Agents Negotiating with Semantic Web Services

Maruf Pasha, H. Farooq Ahmad

Abstract –**Technological advancements in the Web services standards have lead to the development and deployment of broad applications in open and dynamic environment. These standards enable the services to be discovered and invoked dynamically. In current web, due to lack of semantics, agents cannot understand services that radically influence the negotiation, coordination and cooperation among the heterogeneous environments. Hence semantics play a vital role to bring automation in the current web and are highly desirable to enhance the capabilities of Web services. We believe that the vision of the semantic web can be realized if the agents are intelligent enough to process and interpret the semantic content based on the understanding which they have developed about the contents through the use of attached ontologies, since the agents have well defined reasoning, decision making, and interaction mechanisms. Our research objective is to propose new paradigm for interactions among semantic web services and software agents for problem solving. In this paper, our focus is to address the issue of negotiation between agents and semantic web services and present our initial work towards the solution, by presenting how the agents can discover, invoke and negotiate with the Semantic Web services for the autonomous service provisioning and consumption. We have achieved this interoperability, where the agents can efficiently communicate with the Web services in the dynamic and heterogeneous environments.**

Index Terms **—-Agents ,Ontology Gateway, , Semantic Language, Semantic Web services, Web Ontology Language(OWL) .**

I. INTRODUCTION

The aim of the web services (WS) endeavor is to obtain an environment where service customers and service providers can set (negotiate) the terms and conditions of service invocation automatically and then execute the necessary actions according to the prevailing contract. The semantic web adds machine- understandable semantics to data, thus enabling processing on behalf of the human user. Although the new possibilities promised by emerging technologies seem attractive, the Semantic Web with its tools and related technologies like OWL, WSDL, UDDI, SOAP and WS are likely to fall short of realizing an automated interaction and negotiation mechanism [6]. Many challenges as predicted in [20] still lie ahead for the following reasons: (i) Clusters of information and services on the Web are unaware of their context. (ii) These entities are not designed to act upon nor reconcile ontologies; they are passive until an invocation brings them to life. Therefore, they fail to capitalize on a potential intelligence, other capabilities and to learn from past experience to improve system performance. (iii) The services and data being used are not autonomous. That is to say, although the Semantic Web promises to make available to programs the meaning of the content of Web pages, these entities alone will not be able to make decisions, interact, and cooperate with other entities. Agent infrastructure, on the other hand has much to offer in this regard.

An agent possesses the ability to comprehend and interact with its environment. Because of being context-aware, autonomous and able to interpret semantics with the help of ontological knowledge representation, agents are a necessary complement to web services to realize the vision of semantic web. The relation between WS and agent systems has already been mentioned by [24] where a web service is viewed as ‘‘an abstract notion that must be implemented by an agent”. Several arguments have been made to support the idea of integration of WS and agent infrastructure, including [13, 14, 20] but perhaps none more evocative than statements made in [21] which clearly expresses the notion that, “software agents are the running programs that drive WS - both to implement them and to access them as computational resources that act on behalf of a person or organization”. To enable this integration, several core issues are there out of which bidirectional service discovery, service invocation and negotiation are the most pertinent. Some significant work has begun taking place in the research community as regards discovery and invocation of a web service from agent infrastructure and vice versa but not much has been done on the issue of negotiation among the two entities. Our goal is to take the flexible interaction schemes from the Multi- Agent Systems (MAS) research, and utilize them to enable negotiation among semantic WS, a paradigm that supports rigid and mechanical interaction protocols, and agent infrastructure. In this paper, we propose an abstract architecture for conducting such negotiations. Figure 1 illustrates our vision of an autonomous, flexible and interactive environment.

The rest of the paper is organized as follows: Section 2 contains a thorough literature review concerning some highly significant issues regarding interoperability of the two paradigms, particularly negotiation, conversation and interaction patterns among the two entities. Section 3 highlights the merger of different candidate technologies. Section 4 briefly defines the negotiation process in the light of literature.

M. Pasha is with Valoria labs, University of South Brittany, France, on leave from National University of Science and Technology, Pakistan. (e-mail: pasha@univ-ubs.fr)
H.F. Ahmad is with National University of Science and Technology, Pakistan(e-mail: drfarooq@niit.edu.pk)

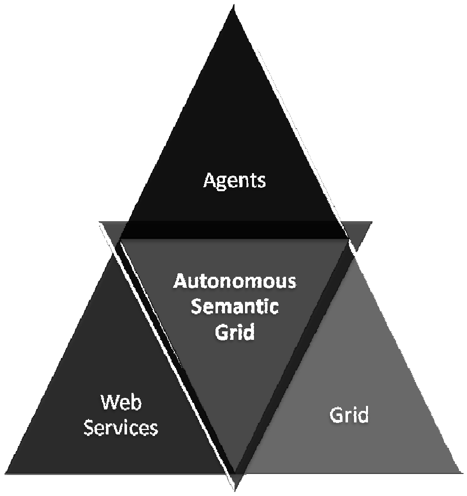

Figure 1: Autonomous Semantic Grid

Role of ontologies in interaction among agents as well as WS has been highlighted in Section 5. Section 6 contains a thorough review concerning some highly significant issues regarding interoperability of the two paradigms. Particularly negotiation, conversation and interaction patterns among the two entities that are captured moreover the details of proposed architecture and steps involved in negotiation have been highlighted. Section 7 highlights conclusion and intended future work followed by references quoted in section 8.

II. Literature Review

Keeping in mind our long- term objective of interoperability between agents and web services we have conducted a thorough study of the capabilities of OWL-S and the potential of semantic web services [17,18]. With OWL-S markup of services, the information necessary for WS discovery could be specified as computer- interpretable semantic markup at the service Web sites, and a service registry or ontology-enhanced search engine could be used to locate the services automatically [7,15].Execution of a Web service can be thought of as a collection of remote procedure calls. OWL-S markup of WS provides a declarative, computer-interpretable API that enables automated WS execution [7, 11, 19].

Given a high-level description of the task by the user, automated composition and interoperation of WS to perform the task is of particular interest to us. With OWL-S, the information necessary to select and compose services would be encoded at the service Web sites [3]. Software agents can be written to manipulate and interpret this markup, together with a specification of the task and thus can be bestowed with the ability to perform the task automatically [7, 9, 10, 11]. WS- Conversation Language (WSCL) and WS-Agreement are two languages to implement Contract Net Protocol (CNP) for negotiation among WS [6]. Yet, the flexibility of negotiation is far-off from that prevalent in the agent infrastructure. FIPA [23] provides detail specifications of Request protocol, Request/ Response protocol, CNP, English Auction, Dutch Auction, Brokering protocol, etc. An exhaustive overview of MAS is beyond the scope of this paper, but essential pointers include [16] and [12]. Two significant alternatives in empowering WS with agents' properties are discussed in [7]. One is to implement a wrapper, which turns a current Web service into an agent like entity. The other alternative is to capture all the functionalities of a Web Service and imbed them into an existing software agent. The authors in [2] propose an architectural model for enabling transparent, automatic connectivity between WS and agent services. A later version of this project [1] uses OWL-S to add semantic aspect to service descriptions. OWL is said to be a candidate content language for ACL messages in MAS and for conducting auctions among agents that use OWL ontologies [8].

In this paper, we contend that the degree of flexibility that persists in agent interaction scenarios can never be achieved in negotiations among WS alone. In order to improve flexibility level, negotiation among semantic WS and agents is highly significant.

III. Merger Of Technologies

The Web is evolving toward machine-readable infrastructure for sharing knowledge among humans as well as machines as discussed previously. Application designers building systems that harness web services are facing many of the same issues that designers of agent systems have been tackling for more than 30 years. We believe that agents have much to offer in overcoming some of the inherent difficulties when dealing with complex and dynamic environments. As pointed out, such environment requires autonomous and conversational components and agents can offer these capabilities. Agents are able decide at run-time which resources and/or service to use for particular task.

In open systems, the structure of the system itself can dynamically change. The characteristics of such a system are that its components and resources are not known in advance, can change over time, and may be highly heterogeneous. The best-known example of a highly open software environment is the Internet. The functionality is almost certain to require techniques based on negotiation and cooperation, which lie very firmly in the domain of agent system. The machine should not just act as a dumb receptor of task descriptions, but should cooperate with the users to achieve their goal. These considerations give rise to the idea of an agent acting as an expert assistant or delegate with respect to some application, knowledgeable about both the application itself and the user, and capable of acting with the user in order to achieve the user's goals. The Grid community has previously focused on interoperable infrastructure and tools for secure and reliable resource sharing within dynamic and geographically distributed organizations. In contrast, those working on agents have focused on the development of concepts, methodologies, and algorithms for autonomous problem solvers that can act flexibly in uncertain and dynamic environments in order to achieve their goals (Foster, 2004). Our vision of the integration of agents with Semantic Web and Grid computing[25] is to lay foundation for self-regulating system, namely Autonomous Semantic Grid as shown in Figure. 1. In the proposed system higher level interaction is pivotal for creation of virtual organization that exhibits characteristics of Autonomous Decentralized System .Theoretical foundations of ADS rely on the principles of autonomous controllability and autonomous coordinatability. These two properties assure online expansion, fault tolerance and online maintenance of the systems. ADS can be realized through software agents, which are autonomous, proactive, and goal-oriented problem solving entities with social ability for high level interaction.

IV. Negotiation Process

Negotiation is an iterative communication and decision making process between two or more agents (parties or their representatives) [4] who: (i) cannot achieve their objectives through unilateral actions; (ii) exchange information comprising offers, counter-offers and arguments; (iii) deal with interdependent tasks; and (iv) search for a consensus which is a compromise decision. The outcome of a negotiation can be a compromise (an allocation) or a disagreement as shown in figure 2. Decision-making rules are used to determine, analyze and select decision alternatives and concessions. Rules of communication determine the way messages are exchanged among negotiators.

Negotiation protocol, on the other hand, defines permissible actions and their sequence, allowable offers, and timing of offers and messages. It may also specify the syntax and semantics of the messages, and mechanisms in which alternatives are determined and assessed, offers are constructed, and concessions are made. In this paper, we focus on introducing intelligent interaction patterns between agents and WS aimed at utilizing the semantics associated with the involved parties. In order to achieve this objective, the existence a common vocabulary (i.e. ontology) is inevitable.

V. Role Of Ontologies In Interaction

A. Multi Agent Systems

As autonomous problem solvers, agents need to develop model of their environment that allows them to reason on how their actions affect their environment and how those changes lead them to achieve their goals [5]. Ontologies provide the conceptual framework that allows agents to construct such models: ontologies describe the properties of entities that agents encounter, and relations between them.

Thus a common vocabulary in the form of ontologies is at the heart of intelligent communication among agents.

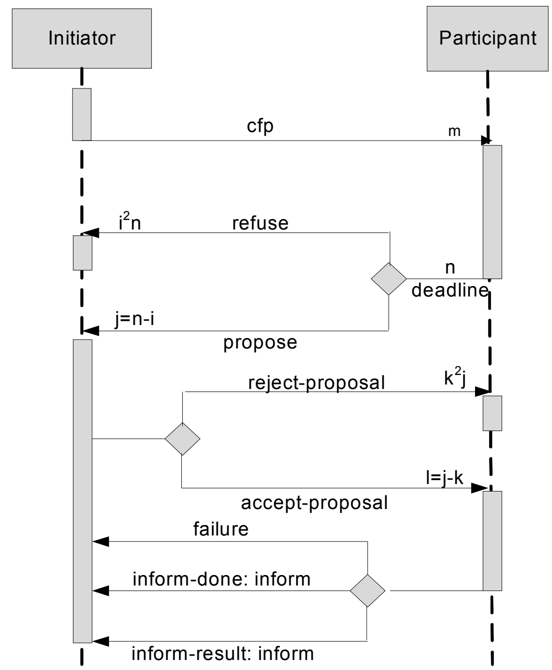

Figure 2: Negotiation Process

B. Semantic WS

The semantic web initiative [20, 21] that addresses the problem of XML's lack of semantics by creating a set of XML based languages, also relies on ontologies that explicitly specify the content of the tags. The Web Ontology Language (OWL) is a forthcoming W3C specification for such a language which will supersede the earlier DARPA Agent Markup language (DAML+OIL) [22]. OWL is an extension to XML and the Resource Description Framework (RDF) enabling the creation of ontologies for any domain and the instantiation of these ontologies in the description of resources. The OWL-Services language (OWL-S) [19] is a set of language features arranged in these ontologies to establish a framework within which the WS may be described in the semantic web context.

VI. Proposed Architecutre

In order to provide an initial implementation of negotiation between agents and semantic web services, we choose one of the FIPA standard interaction protocols for the agent infrastructure i.e. CNP [23].

According to our proposed architecture as shown in figure3, an entity called "Ontology Gateway" (OG) our previous work acts as a broker to conduct the negotiation between the two heterogeneous entities. It is an application (neither an agent nor a web service) that operates in a distributed environment. In its preliminary stage, the gateway currently has "Bidirectional OWL-FIPA Translator". We propose another module called "Negotiation Module" (NM) in the OG which supervises the sequence of messages as they are being exchanged. This would ensure that pre-agreed protocol is followed and whole process appear transparent to the parties involved in the negotiation.

There are several prerequisite tasks that need to be done before carrying out the standard CNP. In our architecture, we propose a "Control Unit" (CU) for handling these initial tasks. Once these have taken place, the control is shifted to the NM within the OG which then carries out the negotiation protocol by communicating messages in a sequence. Current implementation of the Gateway consists only of the bidirectional OWL- FIPA ontology translator.CU has been added so as to supervise the flow of messages from input to internal architecture of the gateway as well as of messages from internal architecture to the output.

This paper discusses only the scenario where an agent asks to conduct negotiation with a semantic web service and not vice versa. Figure 3 explains the proposed architecture of an OG in detail while showing its relation with outside world.

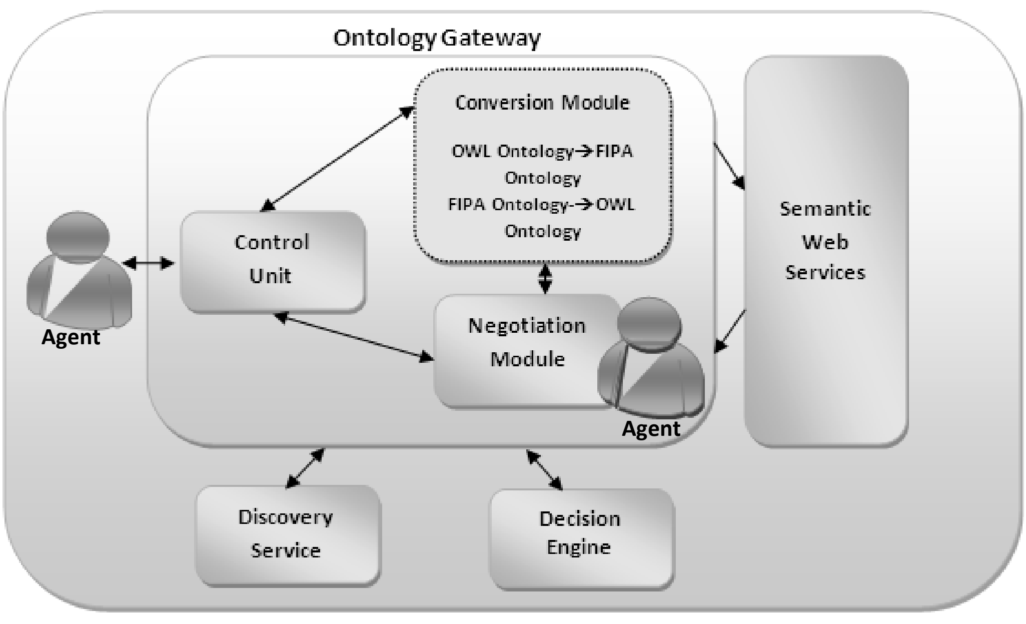

Figure 3: Components of Ontology Gateway for Negotiation

A. Control Unit

First of all, the requesting agent who wants to initiate negotiation invokes the service exposed by OG and sends its reference and message. This message is received by the CU within the OG. The message content contains the description of the service based on which CU identifies other negotiating party (i.e. a semantic web service).

B. Bi-Directional OWL- FIPA Ontology Converter

Since this ontology is written in OWL for a semantic web service, which though is allowed as a valid content language by FIPA but is not as expressive as SL, so there is a need to translate this ontology from OWL to SL. The CU feeds this ontology to the OWL to FIPA ontology translator, which returns the FIPA Ontology equivalent of the OWL ontology fed as an input. This is done with the help of a matchmaking service that returns a reference or handle of that service to the CU. This handle enables the CU fetch the service profile of the service and its ontology, without which semantic understanding is unattainable.

C. Negotiation Module

In order to conduct meaningful negotiation between agent and semantic WS, this module, also known as Mediator, needs the reference of the requesting agent, its FIPA ontology and service profile of the semantic web service. Handles to all of these resources are passed to this module at the time of transfer of control to it by CU. This module shall create an agent at runtime referred to as Runtime Agent (RA), which will be used to query the web service ontology. This agent extracts all negotiable parameters from the profile of the semantic web service with the help of understanding its ontology.

D. Flow of Control during CNP

Once the prerequisites for the implementation of negotiation protocol are set by the CU, it is now time for the NM to initiate the negotiation process according to a formalized protocol (CNP, in this case). The RA acts as Initiator of the protocol and sends a call for proposal to the semantic web service which assumes the role of Participant of CNP. For sending such a request (i.e. a cfp message), NM needs user preferences too. These user preferences can be retrieved with the help of the requesting agent reference that has been passed to NM by the CU.

Along with getting these parameters from the requesting agent, NM needs to have agent's ontology too so as to get a semantic understanding of what these parameters mean based upon a common vocabulary. It can also be accessed with the help of agent's reference that this module has. Such an understanding of semantics will help NM map the information that the requesting agent possesses to the information required by the service method that is going to be invoked as a result of sending a cfp. As a response of this, the participant would return all possible options (proposals) to the RA. Each of this response is the handle of a semantic web service that closely matches the requested service description. These responses are passed to a decision engine. We assume here that the decision engine is an independent component that uses artificial intelligence and semantic deduction rules to choose the best possible option out of many as per the closest match with user preferences.

The decision engine sends the chosen option back to RA which invokes the corresponding service. The service is executed as a result of this invocation and a response is sent to the RA indicating whether the service has succeeded or failed. RA forwards this response to NM which stores the results in its knowledge base and returns control to CU. Finally, the requesting agent is informed of the results of negotiation along with all associated details by CU and the negotiation session comes to an end.

VII. Application Scenario

In this section, we map our abstract level design to a real- life scenario where negotiation is required between an agent application and a semantic web service.

1- In the Agent infrastructure, we choose a meeting scheduler agent that arranges a meeting among two or more persons at the desired time and location by the consensus of all participants. In case of a failure, it signals the participant.

2- As a semantic web service, we consider an airline providing online procedures to do routine tasks. Assuming that a meeting has been arranged by the meeting scheduler agent, next step for the participants is to reach the venue at the required time. We assume that the set location is such that at least one participant has to travel by air to reach that place. In an automated flight selection and reservation, negotiation is an elementary requirement. While choosing one of the available flights, factors such as time, fare, and comfort level are negotiable. We aim to carry out this negotiation between the meeting scheduler agent and the flight service. Given below is a sequence of how control shall be exchanged among various modules of an OG and what actions shall take place in between: -

a. First of all, the meeting scheduler agent sends a message to OG expressing its desire to engage in negotiation. The message is received by the CU. The content of the message contains the description of a flight service. The CU identifies appropriate web service dynamically on the basis of message content.

b. The handle of the service is returned to CU with the help of which CU fetches the profile of the service and its ontology. The CU feeds this ontology to its OWL to FIPA ontology translator, thus acquiring the FIPA Ontology equivalent of the OWL ontology.

c. Then, the CU passes the reference of the meeting scheduler agent and its FIPA ontology along with the profile of the semantic web service to the NM. NM creates RA which extracts all negotiable parameters web service profile. Next NM acquires the user preferences for all the negotiable parameters with the help of the meeting scheduler reference that was sent to it by CU.

d. It is now required to invoke the web service's method that takes user requirements as input parameters and returns a list of matching flights. These parameters shall be passed in a format and sequence expressed in the service profile. RA invokes the web service

e. As a result, RA shall receive a list of flights, closely matching user criteria. These responses are equivalent to the "proposals" as referred to in figure 2. RA forwards these responses to the decision engine to choose a particular flight and respond to RA. RA then invokes the chosen. A response is sent to the RA indicating that the service has succeeded which in turn informs the NM. NM returns control back to the CU after storing results permanently for future use. The CU informs the meeting scheduler agent of the reserved flight and all associated details and the negotiation session comes to an end.

VIII. Conclusions

In this paper we have proposed an enhancement in Ontology Gateway architecture for enabling flexible, autonomous interaction between Semantic WS and agent services. We have also highlighted the technologies and how they come together in order to achieve the whole process. For now we have considered the case where an agent negotiates with the web service. An initial implementation of this architecture has been done and we intend to improve the proposed design so as to cater more negotiation protocols especially, the auction protocols in future. We expect that this initial effort of conducting negotiation via Gateway service bridging agents and WS is only a prelude to exploring the immense potential it offers as a means to compose, invoke, administer and manipulate heterogeneous service populations in future.

REFERENCES

[1] Greenwood, D., Calisti, M., Nagy, 2005, J.: Semantic Enhancement of a Web Service Integration Gateway, AAMAS SOCABE Workshop, Utrecht, Netherlands

[2] Greenwood, D., Calisti, M., 10-13 October, 2004, An Automatic, Bi-Directional Service Integration Gateway, IEEE Systems,Cybernetics and Man Conference; the Hague, Netherlands

[3] Laukkanen, M., Helin, H., July 2003. Composing workflowsof semantic web services. In Proc. of the 1st International Workshop on Web Services and Agent Based Engineering, Sydney, Australia.

[4] Richler, M., Kersten, G., Strecker, S., 2003, Towards a Structured Design of Electronic Negotiations. Group Decision and Negotiation.

[5] Sycara, K., Paolucci, M.: Ontologies in Agent Infrastructure. Carnegie Mellon University, USA

[6] Paurobally, S., Jennings, N. R., 2005, Protocol engineering for web services conversations. Journal of Engineering Applications of Artificial Intelligence.Special Issue on Agent Oriented Software Engineering

[7] Tadiou, K. M.: Semantic Web services Interaction

[8] Zou, Y., Finin, T., Ding, L., Chen, H., 2003 TAGA: Using Semantic Web Technologies in Multi- Agent Systems, IJCAI

[9] Sirin, Hendler, 2003, Semi-automatic Composition of Web Services using Semantic descriptions. 1st Workshop on Web Services: Modeling, Architecture and Infrastructure in conjunction with ICEIS.

[10] Laukkanen, M., Helin, H., July 2003, Composing workflows of semantic web services. In Proc. of the 1st International Workshop on Web Services and Agent Based Engineering, Sydney, Australia,

[11] K Sycara, M Paolucci, A Ankolekar, N Srinivasan,2003,Automated discovery, interaction and composition of semantic web services in Web Semantics: Science, Services and Agents on the World.

[12] M. Luck, P. McBurney, C. Preist. Agent Technology: Enabling Next Generation Computing. http://www.agentlink.org/roadmap, 2003.

[13] Lyell, M., Rosen, L., Casagni-Simkins, L., Norris, D., July 2003, On software agents and web services. In Proc. of the 1st International Workshop on Web Services and Agent Based Engineering, Sydney, Australia.

[14] Maximilien, E., M., Singh, M., P., July 2003, Agent-based architecture for autonomic web service selection. In Proc. of the 1st International Workshop on Web Services and Agent Based Engineering, Sydney, Australia

[15] Sycara, K., Paolucci, M., Soundary, J., Srinivasan, N., May 2004, Dynamic discovery and coordination of agent-based semantic web services. IEEE Internet Computing, 8(3):66–73

[16] Wooldridge, M., 2002, An Introduction to Multi- Agent System. Wiley &Sons.

[17] McIlraith, S., Son, T., C., Zeng, H. 2001, Semantic Web Service. IEEE Intelligent Systems, 16(2):46-53

[18] Martin, D., Burstein, M., Lassila, O., Paolucci, M., Payne, T., McIlraith, S., November 2004, Describing Web Services using OWL-S and WSDL:
http://www.daml.org/services/owl- s/1.1/owl-s-wsdl.html

[19] OWL based web service ontology (OWL-S) 1.2 Pre release
http://www.ai.sri.com/daml/services/owl-s/1.2/

[20] Berners-Lee, T., Hendler, J., Lassila, O., 2001, The semantic web. Scientific American, 284(5):34–43

[21] W3C Semantic Web Activity:
http://www.w3.org/2001/sw/

[22] Language Specification of DAML+OIL 27 March 2001:
http://www.daml.org/2001/03/daml+oil-index.html

[23] FIPA Communicative Act Library Specification:
http://www.fipa.org/specs/fipa00037/

[24] Web Services Architectures, W3C Working Draft 14 May 2003:
http://www.w3.org/TR/2003/WD-ws-arch-20030514/

[25] Maruf Pasha, H. Farooq Ahmad, Hiroki Suguri, "Autonomous Semantic Grid: Architecture and Implementation" ,Open Grid Forum 19,February 2007,Charlottee USA.