

Provider 1 • • • Provider n

Request for service+pref.

Results of service

Delegation of service

Service result

Advertisement of capabilities + para.
Discovery Over the Internet

• Problem:
  How to quickly and broadly implement a solution to allow discovery and lookup communication between widely dispersed systems…

• Our Solution:
  Piggy-back on top of an existing non-proprietary and popular (widely utilized) communications framework that provides global connectivity - Gnutella.
Wide-Area Discovery, Affinity, and Community Aggregation

- Gnutella Peer-to-Peer (P2P) networks provide random connectivity to a large range of other systems over a wide-area: the Internet.
- Agent-to-Agent (A2A) adds task identifiers to messages; these are used to classify hosts into specific communities of specific interests.
- A2A prefers connectivity partners that share the same interests. This affinity causes congregations of Agents in P2P networks to be formed.
- A2A attempts to maintain certain levels of connectivity to related Agents, and to hosts that have higher confidence/probability of providing needed information (either directly, or through their peers.)
- A2A reduces random P2P connectivity, in favor of these links with higher probability for satisfying directed or background discovery and lookup processes.
A2A Agent Architecture

- A **Task** object identifies the A2A community that the Agent will interact with.
- A **Question** object is created when an Agent asks a question to the Task object.
- An **Answer** object is an individual response associated with the specific Question.
Gnutella Peer-to-Peer (P2P) and CMU’s Agent-to-Agent (A2A)

Random Connectivity

Connectivity to Interest Groups:

Copyright Katia Sycara
Agent Languages and Ontologies

Formal Languages that have well defined syntax and Semantics

- To be used both as languages for Web page markup (semantic Web), and
- As agent communication content languages
Semantic Markup for Internet Pages

- HTML is markup for making pages viewable by humans
- XML has more rigorous syntax and allows semantics by agreement
- **DAML** is XML+semantics (provided by RDF)
Ontological Reasoning in RDF

Type constraint violation: The range of owns is Fish.

OR There is no inconsistency: Wanda is a fish!  Mermaid?
Cardinality constraint violation: George can’t have two majors

OR There is no inconsistency: Engineering = Arts & Sciences
Consequences of Semantic Markup

- Semantic markup allows the Web to be used as a vast knowledge base
- Semantic markup facilitates distributed inference
- Semantic markup allows pages to be viewed as “services”
Industry Efforts in Web-Based Services

- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Service Description language)
- E-speak
- ebXML
DAML-S: A DARPA Agent Markup Language for Services

• DAML-S Objectives:
  Provide an upper ontology for describing properties of agents & (Web) services in a semantically meaningful, computer interpretable markup language.

• AI-inspired markup language:
  – tailored to the representational needs of Services
  – expressive power
  – well-defined semantics
  – supports interoperation

• Release of DAML-S version 0.5 June, 2001
  – http://www.daml.org/services/
Automation Enabled by DAML-S

- Web service discovery
  Find me a shipping service that transports goods to Somalia
- Web service selection and execution
  Buy me 500 lbs. rice from www.acmemoo.com
- Web service composition
  Arrange food for 100 people for 2 weeks in Somalia.
- Web service execution monitoring
  Has the rice been ordered and paid for yet?
Upper Ontology of Services
DAML-S Service Profile

• High-level description of a service and its provider
  – description of service (human readable)
  – specification of functionalities service provides
  – functional attributes (requirements and capabilities)

• Profile used for
  – populating service registries
  – automated service discovery via middle agents
  – matchmaking
DAML-S Service Model

Service Model may be used to

1) to perform a more in-depth analysis of whether the service meets a requester’s needs;
2) to compose service descriptions from multiple services to perform a specific task;
3) during the course of the service enactment, to coordinate the activities of the different participants;
4) to monitor the execution of the service.

For non-trivial services, the first two tasks require a model of action and process, the last two involve, in addition, an execution model.
Process Model

• Subclass of ServiceModel and composed of:
  – *process*, which describes a service in terms of its actions
  – *process control model* to allow agents to monitor execution of a service request

• The top level class is Process
  – Subclass of Process is CompositeProcess
  – Properties of Process are: *parameter, input, output, participant, precondition, effect*
Composite Process

• CompositeProcess has an additional property
  – *components*, to indicate ordering and conditional
    execution of processes

• Subclasses of CompositeProcess are:
  – **Sequence**: list of processes to be executed in order
  – **Split**: bag of subprocesses to be executed concurrently
  – **Unordered**: bag of processes executed in any order; all of
    them must be executed
  – **Split+Join**: concurrent execution of a bag of sub-processes
    with some partial synchronization
Composite Process (cont)

• Subclasses of CompositeProcess are:
  – **Choice**: a composite process with additional properties “chosen” and “chooseFrom” – these are useful in specifying new subclasses, such as “choose at least n from m”
  – **Condition**: composite process with an output property (conditionValue) that has a binary range. Useful for specifying test actions
  – **If-Then-Else**: composite process with intended semantics “Test If-condition; if True do *Then*, if False do *Else***”
  – **Iterate**: composite process whose *next process* property has the same value as current process.
  – **Repeat-Until**: subclass of If-Then-Else class
DAML-S: Supporting a Service Grounding

- Service Process
  - Provides a specification of service access information.
- Specifies:
  - communication protocols, transport mechanisms at varying levels etc.
  - E.g., SOAP, HTTP forms, KQML, OAA ACL, Java RMI, RPC, etc.
Current Status of DAML-S

• Initial version of DAML-S on www.daml.org/services/daml-s

• Intended for review and comment from the DAML community

• To be added later
  – Semantics for DAML-S: DAML-S is defined in DAML+OIL and inherits its semantics. Semantics for DAML-S will also be produced
  – Service Grounding Ontology
  – Process Control Ontology to support service execution monitoring
  – Conversational Protocols: current definitions are limited in specifying messages to be exchanged between service participants in carrying out a service
RETSINA - REusable Task-based System of Intelligent Network Agents

• Adaptive, self-configuring collection of distributed heterogeneous autonomous agents that interact with humans and each other over WANs in real-time in a dynamic environment:
  – find each other through discovery in open environments where network connections, information sources and agents can unpredictably appear and disappear
  – access, filter, aggregate and distribute information
  – integrate information management and decision support
  – interleave planning, information management, execution and monitoring
  – anticipate and satisfy human information processing and problem solving
  – notify users and each other about significant changes in the environment
  – discover and interoperate semantically
  – adapt to user, task and situation
  – include “agentified” legacy systems
RETSINA Functional Organization
Some RETSINA Applications

• Aiding Human Teams in joint mission planning (using ModSAF as a simulated battlefield)
• Robot teams for de-mining
• Team Rescue Scenario (NEO)
• Agent-based “on the move” collaboration on mobile devices
• Agent-aided aircraft maintenance
• Agent-based financial portfolio management
• E-commerce in wholesale markets (agent-based auctions and negotiation)
• Agent-based Supply Chain Management
RETSINA Agent Architecture
Distributed Agents Supporting Different Functions

Multilevel fused intelligence reports

Reconnaissance Data

Fused Data from different Battlefield regions

Customized Views for Different Roles

Copyright Katia Sycara
Anytime Anywhere Agents

As Jim tells Brad about the briefing, a map showing the location of the TIC automatically appears on Brad’s laptop or nearby graphical device.

Jim wants to inform Brad that the briefing is to be held at the TIC (Technology Integration Center).

The RETSINA People matcher locates Brad’s Personal Agent.
RETSINA Agent Visualization Tool
Air-Force Aircraft Maintenance - Current Situation

- Aircraft (e.g. F-15) arrive at maintenance airbase

- Periodical maintenance must be completed in 60 days (varies for different aircraft types)

- Mechanics inspect aircraft. For a discrepancy:
  - If repair method is unknown to mechanic:
    • Consults: experienced mechanics, manuals. Else,
    • A 202a form is filled (by hand). Sent to experts
    • Reply (202b) includes repair instructions

- Mechanic repairs and files a report (and the 202)
Aircraft Maintenance Agent Organization
Matching News Articles for NSCP

2. "Microsoft And Netscape同意合并" [Tue, Dec 12, 1995]
3. "Netscape, Sun idle on Internet venture plan" [Thu, Dec 14, 1995]
5. "Netscape, Sun to form joint venture" [Wed, Dec 13, 1995]
27. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
32. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
33. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
34. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
35. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
37. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
38. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
40. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
41. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
42. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
43. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
44. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
45. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
46. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
47. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
49. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
50. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
51. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
52. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
53. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
54. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
55. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
56. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
57. "Netscape, Sun to merge" [Wed, Dec 13, 1995]
Coalition Formation for Volume Discounts
Conclusions

- Agent Technology is key to important aspects of interoperability
- Provide completed specification of DAML-S
- Provide algorithms for matchmaking and service composition for DAML-S
- Applications built using A2A discovery and interaction
- Experimental investigation of A2A scheme regarding scalability, reachability, network congestion and security