

2007

## **An agent-based framework for service level agreement management**

Qiang He

*Swinburne University of Technology*

Jun Yan

*University of Wollongong, jyan@uow.edu.au*

Ryszard Kowalczyk

*Swinburne University of Technology*

Hai Jin

*Huazhong University of Science and Technology*

Yun Yang

*Swinburne University of Technology*

Follow this and additional works at: <https://ro.uow.edu.au/infopapers>



Part of the [Physical Sciences and Mathematics Commons](#)

---

### **Recommended Citation**

He, Qiang; Yan, Jun; Kowalczyk, Ryszard; Jin, Hai; and Yang, Yun: An agent-based framework for service level agreement management 2007.

<https://ro.uow.edu.au/infopapers/2992>

---

## **An agent-based framework for service level agreement management**

### **Abstract**

In the Web services environment, service level agreements (SLA) refer to mutually agreed understandings and expectations of service provision between service consumers and providers. Although management of SLA is critical to wide adoption of Web services technologies in the real world, support for it is very limited nowadays. There lacks adequate frameworks and technologies supporting various SLA operations. This paper presents an agent-based framework which utilises the agents' ability of negotiation, interaction, and cooperation to facilitate autonomous and flexible SLA management. Based on this framework, mechanisms for autonomous SLA formation, recovery, and profiling are proposed and discussed.

### **Disciplines**

Physical Sciences and Mathematics

### **Publication Details**

He, Q., Yan, J., Kowalczyk, R., Jin, H. & Yang, Y. (2007). An agent-based framework for service level agreement management. The 11th International Conference on CSCW in Design (pp. 412-417). USA: IEEE.

## An Agent-based Framework for Service Level Agreement Management

Qiang He<sup>1,3</sup>, Jun Yan<sup>2</sup>, Ryszard Kowalczyk<sup>1</sup>, Hai Jin<sup>3</sup>, Yun Yang<sup>1</sup>

<sup>1</sup>*Faculty of Information and Communication Technologies, Swinburne University of Technology, Australia*

*rkowalczyk@ict.swin.edu.au, yyang@ict.swin.edu.au*

<sup>2</sup>*School of Information Systems and Technology, University of Wollongong, Australia*  
*jyan@uow.edu.au*

<sup>3</sup>*School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China*

*heqiang@gmail.com, hjin@hust.edu.cn*

### Abstract

*In the Web services environment, Service Level Agreements (SLA) refer to mutually agreed understandings and expectations of service provision between service consumers and providers. Although management of SLA is critical to wide adoption of Web services technologies in the real world, support for it is very limited nowadays. There lacks adequate frameworks and technologies supporting various SLA operations. This paper presents an agent-based framework which utilises the agents' ability of negotiation, interaction, and cooperation to facilitate autonomous and flexible SLA management. Based on this framework, mechanisms for autonomous SLA formation, recovery, and profiling are proposed and discussed.*

**Keywords:** Agents, Service Level Agreement, Web Services

### 1. Introduction

With the popularity of Service-Oriented Computing (SOC), research on Service Level Agreement (SLA) becomes a hot topic. SLA refers to the contractual obligations between a service consumer and a service provider, representing guarantees of quality of service (QoS), priority of the service consumer and promises of the service provider. A SLA usually contains three parts:

1. parties involved, including contracted parties, and supporting third parties, such as monitoring parties, auditing parties, etc;
2. a set of service level guarantees, which define the promised metrics of the services, such as response time and availability, and sometimes the penalty and award for the service providers according to their behaviours in the whole process; and
3. interactions among the involved parties in case of deviation and failure of the guarantees.

SLA can be used by potential service consumer to choose the service providers from which its requirements of the service would be fulfilled. At the same time, by introducing SLA-guaranteed system, matured service providers could consolidate their big clients, and meanwhile, newly-rising service providers could attract potential undeveloped clients.

In the sophisticated Web services environment, covering Web services with SLA is challenging since the relationships among clients and service providers become more complex. Web services are designed to provide transparent automated utilisation of heterogeneous resources and applications. Minimised human intervention is desirable in all the functioning of Web services, which is also true for SLA management. The automated formation of SLA requires precise and unambiguous definition of the agreement as well as customisable engines to support automated negotiation on the details of the agreement for both contracted parties. Also, monitoring SLA, including state measuring and compliance verification, is supposed to be automated as the Web services would potentially be in trouble or down because of the inherent unreliability of the underlying Internet and internal infrastructure of service providers. To ensure that the service provider is adherent to its promised service level guarantees, both client side measurement and provider side measurement of the SLA are needed. Alternatively, commissioning the monitoring to an authorised supporting third party is recommended by this paper to figure out the problem of asymmetry measurement. Furthermore, besides formation and monitoring of the SLA, some additional operations are also desirable, such as automatic SLA recovery in case of failure and resumption of Web service provisioning, and credible SLA profiling in reputation evaluation of the services providers.

To address some of these issues, this paper proposes a novel SLA management framework which integrates agent and Web services technologies to support autonomous management of SLA. Based on this framework, mechanisms for operations such as SLA formation, recovery, and profiling are also discussed.

The remainder of the paper is organised as follows. Major related work is introduced in Section 2. Section 3 presents the details of the proposed framework, followed by discussion of the operational mechanisms in Section 4. Finally, Section 5 concludes the paper and outlines authors' future work.

## 2. Related Work

Over the past years, SLA-related research has attracted growing attention from both industry and academia. Significant effort has been placed on specification of SLA and a number of approaches have been released. WS-Policy [11] defines a model to express the static properties of Web Service entities, such as security, privacy and authentication details, as policies, enabling service consumers to select Web services more meaningfully. WSPL [1] implements more complicated operations on policies and a rich set of comparison operators rather than just simple equality matching. Merging of mutually accepted policies enables simple match schema for service consumers and service providers. WSLA [6] [8] adopts the template mechanism as the carrier for parties to rapidly and conveniently establish an agreement. It also provides a set of standard extensions that allow users to define agreement with guarantees for common metrics, e.g. response time, delay and throughput. WS-Agreement [2] aims at standardising the terminology, concepts, and the overall agreement structure. It provides more expressive language than WSLA to support the truly complex nature of the relationship between a service consumer and a service provider. WS-Agreement specification is designed to be independent on negotiation models and management models so that it can be composed with various negotiation protocols.

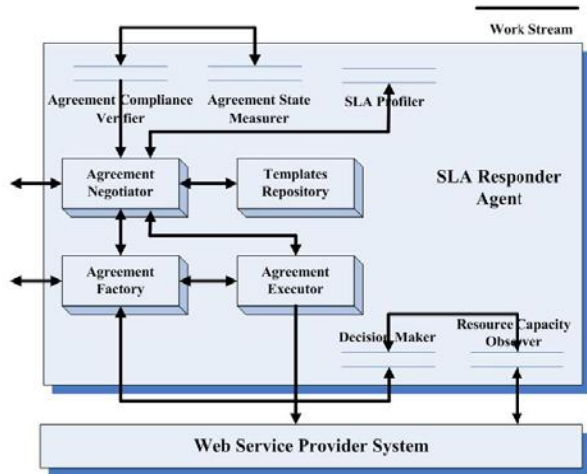
Apart from SLA specification, research on SLA management has been carried out. Focusing on standardising the representation of the SLA, IBM proposes Cremona [7], which is a WS-Agreement based middleware that defines mechanisms to implement interactions between organisational domains. A simple two-step request-response schema is provided for SLA negotiation in Cremona. HP devotes to presenting the requirements of precision and flexibility for SLA specification [5] and analysing the SLA monitoring model from the XML-specific aspect [10]. HP proposes automated and distributed SLA monitoring engine that considers both provider side and client side measurement of SLA and deals with the scenario when Web service providers work and contract with each other to fulfil the customer's request. [9] proposes a SLA negotiation protocol based on the Iterated Contract Net Protocol for job scheduling on the Grid, which divide the SLA negotiation into two levels, meta-SLA negotiation and sub-SLA negotiation. [4] addresses applying provider policy and admission control of service requests onto the underlying network infrastructure via SLA negotiated by agents to facilitate

adaptive coordination customers and service providers in telecommunications domain. [3] presents an agent-based coordinated-negotiation architecture for the SLA negotiations in Web Service composition with end-to-end QoS constraints. These approaches have made contributions to the SLA management research. However, it is clear that comprehensive SLA management over the lifetime of SLA, including SLA formation, monitoring, recovery, and profiling, is still at its infancy.

## 3. Agent-based SLA Management Framework

The framework proposed in this paper is a reusable "primary-complete" architecture that can be specialised to produce customised applications. This framework identifies and represents major functionalities which would be commonly observed in various application domains. One of the innovations of this framework is the adoption of the agent technology in managing SLA. Web services are designed to follow a loosely-coupled integration model to facilitate the seamless integration of heterogeneous systems within and across organisations. There are certain limitations potentially hindering the broader adoption of the Web services technology in sophisticated applications, particularly in Web services composition scenarios in which multiple Web services are required to work together to achieve a goal. It is believed that the agent technology is promising in the area of SLA management in Web services environment, as it offers the complementary abilities to the Web Services technology, such as intelligent operation, interaction, negotiation, and cooperation, which support autonomous operations such as SLA negotiation, recovery, profiling, and so on.

In the proposed SLA management framework and the corresponding operational mechanisms, entities play two distinct roles, i.e., the SLA initiator and the SLA responder. A SLA initiator is the SLA requestor which initialises the SLA formation process when needed. In most cases, the SLA initiator would be the service consumer which handles the details of the service request. In other cases, the service consumer delegates the request to a third party which subsequently becomes the service initiator. For example, in sophisticated and complex scenarios where a composition manager is needed to consolidate multiple Web services to fulfil a request, the composition manager acts as a SLA initiator, on the service consumer's behalf. A SLA responder is normally a service provider which could announce its service level capability as an advertisement in a service registry such as UDDI for service consumers to discover. In general, both the SLA initiator and responder are agents which consist of respective components. SLA operations such as formation, recovery, and profiling can be fulfilled autonomously through interactions between these agents.



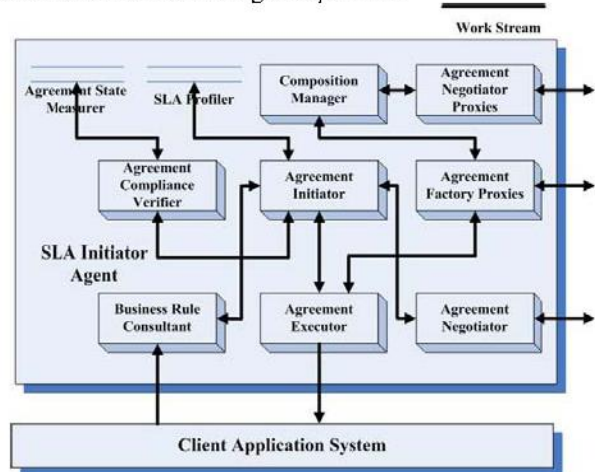
**Figure 1: Architecture of SLA responder agent**

A SLA responder normally interfaces with other parties, for example, the Agreement State Measurer interface, the Agreement Compliance Verifier interface and the SLA Profiler interface of supporting third parties, and the Decision Maker interface and the Resource Capacity Observer interface of the Web service provider system. As shown in Figure 1, the SLA responder agent comprises the following components:

- The Template Repository stores the preset templates for services at different promised levels.
- The Agreement Negotiator negotiates the detailed parameters with the SLA initiator's Agreement Negotiator.
- The Agreement Factory is a domain-independent component which processes the agreement with static and dynamic information, such as party information and agreed SLA parameters, provided by the Template Repository and the Agreement Negotiator respectively.
- The Agreement Executor announces the Web service provider system to enforce the agreement.
- The Resource Capacity Observer interface verifies the resource capacity and reports to the Decision Maker.
- The Decision Maker interface decides whether a SLA request could be fulfilled based on verifying current resource capacity information provided by the Resource Capacity Observer.
- The Agreement State Measurer interface could be implemented by the SLA initiator itself domain specifically, or by supporting third parties to measure and compute the runtime state of the Web service provisioning according to the agreement.
- The Agreement Compliance Verifier interface has two potential implementers, the SLA responder and supporting third parties. It verifies whether the service provisioning conforms to the agreement based on the result of the measurement performed by the Agreement State Measurer.
- The SLA Profiler interface is usually implemented by a supporting third party where historical data of services provisioning are stored and analysed. The

SLA responder uses this interface to report the runtime information of the service provisioning for profiling.

A SLA initiator may also interface with other parties, for example, the Agreement State Measurer interface and the SLA Profiler interface with supporting third parties. As depicted in Figure 2, the SLA initiator consists of the following components:



**Figure 2: Architecture of SLA initiator agent**

- The Business Rule Consultant interacts with application client to translate its requirements of the Web service into machine readable information in order to form a SLA, and provides domain specific information for other components.
- The SLA Profiler interface, similar as that of SLA responder's. Moreover, it can be used to retrieve information about service providers, such as historical and predicted performance.
- The Agreement Initiator initiates the SLA formation with support from the Business Rule Consultant. Also, it fetches additional information of the service provider from the SLA Profiler for provider selection.
- The Agreement Negotiator, similar to that of SLA responder's, performs the SLA negotiating process on behalf of the SLA initiator.
- The Agreement Executor, similar to that of SLA responder's, but deals with the client application system.
- The Agreement State Measurer interface is the same as in the SLA responder.
- The Agreement Compliance Verifier performs similarly as the one in the SLA responder.
- The Agreement Factory Proxies are a set of references to the remote SLA responders' Agreement Factories.
- The Agreement Negotiator Proxies are a set of references to the remote SLA responders' Agreement Negotiators.
- The Composition Manager interacts with multiple Agreement Factory Proxies and Agreement Negotiator Proxies to manage the Web services composition.

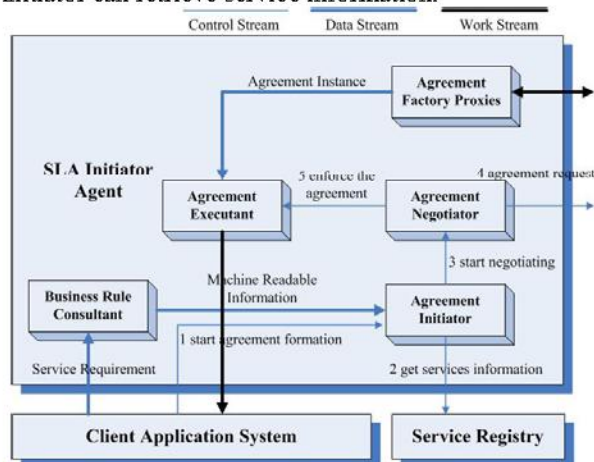
Based on this framework, corresponding operational mechanisms are detailed in Section 4.

## 4. Operational Mechanisms

As discussed earlier that comprehensive SLA management over the lifetime of SLA, including SLA formation, monitoring, recovery, and profiling, is essential. In addition, commissioning the monitoring to an authorised supporting third party is recommended. Therefore, we focus on SLA formation, recovery and profiling in this section to support the framework proposed in Section 3.

#### 4.1 SLA Formation

SLA formation always starts with a SLA initiator requesting a SLA responder to provide a predefined agreement template with fixed, predetermined information, and negotiable elements, as well as rules for SLA initiator to follow when creating the agreement. Then the SLA initiator fills in the blanks in the template according to its practical requirement and returns the template to the SLA responder. The SLA responder could further amend the values in the template, thus, starting the negotiation with the SLA initiator, until an agreement is reached. Alternatively, the service provider can register its services and corresponding templates in a service registry such as UDDI, from which the SLA initiator can retrieve service information.

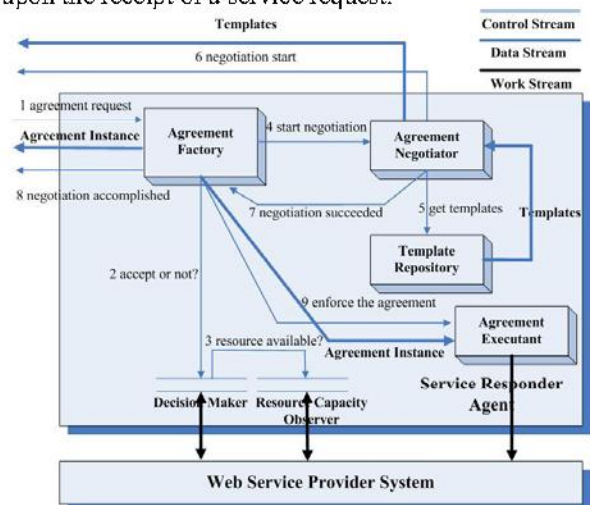


**Figure 3: Interactions in SLA initiator agent in SLA formation**

When the agreement formation starts, on the SLA initiator's side, the domain specific client application system inputs client requirements into the Business Rule Consultant component, and activates the Agreement Initiator component to undertake the agreement negotiation. The Agreement Initiator accesses the service registry or the service providers to request service information, such as the types and the service level guarantees of the services. By doing so, the Agreement Initiator can find out which service providers it is most interested in. The selection depends

on the domain specific business rules from the Business Rule Consultant. When the selection is done, the Agreement Initiator activates the Agreement Negotiator component to send a request message to the SLA responder agent to start the negotiation. If the negotiation succeeds, the Agreement Executor will receive the announcement of consuming the service from the Agreement Negotiator, according to the finalised agreement from the Agreement Factory Proxy. Figure 3 illustrates the interactions between the components in the SLA initiator agent in the SLA formation process.

On SLA responder's side, upon receiving an agreement request, the Agreement Factory component requests the Decision Maker whether the request can be accommodated. The Decision Maker makes the decision by comparing the current resource capacity provided by the Resource Capacity Observer with the agreement request. If the requirements cannot be met, a "Request Rejected" message is sent to the SLA initiator, otherwise, the Agreement Factory advises the Agreement Negotiator to obtain agreement templates from Template Repository and then start the negotiation by sending the templates to the SLA initiator. Subsequently, if the Agreement Negotiator succeeds in making a deal with the SLA initiator, it notifies the Agreement Factory to generate the agreement. Finally, the Agreement Factory sends the agreement to the SLA initiator and notifies the Agreement Executor to enforce the agreement. Figure 4 illustrates the interactions between the components in the SLA responder agent upon the receipt of a service request.



**Figure 4: Interactions in SLA responder agent in SLA formation**

The SLA specifications introduced in Section 2 present different advantages and also reveal respective limitations. Based on comparison of these specifications, WS-Agreement is used in this research, as it is the most recent and comprehensive approach. It is flexible, scalable and expressive in SLA representation so that it can be conveniently extended for a wide range of applications.

## 4.2 SLA Recovery

During service provision, many events, e.g., link failure, resource unavailability and service provider's intended cancelling, may lead to incompliance of the agreed agreement. If not handled properly, these problems may result in severe consequences, such as service interruption, business loss and jeopardy of service provider's reputation. The incompliance detection can be event-driven. The agreement can contain an element like "EvaluationEvent" associated with a service level object, which defines the case in which the expression of the service level objective is to be evaluated. Alternatively, the detection task can be arranged according to an element "Schedule" referring to a schedule defined in the agreement. In the proposed framework, the above two detection mechanisms are performed simultaneously by the Agreement Compliance Verifier component. In this way, if the Web service provider needs to amend or cancel the agreement at runtime due to unavoidable causes, the Web service agent activates the Agreement Compliance Verifier component to inform the Agreement Negotiator component of the incompliance. The Agreement Negotiator immediately sends an "Amend Agreement" message or a "Cancel Agreement" message to SLA initiator. The breaking event such as resource down and link failure would be identified by scheduled detection. The incompliance will cause a renegotiation between the SLA initiator and the SLA responder, or even selection of a new Web service provider for negotiation.

The Agreement Compliance Verifier works with the Agreement State Measurer. Traditionally, incompliance is detected through comparing the runtime values of the service parameters against the agreed thresholds defined in the agreement. In order to improve the performance of the recovery functionality, the Agreement State Measurer could be enhanced by adding the capability of predicting the incompliance, which can be realised by forecasting the trend of the parameters of the service by contrasting runtime information with historical data provided by the SLA Profiler. The specific predicting algorithm is beyond the scope of this paper.

SLA recovery is performed differently, depending on whether the detected SLA incompliance is caused by a cancel on purpose or by a breaking event. If the Web service provider system is incapable of satisfying service provision, it will send a message to the Agreement Compliance Verifier component to announce its incapability and the need to amend or cancel the agreement. If the agreement needs amendment, the Agreement Compliance Verifier will trigger the Agreement Negotiator to contact the SLA initiator to renegotiate the agreement. Upon receiving an "Amend Agreement" message, the SLA initiator will decide whether to start the renegotiation or not, followed by actions such as renegotiation or claim for compensation and selection of a new service provider. If it is a "Cancel Agreement" message from the SLA

responder, the SLA initiator can immediately claim for compensation and start to select new service providers. In both cases, new SLA negotiation will be activated. When the incompliance detected is caused by a breaking event such as link failure, the contracted parties will be informed by the one who is in charge of the detection, typically a supporting third monitoring party. In this case, the SLA responder contacts the SLA initiator to discuss the emergency. The rest of the SLA recovery is just the same as the one in the case of amendment or cancel on purpose. If customised, at the end of the SLA recovery, the SLA initiator will report a negative feedback about the Web service provision to the third supporting SLA profiling organisation through the SLA Profiler interface. Figure 5 shows the interactions between the components in the SLA responder agent.

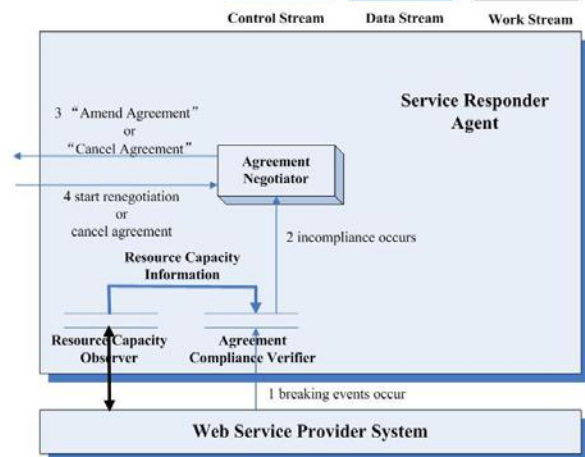


Figure 5: Interactions in SLA responder agent in SLA recovery

## 4.3 SLA Profiling

The reputation of Web service providers becomes a desirable criterion for Web service consumers to judge and select a service provider. The reputation of a Web service provider can be evaluated by judging its past performance. Therefore, in a practical environment, it is necessary to profile the performance of the service providers and to rank them. Another significance of SLA profiling is that it is a feasible mechanism to improve Web service composition. Ranking Web service providers can be achieved by analysing the historical performance of service providers and predicating their performance for the forthcoming task, which is again beyond the scope of this paper.

SLA profiling could be either centralised or distributed. Centralised SLA profiling architecture is easy to deploy, while distributed architecture would have better scalability, availability, fault tolerance, and is more compatible with the distributed Web services environment. It is recommended that SLA profiling system be deployed along with the service registry, usually UDDI, to provide simplicity for deployment and convenience for information retrieval.

Actually, the criteria to rank the Web service providers, the algorithm to estimate their future collaboration and the way to deploy the SLA profiling system are domain specific problems and should be tasked to authorised professional organisations. Thus, this paper only discusses when and how the agent makes use of the SLA profiler interface.

The SLA initiator can consult the SLA profiling system before the negotiation phase. After translating the requirement of the client application, the Agreement Initiator component turns to the SLA Profiler for information about service providers. The SLA Profiler then provides a list of candidate service providers and their preferences. After that, the Agreement Initiator can trigger the Agreement Negotiator to start interacting with the candidate service providers one at a time or simultaneously. The Agreement Negotiator can contact multiple service providers for comparison in order to choose the most appropriate one. The next step goes to the agreement negotiation described earlier. Figure 6 shows how the components work in SLA profiling at the beginning of the SLA formation.

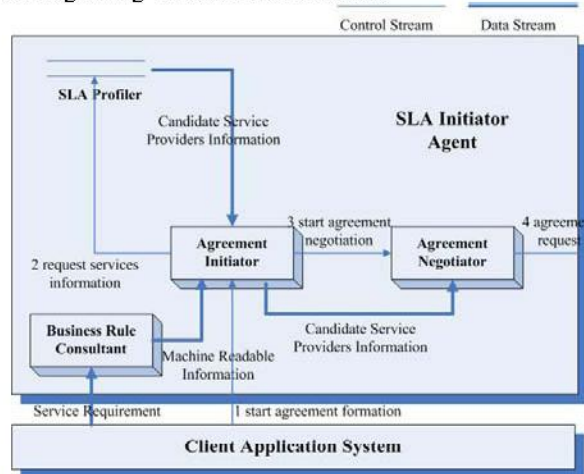


Figure 6: Interactions in SLA initiator agent in SLA profiling

## 5. Conclusions and Future Work

As the Web services technology gained intensive attention in developing business applications, management of service level agreement associated with Web services has become an urgent and thriving research area. Adequate SLA management support should allow for autonomous, dynamic, and flexible SLA operations. This research advocates that utilising the agent technology in solving complicated SLA management problems is a fruitful line to follow. An innovative agent-based framework and corresponding mechanisms for SLA formation, recovery and profiling are presented in this paper. The main functionalities the framework needs to provide and the interactions between the components are identified and discussed.

In the future, a prototype of this framework will be implemented and tested. Integration of the proposed

approach and the existing standards such as WSDL, UDDI and WS-Agreement will be investigated. In addition, more sophisticated behavioural models, such as SLA suspension and resumption, and SLA amendment and cancellation from the SLA initiator, will be studied.

## Acknowledgement

This work is partly funded by the Australian Research Council Discovery Project Scheme under grant DP0663841.

## References

- [1] A. Anderson, "An Introduction to the Web Services Policy Language", in *Fifth IEEE Int'l Workshop on Policies for Distributed Systems and Networks*, 2004.
- [2] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke and M. Xu, "Web Services Agreement Specification (WS-Agreement)", Version 2006/09, World-Wide-Web Consortium (W3C), September 2006.
- [3] M. B. Chhetri, J. Lin, S. K. Goh, J. Yan, J. Y. Zhang and R. Kowalczyk, "A Coordinated Architecture for the Agent-based Service Level Agreement Negotiation of Web Service Composition", in *Proc. of the Australian Software Engineering Conference*, Sydney, Australia, April 18 – 21, 2006. pp. 90-99.
- [4] D. Greenwood, G. Vitaglione, L. Keller and M. Calisti, "Service Level Agreement Management with Adaptive Coordination", in *Proc. of International Conference on Networking and Services (ICNS)*, July 16-19, 2006, Silicon Valley, USA. pp. 45.
- [5] L. J. Jin, V. Machiraju and A. Sahai, "Analysis on Service Level Agreement of Web Services", *Research Report HPL-2002-180*, Hewlett-Packard Laboratories, June 2002.
- [6] A. Keller and H. Ludwig, "The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services", *Journal of Network and Systems Management*, v.11 n.1, March 2003. pp. 57-81.
- [7] H. Ludwig, A. Dan and R. Kearney, "Cremona: An Architecture and Library for Creation and Monitoring of WS-Agreements", in *Proc of 2nd ICSOC*, New York, USA, November 15-18, 2004. pp. 65-74.
- [8] H. Ludwig, A. Keller, A. Dan, R.P. King, and R. Franck, "Web Service Level Agreement (WSLA) Language Specification", Version 1.0, International Business Machines Corporation (IBM), 2003.
- [9] D. Ouelhadj, J. Garibaldi and J. MacLaren, "A multi-agent infrastructure and a service level agreement negotiation protocol for robust scheduling in grid computing". In *Proc. of the European Grid Conference (EGC)*, February 14-16, 2005, Amsterdam, The Netherlands. pp. 651-660.
- [10] A. Sahai, A. Durante and V. Machiraju, "Towards Automated SLA Management for Web Services", *Research Report HPL-2001-310*, Hewlett-Packard (HP) Laboratories, July 26, 2002.
- [11] C. Sharp, J. Shewchuk, A. Vedamuthu, Ümit Yalçınalp and D. Orchard, "Web Services Policy Framework (WS-Policy)", Version 1.0, March 2006.