Agent Interoperation Across Multiagent System Boundaries

Joseph A. Giampapa, Massimo Paolucci, Katia Sycara, Martin Van Velsen

March 7, 2000

1 Overview

One of the problems facing open, multiagent systems [MAS] operating on the Internet is that as the number of MAS architecture-specific agent communities increases, so too does the difficulty of locating and collaborating with agents in communities of different MAS architectures. Each MAS has its own architecture-specific features such as: agent registration, agent capability advertisements, strategy for finding agents, agent communication language [ACL], agent dialogue mediation, default agent query preference, and agent content language, to name a few.

In this exhibit we demonstrate the RETSINA-OAA InterOperator, which is described at length as a long paper in this conference. The RETSINA-OAA InterOperator acts as a connection between two multiagent systems with two radically different agent architectures: the RETSINA capability-based MAS architecture, and SRI’s Open Agent Architecture [OAA]. The task of the InterOperator is to allow any agent in the RETSINA system to access services and information provided by OAA agents, and for any agent in the OAA system to access services and information provided by RETSINA agents.

2 The RETSINA and OAA MAS Architectures

Figure 1, below, illustrates the two multiagent systems: RETSINA is on the left, and OAA is on the right. OAA is organized around an agent called the Facilitator, which manages all the communications between agents in such a way that no two OAA agents directly communicate with each other.

On the other hand, the RETSINA system is constructed on the principle that all agents in the MAS community should be able to communicate with each other directly. Agents in the RETSINA system find each other through a Matchmaker agent, which, in contrast to the Facilitator, does not manage the transaction or communication between agents. The Matchmaker allows agents to find each other and then allows them to interact with each other directly.
3 Principles of MAS Interoperability

The RETSINA-OAA InterOperator, shown above, between the two MASs, “bridges” the two worlds of RETSINA and OAA by advertising RETSINA agents with the OAA Facilitator, advertising OAA agents with the RETSINA Matchmaker, and by enabling agents from both RETSINA and OAA worlds to send messages to each other.

Five principles guided our design of the RETSINA-OAA InterOperator:

1. MAS interoperators should maintain distinct MAS architecture boundaries;
2. MAS interoperators should be scalable in order to preserve the open systems architectures of both participating MASs;
3. MAS interoperators should present an increase in savings relative to the amount of effort that must be invested in their development, so as to have increased functionality for the agent system;
4. MAS interoperators should cross register agent capabilities from one MAS architecture community to another so as to maintain maximum accessibility of both systems to each other’s capabilities. Furthermore, this cross registration should be performed in real time so as to maintain a high fidelity representation of agent community state; and
5. MAS Interoperational Transparency. Agents should dialogue with each other across MAS boundaries without being aware that the interoperator is present.
Figure 2: The InterOperator Synchronizes RETSINA and OAA Advertisements

4 The Cross Registration Process

We presume that agents in multiagent systems join and leave their communities at will. InterOperators are no exception. When an InterOperator joins a community, the first thing it does is attempt to synchronize the advertisements of one MAS system with the advertisements of the other MAS, and vice versa. It does this by issuing a “monitor advertisements” query to the RETSINA Matchmaker and by issuing a “get current solvables” query to the OAA Facilitator. Figure 2 illustrates this type of bi-directional advertisement synchronization.

Once the InterOperator has synchronized the RETSINA Matchmaker with the OAA Facilitator, agents from the two communities can share and request each others’ capabilities. OAA agents may enter the combined community asynchronously, as illustrated in Figure 3, as may the RETSINA agents, as illustrated by Figure 4.
5 Message Passing and Translation

RETSINA and OAA agents may pass messages in any number of ways. Messages may be sent unidirectionally without any expectation for a reply, as simple informative messages. As queries, messages require that query state be maintained somehow. That state may be maintained for a duration of several seconds, as would be appropriate for a human to wait, or for a duration of several minutes,
or even hours. If the duration of a query is for at most several seconds, then we call this an “active” query. Such a query might be for an information agent to report the current weather conditions. Queries that last for several minutes, or even hours, are called “persistent” queries. Such queries are the kind that are registered with the RETSINA Matchmaker and with the OAA Facilitator.

Figure 5 illustrates a RETSINA interface agent, called a Messenger, registering a persistent query with the OAA Facilitator via the InterOperator. The persistent query “path” is indicated by message label 3. Message label 1 indicates the Messenger agent registering with a voice recognition interface agent. Message labels 2 indicate the Messenger agent registering with the other two Messenger agents so that they can share information and query replies with each other.

Figures 6 and 7 are two parts of a sequence of interactions that demonstrate substitutability across MAS boundaries. The sequences of Figure 6 are as follows:

1. A form of interface agent, the Messenger agent determines that it should find an agent that can provide the weather conditions for a certain city.
The Messenger queries the Matchmaker to discover if there are any such agents.

2. The Matchmaker replies that there are two agents that can answer questions about the weather: the RETSINA-OAA InterOperator, and the RETSINA WeatherCNNAgent, not necessarily in that order.

3. The Messenger performs whatever reasoning it can about which of the two agents it should contact. The WeatherCNNAgent should be quicker to contact because there is no need to translate the query, but the weather capability that the InterOperator listed indicates that its web-based information source is faster. Consequently, the Messenger queries the InterOperator, which then receives the query, converts it, and issues it to the OAA Facilitator.

4. The OAA Facilitator invokes whatever agents it can to solve the query. Since it cannot find the indicated city, it invokes another agent to perhaps translate the city name. The scenario then continues in Figure 7.

5. The OAA Weather agent fails, possibly because the information source
Figure 7: OAA Intellicast Weather Agent Fails; Query the RETSINA CNN Weather Agent

could not find weather for the city requested, or possibly because the web source was too slow, and sends an error reply back to the Messenger via the InterOperator.

6. The Messenger then consults the list of agents that the Matchmaker had previously sent it. It has yet to try the CNNWeatherAgent, and so sends that agent the weather query.

6 Agents Unregister Across MAS Boundaries

For completeness, the demo concludes with an illustration of how the RETSINA-OAA InterOperator keeps both the RETSINA and OAA MAS worlds updated on the agents and capabilities available, even when they unregister.
Figure 8: The InterOperator detects, translates, and cross unregisters the retract event for an OAA agent.

Figure 9: The InterOperator detects, translates, and cross unregisters the unadvertise event for a RETSINA agent.