

# The IEEE FIPA Approach to Integrating Software Agents and Web Services

Dominic Greenwood  
Whitestein Technologies AG  
Pestalozzistrasse 24  
8032 Zurich  
Switzerland  
dgr@whitestein.com

Margaret Lyell  
Intelligent Automation Inc.  
15400 Calhoun Drive  
Rockville, MD 20855  
United States  
mlyell@i-a-i.com

Ashok Mallya  
Veraz Networks  
926 Rock Ave., Suite 20  
San Jose, CA 95131  
United States  
ashok.mallya@gmail.com

Hiroki Suguri  
Communication Technologies  
2-15-28 Omachi, Aoba-ku  
Sendai  
Japan  
suguri@comtec.co.jp

## ABSTRACT

In many settings Web services are now perceived as the first choice technology to provide neatly encapsulated functionality for Web-based computation. To date, many standards have been produced and adoption is accelerating across numerous application domains. This uptake has long been recognized by members of software agent community with several approaches reported that explore various means of extending the utility of Web services with the autonomous control offered by agents. This paper reports on the recent work of several members of this community to consolidate their approaches into a common specification describing how to seamlessly interconnect FIPA compliant agent systems with W3C compliant Web services. This work has been conducted within the context of the IEEE FIPA Agent and Web Service Integration working group and will be shortly published as a new FIPA specification.

## Categories and Subject Descriptors

D.2.1 [Software Engineering]: Requirements/Specifications—*software agents, Web services*

## General Terms

Design, Standardization

## Keywords

Software Agent, Web service, FIPA, W3C

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'07 May 14–18 2007, Honolulu, Hawaii, USA.  
Copyright 2007 IFAAMAS .

## 1. INTRODUCTION

This paper reports on the results of preliminary work conducted by members of the IEEE FIPA Agents and Web Services Integration (AWSI) Working Group. This group was formed shortly after the creation of IEEE FIPA with the remit to specify the minimum infrastructure required to facilitate interaction between FIPA compliant agents with W3C compliant Web services. The purpose of this paper is to both inform the community of progress and to elicit comments and feedback before production of the specification proceeds.

The purpose of specifying an infrastructure for integrating these two technologies is to provide a common means of allowing each to discover and invoke instances of the other. One clear benefit of this is that agents can then be used as the orchestrators of dynamically variant Web service compositions, such as those proposed in [12] and [6], all of which are supported by the statement in the Web Services Architecture specification [3] that, "...software agents are the running programs that drive Web services - both to implement them and to access them as computational resources that act on behalf of a person or organization.". Naturally, once the infrastructure for establishing bi-directional connectivity is in place, compositions can also be an amalgamation of Web services and agent services; for example, an agent may act as a synthesized Web service presenting a simple invocable operation to Web service clients that in fact masks a deeper composition behind the agent 'facade' that requires a degree of autonomous indirection.

### 1.1 Web Service Technology

Web services are now a mainstream technology in common use throughout business environments, both within the enterprise boundary and in B2B and B2C modalities. They essentially encapsulate a simple, coherent set of application or business logic that can be exposed as functional capabilities for invocation by client programs via a Web server. The W3C defines a Web service as, "...a software system designed to support interoperable machine-to-machine interaction over a network.." [3]. In their basic form Web services

have an interface expressed in a machine-processable format, the basic representation being WSDL [4]. Interaction with a Web service typically takes place using SOAP [10] messages, in accordance with the WSDL interface description. SOAP messages consist of an XML serialization conveyed using HTTP. In conjunction with these, UDDI [5] is the directory services schema commonly used to register and discover Web services. During the course of this paper we will refer to this standard Web service technology stack as the *WSA stack*.

Due to the huge potential complexities involved with developing a specification covering all known Web service technologies and specifications, of which there are many, we have elected to provide support for only this well-known WSA Web service stack. This has the dual effect of keeping the specification simple yet functional to a satisfactory degree, while maintaining some degree of immunity from the moving target created by the stream of WS-\* specifications<sup>1</sup>.

## 1.2 FIPA Agent Technology

From 1997 until present FIPA has produced a set of specifications [1] describing various aspects related to the life-cycle support of software agents. Quite deliberately, these specifications do not address the form or behavior of agents themselves, which is left to the application designer, but rather focus on communication among agents and middleware support services such as directory management, with the goal of promoting interoperability among agent systems.

In terms of inter-agent communication FIPA specifies the FIPA-ACL language which is used to express messages and allows the specific message content to be expressed using different content languages such as FIPA-SL or FIPA-KIF. FIPA-ACL is grounded in speech act theory resulting in typed messages which represent the action, or performative, the message performs. FIPA-ACL defines 22 of these performatives, otherwise known as communicative acts, with some of the most commonly used being inform, request, agree, not understood, and refuse. These capture the essence of most forms of basic communication. It is stated in the FIPA standards that to be fully compliant an agent must be able to receive any FIPA-ACL communicative act message and at the very least respond with a not-understood message if the received message cannot be processed. Additionally, based on these communicative acts FIPA defines a set of interaction protocols, each consisting of a sequence of communicative acts to coordinate multi-message actions, such as the contract net for establishing agreements and several types of auctions.

A critical choice to be made by this integration work is which of these performatives should be supported within the context of agent to Web service interactions. As Web services conforming to the WSA stack are essentially invocable remote procedures that may return a response, the minimum subset of performatives would consist of Request and Inform used within the context of the FIPA Request interaction protocol. Section 3 discusses further issues relating to the wider support of FIPA-ACL performatives.

## 1.3 Industrial Application

Generally speaking, the purpose of standardization is to establish a common technical standard that organizations

<sup>1</sup>For a partial list see: [http://en.wikipedia.org/wiki/List\\_of\\_Web\\_service\\_specifications](http://en.wikipedia.org/wiki/List_of_Web_service_specifications)

can use to position a specific approach within the market. In many cases such cooperation is fundamentally necessary to develop a market to the point where competition can take over. Both software agents and Web services now have quite mature standards available; the latter now being commonly exploited by many companies in their day-to-day business. However, some gaps do exist, one of which is a FIPA-specific specification defining how FIPA agents should interact with WSA (and other) Web services.

In this vein, one area of considerable emerging interest to industry at large is the on-demand, dynamic composition of applications from Web (and other types of) services. Building on composition techniques used for such things as creating adaptive business processes (such as WS-BPEL), inter-business collaboration (supply chains), and mass customization, agile application composition offers rapid response to customer and business demands by forming ad-hoc applications from clusters of distributed services, perhaps even owned by multiple parties.

Such composite applications require some form of application logic to control service interaction and produce the new, derived functionality. Two examples of extant Web service standards that move toward this are the Web Services Choreography Description Language (WS-CDL) and the Web Services Composite Application Framework (WS-CAF). However, what the AWSI group is targeting with this work is the enabling of FIPA software agents as independent reasoning engines for service composition. Admittedly, the utility of agents is limited when only considering the standard WSA stack without semantic annotations, but once the basic framework of this specification is in place we expect to extend it to incorporate interfaces for manipulating semantically annotated Web service descriptions.

## 2. PROPOSED SOLUTIONS

The initial step taken with this work was the solicitation of input from interested parties. The result was the submission of several guidances and proposals, which are summarized herein. These contributions serve as the source of the requirements identified in Section 4, subject to the evaluation provided in Section 3.

### 2.1 JADE WSIG

The JADE Web Services Integration Gateway [7], WSIG, is an open source add-on for the FIPA 2000 compliant JADE platform. The WSIG supports registration and discovery of WSA Web services by JADE agents, registration and discovery of JADE Agents and agent services by Web service clients, automatic and bi-directional cross-translation of UDDI directory entries into DF directory entries and vice-versa, invocation of Web services by JADE agents, and invocation of JADE Agent services by Web services.

Shown in Figure 1, the WSIG adds a specialized Gateway Agent to a JADE Main Container which acts as the interface point between the agent system and third party Web service technology including a UDDI repository and Axis Web server.

The DF of the hosting JADE agent platform and a third-party UDDI directory expose the standard actions of register, deregister, modify and search. All register, deregister and modify actions performed on either repository are automatically echoed onto the counterpart repository, i.e., if an agent service is de-registered from the platform DF, the

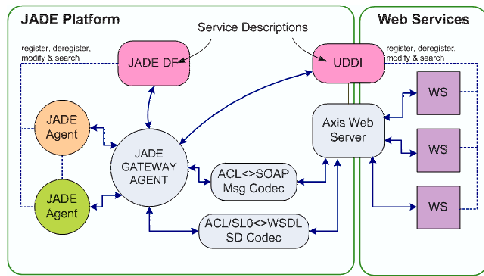


Figure 1: WSIG Architecture

Gateway Agent will ensure that the corresponding tModel is removed from the UDDI repository. This ensures that the two repositories remain consistent.

The WSIG Gateway Agent is responsible for (1) receiving and translating agent service registrations from the DF into corresponding WSDL descriptions and registering these with the UDDI repository as tModels. This also applies to de-registrations and modifications; (2) receiving and translating Web service operation registrations from the UDDI repository into corresponding ACL descriptions and registering these with the DF. This also applies to de-registrations and modifications; (3) receiving and processing Web service invocation requests received from JADE agents. This includes retrieving the appropriate tModel from the UDDI repository, translating the invocation message into SOAP and sending it to the specified Web service. Any response from the Web service will be translated back into ACL and sent to the originating JADE agent; (4) receiving and processing agent service invocation requests received from Web service clients. Processing includes retrieving the appropriate tModel from the UDDI repository, translating the invocation message into ACL and sending it to the specified agent. Any response from the agent will be translated back into SOAP and sent to the originating Web service.

To invoke a Web service, a JADE agent first seeks its service description in the JADE DF (or otherwise, be provided the description), and then sends an invocation request to the Gateway Agent. To invoke a JADE agent service a Web service first seeks its service description in the UDDI (or otherwise, be provided the description), and then sends an invocation request to the Gateway Agent via the Axis Web server.

The JADE WSIG requires no alteration to existing FIPA of WSA-related specifications.

## 2.2 DFKI Agent Messaging Framework

Soto [15] at DFKI, Germany, has proposed a mapping between the addressing and messaging schemes in Web services and those in FIPA compliant multiagent systems. Whereas Web services have been widely accepted and are gaining popularity, FIPA compliant agent systems have had limited success in terms of adoption by the industry because of their narrow applicability. The DFKI paper identifies one of the key reasons for this disparity in adoption as the difference in the level of complexity and sophistication between the two technologies. Web services standardize low-level details of communication among simple services, in contrast with FIPA standards, which standardize interac-

tion protocols among intelligent agents in terms of their beliefs and intentions. This paper briefly describes how a particular standard from the Web services stack –the WS-Addressing standard– can be mapped to FIPA messages to achieve an interoperability between WS-compliant entities and FIPA compliant entities. This is a promising effort in a narrow sub-area of agent and Web service compatibility.

Soto points out how Web services in general, and the WS-Addressing standard in particular, base their addressing scheme on the concept of a physical endpoint address, in contrast with the more generic addressing scheme used among FIPA compliant agents based on Agent IDs (AID). The AID-based scheme facilitates the operation of mobile agents without the agents having to deal with changes in endpoint addresses.

Soto identifies the components that should comprise a FIPA Communication Specification Stack, based on components from both the Web Services and the FIPA messaging standards. In this stack, FIPA speech acts are described in an XML schema and are used as SOAP message types. The Web Services Collaboration Description Language (WSDL) is then used to describe FIPA interaction protocols in terms of SOAP message exchanges.

The DFKI effort aims to concentrate more on determining how agents can specify their reasoning power and the services they offer.

## 2.3 OWL-P

OWL [14] for Protocols and Processes [9], OWL-P, has been recommended as a technology that will contribute to software agents' interaction with Web services and standard Web service technologies. OWL-P provides a framework in which to (1) develop, (2) specify and combine and (3) enact commitment-based protocols. OWL-P components include (1) an OWL ontology to support the specification of protocols elements, (2) a rule-based representation of protocols using [8] for rule expression, (3) an algorithm for combining protocols that is based on an algebra and offers a sound mathematical foundation and (4) an architecture that supports run time activities. These activities include (1) publishing and discovery of protocols (in a UDDI [5] registry), (2) agent incorporation of specific rule sets, and (3) protocol execution through message exchange. A reference implementation of OWL-P makes use of the messaging transport abstraction and uses JMS, together with endpoint bindings that are discovered in the registry. An additional OWL-P feature is provided by its handling of exceptions at the level of commitment violations.

OWL-P may contribute to autonomous software agent-Web service interactions in the cross cutting area of interactions and protocols. OWL-P can offer both a language and a framework for defining and executing the agent-Web service interactions.

Among the specific contributions cited are: (1) A representation of protocols that may be advertised in the UDDI registry, (2) an algorithm that allows protocols to be combined, thus assisting with the development and execution of complex Web services, and (3) a unified framework in which an agents preferences among variants of the Web service interaction protocols can be specified.

## 2.4 WS2JADE

WS2JADE [11] is a toolkit which allows JADE agents to

discover Web services and use them. This is developed by Swinburne University of Technology, Australia. WS2JADE addresses the following areas of agent and Web services interoperability:

- communication languages
- ontologies and service descriptions
- interaction protocols and composition languages

A symmetric integration of Web services and FIPA compliant agent platforms has been proposed by the Agentcities Web Services Working Group [2] as a high-level architectural recommendation. There have been a few implementations following this recommendation. WSDL2JADE [17], available from Sztaki's website<sup>2</sup>, can generate agent ontologies and agent code from a WSDL file. The distinctive feature of WSDL2JADE, as compared to other software with similar functionality, is that it allows deployment of Web services as JADE agents at runtime. In this manner, it provides a greater level of automation in Web service discovery and usage.

WSIG (Web Services Integration Gateway) [7], implemented by Whitestein Technologies, supports bi-directional integration of Web services and JADE (Java Agent Development framework) agents.

In WS2JADE, Web services are visible to FIPA compliant agents through proxy agents which reside in WS2JADE system. Web services are seen by FIPA compliant agents as agent services offered by the proxy agents. The mapping from Web services to proxy agents are many to many. Since in JADE, an agent is often single-threaded, offering the same Web services on different proxy agents allows concurrent accesses to a Web service. Offering more than one Web service on a proxy agent allows related Web services to be grouped together. In WS2JADE, Web services discovery is done through UDDI proxy agents. These agents support special discovery services which can be configured to proxy to any UDDI version 2 servers, including Microsoft and IBM UDDI inquiry servers.

Although WS2JADE has demonstrated its effectiveness in working with different popular Web services, there are areas—for instance, one-way integration—where it can be improved upon. At the moment there is a reluctance toward any agent to Web services implementation as there is still a lack of substantial theoretical work on the topic of agent to Web service integration, especially in the areas of translating agents' stateful communication model into Web services' stateless communication model and building asynchronous interaction framework for Web services. This is a subject of on-going research with current and future work also involving improvements to the semantic processing capability of WS2JADE's ontology management component.

## 2.5 AgentWeb Gateway

The AgentWeb Gateway [13] is an initiative toward achieving dynamic and seamless integration of FIPA compliant software agents and WSA compliant Web services. The Gateway is a middleware layer between the multi agent system and Web services facilitating integration without requiring alteration to existing specifications. The system provides

<sup>2</sup>sas.ilab.sztaki.hu:8080/wsdl2agent/index.html

Service Discovery transformation, Service Description transformation and Communication Protocol transformation.

The Gateway is shown in Figure 2.

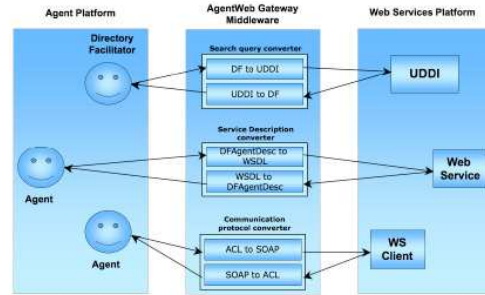


Figure 2: AgentWeb Gateway Architecture

The Service Discovery converter enables FIPA agents and Web services to search for one another. Software agents can discover Web services via UDDI registries and conversely, Web service clients can perform searches for agents and agent services from agent registries such as the FIPA DF. Specifically, the 'UDDI search query to DF search query converter' enables the discovery of Agent platforms by Web service clients via the DF. The 'DF search query to UDDI search query converter' enables the discovery of Web service platforms by software agents via UDDI.

The Service Description converter enables service publishing among Software Agents and Web services. Software Agents can publish services in Web Services registries such as UDDI and Web Services can be published in Multi Agent Systems service registries such as the FIPA DF. Specifically, the 'WSDL to DF-Agent-Description converter' enables Web services to publish their services in the DF of Agent Platform, and the 'DF-Agent-Description to WSDL converter' enables Software Agents to publish their services in the UDDI of a Web services platform.

The Communication Protocol converter component enables service invocation among software agents and Web services. Software agents can invoke Web services and Web service clients can invoke software agents in Multi Agent Systems. Specifically, the 'SOAP to ACL converter' enables Web service clients to invoke Software Agents, and the 'ACL to SOAP converter' enables Software Agents to invoke Web services.

This approach is supported by a mature codebase and has an application layer designed for use with the Semantic Grid Framework.

## 2.6 JBees

JBees [16] is a workflow management system developed at the University of Otago. The central concept of JBees is the use of agents to manage resources involved in the enactment of a distributed workflow to enable adaptability and distributability.

JBees uses Colored Petri Nets (CPNs) to model workflows and utilizes agents to distribute tasks in the workflow. Adaptability is achieved via dynamic process creation, i.e., by instantiating tasks in the workflow as needed, so that tasks can be instantiated based on the latest model, thus incorporating process model changes at runtime.

JBees uses CPNs for modeling workflows, since CPNs have formal semantics which can define executions unambiguously. JFern, a CPN simulator is used in the JBees system to design and execute the CPN models. The JBees framework consists of several types of agents, each with specific tasks. There are management agents which provide an interface to humans for managing processes, storage agents that manage the storage of persistent data such as the process models themselves, process agents that execute one particular instance of a process, resource agents, each of which is the interface for some resource that will be used during a process, resource broker agents that manage resources and resolve contention for resources, monitor agents that gather system data, and control agents that provide the feedback mechanism which gathers data that a human operator can use to re-engineer the process if needed. It is this division of tasks among agents that enables adaptability. The distribution facilitates replication of data and functionality so that the system does not have a central point of failure.

### 3. EVALUATION

Taken as a whole, the proposals cover a range of pertinent issues and supporting technologies that are primarily focused on the integration of FIPA compliant software agents and the WSA stack.

The offered proposals respectively present, either implicitly or explicitly, a viewpoint regarding the interaction and interoperability of Web services and software agents. The DFKI response explicitly raises the issue that suggested mappings may "require a comparison of semantics involved (in both scenarios), the effects any divergence may have and a proposal of how to cope with any divergence". The proposal from Whitestein Technologies advances the need for an Interaction Model that "details the relationship between FIPA compliant agents (and agent systems) and Web services" to reside at the core of the standardization effort.

With the caveat that there are likely semantic mismatches between various aspects of the Web service and software agent paradigms, the proposals focus on specific areas that should be addressed for standardization. Proposal discussions are typically detailed and comprehensive in the technology areas that the proposers are either offering or with which they have experience.

There is a strong consensus among the participants that SOAP messages and FIPA ACL message translations should be standardized. The DFKI proposal relates specific experience in translating FIPA Envelope Message to WSAddressing<sup>3</sup> Envelopes. This is stated more specifically from the University of Otago proposal's call for message translations and "standardization of support for different transport protocols (HTTP, SMTP, JMS etc.)." A Gateway technology is proposed by multiple groups; this has been detailed in the summary section.

A second area of consensus is in regards to the standardization of Interaction protocols. The scope of proposed standardization and the supporting technologies indicate that this area is quite active. The standardization of Interaction

<sup>3</sup>W3C Web Services Addressing Working Group. Web Services Addressing Core Recommendation and the Recommendation on the SOAP Binding dated 2006-05-09. Website information accessed on 11-21-2006 [www.w3.org/2002/ws/addr/](http://www.w3.org/2002/ws/addr/)

protocols include, (1) standardization of message exchange patterns, and (2) standardization of a protocol library from which specific protocols may be composed. This is supported by the technology OWL-P, which was summarized in a previous section. Composition is supported mathematically by a protocol (process) algebra formulation. The scope of OWL-P supports (social) commitments, which is beyond the semantic scope of WSA Web services.

Interaction protocol standardization was addressed from the viewpoint of which types of interactions should be standardized. These include (1) the interactions for Web Service Composition, both Orchestration and Choreography, (2) interactions to support advertisement and Web-service discovery. Within the first area, the proposal from the group at the University of Otago has proposed that Colored Petri Nets mathematics be used as for the formulation. They are utilizing this in their workflow technology, JBees. The proposals that focused on standardization of Interaction protocols supported by technologies clearly stated the underlying mathematics. This is necessary to support protocol verification and error handling.

While not the focal point of the current AWSI standardization activity, multiple responders noted a connection with the Semantic Web technologies. Of course, OWL-P technology was developed as an extension to OWL (and employs SWRL). However, some attention was given to development of standards for semantically richer advertisements. OWL was proposed as the technology to use in order to express the richer service details. The advertisements would be accessible (perhaps by indirection) via a UDDI registry.

After an assessment of the proposals taken as a group, the following remarks are pertinent.

1. The first priority as regards standardization is to address the FIPA ACL to SOAP (and vice versa) translations,
2. Standardization in use of various transport mechanisms would be of great immediate benefit in supporting the interaction of agent systems and Web services,
3. Interaction protocols for the purpose of agent interaction with Web services (and vice versa) are the second primary area that should be addressed by a standardization effort. This area is more complex than the area of message translation.
4. Interaction protocol standardization must be supported by an underlying mathematical model.
5. Complexities are introduced into the standardization effort due to the different semantic levels of the different paradigms. These may be reflected in translations and error handling.
6. Agent technology capabilities may prove to be a significant implementation approach to address service discovery, choreography and orchestration within the world of Web service technologies.

### 4. DERIVED REQUIREMENTS

The central issue addresses by this work is an assessment of architectural additions and modifications to the FIPA standards that are required to enable seamless interoperation between FIPA compliant agents and WSA technologies.

Another important requirement of this work is to avoid modifications to FIPA standards. This section lists functional and non-functional requirements for FIPA-WSA interoperability, identifies requirements among these that are fulfilled by existing FIPA standards, and suggests new mechanisms for meeting requirements that are not met by FIPA standards.

#### 4.1 Non-Functional Requirements

The chief non-functional requirement of the AWSI, as was proposed in the initial submission to FIPA, is that there be no impact of the AWSI standards development effort on existing FIPA specifications. This requirement clearly identifies the task of the AWSI working group, differentiating it from the core FIPA specification development body.

All of the proposals submitted to the AWSI meet this requirement. Even proposals such as OWL-P, which have been developed independent of FIPA specifications, do not require any modification of existing FIPA specifications.

#### 4.2 Functional Requirements

During the initial phase of this work, a set of requirements was drawn which characterize what should be expected from a specification detailing the interoperation of FIPA compliant agents. These mandatory requirements are listed here.

*Discovery.* FIPA compliant agents and Web Services need to be able to find the location, communication method, and services offered by each other for them to interoperate. For FIPA compliant agents to discover Web Services (and vice versa), the following requirements should be fulfilled

1. Translation of WSDL Web Service descriptions into FIPA Agent Service descriptions should be possible.
2. Translation of FIPA Agent Service descriptions to WSDL Web Service descriptions should be possible.
3. FIPA compliant agents should be able to discover Web Services published via UDDI.
4. Web Services should be able to discover agent service descriptions for those FIPA compliant agents that are designed to interoperate with Web Services.

The FIPA Agent Discovery Service Specification (PC00095A) describes the FIPA framework for the registration, advertisement, and discovery of FIPA compliant agents. At its core, the FIPA specification describes a Directory Facilitator (DF) per agent platform. The DF is where FIPA compliant agents register themselves (along with a description of services they offer). There is also an Agent Discovery Service (ADS) element per agent platform. The ADS facilitates discovery of agents across platforms.

The WS2JADE and AgentWeb Gateway technologies both address the service and agent discovery issue. Both are examples of how Web Services and FIPA compliant agents can discover each other without necessitating a change in the FIPA specifications.

*Messaging and Invocation.* For FIPA compliant agents to be able to send messages to and receive messages from Web services, the addressing scheme and message formats of both schemes should be mapped to each other.

1. Translation of FIPA ACL messages into SOAP messages should be possible.
2. Translation of SOAP messages into FIPA ACL messages should be possible.
3. All invocation requests received and sent by FIPA compliant agents from and to Web Services should be encoded as FIPA-Request messages (within the agent system).
4. All responses received and sent by FIPA compliant agents from and to Web Services should be encoded as FIPA-Inform messages (within the agent system).

The FIPA specifications for Agent Communication describe the structure of ACL messages, the communicative acts and their semantics, and the content languages allowed. While content languages are not in the scope of our work, a mapping from communicative acts to SOAP messages is. We propose that the minimum requirement for interoperation between FIPA compliant agents and Web Services is that service invocation requests are mapped to the FIPA-Request act and the responses to this request are mapped to the FIPA-Inform act.

*Interaction.* Whereas messaging and addressing requirements enable communication between FIPA compliant agents and Web Services, a well developed *Interaction model* specifying conversations or long-lived engagements between agents and services is required, as described in Section 3.

FIPA has standardized a library of agent interaction protocols such as the Contract Net Protocol and the Dutch Auction Protocol. Each protocol is described in a UML notation, with ACL messages exchanged between the participants of the protocol.

The development of an interaction model between Web Services and FIPA compliant agents would be based on how ACL messages are mapped to Web Services, which have rudimentary support for performatives in messages at best.

*Optional Requirements.* The AWSI standards will have the following optional requirements:

- The specification should demonstrate how business process languages (such as WS4BPEL) can be used by FIPA compliant agents to construct and execute business processes.
- The specifications should describe extensions to the interaction model specifying the relationships between FIPA compliant agents and Semantic Web Services.

The AWSI working group aims to develop specifications for an *interface* between FIPA compliant agents and Web Services, not modifications to FIPA standards. This position is justified by the intent with which the two technologies were developed. Whereas the FIPA specifications describe how agents and agent systems interact with each other, the Web Services stack standards were developed with a much more generic goal of enabling interoperation and information exchange among software components across the Web. Changing either technology by modifying existing parts would erode their best aspects.

## 5. CONCLUSIONS

As illustrated by the popular 'mash-up' Web sites of the Web 2.0 initiative, it is now increasingly common for developers to construct composite applications from simpler, readily available building blocks. This, and the growing presence of service-oriented architectures in many business and industrial systems are direct indicators that the long forecast move toward network and service oriented computing is well underway. It is against this backdrop that the IEEE FIPA AWSI working group was formed with the mandate to produce a FIPA specification dealing with integration of FIPA compliant agents and Web service technologies.

An important issue underlying this work is whether there is sufficient value in introducing new FIPA specifications at a time when FIPA has yet to build a broad adoption base in industry. The view taken by the participants of this working group is that the presence of new, key specifications such as the forthcoming AWSI Specification are of significance in light of this trend toward service and application composition. It is hoped that by specifying straightforward mechanisms enabling agent and Web service integration we will assist with promoting FIPA and agent systems as technologies that can offer a competitive advantage to businesses. If it is true that application composition is a desirable path forward, then this specification work may just help to position agent technology within the crosshairs of industrial adopters.

The first critical task in producing the AWSI specification is gathering requirements, which has been dealt with in this paper. It is our hope that addressing this paper to the audience of the AAMAS Industrial Track will provoke a dialogue to help clarify these derived requirements before proceeding to the next phase of completing the AWSI Specification.

## 6. ACKNOWLEDGMENTS

The authors would like to thank all those who contributed material to this paper and to all members of the IEEE FIPA AWSI Working Group.

## 7. REFERENCES

- [1] The FIPA Specifications. [www.fipa.org/repository/index.html/](http://www.fipa.org/repository/index.html/).
- [2] Integrating web services into agentcities, November 2003. Agentcities Web Services Working Group.
- [3] D. Booth, H. Haas, F. McCabe, E. newcomer, M. Champion, C. Ferris, and D. Orchard. Web Service Architecture: W3C Working Group Note, February 2004.
- [4] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana. Web Services Description Language (WSDL) 1.1, March 2001.
- [5] L. Clement, A. Hatley, C. von Riegen, and T. Rogers. UDDI Specification Version 3.0.2, October 2004. [www.uddi.org/pubs/uddi\\_v3.htm](http://www.uddi.org/pubs/uddi_v3.htm).
- [6] D. Greenwood, P. Buhler, and A. Reitbauer. Web Service Discovery and Composition using the Web Service Integration Gateway. In *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, pages 789–790, Washington, DC., 2005.
- [7] D. Greenwood and M. Calisti. Engineering Web Service-Agent Integration. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 1918–1925, The Hague, Netherlands, 2004.
- [8] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosf, and M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, May 2004.
- [9] A. U. Mallya and M. P. Singh. Specifying and resolving preferences among agent interaction patterns. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 1361–1368, New York, 2006. ACM Press.
- [10] N. Mitra. Simple Object Access Protocol (SOAP) 1.2 Part 0: Primer, June 2003.
- [11] X. T. Nguyen. Demonstration of WS2JADE. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 135–136, Utrecht, The Netherlands, July 2005.
- [12] D. Richards, S. van Splunter, F. M. T. Brazier, and M. Sabou. Composing Web Services using an Agent Factory. In *Proceedings of the 1st International Workshop on Web Services and Agent Based Engineering*, Sydney, July 2003.
- [13] M. O. Shafiq, A. Ali, H. F. Ahmad, and H. Suguri. AgentWeb Gateway - a Middleware for Dynamic Integration of Multi Agent System and Web Services Framework. In *Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise*, pages 267–270, Washington, DC, 2005. IEEE Computer Society.
- [14] M. K. Smith, D. McGuinness, R. Volz, and C. Welty. Web Ontology Language (OWL) Guide version 1.0, 2002.
- [15] E. L. Soto. Fipa agent messaging grounded on web services. In *Proceedings of the 3rd International Conference on Grid Service Engineering and Management*, September 2006.
- [16] B. Tony, R. Savarimuthu, M. Purvis, M. Purvis, and S. Cranefield. Integrating web services with agent based workflow management system (wfms). In *Proc. of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, Compiègne, France, September 2005.
- [17] L. Z. Varga and Á. Hajnal. Engineering Web Service Invocations from Agent Systems. In *Proceedings of the 3rd International Central and Eastern European Conference on Multi-Agent Systems*, pages 626–635, Prague, Czech Republic, June 2003.