Agent-based Negotiation of Service Level Agreements for Web Service Compositions

Mohan Baruwal Chhetri¹, Sukkeong Goh², Jian Lin³, Jakub Brzostowski⁴, Ryszard Kowalczyk⁵

¹ Swinburne University of Technology
Faculty of Information and Communication Technologies
John St, Hawthorn 3122
Australia
mchhetri@ict.swin.edu.au

² Swinburne University of Technology
Faculty of Information and Communication Technologies
John St, Hawthorn 3122
Australia
sgoh@ict.swin.edu.au

³ Swinburne University of Technology
Faculty of Information and Communication Technologies
John St, Hawthorn 3122
Australia
jlin@ict.swin.edu.au

⁴ Swinburne University of Technology
Faculty of Information and Communication Technologies
John St, Hawthorn 3122
Australia
jbrzostowski@ict.swin.edu.au

⁵ Swinburne University of Technology
Faculty of Information and Communication Technologies
John St, Hawthorn 3122
Australia
rkowalczyk@ict.swin.edu.au

Abstract. This paper presents an agent-based negotiation framework to support the autonomous establishment and maintenance of Service Level Agreements for QoS aware web service compositions. It uses a two-layered architecture for the coordinated negotiation of QoS constraints, in order to ensure collective functionality, end-to-end QoS and the stateful coordination of complex web service compositions. It forms part of the adaptive
service composition management framework which is being developed by the ASAPM Project. We present a prototype implementation of the framework. It is built on top of the JADE agent framework and also uses the WS2JADE plug-in which enables the integration of web services and JADE agents.

**Keywords:** multiagent systems, composite web services, automated negotiation

### 1 Introduction

Recent advances in Web Services technology and the emergence of the Service Oriented Computing (SOC) paradigm is facilitating an environment in which service consumers and service providers can dynamically locate one another, automatically set the terms and conditions of invocation and then execute the necessary actions according to the negotiated Service Level Agreements. If the user request cannot be satisfied by a single web service, then it is/should be possible to combine existing services in a loosely coupled manner such that collectively they can fulfil it.

Service composition is the process of combining several component services and binding them together to meet the specific needs of the user request. The provision of a service composition implies the consumption of a set of services that are dynamically selected from a pool of potential candidate services offered by different providers. The consumer need not be aware of the actual composition (of several heterogeneous Web Services) as long as the functional and non-functional requirements are satisfied. The consumption of these services is commonly governed by an agreement, known as the Service Level Agreement (SLA) which covers the guarantees of a service provision (Keller and Ludwig 2002). A SLA is a contractual obligation between the service provider and the service consumer, which specifies mutually-agreed understandings and expectations (for both functional and non-functional attributes) on the procurement of the service.

Using negotiation as a means of establishing service contracts has been a topic of considerable interest in both the agent community (Faratin et. al. 1998; Brzostowski and Kowalczyk 2005) as well as the web services (Chhetri et. al. 2006) and grid services community (Nassifi et. al. 2005). Agent technology offers the abilities of autonomous operations, learning, adaptation, intelligent and adaptive decision-making, and cooperation and coordination in highly dynamic environments. Hence it is an accepted mechanism in automated negotiations and has been recognised as a promising technology for SLA Management (Jennings 2002; Nguyen and Kowalczyk 2006). Agents act as flexible problem solvers which can make context-aware and context-dependent decisions about the scope and nature of their interactions. Given the highly dynamic and volatile nature of the web service environment, SOC can benefit immensely from multiagent technology.

SLA Management has been the subject of intense research for several years. The research has focussed on different aspects of SLA Management such as SLA specification and languages and SLA lifecycle management (creation, operation, monitoring, termination and exception handling). OASIS has put forward several standards including Web Service Business Process Language (WS-BPEL), Web Service Distributed Management (WSDM), Web Service Quality Description Language (WS-QDL) and Web Service Quality Model (WSQM). Similarly IBM has defined the Web Service Level Agreement WSLA language (Keller and Ludwig 2003). GRAAP working group of the Global Grid Forum has put forward the WS-Agreement specification. It is also working on a WS-AgreementNegotiation specification. Similarly, the ESSI WSMO Working Group (Toma, I., and

---

Foxvog, D. (2006) has provided support for specifying the non-functional properties for Web Services in Web Service Modelling Ontology (WSMO) and Web Services Modelling Language (WSML).

At the same time, significant research has been carried out in developing new mechanisms, models and frameworks for the adaptive QoS Management of Web Services. Lin et. al. (2004) proposed a method for selecting optimal sub-providers from a list of potential providers. Comuzzi et. al. (2005) focussed on the negotiation of web service QoS parameters with the ability to use different negotiation strategies. Similarly, Gu et. al. (2003) proposed an adaptive and dynamic approach to manage composite services by focussing on their functional aspects such as dynamic service discovery and ad-hoc changes as well as their non-functional aspects such as Quality of Service (QoS) requirements. Hence innovative research is being extensively conducted in the area of adaptive SLA Management.

However, one major limitation of most research into SLA management is that it is generally restricted to the provisioning of single services. In the case of web service compositions, the management relies upon manual configuration to determine the QoS constraints on individual services in the service composition. Such systems are inflexible, rigid and not fully automated. There is a need for a SLA Management framework which supports the fulfilment of end-to-end QoS requirements for the service composition through the autonomous and coordinated SLA negotiation and contracting. Similarly within the agent community, research into automated negotiation has been in the context of one-to-one negotiations and one-to-many negotiations for the provision of single services. There hasn’t been much work done in the negotiation of end-to-end QoS constraints for a composite service.

Thus there is a recognized need for a framework which can support the autonomous negotiation of QoS constraints for web service compositions. This paper presents innovative research in the agent-based coordinated negotiation of Service Level Agreements for QoS aware web service compositions by bringing together two complementary technologies - software agents and web services. This paper is organised as follows. The next section briefly describes the Adaptive Service Agreement and Process Management (ASAPM) project, in the context of which this research has been carried out. Section 3 discusses the adaptive Negotiation Framework that we have developed. Section 4 describes our implementation of the ASAPM Negotiation Framework using the Java Agent Development (JADE) Framework (Bellifemine et.al. 1999). Section 5 concludes the paper.

2 Background

The R&D work presented in this paper is carried out in the context of the Adaptive Service Agreement and Process Management (ASAPM) in Services Grid Project (AU-DEST-CG060081) (Wu et. al. 2006). This project aims at developing intelligent agent-based techniques and tools to facilitate the adaptive service management and process management in order to ensure the 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 SLA negotiation and renegotiation, SLA lifecycle management and dynamic service profiling (DSP) while the latter includes process enactment, SLA monitoring, process enactment visualization and mediation for renegotiation and replanning.

Figure 1 shows the reference architecture of ASAPM. It sits between the Application Domain Expert System (DES) and the actual web service infrastructure. The DES is responsible for planning the web service composition in order to satisfy the user request that it receives. ASAPM can either be embedded in the Domain Expert System or used as a stand alone system by the DES. It focuses on providing adaptive, transparent provision of complex services based upon the user request. It consists of the following four components.

- **Negotiation Subsystem** – responsible for the autonomous SLA negotiation and renegotiation of composite web services.
Agent based Negotiation of SLAs for Web Service Compositions

- **SLA Management & Monitoring Subsystem** – responsible for the management of the SLA during its lifetime. It also monitors the actual QoS during enactment and triggers mediation if there is a service failure or QoS violation.
- **Workflow Subsystem** – responsible for the enactment of the service composition, and
- **Mediation Subsystem** – responsible for mediation, either for:
  - renegotiation with contracted provider
  - renegotiation and replacement by new providers, or
  - replanning for generation of new plan to satisfy original user request

![ASAPM Reference Architecture](image)

Figure 1. ASAPM Reference Architecture

The DES provides ASAPM with the abstract service composition as well as the global QoS requirements for SLA negotiation. The Negotiation subsystem receives the list of potential services for each atomic service type within the composition from the DES. It is then the task of the Negotiation subsystem to find the suitable candidates for each service type while ensuring that the end-to-end QoS is satisfied.

3 ASAPM Negotiation Framework

A web service composition consists of several atomic services which collectively satisfy the user request. Each atomic service can be offered by several service providers with varying levels of QoS. From a functional viewpoint the SLA negotiation of a service composition involves two aspects. One is the atomic negotiation between the service consumer (or its representative e.g. ASAPM) and the service providers for every atomic service within the composition. The second 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 – *global coordination* and *atomic negotiation*.

When building automated negotiation systems we need to consider three main areas – *negotiation protocol*, *negotiation attributes* and *negotiation strategies* (Mueller, 1996). The negotiation protocol defines the mechanism and rules of interaction that the negotiating parties need to follow. The negotiation attributes represent the range of issues over which negotiation takes place and an agreement has to be reached. The third area of significance in automated negotiations is the negotiation strategy or decision making model used.
Any software framework consists of frozen spots and hot spots. Frozen spots define the overall architecture of a software system, i.e., the basic components of the system and the relationships between them. These remain unchanged for any instantiation of the application framework. The hot spots represent the parts which can be modified by users of the framework to add code specific to their own application domain or business requirements. For the ASAPM Negotiation Framework the overall architecture of the negotiation system is fixed, i.e., it consists of a two-layered architecture consisting of a Coordinator Agent (CA) and several Atomic Negotiator Agents (NAs). Similarly, the global coordination protocol (which defines the interaction between the coordinator agent and the various negotiator agents) and the atomic negotiation protocol (which defines the interaction between the negotiator agents representing the client and the service providers) are predefined. However, the decision-making strategies which include the strategies for the end-to-end coordination and the atomic negotiations can be modified by the users of the Negotiation Framework. The negotiation attributes also vary depending upon the application domain and the user request.

3.1 System Architecture

Figure 2 shows the two-layered architecture we have proposed for our negotiation system. We have a CA which coordinates the negotiation of the whole composition so that the end-to-end QoS requirements are satisfied. At the atomic level, we have the NAs which conduct one-to-many negotiations with the service providers. They try to achieve the best negotiation outcome for the provided local QoS constraints.

![Figure 2. Coordinated Negotiation Architecture](image)

The CA receives the abstract service composition definition as well as the end-to-end QoS constraints from the DES. The CA then analyses the abstract composition and decomposes the end-to-end QoS constraints into atomic QoS constraints. A list of potential providers for each atomic service is obtained from the Service Discovery Component of the DES. The primary phase of coordinated negotiation involves analysing the service composition and the non-functional requirements. The result of this analysis phase is the creation of a cluster of NAs corresponding to each atomic service within the composition. Each NA receives the local QoS constraints for the atomic service they are trying to procure along with a list of potential providers. For example, in Figure 2, NA_A (negotiating...
Agent based Negotiation of SLAs for Web Service Compositions

for atomic service A) will negotiate with three providers (Provider A₁, Provider A₂, and Provider A₃). Each NA is required to negotiate the most favourable outcome within the stipulated time frame at the end of which it reports its results to the CA. When the CA receives results from all the NAs, it aggregates the atomic negotiation results to see if the end-to-end QoS constraints are satisfied. If the end-to-end QoS constraints are satisfied, then the next step is formation of the SLAs with the selected providers. However, if the negotiation of the service composition is not satisfied the CA has two options. It can either redistribute the adjusted reservation values to carry out the atomic negotiations again, or it can report the negotiation failure to the client (through the DES).

3.2 Agent Architecture

Figure 3 shows the Coordinator/Negotiator Agent architecture. The CA and the NAs have a set of behaviours associated with them. These behaviours reflect the tasks to be performed by the agents. These behaviours are instantiated by the FSM Engine during runtime based upon a behaviour script which specifies the different states of the agent and the events which trigger these states. Similarly the CA and the NAs share a common communication ontology which contains the shared conceptualizations about negotiation. The agents have to follow the Coordinated Negotiation Protocol which defines the rules for interaction between the CA and the NAs as well as between the NAs and the service providers. The negotiation capabilities on the service provider’s side could be implemented in several ways, as long as they follow the negotiation protocol and the negotiation semantics.

![Figure 3. Coordinator/Negotiator Agent Architecture](image)

The coordination and negotiation strategies have been developed as separate libraries which can be plugged into the framework. Depending upon the semantics of the user request, the agent can choose the best coordination strategy or negotiation strategy from a pool of available strategies. We have tried to make the negotiation system flexible by using a component based development framework to support reuse and extension.

3.3 Coordinated Negotiation Protocol

In this research, the negotiation coordination is supported by extending the FIPA Iterated Contract Net Interaction Protocol (ICNIP). ICNIP supports recursive negotiation and allows for multi-round iterative negotiation to find an outcome. As shown in Figure 4, the extension to the ICNIP is the step of triggering negotiation and renegotiation, and negotiation confirmation. The CA triggers the NA to
start the SLA negotiation by sending it the reserve values defining the negotiation attributes and objectives. The NA then negotiates with many providers using the standard FIPA ICNIP, followed by the confirmation of negotiation or the modification of reserve values by the CA.

![Coordinated Negotiation Protocol](image)

**Figure 4. Coordinated Negotiation Protocol**

Table 1 shows the ACL Messages exchanged between the CA and the NA while Table 2 shows the messages exchanged between the NAs and the service provider.

### Table 1. List of Performatives exchanged between CA and NA

<table>
<thead>
<tr>
<th>ACL Performative</th>
<th>ACL Message Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUEST (CA → NA)</td>
<td>Reserve values, Negotiation timeout and Proposal timeout</td>
</tr>
<tr>
<td>INFORM (NA → CA)</td>
<td>Negotiation results and Selected Provider details</td>
</tr>
<tr>
<td>CONFIRM (CA → NA)</td>
<td>Confirmation that selected provider is to be contracted</td>
</tr>
</tbody>
</table>

### Table 2. List of Performatives exchanged between NA and Service Provider

<table>
<thead>
<tr>
<th>ACL Performative</th>
<th>ACL Message Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFP (NA → SP)</td>
<td>Initial Call for Proposal</td>
</tr>
<tr>
<td>PROPOSE (NA ← SP)</td>
<td>Offer made by Provider</td>
</tr>
<tr>
<td>REJECT (NA ← SP)</td>
<td>Message sent by Provider if not interested in negotiating</td>
</tr>
<tr>
<td>NOT_UNDERSTOOD (NA ← SP)</td>
<td>Message sent by either party if message content is not understood</td>
</tr>
<tr>
<td>ACCEPT_PROPOSAL (NA → SP)</td>
<td>Message sent to Provider once Proposal has been accepted by Requestor</td>
</tr>
<tr>
<td>REJECT_PROPOSAL (NA → SP)</td>
<td>Message sent if Proposal not acceptable</td>
</tr>
<tr>
<td>INFORM (SP → NA)</td>
<td>Message sent by Provider when negotiation result is accepted by NA</td>
</tr>
<tr>
<td>FAILURE (SP → NA)</td>
<td>Message sent by Provider if some error occurs after it receives ACCEPT_PROPOSAL</td>
</tr>
<tr>
<td>CANCEL (NA → SP)</td>
<td>Message sent by either party when cancelling/terminating negotiation</td>
</tr>
</tbody>
</table>
4 Implementation of the Framework

We have implemented the Negotiation Framework using the FIPA compliant JADE Agent Framework (Bellifemine et. al. 1999) and WS2JADE (Nguyen et. al. 2005) which enables the integration of JADE agents and web services. The main abstractions in JADE are agents and behaviours. JADE uses the Behaviour abstraction to model the tasks that an agent needs to perform. JADE agents communicate using the FIPA Agent Communication Language (ACL). JADE also provides a library of interaction protocols and generic agent behaviours which we have used as the basis of our implementation. Since the CA and the NAs are supported by JADE, we do not have to redefine the basic services for agent creation, communication, lifecycle management and so on. We only need to embed the decision making and coordination mechanism inside the agents to facilitate adaptive negotiation.

4.1 Agent and Behaviours

We have defined a set of basic behaviours for the CA and the NA representing the different tasks and roles that these agents take on during the process of negotiation. One of the key requirements of adaptive multi-agent systems is the ability to dynamically configure and reconfigure agent behaviours in response to events or changes in the environment. There is a need to be able to control the precise order of behaviour invocation. In order to achieve this, we have developed a Finite State Machine (FSM) Engine on top of the JADE platform, increasing the flexibility and adaptability of the Negotiation System. The FSM Engine also enables loose coupling between the agents and behaviours i.e. behaviours aren’t bound to any particular agent. Table 3 shows some of the behaviours associated with the CA and Table 4 shows the behaviours associated with the NA.

<table>
<thead>
<tr>
<th>Coordinator Agent Behaviour</th>
<th>Associated Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReceiveRequestBehaviour</td>
<td>Receive abstract service composition + Global QoS requirements</td>
</tr>
<tr>
<td>DistributeQoSBehaviour</td>
<td>Perform QoS distribution</td>
</tr>
<tr>
<td>SetupNegotiatorsBehaviour</td>
<td>Setup the Negotiator Agents</td>
</tr>
<tr>
<td>CollectAtomicProposalsBehaviour</td>
<td>Collect the Atomic Proposals from all Negotiator Agents</td>
</tr>
<tr>
<td>CreateContractBehaviour</td>
<td>Create Contract with the client</td>
</tr>
<tr>
<td>FinaliseBehaviour</td>
<td>Perform any cleanup at end of session</td>
</tr>
<tr>
<td>ExceptionHandlingBehaviour</td>
<td>Handle any exceptions</td>
</tr>
<tr>
<td>UserConfirmationBehaviour</td>
<td>Get the user confirmation for the Negotiation results</td>
</tr>
</tbody>
</table>

4.2 Decision Strategies

The decision making strategies have been implemented as a separate library which can be used by the agents. This library can easily be extended to include the new strategies as they become available. It consists of two sets of strategies. One set includes strategies for the QoS distribution, QoS aggregation and the overall coordination of the negotiation. This is used by the CA to ensure that the end-to-end QoS requirements are satisfied. Similarly, the second set of strategies support decision-making in atomic one-to-many negotiations which are conducted between the NA and the service providers. Information on the strategies we have implemented is available in (Chhetri et. al. 2006; Brzostowski and Kowalczyk, 2006a; Brzostowski and Kowalczyk, 2006b).
Table 4. Negotiator Agent Behaviours

<table>
<thead>
<tr>
<th>Negotiator Agent Behaviours</th>
<th>Associated task</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAtomicRequestBehaviour</td>
<td>Receive the local QoS + list of providers</td>
</tr>
<tr>
<td>BroadcastCfpBehaviour</td>
<td>Broadcast CFP to all service providers</td>
</tr>
<tr>
<td>CollectRepliesBehaviour</td>
<td>Collect offers from all service providers</td>
</tr>
<tr>
<td>AnalyseProposalsBehaviour</td>
<td>Analyse the offers to see if they satisfy the QoS</td>
</tr>
<tr>
<td></td>
<td>requirements</td>
</tr>
<tr>
<td>ExceptionHandlingBehaviour</td>
<td>Handle any exceptions</td>
</tr>
<tr>
<td>GetConfirmBehaviour</td>
<td>Get notification from CA about success or failure of</td>
</tr>
<tr>
<td></td>
<td>atomic negotiation</td>
</tr>
<tr>
<td>ContractBehaviour</td>
<td>Confirm final negotiation result with selected provider</td>
</tr>
</tbody>
</table>

4.3 Ontology

We have developed a simple, custom defined ontology for the coordinated negotiation. The Negotiation Ontology which is a subset of the ASAPM Ontology has been built on top of the JADE ontology. It is defined as a singleton object which is shared among CA and the NAs. In its first version, the ontology specifies the interactions that may occur between the CA and the NAs. It mainly consists of agent actions extending the `jade.content.AgentAction` which indicate actions performed by agents. These actions form the meaningful content of ACL REQUEST messages. On executing the actions, the agents which receive the REQUEST message return a INFORM message which contains the result of the performed action. The content of these INFORM messages is a notification message which extends `jade.content.Concept`.

Table 5. Negotiation Ontology

<table>
<thead>
<tr>
<th>Ontology Name : asapm.negotiation-ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concepts:</td>
</tr>
<tr>
<td>NegotiateComposition-Result</td>
</tr>
<tr>
<td>NegotiateService-Result</td>
</tr>
<tr>
<td>Negotiation-Object</td>
</tr>
<tr>
<td>Negotiation-Offer</td>
</tr>
<tr>
<td>Agent Actions:</td>
</tr>
<tr>
<td>NegotiateComposition-Request</td>
</tr>
<tr>
<td>RenegotiateComposition-Request</td>
</tr>
<tr>
<td>NegotiateService-Request</td>
</tr>
<tr>
<td>RenegotiateService-Request</td>
</tr>
<tr>
<td>CancelNegotiation-Request</td>
</tr>
<tr>
<td>ResumeNegotiation-Request</td>
</tr>
</tbody>
</table>

4.4 Proxy Agents

Service providers can expose their negotiation capabilities either as an agent-based system or as web services which can then be registered in and retrieved from a service registry. In the case of the former, the atomic negotiation is pretty straightforward since it takes place between NAs representing the user and the service provider. In the case of the latter, the NAs inside the ASAPM Negotiation
system negotiate with the Negotiation Web Service (NWS) through a Negotiation Proxy Agent (NPA). Using WS2JADE, the NA invokes the NWS through the NPA as shown in Figure 5. Communication between the NA and the NPA takes place using ACL Messaging and between the NPA and the NWS using SOAP messages over HTTP. More details about how web service invocation takes place through WS2JADE are provided in (Nguyen et. al 2005).

Figure 5. Automatic Negotiation between NA and NWS

5 Use case Scenario

This section describes the Dynamic Supply Chain Scenario for Internet services provisioning (DSC scenario) which we have chosen to prototype the automated negotiation framework. The DSC scenario describes a system for the automated Domain Name registration and provisioning of Webspace. A sample request by the client may look like ‘Provide a service for Domain Name registration, Webspace provision, payment etc. whereas the service needs to be available in less than 70 time units (TU) and costs for setting up this services must be below 15 monetary units (MU)’. Based upon this user request, the DES composes an abstract service composition that can fulfil the user request. Figure 6 shows one possible service composition. All the atomic services can be provided by several candidate service providers. We assume that all service providers are able to adjust their QoS parameters (execution time) and cost.

The Negotiation subsystem of ASAPM receives the abstract service composition as well as the end-to-end QoS requirements from the DES. It then queries the Service Discovery component of the DES for the list of prospective providers for each atomic service within the composition. Figure 6 shows that atomic service CheckDomain is offered by planetDomain, UnitedDomains and DomainPro. Similarly the other services are also provided by several providers. Since the abstract service composition is just the logical composition of the necessary service specifications, the task of the Negotiation subsystem is to select concrete service providers from a potentially high number of providers through negotiations. ASAPM negotiates with different service providers in order to find the most suitable configuration of QoS parameter values for all included DSC services.

Figure 7 shows one possible solution found at the end of the successful coordinated negotiation. In this instance, the coordinated negotiation was successful in finding providers for each atomic service within the composition. The aggregated QoS satisfies the end-to-end QoS requirements of the user request. Once service providers have been selected for each atomic service within the composition, the next task is to contract the selected service providers, provided that the end-to-end QoS requirements are met. Once the SLAs have been created, the next step in the service composition lifecycle is the enactment of the concrete service composition with the coordinated monitoring of the service composition to ensure that the contracted QoS requirements are complied with.
Agreement into our framework. Complicated decision making strategies at both the coordination level and the atomic negotiation into coordinated negotiation of SLAs. Future work will be focused on developing stronger and more attribute and multi-attribute negotiations. We have built a simple negotiation ontology which defines significant research has been carried out in the field of autonomous negotiation, there is a clear lack the basic communication acts involved in the negotiation process.

Figure 6. Abstract Service Composition with e-2-e QoS Requirements (Before Negotiation)

Figure 7. Concrete Service Composition with e-2-e QoS Requirements satisfied (After Negotiation)

6 Conclusion

Current approaches to QoS Management lack mechanisms to autonomously and dynamically determine and adjust QoS values for individual services in a web service composition. While significant research has been carried out in the field of autonomous negotiation, there is a clear lack of mechanisms to support the coordination of such atomic negotiations such that the end-to-end QoS constraints for the composite service are satisfied.

In this paper, we have presented an agent-based Negotiation framework to support the coordinated negotiation of Service Level Agreements for QoS aware service compositions. Our negotiation framework supports several negotiation and coordination strategies. It can be applied to single attribute and multi-attribute negotiations. We have built a simple negotiation ontology which defines the basic communication acts involved in the negotiation process.

The Negotiation framework presented in this paper builds a solid foundation for future research into coordinated negotiation of SLAs. Future work will be focused on developing stronger and more complicated decision making strategies at both the coordination level and the atomic negotiation level. The Negotiation Ontology will also be extended to cover a wider range of communication acts. We will also look into integrating the WS-standards of WS-Agreement and WS-AgreementNegotiation into our framework.
Acknowledgements

This work is partially supported by the Adaptive Service Agreement and Project Management (ASAPM) Project. 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.

References


*The Joint conference of the INFORMS section on Group Decision and Negotiation and the EURO Working Group on Decision Support Systems*, Karlsruhe Germany, 2006.