

A Coordinated Architecture for the Agent-based Service Level Agreement Negotiation of Web Service Composition

Mohan Baruwal Chhetri¹, Jian Lin¹, SukKeong Goh¹, Jun Yan², Jian Ying Zhang¹, Ryszard Kowalczyk¹

¹*Faculty of Information and Communication Technologies
Swinburne University of Technology, Australia
{mchhetri, jlin, sgoh, jyzhang, rkowalczyk@ict.swin.edu.au}*

²*School of Information Technology and Computer Science
University of Wollongong, Australia
jyan@uow.edu.au*

Abstract

Recent progress in the field of web services has made it possible to integrate inter-organizational and heterogeneous services on the web at runtime. If a user request cannot be satisfied by a single web service, it is (or should be) possible to combine existing services in order to fulfill the request. However, there are several challenging issues that need to be addressed before this can be realized in the true sense. One of them is the ability to ensure end-to-end QoS of a web service composition. There is a need for a SLA negotiation system which can ensure the autonomous QoS negotiation of web service compositions irrespective of the application domain. In this paper we propose agent-based coordinated-negotiation architecture to ensure collective functionality, end-to-end QoS and the stateful coordination of complex services. We describe a prototype implementation to demonstrate how this architecture can be used in different application domains. We have also demonstrated how the negotiation system on the service provider's side can be implemented both as an Agent based Negotiation System and as a Web Service based Negotiation System.

1. Introduction

With recent advances in Web Service technology it is possible to dynamically publish, discover and invoke web services. The principal objective of the web services effort is to facilitate an environment in which

service customers and providers can dynamically locate and connect with each other, automatically set the terms and conditions of invocation and then execute the necessary actions according to the negotiated contract. If a particular user request cannot be satisfied by a single Web Service, then it is (or should be) possible to combine existing services together to form a web service composition which can then fulfill the request.

Service composition is the process of combining several component services and bundling them together to meet the specific needs of a user request. It has potential benefits from both the web service provider's perspective and the consumer's perspective. Some benefits include rapid application development, service reuse and complex service summation for the former and seamless access to a variety of complex services for the latter [1]. The consumer need not be aware of the actual composition of several heterogeneous Web Services, since all the integration logic is hidden from the user. These are the principle on which Web Service Technology has been developed.

A service composition life-cycle, in general, can be characterized by five phases – planning phase, definition phase, scheduling phase, construction phase and the execution phase [2]. A service composition can initially be described in abstract and then later be made concrete and executable. An abstract service composition consists of base service types without any concrete bindings. The outcome of the construction phase results in the concretization of the abstract service composition, from a set of potentially available component services.

How an abstract service composition is created from the user request is outside the scope of this paper and our research efforts. Significant research efforts are being carried out in this particular direction and in the area of web service composition in general [1] [3] [4] [26]. We are more interested in how an abstract service composition can be transformed into a concrete executable process provided we have the end-to-end Quality of Service (QoS)¹ requirements for the composition and a list of potential service providers for each base service type. The research challenge is to provide a mechanism for the autonomous selection of suitable service providers for each service type within the service composition while guaranteeing the end-to-end (global) QoS.

Although Service Level Agreement (SLA) contract formation and management has been the focus of research for several years, it has mainly been limited to SLA management for single service provision or, in the case of web service compositions, the manual configuration to determine the QoS constraints on individual services. The SLA management for a service composition which involves a single consumer and multiple service providers is still an open research issue. As other service composition related activities become automated, having the negotiation between the service provider and the service requestor conducted via human agents goes against the very nature of Web Service technology. Thus there is a recognized need for an architecture which can support the autonomous negotiation of QoS constraints for service compositions. This paper presents an agent-based approach for the coordinated negotiation and re-negotiation of QoS constraints for service compositions to guarantee end-to-end QoS.

The paper is organized as follow. The next section briefly discusses related work in the areas of SLA Management and Agent-based negotiation. Section 3 discusses the Adaptive Service Agreement and Process Management (ASAPM) project [8], in the context of which this research has been carried out. Section 4 discusses the negotiation of service compositions and the need for coordination as well as the coordinated-negotiation architecture. Section 5 describes the use-case scenario which we have used to demonstrate our coordinated-negotiation architecture. Section 6

discusses the implementation of the architecture and section 7 concludes this paper.

2. Related Work

Research on SLA Management has been actively carried out for the past couple of years [15]. This research has generally focused on SLA specification and definition languages and SLA creation, operation, monitoring and termination. Web Service Level Agreement (WSLA) [16] and Web Service Offerings Language (WSOL) [18] are two approaches which propose the SLA representation. WS-Agreement is a specification of the Global Grid Forum (GGF) [19] which specifies the agreement structure, a negotiation protocol and a monitoring interface. Similarly, WS-Negotiation is an XML based language which contains the Negotiation Message, Negotiation protocol and the Negotiation Decision Making process [24]. It also provides a SLA template model for supporting different types of business negotiations. Service Negotiation and Acquisition Protocol (SNAP) [20] has been proposed as a provision for resource management framework for distributed systems. However, while these standards are still evolving, they do have their limitations. For instance, WS-Agreement only supports two types of messages – *offer* and *agree*. Such simple message types cannot support coordinated transactions between multiple web services because QoS negotiation involves different states other than *offer* and *agree* such as *propose*, *counter-propose*, *reject* and others. There needs to be a well-defined protocol to enable such interactions which does not currently exist in web services. Another major limitation of the major approaches is that they are applicable only to single service provision scenarios and cannot be used automatically for service compositions.

The use of negotiation as a means of establishing service contracts has been a topic of considerable interest for quite some time now within the agent community [5] [22]. Intelligent software agent technology is an accepted mechanism in conducting automatic negotiations [5] [6] [23] and Multiagent systems have been proposed and used for the automated negotiation of resource allocation in grid environments [20] [21]. Given that agents are flexible problem solvers they are able to make context-aware and context-dependent decision about the scope and nature of their interactions. Hence given the similar environments in which web services and software agents operate, web services can only benefit from the features of multi-agent technology. Again, within the community, this agent based negotiation has principally

¹ The Quality of Service is a collection of metrics providing qualitative and quantitative representation of non-functional requirements. They are evaluated in terms of concrete measures and specify the expected service level offered by the service provider.

been in the context of one-to-one negotiation or one-to-many negotiations but for the provision of a single service. There hasn't been much work done in the negotiation of end-to-end QoS constraints. Our work is a first step in this direction.

3. Background

The work describe in this paper is based on research conducted in the context of the Adaptive Service Agreement and Process Management (ASAPM) in Service Grid Project (AU-DEST-CG060081) and the EU FP6 Integrated Project on Adaptive Services Grid (ASG) (EU-IST-004617). This project aims at developing intelligent agent-based techniques and tools to facilitate the adaptive service management and process management in order to ensure collective functionality, end-to-end QoS and the stateful coordination of complex services. This project has two principal areas of research namely *adaptive service agreement management* and *adaptive service process management*. The former includes automated service agreement negotiation and re-negotiation, service agreement lifecycle management and dynamic service profiling (DSP) while the latter includes process enactment, service agreement monitoring, process enactment visualization, and mediated workflow re-planning.

Figure 1 shows the overall architecture of ASAPM. It sits between the application domain system and the actual web service/grid service infrastructure. It focuses on providing adaptive, transparent provision of complex services based upon the user request. The ASAPM architecture consists of four components:

- *Negotiation Manager* – is responsible for the autonomous SLA negotiation and re-negotiation for composite services
- *SLA Lifecycle Management and Dynamic Service Profiling* – manages the SLA during their lifetime and maintains up-to-date service profile which can be used for SLA negotiation and re-negotiation
- *Workflow Enactment, Monitoring and Visualization* – is responsible for the enactment of the service composition, the monitoring of the negotiated QoS parameters, and the visualization of the service composition enactment.
- *Mediated Workflow Re-planning* – is responsible for mediation, aimed at providing alternate plans to satisfy the original user request if any unrecoverable error occurs

The Application Domain System (ADS) is responsible for planning the web service composition in order to satisfy the user request that it receives. The

Composition Planner provides ASAPM with the application service workflow as well as the global QoS requirements for SLA negotiation. The Negotiation Manager can get the list of potential service providers for each base service type within the composition from the Service Registry which is part of the ADS. It is then the task of the Negotiation Manager to find suitable candidates which satisfy the QoS constraints for each service type while ensuring that the end-to-end QoS is satisfied.

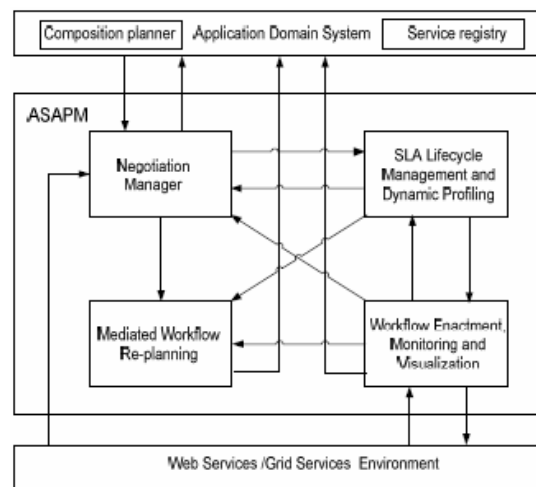


Figure 1: Overall Architecture of ASAPM

4. Negotiation of Web Service Compositions

There are two possible ways in which SLAs can be established for web service compositions. One is the *negotiate-all-then-enact* mode in which SLAs are established for all base service types within the composition before the enactment of the first service. The second mode is the *step-by-step-negotiate-and-enact* mode in which each service type is negotiated for and enacted straightaway. Both have their pros and cons. As a first step, we have chosen the *negotiate-all-then-enact* mode for service composition negotiation.

4.1. Coordinated Negotiation Architecture

The task of the Negotiation Manager is to find suitable candidates for each service type within the service composition so that the end-to-end QoS is ensured i.e. it is responsible for the concretization of the abstract service composition so that the process

becomes an executable workflow plan. Figure 2 shows the architecture we propose to achieve this.

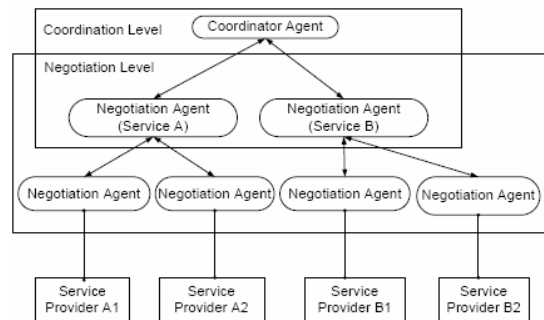


Figure 2: Coordinated-Negotiation Architecture

From a functional view point, the SLA negotiation of a service composition involves two aspects. One aspect is the negotiation between the service consumer and service providers for the QoS constraints of a single service type within the composition. The other aspect is the coordination of these concurrent one-to-many negotiations so that the end-to-end QoS constraints are satisfied. Hence there are two levels of operation – at the coordination level and the negotiation level. The coordinated-negotiation of a service composition can be supported by a Multi-Agent System (MAS).

At the coordination level, the MAS has the Coordinator Agent (CA) which is responsible for the negotiation of the service composition as a whole. It interacts with the Composition Planner (CP) of the ADS to receive the abstract service composition definition as well as the end-to-end user QoS requirements. It then analyses this abstract service composition and decomposes it into base service types. A list of potential providers for each service type is retrieved from the Service Registry (SR) of ADS. Thus primary phase of coordinated negotiation involves analyzing the service composition and the user requirements which results in a decentralized, distributed architecture. The CA creates a cluster of Negotiation Agents (NA) which can be modified and destroyed as required. While the CA has an infinite life-cycle and the knowledge of the global constraints, the lower level NAs have a finite life-cycle and only local level knowledge.

The NAs act as local problem-solvers which have specific roles and predefined capabilities which enable them to fulfill these roles. They act to select the best service provider for the particular base service type

they represent. In doing so, they indirectly contribute to the wider problem solving initiative. Each NA receives the local QoS constraints for the service type they are negotiating along with a list of potential service providers. For example, in Figure 2, NA for Service A will negotiate with two providers (Provider A1 and A2) and NA for Service B will negotiate with two providers (Provider B1 and B2) under the supervision of the CA. Each NA is required to negotiate with the service providers within the stipulated negotiation time-out and a proposal time-out which is provided to it by the CA.

4.3. Role of the Coordinator Agent

In general, the CA is responsible for the analysis of the service composition with a focus of mapping the overall QoS requirements onto the atomic service level QoS requirements i.e. QoS requirements for each service type within the composition. It is also responsible for the coordination of the negotiation for different services in the composition, with the selection of the best combination. The main responsibilities of the CA can be listed as follows:

1. Map the QoS requirements of the user request onto the QoS requirements of all services within the composition
2. Create a cluster of NAs corresponding to each service type within the composition
3. Provide these NAs with local-level QoS constraints, the negotiation timeout and the proposal timeout
4. Coordinate the NAs to negotiate with potential service providers for various services in the service composition in parallel
5. Redistribute the local-level QoS constraints if overall QoS constraints are not met
6. Coordinate the re-negotiation of QoS constraints in the case of service failure or QoS violation
7. Collaborate with the ADS as well as the other components of the ASAPM system.

4.4. Single Service Negotiation

Single service QoS Negotiation refers to the process by which a NA selects and contracts a service provider from a list of potential providers. The key concepts in automated negotiation are – negotiation protocols, negotiation objects, and the decision making model [9]. The negotiation protocols define the rules which govern the negotiation including how and when it ends. The negotiation objects are the different parameters of negotiation such as price, time etc which can be predefined using a *negotiation template* [10]

acceptable by both sides. The decision making model is used for evaluating and generating offers and counter-offers. It is independent of the negotiation protocol and uses a learning mechanism which models the agent's behavior based upon the opponent's previous offers. In order that the negotiation outcome is mutually acceptable, it is necessary that the negotiation is cooperative and competitive.

The NA carries out one-to-many negotiations with multiple service providers using the reserve values for the local QoS constraints given to it by the CA. It is constrained by a negotiation timeout (the negotiation time available) and the proposal timeout (the time it can wait for a response from a service provider for its initial call for proposal).

4.5. Coordinated Negotiation Protocol

The negotiation protocol refers to a set of rules which define the boundaries within which the participants can interact. It covers the permissible types of participants; the negotiation states (e.g. call for proposal, negotiation closed), the events that cause the change of negotiation states, and the valid actions of the agents in the different states of negotiation. In this research we have used the Iterated Contract Net Protocol (ICNP) which is one of the most widely used negotiation protocol provided by the Foundation of Intelligent Physical Agents (FIPA) [11]. ICNP supports recursive negotiation and allows for multi-round iterative negotiation to find a compromise. By exchanging modified proposals and counter-proposals, a more acceptable negotiation result is likely to be reached. While the ICNP protocol is used for the interaction between the actual negotiating agents, the interaction between CA and NA also follows the FIPA ACL performatives.

ACL Performative	ACL Message Content
REQUEST (CA → NA)	<ul style="list-style-type: none"> Reserve values (local QoS constraints) Negotiation timeout Proposal timeout
INFORM (NA → CA)	<ul style="list-style-type: none"> Negotiation results Selected provider details
CONFIRM (CA → NA)	Confirmation that selected provider is to be contracted

Table 1: List of ACL Performatives exchanged between CA and NA

Table 1 shows the ACL messages exchanged between the CA and the NA while Table 2 shows the

messages exchanged between the NAs of the requestor (client) and the service provider. Figure 3 and 4 show the interaction between the CA and the NA, both when the end-to-end QoS constraints are satisfied and when they aren't.

ACL Performative	ACL Message Content
CFP (NA _C → NA _{SP})	Initial Call for Proposal
PROPOSE (NA _{SP} → NA _C)	Offer made by Provider to Requestor
REJECT (NA _{SP} → NA _C)	Reject message if Provider is not interested in negotiating with Requestor
UNKNOWN (NA _{SP} ↔ NA _C)	Message sent by either party if message content is not understood
ACCEPT_PROPOSAL (NA _C → NA _{SP})	Message sent to Provider once proposal has been accepted by Requestor
REJECT_PROPOSAL (NA _C → NA _{SP})	Message sent if proposal is not acceptable by Requestor
INFORM (NA _{SP} → NA _C)	Message sent by Provider when Requestor accepts its proposal
FAILURE (NA _{SP} → NA _C)	Message sent by Provider if some error occurs after it receives ACCEPT_PROPOSAL

Table 2: List of Performatives exchanged between NA for Client and the Service Provider

5. Use Case Scenario Description

The use case scenario which we have chosen for the implementation of the coordinated negotiation architecture is called the **Application on Demand (AOD) System**. In this scenario shown in Figure 5, a user can request for access to a computer application via the Internet. The user provides the following information as input – IP address, Throughput (in MB), Availability (in percentage), Latency (in milliseconds) and the Maximum cost (in dollars). This information is given as input to the **Application Domain System** (which is the ISP). The ADS creates a composite web service based upon this user request and forwards this information to the ASAPM system.

The workflow plan generated by the ADS consists of three service types:

- *Provide_Access_Service* - web service which provides access from the Customer Premise Equipment (CPE) to the Provider Edge (PE).
- *Provide_VPN_Service* - web service which provides VPN connectivity from the PE to the actual host hosting the requested application.

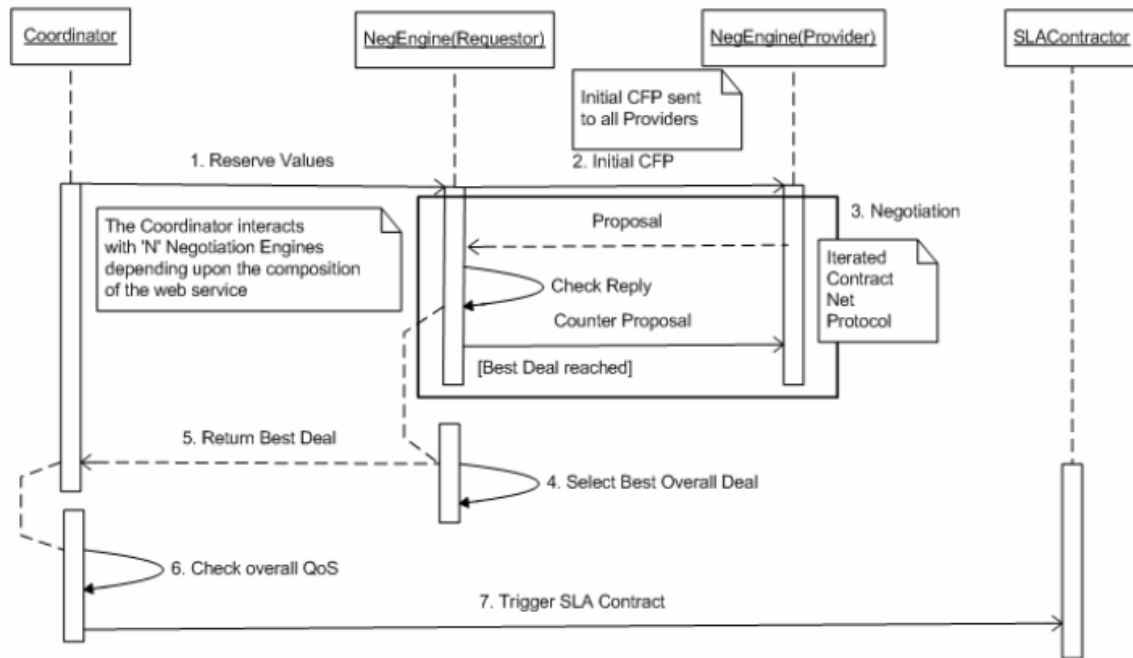


Figure 3: Interaction between CA and NA in case of successful negotiation of global QoS

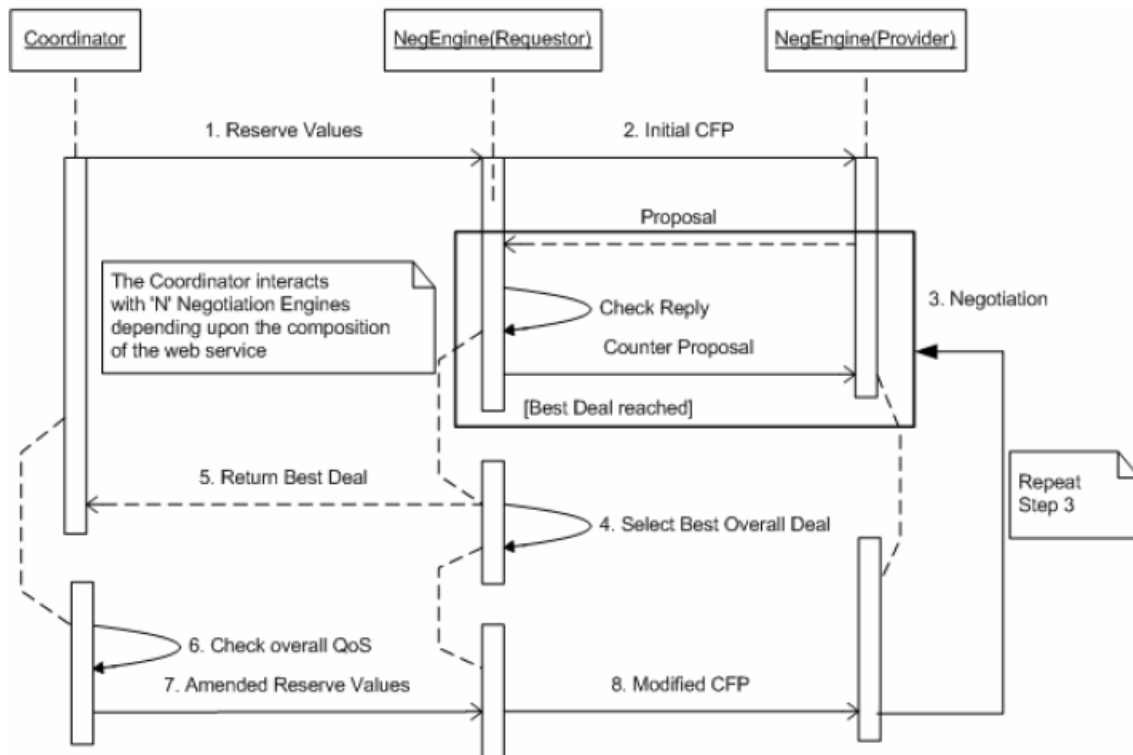


Figure 4: Interaction between CA and NA in case of failed negotiation of global QoS

- *Provide_Application_Access* – web service which provides access to the requested application

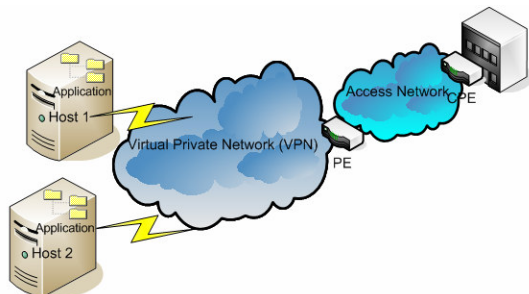


Figure 5: Application on Demand (AOD) System

In the simplest scenario, the composite workflow plan as generated by the ADS will look as shown in Figure 6.



Figure 6: Composite Workflow Plan

After receiving the workflow plan and the user request, the Negotiation Manager of ASAPM retrieves a list of prospective providers for each service type within the composition from the Service Registry of the ADS. The CA is responsible for negotiating the overall (end-to-end) QoS parameters. It creates the corresponding NAs for each service type. In this scenario, since there are three service types the CA creates three corresponding NAs. It then splits the global or end-to-end QoS amongst the three NAs. As a starting point, we have used an equal split of the QoS values amongst all the NAs. The CA passes the following data to each NA:

- Reserve values for each negotiable attribute
- List of prospective service providers (and their corresponding agent location /id)
- The negotiation timeout
- The proposal timeout

Once the CA has created the NAs and provided them with all the data necessary for carrying out the negotiation, it waits for the reply from each NA. The NAs carry out one-to-many negotiations with the Service Providers' NAs until a favorable outcome is

reached or the negotiation timeout is reached. They then return the negotiation results back to the CA.

When the CA has received all the negotiation results, it compares them to see if they satisfy the global QoS. If the global QoS is satisfied, it passes this information on to the SLA Manager which then contracts each service provider. It is then the task of the Enactment, Monitoring and Visualization component to enact the workflow plan in the predefined order while monitoring the actual values against the contracted values. The user is provided connection to the host which is offering the request application. However, if the global QoS is not satisfied, then the CA has to redistribute the reserve values amongst the NAs and return them to the NAs which again try to negotiate with all the service providers.

Note: Since this is an initial prototype of the Coordinated Negotiation Architecture, we have demonstrated a simple scenario in which the global QoS is satisfied.

6. Implementation of the Coordinated-Negotiation Architecture

The Negotiation Manager has been implemented using the FIPA compliant JADE Agent Framework [12] and WS2JADE [13] [14], a toolkit which enables the integration of JADE agents and web services. There are two types of QoS Negotiation scenarios – one in which the NAs representing the user (part of the Negotiation Manager of ASAPM) directly negotiates with NAs representing the service provider (external to the ASAPM), and another in which the NAs representing the user negotiate both with Negotiation Agents and with the Negotiation Web Services using the WS2JADE toolkit which sits on top of the JADE framework. Figure 7 shows a scenario in which both types of negotiation are carried out. The NA for Service A negotiates directly with the Service Provider A-1's NA using ACL Messaging and with the Negotiation Web Service of Service Provider A-2 via the Proxy Agent. In this case, the mapping from the ACL Message to SOAP message is handled by the Proxy Agent.

6.1. Direct Negotiation between Agents

Figure 8 shows the exchange of messages between the CA and the NA at the *coordination level*, as well as the exchange of messages between the NAs at the *negotiation level*. It is a screenshot of the output of the Sniffer Agent which is provided by the JADE platform

for visualizing the message exchange between JADE agents.

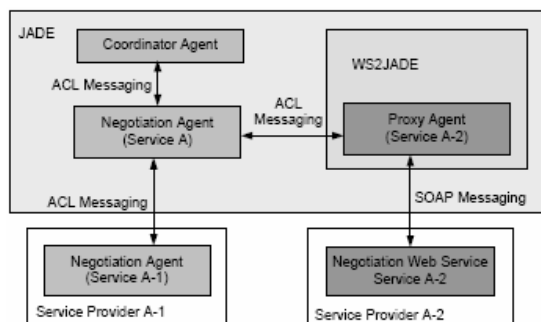


Figure 7: NA for Service Consumer negotiating with NA (for Service Provider A-1) and NWS (for Service Provider A-2) simultaneously

In Figure 8, the CA passes the list of providers and local QoS to the NA. This is indicated by the INFORM message at the start (A full list of the performatives exchanged between the agents is shown in Tables 1 and 2). This NA then issues a CFP to the list of providers it receives from the CA i.e. three in this case. Each provider responds with their own initial proposal which the NA collects. If none of the offers are satisfactory, it issues a modified CFP to all providers. If any provider does not respond within the proposal timeout, that

particular provider is dropped out of the negotiation process. This exchange of messages (proposal, counter-proposals) continues until a favorable outcome is achieved as indicated in the figure. The NA then returns this result back to the CA along with the selected Provider ID.

6.2. Negotiation Web Service

One of the primary reasons why we implemented the negotiation system as a Web service is to address the issue of interoperability among multiple systems. Most negotiation support systems have been implemented as pure MultiAgent systems. We want to demonstrate the flexibility of our negotiation system in which the NAs can not only negotiate with NAs on the service provider's side, but with Negotiation Web Services as well. This provides service providers with the flexibility to expose their negotiation capabilities either as web services which can be discovered, or as an agent-based system. The Negotiation Web Services (NWS) provide uniform interfaces for negotiation without disclosing the internal implementation. The underlying implementation could either be agent-based or non-agent based as long as it follows the ICNP. Like any other web service, a NWS can be published and discovered using standards such as UDDI and invoked using standards such as SOAP.

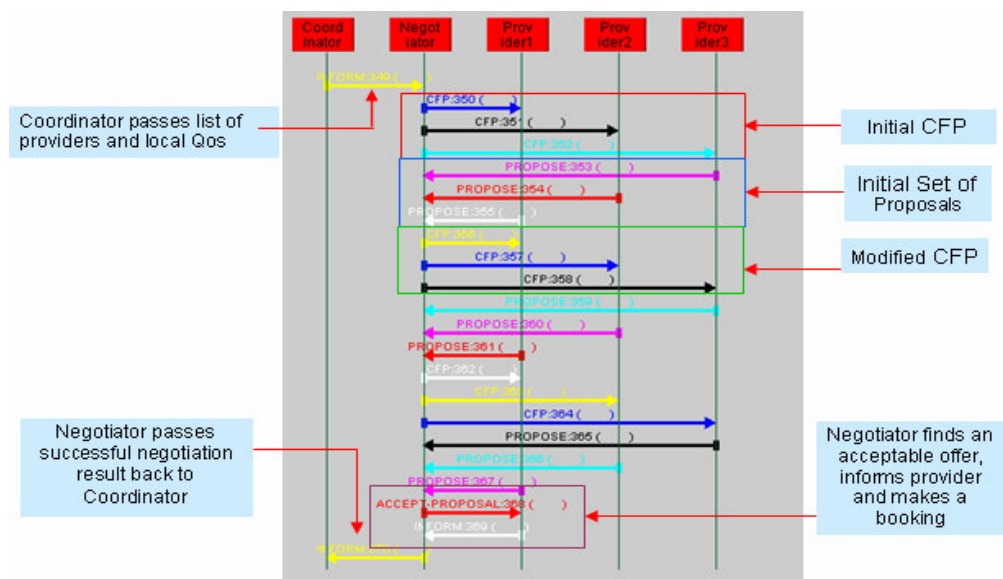


Figure 8: Sniffer output showing the message exchange between the CA and the NAs

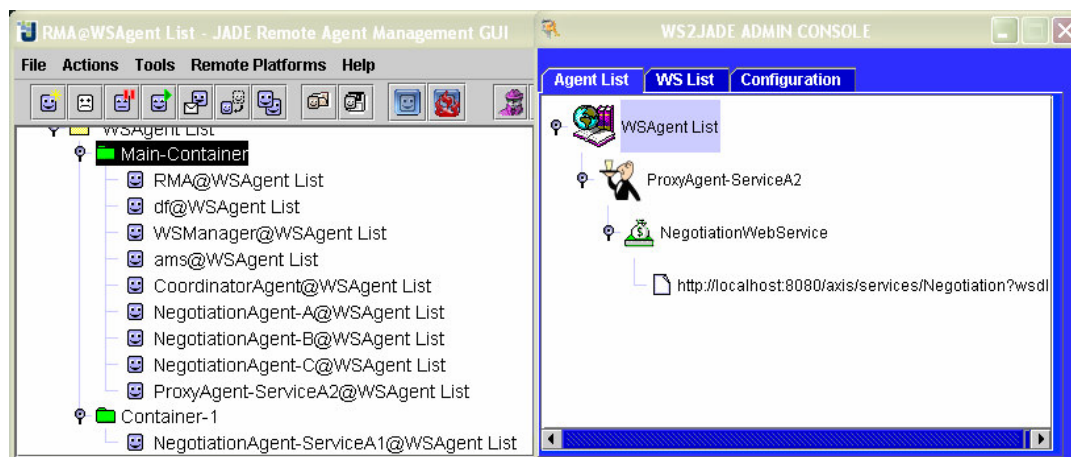


Figure 9: Screenshot showing the CA, NAs for Service types A, B and C as well as the Proxy Agent for the Negotiation Web Service offered by Service Provider A2.

7. Conclusion and Future Work

Service Oriented Computing (SOC) is emerging as a paradigm that utilizes services as fundamental elements for developing distributed applications. It promises organizations the ability to integrate their systems in a seamless manner by composing distributed business applications i.e. service compositions with minimal effort both within and across organizational boundaries. The challenge then, given a user specification, an abstract service composition and a set of component Web Services (for each service type within the composition), is to find the best combination of component Web Services which exactly match or is 'as close as possible' to the user specification.

In this paper, we have shown an agent based approach to address this challenge. We have presented coordinated-negotiation architecture for the negotiation of each individual service type within the composition while ensuring that the end-to-end QoS constraints of the composition are satisfied. We have presented a two-layered approach which enables the reuse of this architecture in any application domain. Moreover, in order to address the issue of interoperability, we have implemented the negotiation systems on the provider's side as a Web Service so that the ASAPM system can negotiate either with intelligent agents or with Negotiation Web Services.

We have used the *negotiate-all-then-enact* approach for the negotiation of the service composition as a starting point. We would like to formalize the protocol for the coordinated negotiation which involves two layers of interaction. This paper has not discussed the

decision making strategies both at the negotiation level and at the coordination level. Future research work will be focused in developing strong and complicated decision making mechanisms at both levels. A more complete prototype will be developed once we have addressed these issues, so that we can demonstrate the overall functionality of the ASAPM System in providing adaptive service agreement and adaptive service management by integrating the Negotiation System with the other components.

8. Acknowledgement

This work is partially supported by the Adaptive Service Agreement and Process Management (ASAPM) in Services Grid project (AU-DEST-CG060081) and the EU FP6 Integrated Project on Adaptive Services Grid (ASG) (EU-IST-004617). The ASAPM project is proudly supported by the Innovation Access Program – International Science and Technology established under the Australian Government's innovation statement, Backing Australia's Ability.

9. References

- [1] N. Milanovic, and M. Malek, "Current Solutions for Web Service Compositions", *IEEE Internet Computing*, 2004, pp. 51-59
- [2] J. Yang, M.P. Papazoglou, "Service Component for Managing Service Composition Life-Cycle", *Information Systems*, Elsevier, 2004, pp. 97-125

- [3] M. Matskin, P.Kungas, J. Rao, J. Sampson, S.A. Peterson. "Enabling Web Services Composition with Software Agents", in the proceedings of the *Ninth IASTED International Conference on Internet and Multimedia Systems and Applications, IMSA 2005*, Hawaii, USA, ACTA Press, 2005, pp. 93-98
- [4] Adaptive Services Grid Project, available online at <http://asg-platform.org/cgi-bin/twiki/view/Public>
- [5] P. Faratin, C. Sierra, and N.R. Jennings, "Negotiation decision functions for autonomous agents", *Robotics and Autonomous Systems*, 24 (3-4), 1998, pp. 159-182
- [6] G. Weiss, "Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence", MIT Press, 1999 pp. 619
- [7] Foster, N. R. Jennings, and C. Kesselmann, "Brain meets Brawn: Why Grid and agents need each other", in proceedings of *Third International Conference on Autonomous Agents and Multi-Agent Systems*, New York, USA, 2004
- [8] Adaptive Service Agreement and Process Management (ASAPM) Project, available online at <http://ciamas.it.swin.edu.au/tiki/tiki-index.php>
- [9] N.R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, C. Sierra and M. Wooldridge, "Automated Negotiation: Prospects, Methods and Challenges", *International Journal of Group Decision and Negotiation*, 2000
- [10] C. Bartolini, C. Preist, and N.R. Jennings, "A Software Framework for Automated Negotiation", *SELMAS 2004*, LNCS 3390, 2005, pp. 213-235
- [11] Foundation for Intelligent Physical Agents available online at <http://www.fipa.org/>
- [12] Java Agent Development Framework available online at <http://jade.tilab.com/>
- [13] Web Services to Agents: WS2JADE available online at <http://www.it.swin.edu.au/centres/ciamas/tikiindex.php?page=ws2jade-proj>
- [14] X. T. Nguyen, R. Kowalczyk, M. B. Chhetri, and A. Grant, "WS2JADE: A Tool for Run-time Deployment and Control of Web Services as JADE Agent Services", in *Software Agent-Based Applications, Platforms and Development Kits*. Whitestein Technologies AG, 2005
- [15] M. Calisti, M. Klusch and R. Unland (Eds.) "Software Agent-Based Applications, Platforms and Development Kits" Whitestein Technologies AG, 2005.
- [16] H. Kreger, "Fulfilling the Web Services Promise", in *Communication of the ACM*, 46(6), 2003, pp. 29-34.
- [17] A. Keller and H. Ludwig, "Defining and monitoring service level agreements for dynamic e-business", in *Proceedings of the 16th System Administration Conference (LISA 2002)*, November, 2002, Philadelphia, USA
- [18] C.K. Hung, H. Li, J. Jeng, "WS-Negotiation: An Overview of Research Issues", in the *Proceedings of the 37th Hawaii International Conference on System Sciences*, 2004
- [19] V. Tosic, B. Pagurek, K. Patel, "WSOL – A language for the Formal Specification of Classes of Services for Web Services", in the *Proceedings of ICWS 2003*, CSREA Press 2003. pp. 375-381.
- [20] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke and M. Xu, "Web Service Agreement Specification (WS-Agreement) 1.1", 2004, available online at <http://www.gridforum.org/Meetings/GGFI1/Documents/draft-ggf-graap-agreement.pdf>
- [21] K. Czajkowski, I. Foster, C. Kesselman, V. Sander, and S. Tuecke, "SNAP: A Protocol for Negotiating Service Level Agreements and Coordinating Resource Management" in *Distributed Systems, 8th Workshop on Job Scheduling Strategies for Parallel Processing*, Edinburgh, Scotland, July 2002."
- [22] L. Nassifi, J.M.Nogueira, M. Ahmed, R. Impey, A. Karmouch. "Agent-based Negotiation for Resource Allocation in Grid" in the *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, Utrecht, Netherlands, 2005
- [23] J. Brzostowski, R. Kowalczyk, "On Possibilistic Case-based Reasoning for Selecting Partners for Multi-attribute Agent Negotiation" in the *Proceedings of the 4th International Joint Conference on Autonomous Agents and MultiAgent Systems*, Utrecht, Netherlands, 2005
- [24] I. Rahwan, R. Kowalczyk, and H. Pham. "Intelligent Agents for Automated One-to-Many Ecommerce Negotiation" in *Proceedings of the 25th Australasian Computer Science Conference*, Melbourne, Australia, pages 197--204, 2002.
- [25] C.K. Hung, H. Lei, J. Jeng, "WS-Negotiation: An Overview of Research Issues" in the *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, Hawaii, USA, 2004
- [26] I. Mueller, R. Kowalczyk, "Service Composition through Agent-based Coalition Formation" in *Proceedings of the Workshop on WWW Service Composition with Semantic Web Services (WSCOMPS05) held in conjunction with the 2005 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Compiegne (France)*, September 2005. pp. 34 - 43.