FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS

FIPA Peer-to-Peer Positioning Paper

Document title	FIPA Peer-to-Peer Positioning Paper		
Document number	f-out-00076	**Document source**	FIPA Peer-to-Peer WG
Document status	Output	**Date of this status**	2000/12/01
Supersedes	None		
Contact	ptp@fipa.org		
Change history			
2000/11/29	Initial draft		
2000/12/01	Minor editorial changes		
2000/12/04	Further editorial changes		

Geneva, Switzerland

Forward

This is an official document of FIPA, the Foundation for Intelligent Physical Agents. Information about FIPA and FIPA documents, including copyright notices, may be found on the at http://www.fipa.org/.

Contents

1 Introduction

This document presents a positioning of agent technology and the FIPA specifications in regards to peer-to-peer computing. It is submitted together with the FIPA Abstract Architecture (see [FIPA00001]) in response to the Peer-to-Peer Working Group's RfP of 10/12/2000.

2 What is Peer-to-Peer Computing?

The peer-to-peer working group defines peer-to-peer computing as "*sharing of computer resources and services by direct exchange*." Current popular applications of peer-to-peer computing include distributed file sharing and distributed processing. However, these systems pose just examples of what truly comprise peer-to-peer computing; as commercial peer-to-peer systems become developed and deployed, they will address applications such as dynamic integration and coordination of systems residing on arbitrary nodes of a network (ranging from wireless devices to server-class computers) in areas such as enterprise application integration, e-commerce, and network management.

Since peer-to-peer computing involves distribution of resources, computation, and services across an entire network of systems, and since control of the participating systems is outside of a central administrator's hands, new issues arise:

- Nodes in a peer-to-peer network may go on- and off-line constantly and arbitrarily.

- Location of services and resources are not fixed; they may move from machine to machine, and machines may move from one domain to another.

- Systems need to have control over how their resources and services are used (i.e. preserve autonomy).

Thus, questions arise as to how to efficiently and robustly deal with such dynamic environments as opposed to the relatively static environments of conventional server-based approaches.

The true power of peer-to-peer systems will not be tapped until many heterogeneous nodes (originally from different applications or providing component services) are able to communicate effectively with one another and compose their services dynamically. This creates significant challenges in:

- Enabling interacting nodes to use new, dynamically created interaction modes to structure their communication or even switch between interaction modes (from a simple query, into a negotiation mode, followed by a step by step transaction for example).

- Describing peer-to-peer communication at an *application level* using domain ontologies, structured message content and semantics to enable communication based on application needs and abstract away from implementation details.

- Leveraging flexible communication tools such as libraries of interaction modes, ontologies and communication "languages" to effectively allow cross application interoperation and composition of heterogeneous peer-to-peer services.

3 The Role of Agents in Peer-to-Peer Computing

The area of multi-agent systems has addressed and provided answers to these questions for many years. Agents are effectively autonomous entities that coordinate their actions. Agent approaches are used for applications requiring

- Localized decision making across distributed systems,

- Dynamic introduction of new data formats, and,

- Dynamic processes, for example, in workflow or business logic.

- High-level communication incorporating flexible interaction protocols, ontology and semantic models for communication.

As such, theories, architectures and techniques developed for constructing agent systems are eminently applicable to peer-to-peer computing. In particular, the Foundation for Intelligent Physical Agents (FIPA) has developed specifications promoting the interoperability between agent-based systems. For this reason we believe that the FIPA Abstract Architecture provides a sound basis for peer-to-peer computing.

4 FIPA's Abstract Architecture

The FIPA Abstract Architecture (see [FIPA00001]) defines an agent to be a computational process that implements the autonomous, communicating functionality of an application. An agent performs actions, such as offering services or providing resources to other agents. As agents are autonomous, they are not de facto required to perform these actions upon request[1]. Instead, getting other agents to perform actions is controlled by passing messages with specified semantics. Message passing is supported by message transport services in the agent infrastructure. It is up to specific implementations as to which message transport services are offered, for example, HTTP, IIOP, XML, etc. Other FIPA specifications address specific instantiations of message transport services and gateways connecting these services. Discovery of which agents perform specified actions is supported by directory services in the agent infrastructure. It should be emphasized that the directory services provide a mechanism to finding other relevant agents, but do not themselves mediate the exchange of messages among agents - exchange of services between the agents is direct (or at least as direct as any underlying message transport service can be).

The companion document to this submission, PC00001H.pdf, explains the FIPA architecture in more detail.

The specific requirements of the RfP addressed by FIPA are as follows:

- *Manage complexity:* FIPA standards for interaction protocols, an agent communication language (FIPA ACL) and content languages allow application designers to concentrate on application level descriptions of their systems. The standards provide tools for interaction and communication that reduce the level of complexity in the environment by abstracting away from implementation details and providing a well defined framework for system interaction.

[1] This is the main extension Agent Oriented Programming provides over Object Oriented Programming (OOP) – in OOP, objects are required to execute the methods upon invocation.

- *Extensible*: The Abstract Architecture provides the core functionality required by peer-to-peer systems. Further functionality can be added by individual reifications of this architecture and as required by specific applications or application areas. In particular FIPA standards follow a modular scheme which support the easy addition of new components in most areas (New content languages, interaction protocols and message transport for example).

- *Site autonomy:* Agents are by their very nature autonomous. FIPA has taken autonomy into account at the core, including providing for local data control and management as well as local decision-making.

- *Secure:* The abstract architecture addresses a number of security issues, including Identity, Access Permissions, Content Validity and Content Privacy. FIPA is currently in the process of establishing specifications for those security issues specific to agent-based systems.

- *I/O:* [Not directly addressed by FIPA.]

- *Accountability:* [Not directly addressed by FIPA.]

- *Multi-language:* The core FIPA specifications are totally language independent. FIPA provides a number of libraries for incorporating and adapting to different message syntaxes (bit-efficient, XML, string-based), message content languages (RDF, KIF, SL), and message transport mechanisms (IIOP, WAP, HTTP). Individual implementations may incorporate further protocols, and the libraries may be extended upon demand. There are no requirements on actual implementation languages.

- *Fault-tolerance and exception management:* As with any large-scale distributed system (especially incorporating autonomous nodes), functioning of the nodes on the network and the underlying transport system can not necessarily be guaranteed. The fault tolerance provided by FIPA infrastructure depends upon the transports used and/or the particular platform implementation. Since the needs of applications differ, fault tolerance and exception handling at the application level is most often handled by the implementation of the interaction protocols being used.

- *Scaleable:* Through the provisioning of dynamic directory services, FIPA-based systems are infinitely scaleable.

- *Simple namespace management:* Agents have globally unique names; the FIPA Abstract Architecture allows the use of any existing or future namespace management system.

- *Resource discovery & management:* Provided by the directory service and associated registration, modification and search constructs.

- *Persistence:* [Not directly addressed by FIPA.]

5 Perspective

Drawing upon the abstract architecture, FIPA has developed a number of concrete specifications addressing the issues of agent management, agent communication and agent message transport. While not directly relevant to this RFP, these specifications may be suitable for future work of the Peer-to-Peer Working Group.

The FIPA specifications are mature: a number of implementations of the specifications are available publicly and open-source, and interoperability has been demonstrated.

FIPA believes the FIPA Abstract Architecture provides a sound basis for important aspects of peer-to-peer computing. It also provides a link to additional FIPA specifications that may be relevant in the area of peer-to-peer applications.

6 About FIPA

FIPA is an international organization dedicated to the promotion of intelligent agents in industry, and currently has over 60 members worldwide. Founded in 1996, FIPA published its first set of specifications in 1997. FIPA members and non-members have developed over 10 implementations of various aspects of the FIPA specifications, many of which are public and/or open-source. More details about FIPA and the FIPA process can be found on our Web site at http://www.fipa.org/. In particular, all interested parties are invited to attend the 20th meeting of FIPA in Phoenix, Arizona from 29th January 2001 to 2nd February 2001 kindly hosted by Intel. Please contact the FIPA Secretariat (secretariat@fipa.org) for more information.

7 References

[FIPA00001] FIPA Abstract Architecture Specification. Foundation for Intelligent Physical Agents, 2000.
`http://www.fipa.org/specs/fipa00001/`