Specifying Protocols for Multi-Agent Systems Interaction

STEFAN POSLAD
Queen Mary, University of London

Multi-Agent-Systems or MAS represent a powerful distributed computing model, enabling agents to cooperate and complete with each other and to exchange both semantic content and a semantic context to more automatically and accurately interpret the content. Many types of individual agent and MAS models have been proposed since the mid-1980s, but the majority of these have led to single developer homogeneous MAS systems. For over a decade, the FIPA standards activity has worked to produce public MAS specifications, acting as a key enabler to support interoperability, open service interaction, and to support heterogeneous development. The main characteristics of the FIPA model for MAS and an analysis of design, design choices and features of the model is presented. In addition, a comparison of the FIPA model for system interoperability versus those of other standards bodies is presented, along with a discussion of the current status of FIPA and future directions.

Categories and Subject Descriptors: C.2.4 [**Computer-Communication Networks**]: Distributed Systems; I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Standards*

General Terms: Standardization

Additional Key Words and Phrases: Multi-Agent systems, autonomy, semantics, social interaction, specifications, deployment

ACM Reference Format:

Poslad, S. 2007. Specifying protocols for multi-agent system interaction. ACM Trans. Autonom. Adapt. Syst. 2, 4, Article 15 (November 2007), 24 pages. DOI = 10.1145/1293731.1293735 http://doi.acm.org/ 10.1145/1293731.1293735

1. MOTIVATION

Multi-Agent Systems, or MAS, are distributed systems, composed of a number of autonomous software entities called agents. In theory, MAS are usually characterized in terms of internal behaviors and external interaction between agents. The main properties for characterizing the internal behavior of agents are: the type of cognition and performance measure they utilize in choosing how to act, for example, reactive, model-based, goal-based, and utility based;

Author's address: Department of Electrical Engineering, Queen Mary University of London, Mile End Rd., London E1 4NS, UK.

how adaptive they are; how they characterize the environment in which they are situated, including their computation infrastructure and their social environment with other agents; and the degree of autonomy of their actions in their environment with respect to other agents, a human owner, and their execution. The main properties for characterizing the external observable behavior inter-agent interaction are: how they interact to share tasks and share information; how they interact as part of different types of social organizations; and their degree of cooperation with other agents.

MAS represent a powerful model to solve distributed computation problems, including being able to adapt their operation in open and dynamic environments in which the content and workload are continuously changing. MAS are able to utilize other agents for cooperative distributed problem solving when individual agents don't wish to or can't perform tasks within certain constraints or don't have the competency to perform tasks by themselves. Agents can cooperate by using an Agent Communication Language (ACL) to support the sharing of a rich agreed understanding concerning the semantics of the message content and the semantics of the communication context of the message content. The semantics of the content typically defines shared information and tasks with respect to some domain such as a specific service or application domain. The semantics of the communication context explicitly defines relationships between a particular message in relation to a context such as: the sender agent's current work flow, an agreed interaction protocol, goals and plans, or with respect to a receiving agent's beliefs, desires, and intentions. Agents can be designed to operate more competitively in marketplaces than cooperatively, so that agents can select services from a variety of other agents that best fit the constraints in meeting their goals.

Although their main internal and external behaviors characterize them as a unique type of distributed computation system, in practice MAS need to be supported by a generic distributed computer infrastructure, or set of middleware services. Software agents need to be able to transport messages, to discover which service capabilities others can offer them, and to be able to invoke other providers' services. They also need an execution environment and management services, including security and storage. These middleware services are provided by an agent's Information Communication Technology (ICT) environment, often called the MAS platform, in which the agents are embedded. The agent platform itself may not be modeled in terms of agents because mature public service specifications already exist for some types of service, for example, service discovery, data storage, message transport, etc. Wrapping or replacing these with the agent equivalent may introduce disadvantages such as an extra overhead in message invocation and less robust service designs. Hence, agents need not only to interact with other agents but also to invoke nonagent software services. Agent interaction is often local within an agent platform but it could involve remote interaction with agents in one or more agent platforms, that is, system of system models. MAS are likely in the future to remain quite heterogeneous; for example, the execution environment is often programming language–specific or operating system–specific. MAS also need to act as open systems of systems, including other MAS and non-agent services, and to support some degree of autonomy so that systems and their members can dynamically appear, disappear, and be upgraded.

Given the potential benefits of MAS interaction and the rich range of heterogeneous internal and external agent behaviors and agent platforms, it is apparent that interoperability between heterogeneous agents and heterogeneous MAS is a major challenge. The drive and ability of agents to interoperate in open and dynamic information and service MAS environments is facilitated by the use of public and standard specifications for MAS.

The purpose of this paper is to present a critical analysis of the FIPA (Foundation for Intelligent Physical Agents) standard specifications for MAS interoperability. The FIPA MAS models are as relevant and as valuable for modeling heterogeneous distributed computing services today as they have been in the past. In fact, it can be argued that the potential of the FIPA MAS model is greater today because only now a basic ubiquitous computing and communications infrastructure for multiple-agents to reside in, and real drivers to deploy them is emerging. The drivers for MAS are an interconnected and digitally augmented world that contains more functionally complex networked digital devices and services that are more interoperable, heterogeneous, and adaptive. These drivers lead to the emergence of several Service-Oriented Architectural (SOA) models of distributed computing, developed within multiple standards forums. SOA models allow their individual components to have some local autonomy and decentralised management but to be able to be orchestrated along multiple dimensions such as information, services, processes, types of enterprise organisational interaction and types of management and to balance meeting local versus global needs.

There is no reported work that has taken a holistic approach to analyze MAS interaction models, that has analyzed a comprehensive range of MAS specifications as specified by the main MAS standards activity, FIPA, and that has critiqued the various design factors and trade-offs in creating MAS inteaction specifications. These are the objectives of this article.

This article is organized as follows. The motivation for modeling semantic type MAS interaction models has already been given in this section. An overview of the FIPA MAS Agent Interaction Protocol Suite is given in Section 2. Next, in Section 3, an analysis of the important design choices and features for the FIPA AIPS model is performed. In Section 4, a comparison of the FIPA model for interoperability versus that of other standards is given. Finally, a discussion of the current status of FIPA and future directions is given in Section 5.

2. SPECIFYING MAS CONTENT AND CONTEXT EXCHANGE

2.1 MAS Properties

It is proposed that for specifications to support rich and flexible interoperability, they should specify: communication protocols not algorithms; messages that are machine readable and machine interpretable, that is, semantics and syntax; an exchange of message context as well as message content; and support for open systems, that is, systems of heterogeneous systems. Specifying message exchange as a protocol focuses on specifying a set of rules that messages must obey to be correctly formed whereas algorithms specify how to generate the messages defined by a protocol. Different algorithms, for example, implemented in different agent toolkits, can be used to generate the same message protocol and to be interoperable. Sometimes researchers state a heuristic that "[Agent] Communication can be best modeled as the exchange of declarative statements" [Genesereth and Ketchpel 1994], however, statements can be declarative without supporting definitions or supporting rules, that is, lacking some of the basic constructs needed to define a protocol.

In specifying syntactic message exchange, the focus is on explicitly defining the correct construction of the message, as a particular pattern of bytes or even bits. However, the interpretation, or semantics, of a byte is not usually explicitly defined in a way that is unambiguously machine-interpretable, for example, protocols that are defined as bit steams grouped into fields. There is a range of expressivity of semantic representations from lighter semantics such as tuple lists, through to simple nested hierarchies such as XML, to class-property graphs or objectivist type semantics such as frame-based knowledge systems and RDF-S, to heavier cognitive type semantics that include logical constraints such as OWL, the Ontology Web Language.

A message context is needed in order to guide the semantics so that congruence about the semantics can be achieved between the sender and receiver and to enable an agent to orient the semantics to a specific application or circumstance. For example, the context of a message could be used to identify a message and to be able to check that the message exchange is reliable, idempotent, timely, process-driven, and coordinated and that it adheres to organizational policies. It would indeed be very challenging to make senders and receivers stateless, holding no context information, and for the whole context to be defined in the message itself, by referencing uniquely identifiable resources to hold the context information, that is, to use a REST Model [Fielding 2000]. Hence the senders and receivers for Semantic MAS interaction tend to be stateful. Semantic MAS interaction can be specified along the following dimensions:

1. Internal agent behavior: action selection and execution;
2. External (agent) interaction to exchange the:
   a. Content of the interaction including both information and tasks;
   b. Context of the Interaction and its relation to an agent organization;
3. System, or platform, services: message transport, discovery, action execution, management and interplatform interaction.

2.2 The FIPA Agent Interaction Protocol Suite Model

The FIPA agent standards forum [Poslad and Charlton 2001] focuses on specifying protocols for external interaction and platform services rather than on the internal agent behavior, as the latter is not easily accessible and observable, in contrast to external agent interaction. In addition, suitable algorithms to specify internal behaviors of different types of agent are difficult to standardize, as they are often a combination of being problem specific, being application

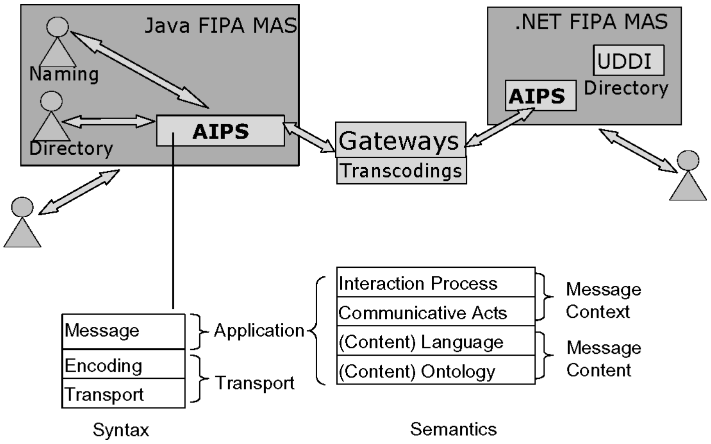


Fig. 1. FIPA specifies MAS interaction using specifications for an AIPS and MAS platform.

specific, and needing to be kept private by organizations rather than being made public.

The FIPA (agent interaction) model, often referred to as the FIPA-ACL, is really more of an Agent Interaction Protocol Suite (AIPS) rather than a single declarative agent communication language. The AIPS contains several distinct semantic protocols for agent communication including: Interaction Process, Communicative Acts, Content Logic, and Content Ontologies. The AIPS also defines several distinct syntactic protocols to specify the structure of messages, their encoding and their message transport, see Figure 2. These are explained in more detail as follows.

The Communicative Act CA protocol is the heart of the FIPA-ACL model and defines communication as a set of different CAs based upon speech act theory [Searle 1969]. Additional parameters are needed to define CAs, for example, a request type CA requires a subaction such as register in order to communicate a request to register a service description. The CA protocol defines a logical semantic definition for each type of CA in terms of the BDI context exchanged between sender and receiver agents. The Interaction Process or IP protocol, referred to as the interaction protocol by FIPA, allows a CA to be used as part of defined message sequences or workflows, to support information exchange and task delegation. The IP provides an alternative process-oriented context for message exchange. FIPA has standardized several interaction protocols such as request, recruit, subscribe, auctions, and the contract net. The content ontology protocol, referred to as the ontology language by FIPA, is used to refer to one or more application Ontologies, to define objectivist type semantics for the content. No specific ontology representation is specified. The content logic protocol, referred to as the content language by FIPA, defines general logical formulas, predicates that compute truth values and algebraic operations for combining and selecting the application ontology concepts in the message. The AIPS separates these from the applications concepts themselves that are defined in the ontology protocol. FIPA has standardised one type of content

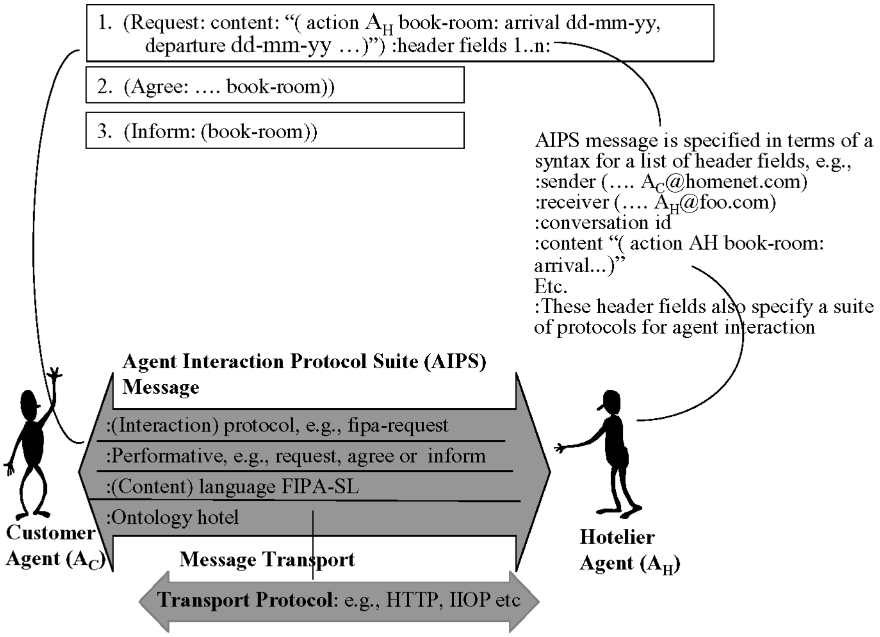

1. (Request: content: "(action A_H book-room: arrival dd-mm-yy, departure dd-mm-yy …)") :header fields 1..n:
2. (Agree: …. book-room))
3. (Inform: (book-room))
AIPS message is specified in terms of a syntax for a list of header fields, e.g.,
:sender (…. A_C@homenet.com)
:receiver (…. A_H@foo.com)
:conversation id
:content "(action AH book-room: arrival...)"
Etc.
:These header fields also specify a suite of protocols for agent interaction
Agent Interaction Protocol Suite (AIPS) Message
:(Interaction) protocol, e.g., fipa-request
:Performative, e.g., request, agree or inform
:(Content) language FIPA-SL
:Ontology hotel
Customer Agent (A_C)
Hotelier Agent (A_H)
Message Transport
Transport Protocol: e.g., HTTP, IIOP etc

Fig. 2. MAS interaction is specified in terms of a list of header fields to define the particular AIPS protocols used and other fields such as sender, receiver, etc.

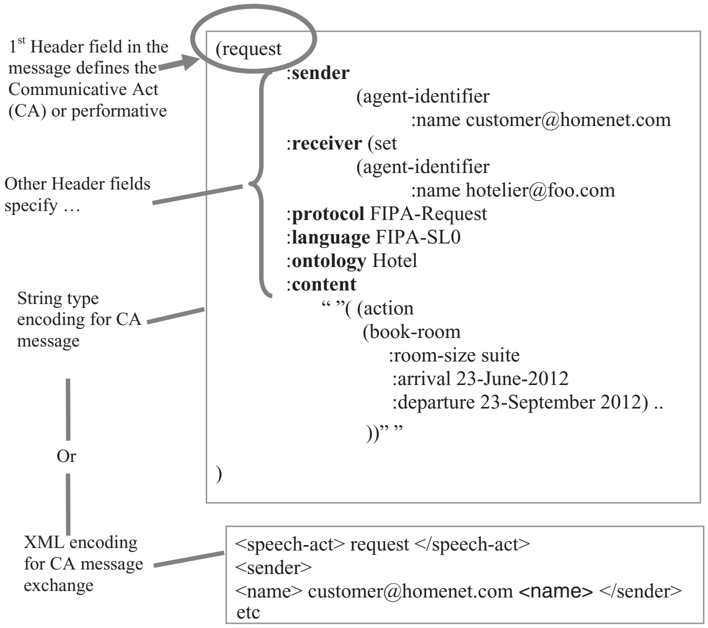
expression, FIPA-SL, [FIPA 2002]. Other types of content expression have also been specified, such as W3C-RDF and a constraint language but these did not mature into standard specifications.

Messaging protocols are often defined in terms of a simple syntactical (metadata) structure for messages such as an ordered sequence of key-value fields, for example, IETF TCP/IP protocols or hierarchical data structures, for example, the W3C Extensible Markup Language XML. Metadata to aid agent communication typically includes the sender and receiver names bound to network addresses, the message type (the FIPA Communicative Act type), the type of interaction (process) it is part of, the content ontology and content logic being used, time outs for replies and message IDs to identify other messages it is in reply to. The encoding protocol enables message exchange to use multiple encodings such as the Stings, XML, and bit-efficient, the latter was designed for use over lower bandwidth network links. FIPA Specification FIPA00084 defines the use of multiple reliable asynchronous message transport protocols for message exchange, for example, asynchronous HTTP [FIPA 2002].

The Semantic Language (SL) is used to define the semantics for the FIPA CAs as a logic of mental attitudes and actions, formalized in a first-order modal language with identity; see the FIPA Communicative Act Library Specification [FIPA 2002] for details of this logic. In order for a CA to be planned or intended by the sender, both preconditions (the reasons for which the act is selected) and the (post)conditions that should be satisfied when the act is completed, have to be specified.

< c, REQUEST (h, *a*)>
FP: FP (*a*) [c \h] $\wedge$ Bc Agent (h, *a*) $\wedge$ B_c $\neg PG_h$ Done (*a*)
RE: Done (*a*)
Where:
a is a any action expression
FP (a) denotes the feasibility preconditions of *a*
*FP (a) [*c\ h*]* denotes the part of the FPs of *a* which are mental attitudes of *i*.
c and h are schematic variables which denote agents A_C and A_H
Done (*a*) means that *a* has taken place
Agent (h, *a*) is a proposition meaning h denotes the only agent that ever performs (in the past, present or future) the actions which appear in action expression *a*.
$\neg PG_h$ means that h does not (already) have the Goal or objective G of Proposition P in this case doing action a already.

Fig. 3. Example of 1st-order modal logic formula to define the semantics of a request type communicative act message.


1st Header field in the message defines the Communicative Act (CA) or performative
Other Header fields specify …
String type encoding for CA message
Or
XML encoding for CA message exchange

```
(request
      :sender
            (agent-identifier
                  :name customer@homenet.com
      :receiver (set
            (agent-identifier
                  :name hotelier@foo.com
      :protocol FIPA-Request
      :language FIPA-SL0
      :ontology Hotel
      :content
         " "( (action
              (book-room
                 :room-size suite
                 :arrival 23-June-2012
                 :departure 23-September 2012) ..
              ))" "
)
```

```
<speech-act> request </speech-act>
<sender>
<name> customer@homenet.com <name> </sender>
etc
```

Fig. 4. Syntax for exchanging CA type messages.

For a given act, the former is referred to as the rational effect or RE, and the latter as the feasibility preconditions or FPs, which are the qualifications of the act. For each CA, its semantics have been defined in terms of the intentional state expressed by the sender agent and the associated RE and FP for selecting and intending the CA sent. Figure 3 defines the message semantics for a sender-expressed intention for an example request type CA message from a sender agent A_C to a receiver agent A_H. In a typical computational framework that implements the FIPA CA model, messages are represented in a string or XML type encoding for exchange (Figure 4).

Most messaging protocols also describe and define active architectural elements that act as messaging nodes to send and receive messages. Three main types of active architectural messaging elements are normally defined: message initiators or communication clients to request services; participants that receive message requests and act as service providers; and mediators such as brokers or directory services that act between these [Decker et al. 1997]. FIPA originally specified three distinct types of active mediator elements: a message transport service, a directory service, and a joint name and management service. For reasons of messaging efficiency, messaging was not modeled as a distinct active service entity in later architecture specifications such as the abstract architecture specification FIPA0001 [FIPA 2002]. The abstract architecture specification can be reified into different concrete agent platforms; see Figure 1. A specific type agent platform model that reifies the abstract architecture model is defined in the Agent Management Specification, FIPA00023 [FIPA 2002]. This defines an agent based directory service and an agent based name and agent management service. These can be queried to discover other agents on the platform and their status.

3. FEATURES AND CONSTRAINTS OF THE FIPA MODEL

There is no holistic semantic model to link across the individual protocols in the AIPS, for example, to make it easier to cross-check that the request type CA defined by a message protocol instance has a matching subaction defined by a content logic protocol instance and that a query type CA defined by a message protocol instance has a matching free variable defined by a content logic protocol instance to hold the results of a query. This lack of holistic structure is further complicated by the lack of a common ontology representation across the different protocols in the AIPS. For example, the CA is defined using a modal logic, the IP is defined using AUML that has no machine-interpretable semantics, and the message structure is defined as a syntax. Cranefield et al. [2005] have proposed an ontology that can be used to interlink the concepts in different ACL sublayers in order to have an online representation of the structural semantics of the whole ACL. However, there is a downside in using a single heavy ontology representation for the whole message. This makes the parsing of messages more complex. it limits the performance of high-throughput type messaging applications and it restricts the logic reasoning, for example, to handle first order predicate logic but not a logic of uncertainty.

3.1 Communicative Act Protocol

3.1.1 *Use of BDI Semantics for CA.* The CA semantics have been formally defined in terms of a modal logic. There are several advantages for using such a formal logic model for communication such as greater expressiveness. For example, the ability to differentiate between the two truths that an agent knowing some specific action or set of actions achieves a goal versus an agent believing a goal can be achieved, without knowing a specific set of actions to help achieve it [Louis and Martinez 2004].

FIPA CA semantics are viewed by the sender's mental attitude or BDI Model. Although there are principles for transferring the mental attitude in terms of belief and intention from the sender to the receiver, inherent in the model, the actual interpretation of the sender's intentional effect in the receiving agent is considered to be relative to each agent and to be customizable by each agent. Algorithms for BDI rule engines to interpret the sender's intent at the receiver are not specified by FIPA, although, this has been proposed and discussed several times at FIPA meetings.

The meaning of a CA varies depends on the context. Standardizing the semantics of a CA is likely to work for some, but not all, contexts. Examples of this include: an agent that does not need to be strictly sincere, an agent that wants to confirm a previous arrangement, an agent that is continuously updating information, and an agent that wants to withdraw specific information from selected agents rather than disconfirm information [Pitt and Mamdani 1999]. Reed et al. [2002] also present a similar argument not to fix the semantics of individual CA absolutely but to specify a framework that allows some dimensions of a CA to vary at runtime. They propose an approach they call semantic fixing in which the preconditions and postconditions of a CA and the CA beliefs can be varied and controlled depending upon the position an agent takes and its freedom to act in relation to norms that are established during contract type interaction. Additional subinteractions are, however, needed, which may involve multiple rounds and voting in order to agree on the semantics of the CA between parties. This however can significantly increase the computation overhead to the interaction and this extended approach proposed to fix the semantics may still not be universal.

Agents are presumed to act sincerely in interactions, to always speak the truth and believe each other. The sincerity assumption is used in many MAS models as it is easier to design cooperative interaction when this is assumed to be true. Agents cannot seek to lie about the beliefs they communicate. This stance however seems unrealistic in actual ecommerce.

Other criticisms and limitations of the BDI model are as follows. There are a number of different variations of BDI theories in terms of the number and choice of modalities, for example, intention being entailed as a modality or defined directly as a modality. BDI models have incomplete axiomizations and can be computational complex or even intractable. The BDI model focuses on private belief and intention transfer between individuals. It doesn't take into account third-party or societal interaction and associated constraints. BDI models seldom focus on pragmatic issues such as belief and intention management, for example, how beliefs are established, how to deal with inconsistent, partial, probalistic, conflicting, cyclic, and belief precedence in an open system. These can make the model computationally complex or even intractable. It is also not clear how a sender's mental attitudes perceives non-agent operations, for example, those that result in inexplicable failures.

Some language constructs such as the temporal constructs *Before* and *After* are implicitly referenced but not formally defined in the CA semantic specification. In addition, the semantics is underspecified in the sense that whilst receiving agents receive CAs concerning the intentions and beliefs of the sender, receiving agents are free to carry out their internal actions, such as changing beliefs, which may be consistent or inconsistent with the sender's CA. Also, the sender agent receives no information that the intended effect of, that is, the goal of, or reason for doing, the action, has resulted [Pitt 1999]. Hence Wooldridge [2000] argues that the semantics are not verifiable.

The FIPA ACL semantics focus on transferring the sender's mental attitude to one or more receivers but models of society or third parties are not considered [Singh 1998]. In a society type semantic model, agents play different roles within society and these roles define associated social commitments that constrain how agents playing a role must act and communicate. For example, one agent can allocate a task to other agents, which is consistent with its mental attitude but the task may not be allowed because of organizational constraints, that is, the agent does not have the authority to carry out the task even though it has the capability to do so.

3.1.2 *Use of Alternative (to BDI) Semantics for FIPA-ACL.* *Contract programming model semantics*: Agent communication can be specified without being formally specified. There are many proprietary MAS that interact in closed systems in this way. For example, the KQML or Knowledge Query Meta Language model uses a type of programming by contract model to specify its semantics in terms of preconditions, postconditions, and completion conditions for each of the KQML CA. Establishing the preconditions, specifies a filter or constrains for triggering event handling. The postconditions describe the states of the interacting parties, assuming successful completion. The completion conditions define the state of what actually happened.

IP Context as CA the Semantics: the FIPA Interaction Protocol model makes a rudimentary attempt at a social model in the sense that the interaction is related to the organizational roles of the interacting parties and the semantics of each CA in an IP is interpreted within the context of the IP.

Semantic commitments based upon social conventions: Jones and Parent [2003] argue that agents' commitments often confuse two kinds of norms called "preservative" and "constitutive." The first are the kind that antecedently control existing activities, for example, traffic regulation, while the second are the kind that create or constitute the activity itself, for example, the rules of the game. Hence Jones argues for a model of communication acts based not on intentions, or commitments, but on public conventions. This type of model has also been taken up in a proposed new subfield of MAS called Normative Multi-Agent Systems, specifications of MAS based upon normative behaviour (as it should be) rather than on actual behaviour (as it is), that held its first main conference in 2005. An introduction paper by Boella et al. [2005] quotes earlier work by Meyer and Wieringa [1993], which explains that normative systems are intimately related with deontic logic. Deontic logic provides a means to specify what should happen if illegal but possible behaviors occur. New modal operators are used to indicate the status of behavior: that is whether or not it is legal (normative). Deontic logic type semantics were discussed by FIPA in the early 2000s, but this did not mature to input into any standard specifications.

3.1.3 *The Definition of the Standard CA Set.* FIPA has standardized a core set of CA given in Table I. There are several distinct but related classifications

Table I. Types of FIPA Communicative Acts (CAs)

Communicative Act (CA)	Base CA	Assertive	Commisive	Directive	Mediate	Phatic	Query
accept-proposal	inform			X			
agree	inform		X				
cancel	disconfirm					X	
cfp	query-ref			X			
confirm	confirm	X					
disconfirm	disconfirm	X					
failure	inform					X	
inform	inform	X					
inform-if	inform	X					
inform-ref	inform	X					
not-understood	inform					X	
propagate	inform				X		
propose	inform			X			
Proxy	inform				X		
query-if	request						X
query-ref	request						X
refuse	disconfirm; inform					X	
reject-proposal	inform			X			
request	request			X			
request-when	inform			X			
request-whenever	inform			X			
subscribe	Request -whenever						X

for types of CA that focus mainly upon the intentional stance of the CA in terms of: asserting the truth of information, directing or delegating actions, promising to commit to future actions, expressing the mental state of the sender and declarative acts by the sender. In Table I, an extended classification for CAs is used in part based upon the generic communication functions of Roman Jakobson, quoted in Ferber [1999] and in part based upon a CA classification according to intentional stance. The main additions made by Jakobson are the definition of phatic type CAs that seek to establish, check, prolong and interrupt, that is, to help control communication. In addition, a distinction is made between mediated directives and nonmediated directives. The extensibility of the CA model can be achieved by composing new composite CAs out of existing ones; for example, see the refuse CA in Table I.

There are, however, some types of CA, missing from the FIPA core set as follows.

—FIPA CAs are mostly assertives and directives such as queries and requests; commissives that promise a commitment to some future action can be simulated (for example, an agree with the request interaction is used as something akin to promise to try to fulfil the request in the future; there are no permissive or prohibitive type directives which are used to manage, to authorise access to, infrastructure components; there are no declaratives or poetic expressive CAs [Singh 1998].

Table II. FIPA Interaction Protocols

Interaction Protocol	Task/ Info-sharing	Push/ Pull	1-1/ 1-m Receivers	Other Features
Request	Task	Pull	1-1	Cancelable (by initiator)
Request-when(ever)	Task	Push	1-1	Cancelable
Query	Info	Pull	1-1	Cancelable
Contract-Net/Iterated CN	Task	Push	1-m	Cancelable, iterated version is a multi-round IP
English/ Dutch Auction	Info	Pull	1-m	Cancelable
Broker	Info	Pull	1-m	Cancelable
Recruit	Task	Pull	1-1	Cancelable
Subscribe	Info	Push	1-1	Not cancelable
Propose	Task	Pull	1-1	Not cancelable

—Labrou et al. [1999] argue that FIPA CA to support facilitation such as broker, recommend, and recruit are missing. However, since that time, the FIPA CAs *Propagate* and *Proxy* have been added and the *Brokering* and *Recruitment* IPs have been defined.

—Although, FIPA CAs include some phatic type CAs, for example, the refuse, failure, not-understood, agree, and cancel CA, this is incomplete. FIPA for example does not support specifications for basic commands such as poll and check and for controlling modalities such as blocking, immediate, etc. [Charlton et al. 2000; Elio and Petrinjak 2005].

—KQML has two subtypes of assertive type of CA, tell (a message to create, delete or modify information and reply (a synchronous message to answer an earlier message) whereas the FIPA-CA only includes inform.

3.2 Patterns of CA: IP Model Features

In practice, an individual CA is frequently used in interaction patterns. A designer of a MAS has to decide whether to let the semantics of the individual messages determine the semantics of the conversations versus letting the semantics of the individual messages be determined to some extent by the context of the interaction and being able to vary this between different conversations. In practice, agent practitioners use both approaches. The simplest way to design interactions is to use prespecified protocols or interaction stereotypes. Agents can nevertheless engage in meaningful conversation with other agents, simply by carefully following the known protocol that relates to each other's roles in the interaction and organizations. FIPA has standardized a set of stereotypical conversations or IPs that go beyond the synchronous request-reply or asynchronous notification type of interaction used in conventional distributed computing; see Table II. The standard IPs have some limited flexibility so that conversations can be canceled by the initiator, can be refused, or can be failed by other participants. At the application level, the IP model is also extensible because interactions can be nested inside other interactions; for example, one interaction may need to require authorization from another authority in order to undertake some requested action.

A more flexible approach is to generate interactions on the fly depending on the current status and communications context, but this is computationally intensive and often avoided in practice. There are several other useful candidates for Standard IPs, for example, to support unmediated agent introductions and voting.

Phatic communication aspects such as the effects of canceling actions, asynchrony, and abnormal or unexpected IP termination and how to explicitly signal a switch to nested IPs are not explicitly addressed. There is also an issue concerning the BDI semantics of the cancel CA this is defined to cancel any single CA that has duration. This causes problems when we want the CA to affect a IP that has duration and that is made up of individual CAs that individually have no significant duration, that is, we can't cancel, or prevent from completing, an individual CA that is already done, for example, register an instance of the request CA in the Subscription IP. IPs may need to be designed at an individual level to be reversible.

3.3 Agent Platform (AP)

As mentioned previously, it is important not to replace current mature services with agent equivalent ones unless the benefits outweigh the disadvantages. In the early versions of the FIPA specifications (1997–1998), the message transport specification was modeled as an agent, but there is a downside to this—lower efficiency. Transferring a single message between agents always required sending at least two messages, one to ask the agent transport agent to send a message, the other for the agent transport to actually send the message. Hence, in later FIPA specifications, the message transport is modeled as a nonagent service.

In some cases, it is useful to specify service interfaces at a more abstract level that can be reified into different specific concrete specifications because concrete specifications evolve or because service specifications need to adjust to the constraints of particular computing infrastructures. In early versions of the Agent Transport Specification, FIPA specified the use of a single so-called baseline message transport, the Object Management Group IIOP transport, which was ideal for use in low volume transaction, wire-line, and private networks (without firewalls). However, when FIPA agents were being considered for use via firewalls, for high-transaction processing, and for wireless environments, the IIOP transport was considered to be less than ideal. It then became clear that agent service interfaces needed to be more flexible and to be able to support multiple message transport protocols.

There are several restrictions on the current agent management and middle services. Many MAS models, such as the Abstract Architecture model, mandate the use of a directory service, but they may not be needed for peer-to-peer action. The agent management interface is defined using a frame-based ontology. The current directory service modeled as the DF agent in the FIPA agent management specification does not support additional types of interaction such as query and subscribe interactions. DFs cannot be federated to support the service registrations of nonservice agents such as user agents. The service ontology for an agent specifies the interaction protocol an agent supports but it does not include the agent role within the interaction. The DF service ontology also does not define competence or reliability of the service provider. FIPA has attempted to support a system of systems model but it has only specified a flat organization for multiple directory services.

Transport issues include the assumption of a reliable transport so that messages cannot get out of order. Fields such as an in-reply-to field has been added to the ACL message header to support message management by defining an identifier for instances of IPs but no standard syntax for the structure of such fields is specified.

4. FIPA SPECIFICATIONS IN RELATION TO OTHER STANDARDS

There are currently about ten to twenty major standards initiatives that are involved in standardising distributed system interoperability and FIPA is only one particular type of these. For example, Singh and Huhn [2005] list nine different standards bodies that are developing standards for SOAs: IETF (Internet, messaging, and communication management protocols), OMG (UML modelling language, CORBA, and MDA architectures) W3C (XML, WSDL, SOAP, RDF, and OWL), OASIS (UDDI and BPEL4WS), UN/CEFACT (ebXML), WS-I (Web Service Interoperability profiles), BMI.org (Business Process Modeling Language), WfMC (Work flow Management Coalition), and FIPA. To this at least SQL (relational database interoperability), the OGC (Grid and utility computing standards) and Rosettanet (open e-business process standards), can be added. Further the W3C standards can be grouped into three sub-groups: the HTML Web, the XML Web with Web Services and the Semantic Web (RDF-S, OWL) with associated services.

It would be useful to characterize and compare and contrast these, to assess what the commonalities are and what the key differences are, and then perhaps to suggest some future directions to evolve particular standards. There are two main ways to approach such an analysis. Firstly, a new canonical reference framework could be defined; a large multidimensional space where the main features of each standard and multiple levels of communication and interaction are mapped and then compared. Secondly, one of the standards could compare only its own features against the others. In this subarticle, the scope is limited, so the focus is on the second approach, the features of FIPA are used as dimensions to selectively compare features of other standards bodies, that is, to use the analysis of the combined FIPA AIPS and AP model; see Figure1.

The majority of the standards bodies listed earlier base their message transport specifications upon HTTP transport and XML encoding and exchange formats. The other smaller group of OMG, FIPA, and IETF are more abstract and have benefits of being able to more flexibly interchange protocols and encodings for use in particular environments, for example where the use of the more verbose XML, that is, compare the N3 encoding of RDF compared to the equivalent XML encoding for RDF. At a syntactic level of interaction, FIPA has developed more (bit) efficient encodings that can be used for interaction over low-bandwidth (wireless) transports via mobile devices to services. This was demonstrated, in the CRUMPET project [Poslad et al. 2001].

At the application level of interaction, there are several important distinctions between FIPA and other approaches. The Web and SQL are based upon a minimalistic approach to specifying the set of communication primitives. Much Web interaction is based upon two communication primitives; a synchronous request-reply and asynchronous reply type notifications. SQL information exchange is based upon the so called CRUD set of communication primitives of Create, Read or query, Update, and Delete. In contrast, FIPA defines about twenty different communication primitives. Less may seem better as this minimizes the redundancy and the confusion in selecting the use of overlapping communication primitive. In the Web and conventional distributed computing models, protocols differentiate between different types of response using single-valued application protocol specific response codes. This makes the control and the transcodings more complex when services that use different protocols need to be orchestrated. In contrast, in the FIPA AIPS model, control type messages are abstracted out of the application specific protocols into a generic CA subset of phatic control messages such as failure, refuse, not understood, agreed, etc. These can be used in a standard way across heterogeneous applications and protocols, easing control and coordination. The FIPA CA model naturally allows more semantic context to be included in messages. This can give applications more understandable information about unexpected events. In addition, the richer set of FIPA CA primitives can lead to more flexible interaction processes.

Businesses and applications typically use specific patterns of individual communication primitives and this is currently an area of high activity across many standards bodies. Here several related terms apply, some standards bodies focus on processes, others on work flows, others on coordination, others on conversations, others on orchestration (centralized controller), and others on choreography (decentralized control). The semantics of these approaches is not clear, and hence how they differ is not clear. A comparison is also compounded by the high state of flux of different specifications as interest in them changes within a short space of time.

FIPA has defined a set of reusable interaction patterns and processes that are generic and can be conveniently used across heterogeneous application domains and protocols. The semantics for the FIPA interaction process models is currently weak and based upon AUML graphs. An example of the lack of semantic expressiveness of the FIPA IP model representation is that it is typical in an English Auction that the participant making the last bid does not send any answer to the next CFP, but it is not clear how best to specify this. It seems apparent that other interoperability standards have specified more explicit support to flexibility and dynamically change interactions. Van der Aalst et al. [2002] present a pattern based analysis of workflow and communication languages such as BPML, BPEL4WS, XLANG, WSFL, and WSCI. These use different explicit patterns for splitting and merging interactions, something that the FIPA IP model does not explicitly define.

FIPA has focused from the onset on supporting an associated semantics for content (information and task) exchange and for services. This whole area is still maturing, and there may well be at least a decade before there is widespread exchange and agreements for semantic data and services—this is mainly because agreements for a diverse ICT world will become more challenging. Currently, the majority of business interaction standards are still very much syntactic rather than semantic. There is a key difference in the way that the semantics of the content has been tacked in the FIPA Model compared to the Semantic Web. The Semantic Web seems to have focused on two models: the use of a lightweight ontology without logic, such as RDF and RDF-S, and on the use of a heavy ontology with a specific inbuilt logic such as OWL, the Ontology Web Language. In contrast, FIPA from the onset has specified a model where application-specific ontologies can be associated with multiple type logics. The latter has the advantage that, for example, in the domain of weather predictions, the weather domain can be associated with uncertainty logic to answer questions such as how likely it is be to be both windy and wet in Summer, in the UK.[1] Whereas in the domain of transport, temporal reasoning can be used to ascertain that lunch will be finished before an afternoon meeting starts. Note there is still a lack of consensus within the Semantic Web how to model and combine rule type reasoning within the Semantic Web model, over 5 years since it was first proposed [Horrocks et al. 2005]. It is not envisaged that a single universal logic can be created to serve diverse reasoning needs.

It is, however, certainly true that a number of standards initiatives have put in a significant effort into maturing semantic service models. The motivation to evolve from Web services is that by using semantic annotations to describe services and resources, the tasks of service discovery, selection, negotiation, and binding could be automated. Petrie et al. [2007] report the results of a challenge cup exercise to improve better approaches and to develop a common understanding of the various technologies intended to facilitate the automation of mediation, choreography, and discovery for Web services using semantic annotations. The trade-offs are explored among existing approaches such as DIANE (a method for automated service matchmaking, selection, binding, and invocation, used in the discovery scenarios), WebML/Webratio, jABC/jETI, WSMX (Web Service Modeling eXecution environment, the reference implementation of WSMO, the Web Service Modeling Ontology), METEOR-S (Semantic Web Services and Processes project at the University of Georgia), and SAWSDL (Semantic Annotations for WSDL). These six approaches are being implemented and analyzed with respect to two scenarios: a mediation scenario concerns making a legacy order management system interoperable with external systems, and a discovery scenario concerns the dynamical discovery, selection, binding, and invocation of the most appropriate shipment service for a set of given shipment requests. This is still under investigation. FIPA's semantic service model in comparison is fairly simple and predates the use WSDL or OWL-S. It is based upon an agent directory service that supports a request IP and a frame-based service description ontology. Lyell [2002] reports on the development of a gateway between Web Services and FIPA MAS, which allows agents that offer services external to an AP, to advertise them in a UDDI Registry. A mapping

[1] The Summer of 2007 was unusually one of the wettest on record in the UK.

was presented that allows FIPA "Service Description" to be mapped into UDDI entries in an ad hoc manner.

FIPA has focused on specifying and exchanging a semantic communication context for content for generic domain. This is highly controversial and is in contrast to a history of hard-coding service processes and the process context into specific application services. FIPA has proposed two different semantic communication context models to date: one based upon a modal logic of belief-intention for CAs and one based upon an interaction process model.

Finally, the combination of the FIPA AIPS and AP models, in contrast to the Web Service and Semantic Web models, naturally form a holistic model, that can use multiple transports and message encodings and that can support multilogic semantics for both content and communication context exchange.

5. DISCUSSION OF FIPA'S CURRENT STATUS AND FUTURE DIRECTIONS

The development of FIPA MAS standards has involved a variety of stakeholders trying to specify protocols that are expressive enough for designers and implementers to use whilst being able to be embedded in existing infrastructures. Often, standards may need a maintenance phase and may not work well initially in specific applications. Standards may not be able to always guarantee consistent design and interoperability in an open service environment. The benefits of standards as a key enabler to support interoperability and open service interaction in practice lead to a critical mass of users and to promote greater uptake.

FIPA (the Foundation for Intelligent Physical Agents) was established in 1996 as an international non profit association of companies that agreed to share efforts to produce standard specifications of generic agent technologies that were: produced in a timely fashion, internationally agreed, and usable across a large number of applications, so that a high level of interoperability across applications is achieved. Since then, FIPA has involved at various stages more than 60 members from more than 20 different countries-worldwide and generated a set of specifications that went through 3 cycles of review: FIPA97, FIPA98, and FIPA2000. Several distinct agent platforms, applications, and hundreds collaborative projects have used the FIPA specifications. The core set of the specifications, available from FIPA [2002], have been used for a number of years, and they are robust and effective enough to be promoted to Standard and believed that these specifications were now stable, mature, well understood and ready for commercial implementation and deployment. In 2005, FIPA became a new IEEE standards activity.

The main impact of FIPA can be summarized as:

—Standard specifications for MAS AIPS and AP with the FIPA ACL still being the most widely used model for MAS.

—Many project deployments in, for example, FACTS, MARINER, Agentcities, etc. In an Internet survey carried out in 2003 by the author, 80 different projects were identified that reported the use of FIPA.

—Ten to twenty different FIPA MAS toolkits including five open source ones have been developed. Of the open source ones, one is still living (JADE) but there are still several commercial ones that support FIPA such as JACK. JADE has a wealth of third-party extensions including a Protégé Ontology environment plug-in to allow Ontologies visually modelled off-line, to be exported for use with JADE agents.

—JADE Board: five telecom companies set up a forum to promote the JADE FIPA toolkit for mobile telecoms applications.

—JCP or Java Community process has developed JAS, the Java Agent Service, JSR87, a reference API for the FIPA abstract architecture and FIPA message protocol.

—FIPA MAS interoperability and FIPA compliance have been tested in several trials and projects.

—AUML, the Agent modelling language, originated in FIPA Modelling TC.

—The FIPA model that can reuse other standard specifications, e.g., XML message encodings and HTTP transport.

5.1 FIPA Projects

One of the first major projects to test the FIPA97 and FIPA98 specifications was the EU FACTS project, 1998–2000. This used three main domain applications to test the standards in real-life scenarios: personal travel market (PTM), electronic trading, and audio-visual entertainment and broadcasting (AVEB). Núñez-Suárez et al. [2000] report their main experiences in developing the PTM application as follows: the FIPA model reduced the amount of work required to attain application level interoperability; FIPA support provides an open distributed computing environment to integrate disparate components and while the use of technologies such as interaction protocols, agent communication languages, and ontology may be common within the agent community, they are not common within the mainstream software development industry. They also reported that the CA protocol was intuitive and comprehensive enough for use in most cases and that there was a transport interoperability problem in using the IIOP transport in a multivendor environment. Charlton et al. [2000] reported experiences in developing the FACTS AVEB application that focused on the semantics of the MAS communication. ACL semantics is based on exchanging a private model of beliefs and intentions. Conventional application developers find these complex to build and operate and are often unable to distinguish between intentions and desires in the implementation. ACL semantics is based upon a social agency or organizational model that can be used to overcome the problems of mental agency model. Note since this time, FIPA has specified a set of predefined interaction protocols with an associated social agency model. Error handling is a concern when agents are autonomous. Agents can delay responces, causing the sender to be uncertain about what has happened. Since this report, FIPA has added time-out fields to the protocol headers. Also there is a limited support to debug and monitor platforms but this has since been tackled in MAS toolkits.

Pitt and Mamdani [1999] report their experiences of using the FIPA specifications in the MARINER project to develop a MAS for load control in intelligent networks. Their primary criticisms of the FIPA ACL standards are firstly, that it is not clear whether the ACL semantics should be interpreted as an informative specification providing guidance to developers, or as a normative specification providing conditions that the agents themselves are responsible for satisfying. Secondly, any communication semantics will vary depending on the context and therefore will be difficult to standardise. They argue that interaction protocols or conversations should arguably be the focal point for standardisation of the ACL as these can provide the context for communication.

The main objective of the Agentcities.rtd project, 2001–2003, was to create a global open system to provide the conditions in which to test: FIPA based agents, services and other technologies such as delegation, coordination, modelling of dynamics and in particular communication based on formal semantics [Willmott 2002]. The main achievements were a network architecture model, a reference model for semantic service interoperability in open environments, a framework for service composition in an open environment, a test suite for FIPA-compliant Agent Platforms and several multidomain ontology-based agent services and a live testbed network that was actively used. In an associated Agentcites.Net project, towards 150 autonomous MAS became hooked up to the testbed.

5.2 FIPA Tools and Software APIs

Application developers typically use agent toolkits, rather than having to develop their own software implementations of the FIPA specifications, and layer their application software on top of it. This eases development and the amount of testing, assuming the agent toolkits undergo some form of evaluation. During the lifetime of FIPA, tens of FIPA agent toolkits have been developed that have implemented sets of FIPA specifications. Here we mention some of the main open source initiatives: JADE [Bellifemine et al. 1999], FIPA-OS [Poslad et al. 2000], and ZEUS [Nwana et al. 1999] as these were used in the FIPA interoperability tests. In addition, a JCP or Java Community process developed JAS, the Java Agent Service, JSR87, reference API for the FIPA abstract architecture specification that has been implemented in the KAoS agent toolkit [Bradshaw et al. 2004]. These toolkits typically support implement and provide APIs and tools as follows:

—APIs and implementations of codecs to parse FIPA ACL messages in accordance with the FIPA CA and other related ACL specifications.

—APIs to support agent management as defined in specification FIPA 00023 [FIPA 2002].

—Different types of Agent templates for producing agents that can then communicate with each other using multilayered support.

—Message and conversation management to support interaction protocols.

—Dynamic platform configurations to support multiple agent platforms, multiple types of persistence, enabling integration with legacy persistence software, and multiple encodings.

—Abstract interfaces and software design patterns.

—Diagnostics and visualization tools.

Software APIs and parsers for computational models of the specifications can be readily developed to exchange ACL messages at a syntactical level. One of the key technical challenges is how to implement software that can handle the ACL messages at a semantic level. Putting aside for the moment the concerns about the validity and verifiability of the CA semantics, Louis and Martinez [2004] have reported their effort to develop a computation framework to handle the CA semantics. This has been implemented as an add-on to JADE. This framework has four elements: an activity loop to send and receive messages, reflexive operations to access and update their beliefs, a semantic representation (SR) to process messages with beliefs expressed in FIPA-SL and semantic interpretation principles (SIP) to produce and consume SRs. They consider three main ways to implement their model: using a dedicated inference engine for modal logic, but these tend not to be publicly available; using a general purpose rule engine to help process messages, but this requires some mapping of the semantic model to the rule engine rules; and finally developing a proprietary ad hoc rule engine. Note that their framework focuses on belief transfer but not intention transfer, although they claim that the latter can be easily added.

An alternate approach is that most agent application developers choose in practice not to use and implement the CA semantics but to use some informal semantics based upon the interaction context. The intentional semantics of an agent, that is, the meaning of the CA, will be determined by the state an agent is in, in relation to a priori agreed interaction protocol (IP) and the type of the next received CA or sent CA. In Pitt and Bellifemine [1999], a JADE implementation of such IP driven intentional semantics is described.

5.3 Interoperability Testing and FIPA Compliance

Several authors, such as Wooldridge [2000], Pitt [1999], and Louis and Martinez [2004] have pointed out that conformance to the specification only requires the sender to respect the feasibility condition in order to send the message but does not require the recipient of a message to respect the rational effect part of the CA semantics. There are other challenges to do with the intentional model mentioned earlier. There is also the further issue of computation tractability when verifying open service interaction, where a system interacts with its environment and whose behavior depends on this interaction, an issue not considered in the above references. These issues make computational models and conformance testing of MAS to the formal semantics, to be challenging in practice.

The concern of application developers and users is less whether the application communication can be formally verified against some underlying theoretic model than whether they can apply a model and that the model performs as expected. In practice, the computation implementations of the models, not the theoretical models, can be evaluated to pass a series of tests. The construction of these ad hoc tests and the specification of the test conditions and test properties is application-specific.

There are several ways that implementations of the specifications were tested in practice. The specifications are tested as part of the experimental phase in progressing a specification from preliminary via experimental to standard. Specifications are tested as part of good development practice within specific projects that used these specifications. In addition, several specific tests of the specifications were carried out at selected FIPA meetings, in 1999 and 2001, in which the interoperability among the JADE, FIPA-OS, and ZEUS agent platforms was tested. The results of this testing demonstrated that the FIPA specifications were considered fairly thorough, in that, only minor changes were necessary to enable the "core functions" to be tested and to interoperate.

5.4 Current Activities and Future Directions for FIPA

After the intense interest in MAS and its standardization by FIPA in the late 1990s and early 2000s, the widespread deployment of MAS seems to be still be some time off in the future. In 2004, as an output from the EU Agentlink3 project, an updated roadmap for agent technology was published by Luck et al. [2004]. In this article, the authors state several judgements and predictions, including phased development of MAS from closed MAS in about 2004, to some cross-domain closed systems, through to open systems in specific domains, and finally to truly open and fully scalable open MAS in the 4th phase, circa 2009 onwards. Already we are starting to see more mainstream standards bodies undertaking applied research and development that were traditionally the sole domain of the MAS community. They are putting significant human effort into it to progress particular elements, such as ontology-based information exchange, and more recently semantic services. Recently, it seems that Semantic Web services are being driven more by the Web service than the Semantic Web community. This leads to the natural question about whether, the specification of MAS should stop within a dedicated organization such as FIPA or be absorbed into more mainstream efforts that certainly have greater resources, are often more business driven, and are based upon actual versus perceived needs.

However, a whole raft of MAS R&D challenges issues still remain to be tackled. A sensible strategy seems that the issues of defining semantic information exchange and using semantics to automatically discover and invoke services is best left to continue to progress outside of FIPA with FIPA making explicit how agents can best utilise this effort.

There are two central issues: First, what should FIPA do with its current specifications—should it draw a line or try to evolve them in an evolutionary versus revolutionary way? This could lead to incompatibilities with existing specifications, based upon some of the limitations highlighted earlier in this article. The opportunities for improvement include defining mechanisms to more flexibly synthesise new CAs and developing a specification of a multilateral view of CA semantics rather than a single one such as mentalistic attitudes. More flexible agent middleware service interaction could be proposed that leverages models such as directory services that have been advanced in other standards bodies.

Secondly, the novel and fruitful areas that FIPA can profitably work on need to be identified. These are illustrated by current FIPA activities in the following work groups; see [FIPA 2007] for more detail about these and how to become involved. The Agents and Web Services Interoperability Working

Group's (AWSI WG) main objective is to fill the communication gap between agents and Web services. Agents should be able to locate, negotiate and interact with Web services seamlessly and vice versa. The Human Agent Communications Working Group (HAC WG) is working to propose extensions, that is, additional CAs to the FIPA AIPS to facilitate agent communication with humans in general. The objective of the Mobile Agents Working Group (MA WG) is to define new specifications for communication interfaces and network protocols for efficient, reliable, and secure code and data relocation, location tracking, and location transparent communication, infrastructure management in the form of overlay networks for agent server discovery and dynamic definition of itineraries and interoperability between different mobile agent toolkits at runtime. The objective of the P2P Nomadic Agents Working Group (P2PNA WG) is to develop and define a specification for P2P Nomadic Agents, capable of running on small or embedded devices, and to support the distributed implementation of applications for consumer devices, cellular communications and robots over a pure P2P network.

ACKNOWLEDGMENTS

The opinions expressed in this article are those of the author. Past discussions by the active FIPA membership, in particular by members of the Review of the FIPA Specifications Study Group (ROFS-SG) have made valuable contributions to this article, and are acknowledged. The constructive criticism of the reviewers to help focus this article is also gratefully acknowledged.

REFERENCES

AGENTCITIES.RTD Project. 2002. Deliverable D1.4 Final Report. Available from http://www.agentcities. org/EURTD/.

BELLIFEMINE, F., POGGI, A., AND RIMASSA, G. 1999. JADE—A FIPA-compliant agent framework. In *Proceedings of the 4th International Conference and Exhibition on the Practical Application of Intelligent Agents and Multi-Agents (PAAM' 99)*, 97–108.

BOELLA, G., VAN DER TORRE, L., AND VERHAGEN H. 2005. Introduction to normative multiagent systems. In *Proceedings of the 1st International Symposium on Normative Multiagent Systems (NorMAS05),* 1–7.

BRADSHAW, J.M., ET AL. 2004. Making agents acceptable to people. In N. Zhong and J. Liu, Eds., *Intelligent Technologies for Information Analysis: Advances in Agents, Data Mining, and Statistical Learning*, Springer-Verlag, 355–400.

CHARLTON, P., CATTONI, R., POTRICH, A., AND MAMDANI, E. 2000. Evaluating the FIPA Standards and its role in achieving cooperation in multi-agent systems. In: *Multi-Agent Systems, Internet and Application, Minitrack Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*.

CRANEFIELD, S., PURVIS, M., NOWOSTAWSKI, M., AND HWANG, P. 2005. Ontologies for interaction protocols, In *Ontologies for Agents: Theory and Experiences*, V. Tamma, S. Cranefield, T. Finin, and S. Willmott, Eds., Basel, Birkhäuser, 1–17.

DECKER, K., SYCARA, K., AND WILLIAMSON, M. 1997. Middle-agents for the Internet. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, Nagoya Japan, 578–583.

ELIO, R. AND PETRINJAK, A. 2005. Normative communication models for agent. *Auton. Agents Multi-Agent Syst. 11*, 3, 273–305.

EHRLER, L. AND CRANEFIELD, S. 2004. Executing agent UML diagrams. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04)*. ACM Press, 906–913.

FERBER, J. 1999. *Multi-Agent Systems: An Introduction to Artificial Intelligence*. Addition-Wesley.

FIELDING, R. T. 2000. Architectural styles and the design of network-based software architectures, PhD thesis, University of California, Irvine.

FIPA SPECIFICATIONS. 2002. Multi-agents system standard specifications. http://www.fipa.or/specifications.

FIPA WORKING GROUPS AND STUDY GROUPS. 2007. Multi-agents system standard specifications. http://www.fipa.org/subgroups.

GENESERETH, M. R. AND KETCHPEL, S. P. 1994. Software agents. *Commun. ACM, 37*, 7, 48–53.

HORROCKS, I., PARSIA, B., PATEL-SCHNEIDER, P., AND HENDLER, J. 2005. Semantic Web architecture: Stack or two towers? In *Principles and Practice of Semantic Web Reasoning (PPSWR05)*. Springer Verlag, 37–41.

JONES, A. J. I. AND PARENT X. 2003. Conventional signalling acts and conversation. Workshop on agent communication languages. In *Advances in Agent Communication*: *International Workshop on Agent Communication Languages,* Dignum, F., Ed. Lecture Notes Computer Science, 2922, Springer, 1–17.

LABROU, Y., FININ, T., AND PENG, Y. 1999. The current landscape of agent communication languages. *Intell. Syst. 14*, 2, 45–52.

LOUIS, V. AND MARTINEZ, T. 2005. An operational model for the FIPA-ACL semantics. In *Proceedings of the AAMAS Workshop on Agent Communication (AC05)*. 101–113.

LUCK, M., MCBURNEY, P., AND PREIST, C. 2004. A manifesto for agent technology: towards next generation computing *J. Autonom. Agents Multi-Agent Syst. 9*, 3, 203–252.

LYELL, M. 2002. Interoperability, standards, and software agent systems. In *Proceedings of the 23rd Army Science Conference*, Orlando, FL.

MEYER, J-J. AND WIERINGA, R., EDS. 1993. Deontic logic: A concise overview. In *Deontic Logic in Computer Science: Normative System Specification*. John Wiley, 3–16.

NÚÑEZ-SUÁREZ, J. 2000. Experiences in the use of FIPA agent technologies for the development of a personal travel application. In *Proceedings of the 4th International Conference on Autonomous Agents*, 357–364.

NWANA, H., NDUMU, D., LEE, L., AND COLLIS, J. 1999. ZEUS: A tool-kit for building distributed multi-agent systems. In *Appl. AI. J. 13*, 1, 129–186.

ODELL, J., VAN DYKE, P.H., AND BAUER, B. 2001. Representing agent interaction protocols in UML. In Ciancarini, P. and Wooldridge, M., Eds., *Agent-Oriented Software Engineering*, Springer, 121–140.

PETRIE, C., MARGARIA, T., KÜSTER, U., LAUSEN, H., AND ZAREMBA, M. 2007. SWS challenge: Status, perspectives and lessons learned so far. *Special Session Comparative Evaluation of Semantic Web Service Frameworks at the 9th International Conference on Enterprise Information Systems (ICEIS07)*.

PITT, J. AND MAMDANI, A. 1999. Some remarks on the semantics of FIPA's agent communication language. *Autonom. Agents Multi-Agent Syst. 2*, 4, 333–356.

PITT, J.V. AND BELLIFEMINE, F. 1999. A protocol-based semantics for FIPA'97 ACL and its implementation in JADE. In *Proceedings of the 6th Congress of the Italian Association for Artificial Intelligence (AI*IA'99)*.

POSLAD, S. J., BUCKLE, P., AND HADINGHAM, R. 2000. The FIPA-OS agent platform: Open source for open standards. In *Proceedings of the 5th International Conference on the Practical Application of Intelligent Agents and Multi-Agents (PAAM00)*, 355–368.

POSLAD, S. AND CHARLTON, P. 2001. Standardizing agent interoperability: The FIPA approach. In M. Luck, V. Marík, O. Stepánková, and R. Trappl, Eds. *Multi-Agent Systems and Applications,* Lecture Notes Computer Science, vol. 2086, Springer Verlag, 98–117.

REED, C. A., NORMAN, T. J., AND JENNINGS, N. R. 2002. Negotiating the semantics of agent communication languages. *Comput. Intel. 18*, 2, 229–252.

SEARLE, J. R. 1969. *Speech Acts*. Cambridge University Press, Cambridge, UK

SINGH, M. 1998. Agent communication languages: Rethinking the principles. *IEEE Comput. 13*, 12, 40–47.

SINGH, M.P. AND HUHN, M.N. 2005. *Service-Oriented Computing: Semantics, Processes, Agents*. John Wiley & Sons, Ltd.

Van der Aalst, W.M.P., Dumas, M., Ter Hofstede, A.H.M., and Wohed, P. 2002. Pattern based analysis of BPML (and WSCI). Tech. rep. FIT-TR-2002-05, Queensland University of Technology, Brisbane.

Willmott, S. Calisti, M., and Rollon, E. 2002. Challenges in large-scale open agent mediated economies. Lecture Notes in Computer Science, vol. 2531, Springer Verlag, 325–340.

Wooldridge, M. 2000. Semantic issues in the verification of agent communication languages. *Autonom. Agents Multi-Agent Syst. 3*, 1, 9–13.

Received August 2006; revised May 2007; accepted August 2007