Agent Standards

Ian Dickinson
Agent Technology Group
Hewlett-Packard Laboratories, Bristol, UK
ijd@hplb.hpl.hp.com

Abstract

This paper summarises a presentation made to a meeting on Intelligent Agent Technology organised by the EPSRC Community Club in Advanced Computing Techniques. The paper briefly reviews the needs, objectives and difficulties of standardisation for intelligent agents, then discusses one particular current activity: the Foundation for Intelligent Physical Agents (FIPA). In particular, the FIPA 97 standard is discussed, and the main features of the three key technology areas are introduced. Ongoing work in the FIPA programme is also outlined.

1 Introduction

The field of *Intelligent Agents* is a hot topic in computer science at the moment. Evidence for this is the proliferation of books, periodicals, mailing lists and WWW sites devoted to agents of one kind or another. Like many new technologies, the jargon of the domain is often used rather imprecisely, and many good and wondrous things are promised, not all of which are plausible or even necessarily desirable. Nevertheless, the underlying belief amongst agent researchers is that the set of techniques, technologies and computational metaphors that form this rather loosely defined field do, in fact, promise significant steps forward in designing and building interesting, novel computational systems. Consequently the field is very active, and a great deal of research is underway in both academia and industry.

A significant class of agent systems consists of open, distributed architectures in which independently developed and executing agents interact socially to solve problems or provide services to end users. Being open, such architectures depend on precise standards that allow the components to inter-operate. The question then arises: when can or should these standards be defined? One school of thought is that the intelligent agents field is still too immature, that the underlying technologies and models are still very much under-developed. Evidence for this view is that there is not even a clear definition of what an agent actually is: most agent researchers will have their own definition. Standardisation, argued from this point of view, can be a barrier to progress by imposing an artificial consensus and restricting design freedom. An alternative view is that the attempt to define standards will further the field by encouraging a market for agent systems, having the twin benefits of building critical mass of applications and services more quickly, and engaging the resources of commercial R&D to complement academic research.

Some standardisation activities are underway, although their eventual success is still an open question. This paper investigates some of these activities, and in particular reviews one such: the work of the Foundation for Intelligent Physical Agents (FIPA).

The opinions expressed in this paper are those of the author, and should not be viewed as official statements of any of the bodies mentioned. In particular, although the author has been an active contributor to FIPA since June 1996, this document does not represent an official statement of the views of the Foundation for Intelligent Physical Agents.

2 Agent Standards: Rationales and Objectives

A typical intelligent agent scenario runs as follows:

> *You are editing a file, when your PDA [Agent] requests your attention: an email message has arrived, that contains notification about a paper you sent to an important conference, and the PDA correctly predicted that you would want to see it as soon as possible. The paper has been accepted, and without prompting, the PDA begins to look into travel arrangements, by consulting a number of databases and other networked information sources. A short time later, you are presented with a summary of the cheapest and most convenient travel options.* [1]

Many assumptions underlie such scenarios. One key assumption is that the different agents and other information sources on the network are able to find each other (possibly only from knowing what they need, not who they need) and communicate effectively.

Other assumptions are that this distributed system of interacting agents will be developed by different vendors, with different implementation strategies and constraints. The agents will be widely distributed across large-scale networks such as the Internet. The overall properties of the system will emerge from the interactions of independent agents, making it more robust and flexible. The agent universe is open and heterogeneous.

Clearly, such openness will require standard interfaces between the components (i.e. the agents and other active entities). There are several approaches to achieving this level of standardisation, as discussed below.

First, it can be observed that most, if not all, existing approaches are essentially bottom up. That is, the technological solutions are addressed to basic infrastructure needs common to distributed applications: message passing, identification and naming, addressing, mobility, etc. Such needs are generally tractable, but suffer from two related criticisms: first they do not really differentiate themselves from other distribution technologies (e.g. CORBA), and second taken together they do not define an entity we may plausibly describe as an agent.

Indeed, it is the attempt to go beyond the straightforward distributed system properties that starts the now well-worn controversy about whether this or that software entity is an agent. Nevertheless, this barrier must be crossed to realise the benefits of flexible service composition, and social interactions with and between agents, as suggested by the scenario above.

2.1 Current standardisation activities

Standards arise in two ways. *De jure* standards are created by consensus and debate among interested parties specifically to be the standard. *De facto* standards arise from the widespread adoption of a particular technology created initially by one or a few of the interested parties.

In the intelligent agents field, standardisation activities include:

- **ARPA Knowledge Sharing Effort (KSE)**
  The KSE is a consortium to *"develop conventions facilitating sharing and reuse of knowledge bases and knowledge based systems"* [2]. The outputs from the KSE are specifications for the Knowledge Querying and Manipulation Language (KQML) and the Knowledge Interchange Format (KIF), and Ontolingua, a tool for specifying ontologies.

  It is not clear that the KSE itself is still active, though many individual research groups continue to use and refine the core tools. No body, however, appears to be taking overall responsibility for continuing development of the original standards.

- **OMG Mobile Agent Facility (MAF)**
  MAF is a proposal from a consortium including IBM, General Magic, Crystaliz, GMD-Fokus and The Open Group. It was jointly submitted as a proposal to the Object Management Group, the custodian of the CORBA standard, for an extension to the CORBA standard to include objects that can migrate and retain their computational state [3]. The MAF submission was debated by OMG, but its current status is unclear. The MAF documents are no longer reachable from the home page of General Magic, though there is still a reference on the IBM site pointing to a page at the OMG [12].

- **The Agents Society**
  The Agents Society was set up *"to assist in the widespread development and emergence of intelligent agent technologies and markets"* [4]. References on the web site include the "Common Agent Platform/Simple Agent Transfer Protocol" and the MAF (above), but the documents no longer seem to be being updated, and it is again unclear what their status is.

- **FIPA**
  FIPA is discussed extensively below.

In addition to explicit standardisation activities, *de facto* standards may arise from widely used and adopted technologies. Reviewing the traffic on the software agents mailing list [5], some example technologies that are often mentioned are IBM's *Aglets* [6], University of Stuttgart's *Mole* [7] and Stanford University's *JatLite* [8]. Features common to each of these are that they are Java based and provide mobility for objects with state. In addition, JatLite includes a KQML messaging and routing facility. While each of these may be good technologies or good implementations in their own right, it seems doubtful that they have the generality or widespread support to form a *de facto* standard by themselves. Indeed, no extant agent package or development tool currently seems likely to step up to this challenge.

2.2 The Foundation for Intelligent Physical Agents (FIPA)

FIPA [9] was started in December 1995 in the belief that some aspects of agent technology were, in fact, mature enough to consider standardising. The standardisation process would follow the format of successful efforts in MPEG and DAVIC by forming an international consortium of academic and industrial organisations to jointly develop public standards for the external behaviour of, and interfaces to, agents and agent systems.

An initial meeting was held in April of 1996, which established a work programme for 1996 and '97. The work programme culminated in the publication of an initial draft standard, FIPA 97, at the Munich meeting in October 1997. All FIPA documents and meeting outputs are publicly available on the web [9], and comments and review have been sought at each stage of the process.

At the time of writing, FIPA comprises 35 member organisations from 11 countries, plus appointed academic fellows.

The remainder of this document reviews the FIPA 97 standard.

3 The FIPA 97 Standard

The FIPA 97 standard comprises two principal components. Parts one to three of the standard are *normative technology references*, and specify a technology platform for the development of open multi-agent systems. Parts four to seven present a set of *informative reference applications*, intended to motivate the technologies in parts one to three, and provide additional context for their use. This paper principally sets out to review the technologies in the standard. Reference applications are only briefly introduced.

3.1 Objectives of the standard

A key concern with standardisation activities in general is that the time it takes to reach consensus and complete the standard is too long, and acts as a brake on progress rather than an enabler. Accordingly, the FIPA work plan, which was laid out in April 1996, fixed the release date for the initial standard to be October 1997. The intent was that the scope and ambition of the FIPA 97 standard should be so contained as to make the release date feasible and practical. Future versions of the standard will progressively both broaden and deepen the scope of the standard.

Other key objectives of the standard are:

- To enable the necessary interactions in an open multi-agent system. Succinctly, these are to facilitate communications between agents, and provide the basic services needed to construct an agent society. In addition, it is recognised that any practical system will be an integration of agent-based and non-agent software components.
- To specify only the external behaviour of system components, leaving the implementation details and internal architectures to developers.

3.2 Reference applications

As mentioned above, part of the FIPA 97 standard is devoted to the description of four *reference applications*. These are provided to motivate and illustrate the use of the technologies in the standard, and are included on an informative basis. The reference applications will be used by individual or groups of FIPA members to implement and test the technology components of the specification over the course of 1998.

The four reference applications are:

- *Personal Travel Assistance*
  Using agents to assist the traveller both to plan a trip, including choosing itineraries, choosing modes of transport, and booking tickets and accommodation, and to complete the trip successfully by giving on-trip advice.

- *Personal Assistant*
  Using an agent to assist the user with typical regular activities in the office, including diary management, email sorting, workflow, etc.

- *Audio/Video Entertainment and Broadcasting*
  Using agents to assist the consumer of digital entertainment (both broadcast and on-demand services). Typical of such assistance is selecting from available programmes: consider the day not far away when the consumer can choose between hundreds of channels of terrestrial, cable and satellite digital TV or watch any movie at any time, and consequently has a TV guide the size of the phone book!

- *Network provisioning and management*
  Using agents to provide dynamic virtual private networks (VPN's).

3.3 FIPA 97 Technologies

The three technology areas from the standard are presented below. For ease of presentation, part two *Agent Communication Language*, is described first. Later sections describe *Agent Management* and *Agent/Software Integration*

3.3.1 Part two: Agent Communication Language

In a multi-agent society, agents must communicate with each other to collaborate on the completion of tasks and goals. Communications protocols such as TCP/IP provide standards that facilitate the basic transfer of information from one place to another, but this is insufficient to support the behaviours associated with agents, such as social ability.

A common paradigm in agent communications is *message passing*, where agents communicate by formulating and sending individual messages to each other. The FIPA Agent communication Language (ACL) specifies a standard message language by setting out the encoding, semantics and pragmatics of the messages.

The standard does not set out a specific mechanism for the transportation of messages. Since different agents, which have a legitimate need to inter-operate, will run on different platforms and use different networking technologies, the messages are encoded in a textual form. It is assumed that the agent has some means of transmitting this textual form, such as TCP/IP, SMTP, etc.

Primary features of ACL

A typical instance of an ACL message is shown below:

```
(inform
    :sender buyer1
    :receiver hpl-auctioneer
    :content
       (price (bid lot07) 150)
    :in-reply-to round4
    :reply-with bid5
    :language sl
    :ontology hpl-auction
)
```

Features of the message structure are:

The communicative act type (***inform*** in this case) denotes the principal meaning of this message. In particular, inform allows an agent to let another know that it believes some proposition to be true. Communicative acts are modelled on speech act theory [10]. The act of making the utterance performs the intended action. The FIPA 97 standard contains twenty such communicative acts.

The body of the message is a sequence of key-value pairs representing the *parameters* of the message. The parameters are used to determine the meaning and pragmatics of the message. In particular, the **:content** parameter denotes that which the communicative act applies to. In the example shown, the buyer informs the auctioneer of the proposition that its bid price for the lot is 150.

The semantics of the message are specified formally in the document using a logical formalism known as SL. SL is a modal logic which also includes a model of actions. In addition to the standard logical primitives, SL contains modal operators for belief, intention, having a goal and being uncertain. Action operators include composition, sequencing, alternatives, feasibility and completion of actions. The formal semantics specified for the communicative acts is the authoritative statement of their meaning. However, a full narrative description of each act is also included.

One other communicative act that will be noted here is the ***request*** act, which allows one agent to ask another to perform an action on its behalf. The request act is used extensively by parts one and three of the standard, as shown below.

While individual messages are specified in detail, it is also recognised that inter-agent conversations often take common forms. These forms are supported in the standard through *protocols*, which are patterns of messages exchanged by two or more agents. Protocols included range from simple query-request protocols, to the well-known *contract net* negotiation protocol and English and Dutch auctions.

Review of ACL features

The definition of a formal semantics for the messages is a strong factor in favour of the FIPA ACL. The level of precision delivered by formal specifications will be essential if two wholly independent agents are to be able to communicate effectively. However, there are some remaining issues. The SL formal model contains a number of subtleties that may not be immediately apparent to developers unfamiliar with such approaches. No solution exists for formal compliance testing that would validate an implementation as conforming to the specification. It is also not yet clear the extent to which the formal model can be used solely as an *external* description of the agent's behaviour, or whether the mental attitudes in the model will dictate elements of the internal cognitive architecture of the agent.

The syntax of the ACL is very close to the widely used communication language KQML. Amongst the feedback on early draft releases of the standard this was a strongly requested feature. However, despite syntactic similarity, there are fundamental differences between KQML and ACL. It is arguable therefore that the degree of similarity is now confusing.

The syntactic encoding of the messages is easy for human developers to read, and hence convenient when debugging systems. However, it is quite inefficient, an unsuited to communications channels severely restrict available bandwidth or message length. For example, agents who use the GSM short message service (SMS) to communicate when roaming are limited to messages of 256 bytes.

3.3.2 Part One: Agent Management

A better title for this part of the FIPA 97 standard might be "*Agent Platform Services*", since it deals less with the direct management of agent but rather the services needed to construct multi-agent societies. In particular, it sets out to provide standard means for:

- Defining an agent community
- Entering and leaving the community
- Meeting other agents within the community (which includes advertising services and resource discovery)
- Communicating with agents in distant communities

Primary features of Agent Management Services

Once again, key principles are to mandate only external behaviour, leaving design decisions to implementors, and to support heterogeneous open systems.

The key to defining an agent community, referred to in the standard as a *domain*, is the directory services that support the community. Two types of directory service are defined:

- a *white pages* service maps agent identifiers to transport addresses, using whatever transport mechanism is required. The white pages service is provided by an *Agent Name Server* (ANS)
- a *yellow pages* service maps between agent identifiers and descriptions of the services they can provide. A *Directory Facilitator* (DF) provides the yellow pages service.

A given community is then simply defined as the set of all agents registered with a directory facilitator, providing a service-oriented view of the agent universe.

Both the directory facilitator and the agent name server are full-fledged agents, so communication with them uses the ACL. In particular, the standard defines a set of actions that can be embedded in a *request* communicative act to solicit the directory services to perform actions such as registering an agent, updating the registration, and searching.

The standard also specifies the *Agent Platform* as a logical place within the agent universe. A given platform, which may or may not correspond to a single host computer, contains at least one DF and ANS agent. Conceptually it is also where debugging and system management tools are introduced.

Intra-platform communications mechanisms (i.e. between agents on the same platform) are not specified. However, in order to bootstrap inter-platform communications (i.e. message passing between agents on different platforms, which may initially be unaware of each other), a normative default must be specified. This is the *agent communication channel* (ACC), and the specified default mechanism is the CORBA Internet Inter Orb Protocol (IIOP).

Review of Agent Management Features

The given directory services are the heart of the standard, and these are simple and straightforward and adequate for most purposes. The effectiveness of the directory services is predicated on the power of the service description language used to characterise registered services. Whether the agents are truly able to discover resources they weren't aware of at compile time will depend on this language. However, as such a language is domain dependent, it is not part of the standard.

The choice of CORBA IIOP is good since many distributed computing problems have already been addressed in the CORBA standard, and it would be unnecessary and wasteful to duplicate those decisions in the FIPA standard. However, some developers may feel that IIOP is too burdensome for their platforms (e.g. agents running on PDA devices).

Two features are mentioned in passing in this version of the standard, but not properly supported. These are *security* and *mobility*. This was inevitable given the process by which the standard was developed. Future versions of the standard will address these needs properly. Of the two, security is arguably the more important (and unquestionably the more difficult!).

3.3.3 Part three: agent/software integration

For the foreseeable future, agent based systems will have to inter-operate with non-agent based software systems. These range from legacy systems like billing systems and customer databases, through device controllers, to common desktop applications such as word processors. An agent system of any size or scale will have to undertake to utilise these resources in some way.

In the past agent system designers have typically tackled this problem in an *ad hoc* way, inventing a custom solution each time. Recognising this, the FIPA 97 standard seeks to provide a general mechanism, which will give a degree of universality and reusability to the non-agent software systems within the agent society.

Primary features of Agent/Software Integration

The standard defines two roles for agents that manage non-agent software or services: a *wrapper agent* which encapsulates the non-agent system and lifts it up to the society level, and a *request broker*, which is a directory service for the underlying software services so encapsulated.

The underlying model is one of service instances. An agent wishing to make use of a wrapped software system can query the request broker to discover which services are currently registered and available. It can then request the wrapping agent to make an instance of the service available for the agent's use, and then utilise the service's capabilities. The separation of brokering, connecting and invoking services allows various points at which resource management can take place. Since the wrapper agent is an agent in its own right, the service-invoking agent may have to negotiate for access to the underlying service.

Again, the interface to the wrapper and request broker roles is through actions defined in the standard which are given as the arguments to ACL **`request`** messages.

The actual connection between the wrapper agent and the underlying software system is regarded as a design/implementation issue and not part of the standard. Similarly, the details of the language used to encode services is regarded as a domain specific issue.

Review of agent/software integration features

Integration with non-agent systems is, and will continue to be, essential for real-world agent systems. The identification of the wrapper and request broker roles, and the actions they support, lends a useful consistency to typically *ad hoc* approaches.

The main design challenge for the developer still remains, however. How to wrap a given service in a way that conforms to the semantics of the wrapper, and how to connect to it, are still issues the developer must overcome.

As with the Agent Management standard, the service description language is essential to the effective use of the actions defined in part three, yet such languages are left entirely open as a domain issue.

3.4 Principal Open Issues

This section summarises the issues left open in the current standard, which are intended to be addressed during the next iteration of the FIPA work programme.

The existing standard is unproven

Although some of the underlying technologies presented in the standard were contributed by member organisations from their current research work, it remains true that taken as a whole the standard has not yet been implemented and is therefore unproven. FIPA is aware of this issue, and is encouraging individuals and groups of member organisations to undertake *field trials* during 1998. These field trials will take the form of implementing one or more of the reference applications to a standard where they can be demonstrated. The results of the field trials will be used to refine the existing standard as problems are identified and resolved. Ultimately, a second, updated, release of the current standards documents will be made ready in October 1998.

The main open technology issues are:

- *Security*
  Concerns over security affect every component of an open agent system. Issues that need to be addressed include: authentication (of agents and users), privacy of communications and user's personal profile information, non-repudiation, trust, auditing, accountability and defence against malicious or incompetent agents.

- *Ontology description and sharing*
  An ontology gives the meanings of symbols used in a message. Currently, ontologies are often specified informally (this is true of the FIPA standard) or implicit in the agent implementation. For true inter-operation, agents will need explicitly encoded, sharable ontologies.

- *Conformance to ACL formal semantics*
  There is currently no mechanism to verify formally that a given agent conforms to the ACL specification, since internal mental states are used in the semantic specification.

- *Support for mobile agents*
  While mobility is not a defining characteristic of agency, a class of useful, interesting agents are mobile, and mobility is an often requested feature. The issue is how to specify mobility without determining internal architecture, either of agents or agent platforms.

- *Balance between design freedom and testability/verifiability*

On a more general note, it seems clear that the current standard addresses the reasonably tractable enabling technologies necessary for constructing multi-agent systems. However, the hard issues, and the success and effectiveness of multi-agent systems overall, often lie more at the level of knowledge and social interactions between agents. It is far from clear what role, if any, standardisation has to play at this level. Successful implementation of the FIPA 97 standard will do much to provide the platform for the continued exploration of these issues.

3.5 FIPA work programme for 1998

In addition receiving and reviewing the results of the field trials, the FIPA work programme for 1998 includes a call for technologies to contribute to a new standard document (i.e. in addition to the update release of the FIPA 97 standard). A call for proposals has been issued [11], containing the following:

- Extensions to the three main technology areas (see open issues, above)
- Specific enabling technologies for the existing reference applications
- Proposals and enabling technologies for two new reference application areas: manufacturing technology and electronic commerce.

Responses to the call for proposals will be reviewed at the January FIPA meeting in Palo Alto.

4 Conclusions

Standardisation has been suggested as a good mechanism to facilitate the development of both agent research and agent based software products. Two principal benefits are investment by industrial corporations, thus building critical mass more quickly, and the inter-operation of systems or components (e.g. individual agents) from different vendors.

FIPA has released the first version of a standard for the enabling technology for agent communities. It is intended to be the basis for ongoing work, both within FIPA and in the agent research community at large. At the time of writing, FIPA appears to be the only *active* agent standardisation activity underway.

Successful, useful agent systems will require more than the technology specified in the FIPA standard, however. In developing agent applications, designers must also tackle issues of knowledge representation, and interaction at the knowledge and social level. It may be that standardisation can assist this process, but this need is not being tackled at present.

5 References

[1] Wooldridge M. & Jennings N., "Intelligent Agents: Theory and Practice", *Knowledge Engineering Review* Volume 10 No 2, June 1995, Cambridge University Press. Also viewable online at: http://gryphon.elec.qmw.ac.uk/dai/pubs/KER95/

[2] Neches R, "The Knowledge Sharing Effort", http://www-ksl.stanford.edu/knowledgesharing/papers/kse-overview.html, 1994

[3] "Joint Submission: Mobile Agent Facility Specification", http://www.genmagic.com/agents/MAF/mafdraft7.pdf, 1997

[4] Agent Society Home Page, http://www.agent.org/, 1997

[5] Software Agents Mailing List http://www.cs.umbc.edu/agentslist

[6] IBM *Aglets* http://www.trl.ibm.co.jp/aglets/

[7] University of Stuttgart *Mole* http://www.informatik.uni-stuttgart.de/ipvr/vs/projekte/mole.html

[8] Stanford University Centre for Design Research *JatLite*, http://java.stanford.edu/java_agent/html/

[9] Foundation for Intelligent Physical Agents http://drogo.cselt.stet.it/fipa/

[10] Austin J. L. How to Do Things with Words. *Clarendon Press* 1962

[11] FIPA's Third Call for Proposals http://drogo.cselt.stet.it/fipa/munich/cfp3.htm, 1997

[12] OMG TC Work in progress CF RFP 3 http://www.omg.org/library/schedule/CF_RFP3.htm, 1997