INTEROPERABILITY, STANDARDS, AND SOFTWARE AGENT SYSTEMS

Margaret Lyell
The MITRE Corporation
McLean, VA 22102

Future Combat Systems (FCS) will involve network-centric approaches and will include autonomous robotic systems, sensor platforms, and reconnaissance and surveillance capabilities, among others. Among the technologies that have been put forth to achieve the objectives of FCS is that of software agent systems.

Jennings (2001) has suggested a software agent-based system as a viable framework for developing complex systems. He states that decomposing the complex system into components "in terms of the objectives they achieve" is both a natural approach and a characteristic of agent-based systems. Recent efforts have addressed the software engineering aspects of multi-agent systems.

The Foundation for Intelligent Physical Agents (FIPA) (FIPA-SPEC, 2002) is engaged in ongoing standards development for multi-agent systems. FIPA standards cover the areas of agent management / platform specification, agent communication, and agent message transport. In addition, standards for specific applications, such as a 'wrapper agent', are provided. The development of software agent standards has spurred reference platform implementations.

Yet, all applications of interest to FCS are not necessarily agent-based. Applications also may be developed in the context of the Java 2 Enterprise Edition (J2EE) framework, a Web-centric, client-server component architecture (J2EE-SPEC, 2002). The J2EE framework also includes Application Programmer Interface (API) suites, among which is the Java Message Service (JMS). Typical elements of the J2EE platform include a browser, servlet, and Enterprise Java Beans (EJBs). The present work focuses on the interoperability of FIPA-compliant software agent-based applications with applications developed for the J2EE platform. In this investigation, two modes of interoperability are considered: (1) that provided by messaging via the Java Message Service (Lyell et al, 2001) and (2) that provided by a Web-Services framework. The overall investigation is specification based. Prototype development is also done in order to elucidate the issues involved. For the prototyping efforts, the WebLogic implementation of the J2EE platform is utilized. The FIPA-OS (FIPA-OS, 2002) platform is taken as the exemplar of a FIPA-compliant agent development environment.

FIPA specifications for a FIPA-compliant platform mandate an Agent Management System (AMS) and a Directory Facilitator (DF). The AMS controls the agent lifecycle. The DF serves as a 'yellow pages' registry.

Prior to a discussion of the interoperability modes, brief results regarding the performance of the FIPA-OS v2.0 agent platform are given (Lyell et al, 2002). One set of performance experiments addressed the maximum number of software agents that could reside on a FIPA-OS v2.0 platform when these agents were in full peer-to-peer communication, and using the FIPA- REQUEST Interaction protocol. The instantiation of this protocol consists of a conversation represented by a sequence of three messages of type 'request', 'agree', and 'inform'. Each software agent on the platform initiates an interaction with every other agent, thus setting off a series of message exchanges. Each agent repeats this action every 35 seconds over a total time period of 300-400 seconds for each experimental run. For these conditions, it was found that a maximum of 18 software agents could reside on the FIPA-OS v2.0 platform before the platform would cease to function. However, the practical number is less than 18 agents, as over 28 percent of the conversations incurred lost messages with the specified agent loading and communication pattern. In this case, all agents resided on one machine.

If the agents registered with a single FIPA-OS platform are distributed across two machines, a maximum agent loading of 24 can be achieved. A change in the communication pattern to that of a hierarchical design, as shown in Figure 1, increases the maximum number of agents that the FIPA-OS platform can support to 73.

The FIPA specifications pertain to software agents and multi-agent systems. Communication is at the crux of agency. FIPA-compliant software agents communicate with other FIPA-compliant software agents via an exchange of FIPA ACL (Agent Communication Language) messages. The FIPA ACL messages are exchanged according to FIPA-specified (but extensible) Interaction protocols, in a conversational manner. Each of FIPA ACL messages contains header information and message content. The header information is semantically meaningful; FIPA specifications mandate that an ACL is invalid without a performative, such as 'query', 'agree', or 'inform'.

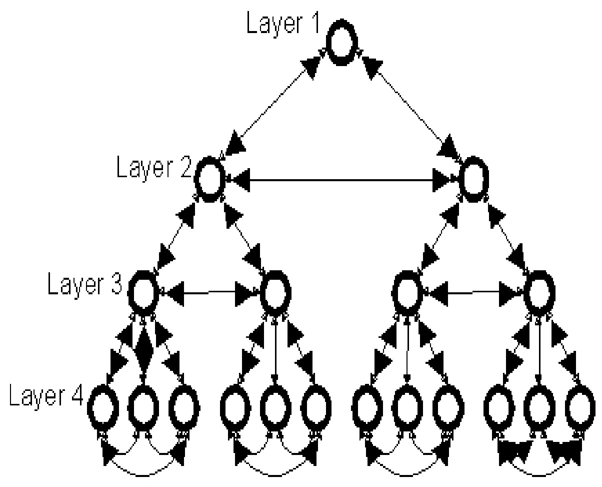

Fig. 1: Hierarchical communication pattern

The FIPA specifications permit software agents to utilize resources. Resources include 'non-agentized' software applications. A FIPA agent does not seek to communicate with a resource as it would a fellow software agent. There is a semantic mismatch between FIPA agent ACL messages, and, say, messages used in the Java Message Service (JMS). It is the FIPA Transport-Message that conceptually maps to a JMS message.

Consider the scenario in which an application that has been developed in a software agent framework produces information that is useful to (non-agent) application clients of the J2EE framework. How should the clients access this information? One approach is to use enterprise messaging, specifically, JMS, to act as a bridge between the software agent system and clients of the J2EE system. (The J2EE system provides the JMS functionality; clients must access a JMS connection.) A specific software agent publishes the information product on a JMS topic channel; interested clients listen on that same channel. The information product must be expressed in a form that the client understands. Clients receive the information product from the agent system via a JMS message.

There are limitations in using messaging as a mode of interoperability between the two frameworks. This mode of interoperation, which is strictly asynchronous, does not support pro-active, or pull, behavior by the client. Moreover, an out-of-band communication has to take place in order for the client to have knowledge of the particular topic channel that carries the information product.

A second approach that alleviates the aforementioned drawbacks utilizes Web Service technology, given by the {SOAP, UDDI, WSDL} suite (SOAP-REF, UDDI-REF, WSDL-REF, 2002), as an interoperability bridge between the multi-agent system and the browser client. Consider the scenario in which a browser client seeks temperature or windspeed information that is offered as a SOAP service by a FIPA-compliant software agent. This mode of interoperability introduces a request-response element into the here-to-fore asynchronous communications.

With Web Services, a registry (UDDI) that is independent of both the J2EE platform and the FIPA-compliant multi-agent system is used. Software agents that offer a SOAP service must register these services with the UDDI registry. Each agent that provides a SOAP service must exhibit an 'agentized' WSDL file(s). For example, in an agent's WSDL file, the 'accessPoint' is given by an agent's name, and the 'binding transport' is given as 'JMS-FIPA'. Thus, a software agent that offers SOAP services typically will register with both the DF and the UDDI registry. A still extant issue is how an agent's FIPA 'Service Description' should be mapped into the UDDI entries in a standardized manner.

Prototypes were developed to investigate both interoperability approaches. An architectural element, that of the gateway (into and out of the FIPA-compliant multi-agent system), was utilized in both prototyping efforts. Gateways were instantiated as software agents. The gateways are knowledgeable about both JMS and FIPA ACL messages. In the Web Services interoperability mode prototype, the gateway agents also act as elements of a novel SOAP-JMS binding.

Work is continuing on the identification of infrastructure elements and design elements that facilitate interoperability.

REFERENCES

FIPA-OS Platform at web-site
http://sourceforge.net/projects/fipa-os/
FIPA-SPECifications at http://www.fipa.org.
J2EE-SPECifications V1.3 available at
http://java.sun.com/j2ee/docs.html
Jennings, N., "An Agent-Based Approach for Building Complex Systems", *Comm. ACM*, Vol. 44, No. 4, pp. 35-41, 2001.
Lyell, M. et al, MITRE Technical Report 02MTR0000032, May 2002.
SOAP-REF V1.1 . at web-site
http://www.w3.org/TR/SOAP/
WSDL-REF, V1.1
http://www.w3.org/TR/wsdl
UDDI-REF, V3.0 at
http://www.uddi.org/specification.html