FIPA Agents Messaging grounded on Web Services

Esteban León Soto
German Research Center for Artificial Intelligence (DFKI)
Multiagents Systems Group
Esteban.Leon@dfki.de

Abstract: Messaging mechanisms for Web Services are tending towards an agent paradigm which is based on very flexible addressing and semantics of messages on top of communication protocols that aim at macro objectives shared by a group of participants. Recent improvements in Web Services make it possible to integrate agents and Web Services in a common messaging framework that includes both architectures integrally. The present work describes the integration based on the properties of both scenarios. The proposed integrated architecture is discussed and related work reviewed to conclude with a numerous set of research areas that are enabled by this elemental step.

1 Introduction

Web Services are a very well accepted tool for interoperability and it has been vastly used for integration of heterogeneous distributed systems. They have been used for *Remote Process call* (RPC) primarily, a mature technique nowadays. In the recent time Web Services have started moving towards agent–like models. This is partly because agents are one of the preferred ways for implementing distributed systems with a higher level of complexity and where communication and autonomy play an important role [Jen01]. This has been increasingly the objective for Web Services as well. There is a big interest to create Service Oriented Architectures (SOAs) with services that are delegated a goal and achieve it autonomously and that allow for wider integration of different systems[Bar03].

The integration of Web Services and agent technologies has been very difficult because of the Web Services having this RPC tendency. Now, new specifications are available that allow for a more agent-like communication, for instance Web Service Addressing [W3C06a] (WS-Addressing) and Web Services Choreography Description Language [W3C05] (WSCDL) among others. In the present work, the issue of integrating Web Services and agent technologies will be approached again, this time taking into account the significant improvements in Web Services and integrating both technologies at the messaging level.

For this integration a specification for an agents system must be used. The Foundation for Intelligent Physical agents (FIPA) [FIP02a] supplies one of the preferred and more extensive agent specifications available. For the purpose of this work the importance of using FIPA specifications relies on the wide acceptance and usage they have in the multiagents systems community. The majority of investigations about agent communication are implemented using these specifications, some of them beeing very relevant to the Service Oriented community, specially those about dynamic planning of services (composition of services), conversations patterns and protocols or autonomous negotiation. Appart of that, FIPA Interaction Protocols Specification [FIP02e] is the only formal definition of generic peer-to-peer agent protocols [She03].

An integration of the messaging of a FIPA platform with Web Services will be proposed by grounding the FIPA abstract specifications using Web Services standards. This is done focusing on messaging only, other relevant issues like description and discovery are left outside the scope of this paper: they are treated as a special type of contents to be transported using the proposed messaging. A quick overview of important research work concerning description of agents in a Web Service scenario can be found in the related work mentioned in the present article.

As a first step, the common ground of Web Services and FIPA specifications is laid out by characterizing them in sections 2 and 3 respectively and comparing them in section 4. Later the integrated messaging architecture is introduced in section 5 and discussed in section 6. Finally a summarizing conclusion is provided in the last section.

2 FIPA Specifications

The Foundation for Intelligent Physical Agents (FIPA) [FIP02a] provides a stack of specifications for the communication between agents. It specifies in different layers the organization of the communication. Figure 1 shows how communication specifications in FIPA are built one on top of another.

Interaction protocols [FIP02e]
Speech Acts [FIP02d]
FIPA ACL Message
Transport Message
Message Contents Representation [FIP02b]
Message Envelope Representation

Figure 1: FIPA Communication Specifications Stack

Looking at it from the bottom up, first we can see the representation of the message envelope, which contains transport specific information required in every single message, independently of their context. Above it, a way of representing the contents is also specified. These concrete representations, envelope and contents, can be based on any appropriate representation technique, like bit efficient, Strings following a specific syntax, XML, etc. Message contents and envelope form together a Transport message which is the realization of a FIPA ACL Message [FIP02c]. Then a set of ACL message types, called speech acts, defines the different kinds of messages that agents will use, each of these messages having a clear semantic definition in correspondence to speech act theory [Sad91]. There are certain well known situations in which a specific sequence of speech acts are expected. These sequences are called protocols and they are a pre-agreed way of how these message exchanges should be preformed to serve a specific purpose.

2.1 FIPA Message Envelope

The envelope is an information structure that contains some fields for the addressing of the message, message correlation, semantic, representation and context information. The structure is shown in Figure 2.

Sender (AID)	Receiver (AID)
Reply-To (AID)	Reply-By
Conversation-ID	Reply-With
In-Reply-To	
Communicative-act	Protocol
Encoding	Language
Ontology	
User-Defined	
Content	

AID:
name
address
locator

Figure 2: FIPA Message Envelope[FIP02b] and Agent ID (AID)

2.2 Speech act

FIPA[FIP02d] has specified the structure and semantics of a basic set of speech acts. Each speech act is an information structure associated with a specific semantic, like a query, an assertion, a request, etc. They are defined in terms of preconditions and consequences that they have.

These speech acts are defined in terms of the preconditions to hold for the action to be performed and the changes that the action produces. For instance, the `inform` speech act, the action of informing that a proposition is true, is defined by the preconditions that the sender must hold that a proposition p is true, is uncertain whether the receiver knows the true value of p or not and intends that the receiver comes to believe that the proposition p is true. The complete set of FIPA speech acts defined can be found in [FIP02d].

2.3 Interaction Protocols

In a multi agents system messages are expected to be exchanged between agents within a context and as part of an interaction. For this reason FIPA provides a set of interaction protocols which are specifications of message exchange sequences that serve a certain purpose.

Interaction protocols are a mechanism used to describe the different options of message exchanges that appear in the sequence of the interaction. The options vary depending on the situations the interaction is involved in. Within a conversation, the path that the conversation takes can be different from one situation to another, but still belonging to the same type of interaction.

FIPA provides a set of protocol descriptions [FIP02e] that describe from simple conversations, like an agent requesting another agent to perform an action, to complex interactions between several participants like auctions or brokering.

Interaction protocols are described using sequence diagrams [OvDPB00] which are graphical representations of the protocol. The semantics of these diagrams is specified in Agent-UML [BMO00], an extension of UML with enhancements for agent specific issues like multithreaded lifelines, roles and message semantics.

3 Web Services Standards overview

The world wide web consortium [W3C] is the organization in charge of the specification of Web Services. This section provides a quick overview of the standards relevant to the subject of the present contribution and describes the properties used in the discussions further in this document.

3.1 Description

Web Services standards provide several description mechanisms. One of them is used to describe complex interactions, it is based on the description of services, both of them will be described briefly bellow.

3.1.1 Service description

Web Services description language (WSDL) [W3C02] is a specification for the description of services in terms of the messages that they interchange and their structure. These descriptions are composed of a specification of the types to be used inside messages, the messages that contain them, which messages are used in which operations, and port types which are collections of operations that are related to a specific task. Apart of that, WSDL provides the mechanism for specifying a binding for these port types that tells how are they going to be grounded, for instance using SOAP [W3C03]. Finally, there is the possibility for creating collections of these bound port types as a service and associate a location for the service to be reached.

3.1.2 Complex Interactions description

In the last years a standard for describing complex interactions between Web Services has been developed. Web Services Choreography Description Language (WS-CDL) [W3C05] has reached a stable stage and entered its last phases. It permits the description of processes that involve different participants, in terms of the messages that are exchanged. It is based on WSDL descriptions to represent the endpoints and XML types to represent the messages. It provides also mechanisms for describing the choreographies independently of the actual services that will participate replacing them with the concept of role. Similarly the concept of token is introduced to make it independent of the actual types to be used to define messages and information. It provides resources convenient for the reliability of the collaborations like process blocks for exception handling, finalizing processes and synchronization between participants.

3.2 Communication

Web Services also provide standards for the messaging between systems. The message representation mostly used in Web Services is SOAP. In the last moths, a new standard was provided for a more precise definition of messaging called WS-Addressing, both are depicted below.

3.2.1 SOAP

Simple Object Access Protocol (SOAP) is a specification for messaging using XML. It defines the structure of a message organizing it in two main parts, the headers block and the body. The first one is used for annotating the message with information mainly relevant for the messaging and process of the message and the body is used for containing the information that is to be provided to the endpoint of the message.

3.2.2 Addressing

Web Service Addressing (WS-Addressing) [W3C06a] is a specification for the concrete headers used for the addressing of a message. SOAP provides messaging using headers and WS-Addressing specifies what headers are to be used and their semantics. These set of headers are a set of parameters used for supporting messaging and compose what is known as an Endpoint Reference(EPR). An EPR can be extended with application specific parameters. The fields in an Endpoint Reference are shown in figure 4 in the right column with some added parameters which appear underlined.

4 Comparison of both specifications

Aspect	**Web Services**	**FIPA**
Syntax and Structural definition (Grounding)	Detailed	Detailed
Adoption	Vast	Little
Semantic definition	Little, open room for application semantics	Well grounded, open room for application semantics
Communication stability support	Explicit	Simple and not always explicit
Communication complexity	RPC only moving towards complex interactions	Targeted at complex dialogs
Complex Conversations support	Immature	Mature
Reasoning capability of the participants	Reduced	Complex
Identification	Address	Instance ID
Statefulness	Stateless	Stateful

Table 1: Web Services and FIPA Specifications comparison

The properties described in the previous sections allow a comparison of both specifications that will serve later as a base for the proposed integral architecture. Table 1 summarizes them.

The first aspect to compare is the low level groundings, present in both. The goal of Web Services is to have a specification that allows communication between systems using XML. Web Services are normally grounded using SOAP for the XML message structure and XML for the contents. It has well established functional description languages that allow detailed descriptions of services and collaborations between them. FIPA provides a set of specifications that go from the basic grounding to the abstract layers of communication like semantics and complex interactions. It has grounding specifications in different formats including one for XML, for the envelope as well as for the contents. The architecture allows for addition of new groundings. FIPA specifications go further up in the abstraction level and provide specifications for semantic and structure of high level concepts which are left completely open by Web Services which focus on bare messaging. Web Services enjoy a wide adoption among a vast amount of application areas, since their interest is interoperability for any application area. FIPA specifications on the other hand, are focused the interoperability and specification of agents and therefore have moderate to good adoption inside the multiagents community but very low adoption outside of it. The wide adoption of Web Services is the most attractive reason why to produce an integrating architecture.

FIPA specifications have given lower importance to some key concepts for low level communication, like acknowledgement of messages or detailed exception handling description. In FIPA the sending of a message and not receiving an error is intrinsically perceived as a successful transfer of the message, which is not always the case in conventional communication systems. In Web Service specifications, exceptions for instance, are part of the basic vocabulary for the description of choreographies [W3C05]. Even though FIPA provides speech-acts and protocols for handling exceptional situations, these are left relatively lax and not targeted to be as explicit and detailed as in Web Services.

Communication complexity has been inherently very high in multiagents systems from the very beginning. Communication between autonomous entities is after all one of the main subjects studied in this area. Web Services on the other hand have been traditionally simple, as far as conversation patterns is concerned. The majority of usages are simple request-response patterns. Even though Web Services are suitable for more complex communication patterns and recently even provide a description language for such conversations, the deployment of complex communication systems is still not dominant in this area. Due to the recency of Web Services and its complex collaboration specifications, their usage for complex interaction processes is still immature in comparison to multiagents systems. This is also related to the complexity level of the participants in the conversations: FIPA specifications target autonomous agents expected to have enough reasoning power to manage complex conversations in opossite to Web Services, where the objective is to provide interoperability between heterogeneous systems which not necessarily support complex reasoning capabilities concerning communication.

Web Services, in current practice, are focused on simple RPC-like function performing interactions only. Participants are traditionally simple (to the outside) and messages can carry complex information but their interaction related semantic has been kept simple and its consequences within the group or participants also. FIPA is focused on complex interactions with messages having simple to complex contents and semantics and participants that can be simple but also complex reasoning agents.

Finally, another aspect relevant to compare is the statefulness of the agents participating in conversations. Agents are in essence stateful and FIPA specifications treat them like that. Agent identification is part of the lower communication layers. Web Services focused on addresses and left out the actual agent providing the service. Web Services, are focused on the messaging only and treat the services as stateless instances. This is reflected clearly, for instance, in the way Web Services and FIPA specs identify participants, namely using Endpoint References (addresses) [W3C06b] and agent instance names [FIP04] respectively.

5 Integrated messaging architecture

Taking into account both specifications, their properties, features and goals, an integration will be proposed which consists in a different grounding for FIPA specifications. A messaging stack is proposed for allowing FIPA messaging over Web Services that connects both specifications at the Message Envelope level. The details are shown next and discussed in section 6.

5.1 FIPA-WS Messaging Stack

The proposed messaging stack is shown in figure 3. At the very bottom is the basic transportation layer. This layer is composed of the different network transportation protocols already used in both architectures. On top of that, the XML-based Web Services messaging is implemented using SOAP [W3C03] which is a service oriented messaging specification very similar to messaging in multiagents systems. WS-Addressing [W3C06c] is used as the Web Service standard envelope specification. Based on it and on the FIPA Message Envelope specification an envelope structure is proposed as an addition both specifications (see section 5.2).

The set of FIPA speech acts is defined using XML-Schema. This schema is used as a types specification for the message types to be transported using SOAP. The speech acts schema introduces the expected structure for the message content that corresponds to the semantic of the performative and therefore are to be used in the (WSDL) descriptions of the services to define the message types. A suitable XML schema of the complete speech act library is provided at [Est06].

Using these speech act schema types and WS-CDL the different FIPA protocols can be defined resulting in a set of machine readable protocol descriptions capable of representing the semantic contents of FIPA messages. These protocol descriptions can be used recombined to describe more complex interaction scenarios or as basic template for application specific interaction protocols.

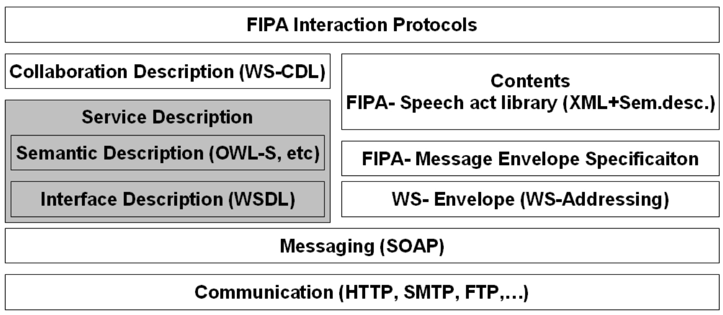

Figure 3: FIPA Communication Specifications Stack

5.2 FIPA Message Envelope using WS-Addressing

The most important detail about the FIPA-WS Stack is the representation of a FIPA Message Envelope using WS-Addressing and SOAP. Figure 4 shows an example of one possible mapping. It has two columns listing field names on each side. On the left side are the FIPA Envelope Fields and on the right side the fields of a WS-Addressing envelope. WS-Addressing allows to extend the properties set with *reference parameters*. The right side makes use of that and adds some fields to hold information present in the FIPA Message Envelope Specification. In the middle there are associating arrows that represent a mapping between the fields in each specification.

WS-Addressing, as stated in its name, is targeted to support messaging based on the address of the Endpoint. It does not take into account for any purpose, the identity of the agent behind the endpoint. It is therefore not possible to ensure using WS-Addressing only, that the same agent instance will be targeted at all times using the

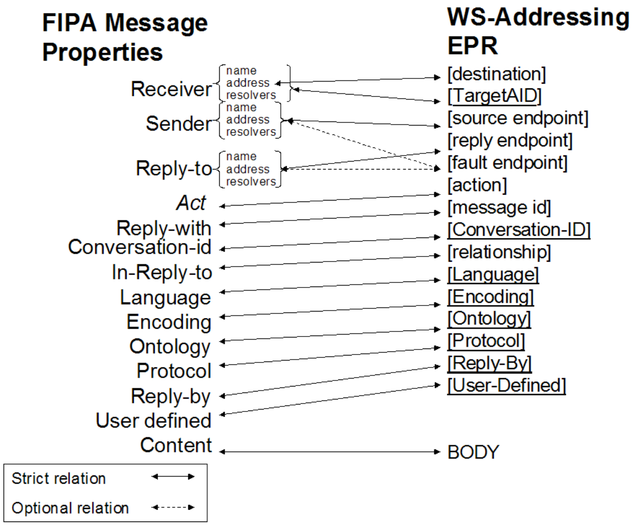

Figure 4: A FIPA and WS-Addressing Mapping

same Endpoint-Reference. On the other hand, FIPA messaging mandates to specify the ID of the targeted agent in the envelope. Therefore the context of the multiagents application and the conditions in which the communications will be performed will be relevant for the definition of such a mapping for an agent platform. This issue will be discussed further in section 6

5.3 Architectural integration

Finally, having a successful stack implementation and a suitable WS-Addressing mapping, it is possible to provide an agent platform that communicates using Web Services. It allows to address agents on remote platforms and perform complex dialogs with them in the same way as done currently using the existing FIPA groundings.

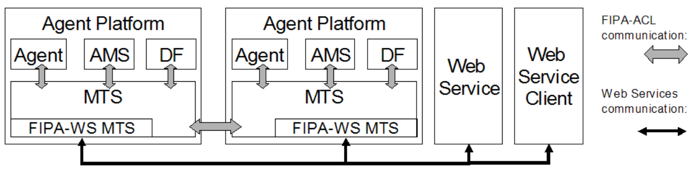

Figure 5: FIPA and Web Services architectural integration

6 Discussion

Web services can be a good grounding for FIPA specs, they provide a widely adopted communication infrastructure and are focused on the representation of information and its transport leaving free room for application semantics. FIPA is specified in a way that the semantics can be grounded using different implementations. Apart of that, messaging works very similarly in both cases.

In the last years Web Services started moving from simple RPC to complex conversations between two or more parties. This increases complexity, because of the increase in the semantics of messages and because concepts like social commitments, turn taking, etc. appear. This RPC nature of Web Services influenced also most of the integration approaches done for FIPA architectures as mentioned in section 6.1. But the latest improvements in Web Services messaging and conversations (WS-CDL and WS-Addressing) allow a smoother WS-compliance for a FIPA grounding which goes beyond request-response situations. These improvements in Web Services signal a movement towards a technology more similar to multiagents systems which implies that problems in Web Services will become more similar to the ones studied for agents. Therefore, this integrated communication architecture provides a platform to port concepts from the agent community to the Web Services like dialog games, social commitments, security, negotiation, etc. which provide solutions for these oncoming difficulties. The multiagents community has studied these issues in detail and gained important experience, one of the most important ones being Agentcities [Age02a], a project in which a worldwide network of agents was created.

Simplicity has been an advantage for the acceptance of Web Services, this acceptance is very helpful for mutliagent systems technologies, but at the same time these technologies will help the further adoption of Web Services in more complex scenarios.

There are some technical issues that must be taken into account for creating an integrated architecture, depending on the system that will be implemented some variations can be done when solving them. The first case is how to handle the statefulness of agents, which are identified uniquely throughout their lifetime. In the mapping exposed in section 5.2 this was solved by adding extra parameters to the WS-Addressing EPR. The difference in the mechanism for message correlation was also solved this way. For example, another way for solving the identity mechanism of agents is to assign them a unique address that will belong to them during their complete existence. In cases where agents do not move in a way that the address would not apply anymore, this would be a simpler solution. It could be enhanced also with a mechanism for message forwarding so that the agent can still be found under the same address even after moving away.

Another issue is the strong variation on reasoning power manifested by participants in such an integrated architecture. Very simple request-response supporting participants would have to interact with complex agents that cope with a wide variety of message types and that are not deterministic in their behaviour. Therefore it will be very important to include in the definitions of the participants the interaction complexity that they support. The messaging mechanism has some requirements that will not always be fulfilled or supported by very simple participants. For instance, messages should include always the type of protocol they are part of, but very simple third party services will not mention such information. Therefore the interpretation of messages should be done, taking into account some assumptions, like in this case, that a not mentioned protocol implies a simple request-response, for example.

The present solution provides a speech act library in XML, where the structure of the different speech acts is defined. Still a semantic description of them is still missing. FIPA already provides definitions for them based on the beliefs of the sender and receiver of the message. [CV02] Mentions that it would be more appropriate to define them in terms of commitments that result from performing the speech act. This would go in more synergy to the way protocols are described, from the perspective of the collectivity and not from that of an individual.

6.1 Related work

Web Services have been one of the prefered technologies for integrating complex network of heterogeneous peers. In [Pas04] Web Services where used to enable platform independent communication for in a grid system.

The different integration of agents and Web Services strategies share in principle the same idea of a wrapper or adapter module that bridges both technologies. This is actually the recommended way to integrate heterogeneous systems to a Web Services architecture [Bar03]. The lack in Web Services, in the past, of support for complex conversations was also a reason to approach the integration with agents using wrappers[Jen01]. The experience gained in the Agentcities project [Age02a] proposes to enable interoperability using a Gateway [Age02b] for the interaction of services and agents. Several solutions have adopted the Gateway approach [GC04], [CKM+03], [SH05]. Most of them focused on simple Web services request-responses not enabling more complex interactions. Even though, significant results was achieved in the mapping of description [GC04] and the complex semantics [NK05], [GNC05] that not always follow the same principles in both areas.

One approach for supporting complex conversations with partners that possess little reasoning power was proposed in [AGP03] which can be described as an orhcestrator services that is in charge of performing the reasoning about the dialogs it manages and guides the participants about the options available to proceed in the dialog. Still, this implies dependency of the participants to this orchestrator which would hinder the autonomy of them which is not always desirable.

The idea in this paper is to take advantage of the latest advances in Web Services and go beyond a simple Gateway and merge them in a more integral way.

6.2 Future work

The present work provides only a solution for an integration at a low level, still some issues at higher levels are open. As discussed previously, the description of services and agents should inform about their reasoning power as well, but how this can be done is still open, one first approach would be to mention the speech acts, protocols and ontologies that a party understands. The general problem about agents description in Web Services is left open in this article since it is part of another related study area, the description, publication and discovery of services. In the contexts of the present work, descriptions remain as one kind of contents the messages can have and discovery as a specific kind of protocol.

Another situation that needs consideration is the unique identification of services when the systems that will interact are not known at all and could have different identification assigning mechanisms. This issue is very common in peer to peer networks of heterogeneous systems [JeGSS04] [FP04] [ZZ04]. Relevant work has been done in this area using FIPA agents. Porting such a solution to this integrated architecture would provide interesting insights for Web Services based peer to peer networks.

A machine readable semantic description for the provided speech act library is another issue that will demand detailed studying. FIPA provides one based on the beliefs of the participants, but a description of them using OWL-S would be very convenient also.

Apart of these issues, using WS-CDL provides the possibility to describe FIPA protocols and enrich them with exception handling and communication reliability support. These descriptions could be recombined to produce more complex conversations during design time or runtime as done by sophisticated multiagent systems.

This architecture is being currently implemented using JADE [Til03] and will be made available to the community. The implementation will be used for further experiments concerning the integration of Web Services and agents as mentioned in this section.

The results of this work will be of relevant contribution for the specification of a communication framework based on Web Services, capable of supporting participants of different processing power (like pursued in FIPA specifications). This is an important aspect in the integration of heterogeneuos autonomous agents in a grid environment.

7 Conclusion

The resent development of Web Services and the tendency shown towards an agent paradigm has provided the possibilities for integrating agents and Web Service messaging in a way that allows complex conversations over Web Services in opposite to the majority of previous solutions that were restricted to request-response interactions.

The properties that both scenarios exhibit have been described and compared and an integration has been produced based on these arguments. This integration takes advantage of the similarity between the messaging frameworks in both architectures and using a mapping between concepts both architectures are integrated at a messaging level.

The integration at the messaging level serves as the first step towards a complete integration. It is the start point for further investigation about integration of Web Services and agents and serves as experimentation platform for other solutions for issues present in higher levels of the architecture, like description, discovery or complex conversations.

The experience using agents for communication of autonomous systems, that guarantee certain constraints in security and independence can be applied to the development of Service Oriented Architectures(SOA), specially those where the communication will be performed between systems belonging to different organizations or that for some reason must remain independent of each other or that cannot be lead by another system in an orchestration fashion. A common and at the same time flexible and simple communication specification is required for the success of such a SOA or grid systems of this kind, as learned in the Agentcities project [Age02a]. Web Services are already moving towards this end and the integration provided here leverages this agent-oriented paradigm in the Web Services scenario and allows a way to form grids of autonomous Web Services.

This integration would not only allow complex conversations between participants, but would also bring agents the possibility of providing services that, in spite of having a simple interface, are of high complexity like, for instance [DW05], services that are not delegated a task to perform but a goal to achieve, where the tasks needed for achieving it should be resolved by the service itself, the kind of services that differentiate common Web Services from agents.

References

[Age02a] Agentcities. Agentcities. http://www.agentcities.org, 2002.

[Age02b] Agentcities Web Services Working Group. Integrating Web Services into Agentcities, 2002. http://www.agentcities.org/Activities/WG/WebServices/.

[AGP03] L. Ardissono, A. Goy, and G. Petrone. Enabling conversations with web services. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 819–826, New York, NY, USA, 2003. ACM Press.

[Bar03] Douglas K. Barry. *Web Services and Service-Oriented Architectures*. Morgan Kaufmann, 2003.

[BMO00] B. Bauer, J. P. Müller, and J. Odell. Agent UML: A Formalism for Specifying Multiagent Software Systems. In *ICSE 2000 Workshop on Agent-Oriented Software*, 2000.

[CKM+03] Francisco Curbera, Rania Khalaf, Nirmal Mukhi, Stefan Tai, and Sanjiva Weerawarana. The next step in Web services. *Communications of the ACM*, 46(10):29–34, 2003.

[CV02] Marco Colombetti and Mario Verdicchio. An analysis of agent speech acts as institutional actions. In *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1157–1164, New York, NY, USA, 2002. ACM Press.

[DW05] Ian Dickinson and Michael Wooldridge. Agents are not (just) Web Services: considering BDI agents and Web Services. *Workshop on Services-Oriented Computing and Agent-Based Engineering at AAMAS 05*, 2005.

[Est06] Esteban León Soto. FIPA Speech Acts Schema. http://www.dfki.de/ estebanl/FIPA/speechActs.xsd, 2006.

[FIP02a] FIPA . Foundation for Intelligent Physical Agents. On line, 2002. http://www.fipa.org.

[FIP02b] FIPA. FIPA Agent Message Transport Envelope Representation in XML Specification. Technical report, Foundation For Intelligent Physical Agents (FIPA), http://www.fipa.org/specs/fipa00071/, 2002.

[FIP02c] FIPA. FIPA Agent Message Transport Services Specification. Technical report, (FIPA) Foundation for Intelligent Physical Agents, http://www.fipa.org/specs/fipa00067, 2002.

[FIP02d] FIPA. FIPA Communicative Act Library Specification. Technical report, Foundation For Intelligent Physical Agents (FIPA), http://www.fipa.org/specs/fipa00037, 2002.

[FIP02e] FIPA. FIPA Interaction Protocols Specification. Technical report, Foundation For Intelligent Physical Agents (FIPA), http://www.fipa.org/repository/ips.php3, 2002.

[FIP04] FIPA. FIPA Agent Management Specification, 2004. http://www.fipa.org/specs/fipa00023.

[FP04] Andrew Fritz and Jehan-FranPáris. Maille Authentication, A Novel Protocol for Distributed Authentication. In *Security and Protection in Information Processing*, 2004.

[GC04] Dominic Greenwood and Monique Calisti. Engineering Web Service - Agent Integration. In *Systems, Cybernetics and Man Conference*. Whitestein, IEEE, 2004.

[GNC05] Dominc Greenwood, Jozef Nagy, and Monique Calisti. Semantic Enhancement of a Web Service Integration Gateway. *Workshop on Services-Oriented Computing and Agent-Based Engineering at AAMAS 05*, 2005.

[JeGSS04] William K. Josephson, emin Gün Sirer, and Fred B. Schneider. Peer-to-Peer Authentication with a Distributed Single Sign-On Service. In *Peer-to-Peer Systems III*, volume 2872 of *LNAI*, 2004.

[Jen01] Nicholas R. Jennings. An agent-based approach for building complex software systems. *Commun. ACM*, 44(4):35–41, 2001.

[NK05] Xuan Thang Nguyen and Ryszard Kowalczyk. WS2JADE: Integrating Web Services with Jade Agents. *Workshop on Services-Oriented Computing and Agent-Based Engineering at AAMAS 05*, 2005.

[OvDPB00] J. Odell, H. van Dyke Parunak, and B. Bauer. Representing Agent Interaction Protocols in UML. In *International Workshop on Agent-Oriented Software Engineering*, 2000.

[Pas04] Serena Pastore. Using Web Services in a Grid Infrastructure. *Grid Services Engineering and Management*, 2004.

[Sad91] M.D. Sadek. Dialogue Acts are Rational Plans. *ESCA/ETRW Workshop on the Structure of Multimodal Dialogue*, 1991.

[SH05] Munindar P. Singh and Michael N. Huns. *Service-Oriented Computing Semantics, Processes and Agents*. WILEY, 2005.

[She03] Onn Shehory. Robustness Challenges in Peer-to-Peer Agent Systems. In *Agents and Peer-toPeer Computing*, 2003.

[Til03] Tilab. Java Agent Development Framework, 2003. http://jade.tilab.com/.

[W3C] W3C. World Wide Web Consortium. http://www.w3.org.

[W3C02] W3C. Web Services Description Language, 2002. http://www.w3.org/2002/ws/desc/.

[W3C03] W3C. SOAP Simple Object Access Protocol, 2003. http://www.w3.org/TR/soap/.

[W3C05] W3C. Web Services Choreography Description Language, 2005. http://www.w3.org/TR/ws-cdl-10/.

[W3C06a] W3C. Web Services Addressing. http://www.w3.org/2002/ws/addr/, 2006.

[W3C06b] W3C. Web Services Addressing 1.0-Core. http://www.w3.org/TR/ws-addr-core, 2006.

[W3C06c] W3C. Web Services Addressing 1.0-SOAP Binding. http://www.w3.org/TR/ws-addr-soap, 2006.

[ZZ04] Yuqing Zhang and Dehua Zhang. Authentication and Access Control in P2P Network. In *Grid and Cooperative Computing*, 2004.