Notice of Violation of IEEE Publication Principles

"Research on SLA Management Model in Service Operation Support System"
by Ke Xu, Xiaoqi Zhang, Meina Song, Junde Song,
in the Proceedings of the 2009 International Conference on Wireless Communications, Networking and Mobile Computing (WiCom2009)

After careful and considered review of the content and authorship of this paper by a duly constituted expert committee, this paper has been found to be in violation of IEEE's Publication Principles.

This paper contains significant portions of original text from the paper cited below. The original text was copied without attribution (including appropriate references to the original author(s) and/or paper title) and without permission.

"Service Level Agreement Monitor (SALMon)"
by David Ameller and Xavier Franch
in the Seventh International Conference on Composition-Based Software Systems, 2008. (ICCBSS 2008), February 2008, pp.224-227

Research on SLA management model in Service Operation Support System

Ke Xu[1], Xiaoqi Zhang[2], Meina Song[3], Junde Song[4]
School of Computer
Beijing University of Posts and Telecommunications
Beijing 100876, China
permit@263.net[1];alphazxq@gmail.com[2]; mnsong@bupt.edu.cn[3]; jdsong@bupt.edu.cn[4]

***Abstract*—In the web services environment, Service Level Agreement (SLA) refers to mutually agreed understandings and expectations of the evaluation criteria for service quality between service providers and customers.**

In this paper, we put forward a complete SLA management model including the lifecycle definition of SLA management for web services, QoS (Quality of Service) modeling and its format description and architecture of SLA monitor. The SLA management lifecycle in the field of web services is defined first. Then we brought forward a QoS model and its formal description which focus on both technical and non-technical aspects of the services. A SLA monitor in accordance with SOA design principle which comprises a monitor, analyzer and arbiter is proposed. Finally we conclude and discuss some future work.

Keywords-SLA; SOSS; QoS; Monitor

I. INTRODUCTION

Along with web-based services becoming richer, to offer differentiated, customizable and QoS guaranteed services is becoming critical for service providers to survive. Both individual and enterprise users are increasingly concerned about QoS issue. Detail of the services, especially QoS terms and metrics must be defined detailedly in the contract between service providers and customers to make the service transaction proceed smoothly.

A service level agreement (SLA) is a contract that includes specifications on a set of agreed service quality management objectives (or service level objectives, SLOs) [1]. Service quality indicators comprise availability, reliability, performance et al. and associated set of QoS metrics may include concurrency, throughput and response time et al. refers to web services.

This work is supported by the National Key project of Scientific and Technical Supporting Programs of China under Grant No. 2008BAH24B04, the innovation technology star program of Beijing under Grant No. 2007A045 and Program for New Century Excellent Talents in University under Grant No.NCET-08-0738.

To set up SLAs between service providers and customers could bring many benefits. 1). Avoiding misunderstandings between both actors by clarifying contract-specific SLA terms and objectives. 2) The penalties for non-conformance could be detailed in the contract in case of SLA violations occur. 3) Service performance data are collected and reported for the customers to make choices and service providers to optimize their services.

Based on study on Operation Support System (OSS) model in the field of telecommunications, we put forward our Services Operation Support System (SOSS).SOSS is designed as a web services management platform for services registration, provisioning, delivery, monitoring and some operational supporting requirements such as authentication, authorization and accounting. SOSS covers some functions which UDDI [2] could provide but maybe more complicated than it. In this paper, we focus on some features of SOSS, mainly the service contract management and monitoring on QoS of web services. Figure 1 shows the general architecture of SOSS and its related actors.

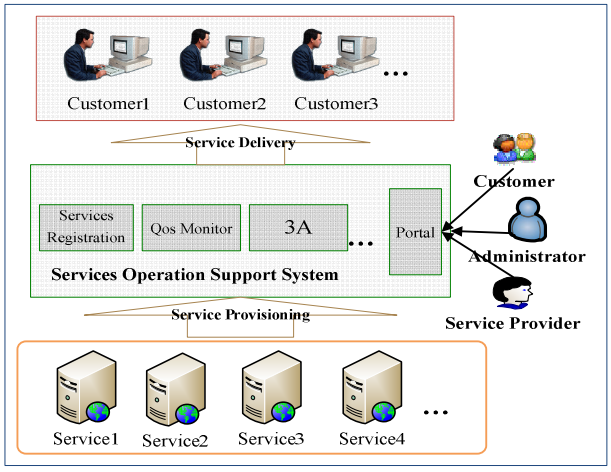

Figure 1. The design of SOSS

As with the SLA itself, the display and interpretation of the report data should be clear and concise [3].Who collects performance data and generates the reports is an issue that is

978-1-4244-3693-4/09/$25.00 ©2009 IEEE

coming under scrutiny. Options include the provider of the service, the customer, or a trusted third party. With each of the options there are different costs associated and a level of trust has to be developed to avoid conflicts. In this paper, SOSS acts as a trusted 3-rd party and is responsible for the performance monitoring and violation checking.

In section 2 of this paper, we describe the SLA management lifecycle in the field of web services based on the study of SLA management lifecycle in telecommunication filed. In section 3, we present our QoS model and formal description of SLA with definition of quality indicators and metrics. In section 4, a monitor for QoS monitoring and SLA violations checking is proposed. Finally there is a section for the conclusion.

II. SLA LIFECYCLE FOR WEB SERVICES

SLA management is the maintenance of an entire business process, has the characteristics of lifecycle. The management of SLA requires interactions between many processes. Various stages must be considered [4]. According to TM Forum, the lifecycle of SLA management maybe as follows [3]:

- The Product/Service Development stage. This stage consists of the identification of the customer needs and the network capacities. From that, service templates are prepared.
- The Negotiation & Sales stage where an SLA is negotiated with a customer. Resource reservation is also used to check with the planning if the SLA can be supported.
- The Provisioning stage. This stage consists of the resource provisioning (i.e. network and service provisioning) and the service activation.
- The Assurance stage which is in charge of monitor validates and reports the SLA, detect SLA violations and handle them.
- The Assessment stage composed of two parts. Assessment with the customer (to check its satisfaction and to identify evolution of its requirements) and internal operator assessment (to check the overall service quality, key problems et al.).

Specific tasks in the stages of the lifecycle above-mentioned are for the field of telecommunications. Considering of the characteristics of web services and the specific needs of SOSS, we describe the SLA management lifecycle in the field of web services as follows shown in figure 2.

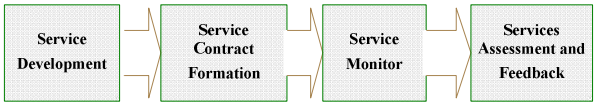

Figure 2. The SLA management lifecycle

- The Service Development stage. In this stage, service providers identify the customers' needs, develop their services consistent with the specifications on services provisioning and define the SLA template. To be integrated in SOSS, services should comply with some specifications on services interface and data exchange format.SLA template is the baseline of negotiation and service ordering between service providers and customers.
- The Service Contract Formation stage. In this stage, service provides register their services, customers searching services for their needs and put forward an order request once they find appropriate services with consistent functions, acceptable SLAs and charges, services providers deal with the orders and confirm the contracts. Web-based accessing channel is provided by SOSS to both service providers and customers for those processes mentioned above.
- The Services Monitoring stage. This stage consists of services activation, QoS monitoring, SLA violations detecting, performance data collecting and report generating. SOSS is in charge of all these processes.
- The Services Assessment and Feedback stage. Based on the performance data and reports together with feedback from customers to assess the overall service quality. Some data analysis could be applied in this stage for the customers' to identify evolution of their requirements and service providers to optimize their services.

In the following two sections, we will focus on QoS model and formal description of SLA, the monitor which is in charge of performance monitoring, violations checking and handling.

III. SLA MODELING AND FORMAL DESCRIPTION

Some solutions for SLA modeling and formal description have been brought forward from both academic and industry, such as WSLA [5] (Web service Level Agreements) by IBM, WSML [6] (Web Service Management Language) by HP and WSOL [7] (Web Service Offering Language) by Carleton University.

WSLA and WSML describe the details of QoS constraints but do not concern about functional constraints and other binding norms of web services. They assumed that both services providers and customers must have the measurement and management infrastructure that makes the solutions complicated and costly.

WSOL describes the web service class, various constraints (functional constraints, QoS, and access permissions) and the management statement (signed agreements, cost per access, fine as well as management responsibilities). Based on WSDL [8] (Web services description Language), further describing of the provisioning of web services using WSOL makes the monitoring and assessment of web services feasible.

To implement SLA, the primary question is: "What do we want to monitor?" To answer this question, we have built a QoS model [9] including some technical characteristics and non-technical characteristics based on quality-related standards especially in the domain of web services. This model was part

of our participation in a National Key project of Scientific and Technical Supporting Programs of China, CSIS (Common Services Integration System), in which we participated with the responsibility of identifying and classifying the characteristics needed for defining the quality of Web services. The simplified QoS model (shown in figure 3) is partly based on the ISO/IEC 9126 standard [10]. However, since this standard focuses just on the technical aspects of software, we have utilized some previous work to enlarge this model including non-technical aspects.

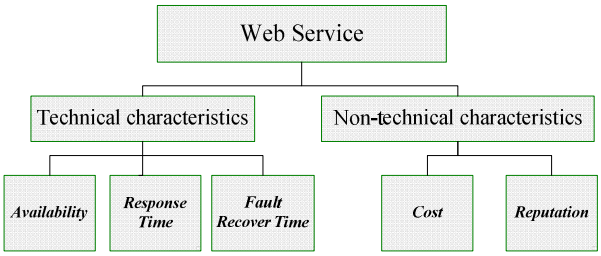

Figure 3. The simplified QoS model for web service

We define the SLA-based QoS model as (1):

$$Q(S) = \{A, T, F, C, R\ldots\}. \quad (1)$$

In this model, "A" represents availability, "T" represents response time, "F" means fault recovery time, "C means service cost and "R" stands for reputation.

Each indicator in this model comprises four items as in (2).

$$\{Name, ComparisionOp, Val, Unit\}. \quad (2)$$

"Name" means the name of the indicator, "ComparisionOp" stands for comparison operator, "Val" represents bound value and "Unit" stands for metrics unit. There is an example of the model showed in the following table.

TABLE I. AN EXAMPLE OF QOS MODEL

	Name	*ComparisionOp*	*Val*	*Unit*
A:	***Availability***	>=	***0.7***	***null***
T:	***ResponseTime***	<=	***10***	***ms***
F:	***FaultRecoverT***	<=	***500***	***ms***
C:	***Cost***	=	***55***	***yuan***
R:	***Reputation***	=	***6***	***null***

Some features of WSOL such as service class, constraints and the management statement could support the model appropriately and we choose related subset of WSOL for the formal description of the SLA model. For example, instance model showed in table1 can be described as the following XML document shown in figure 4.

```
47 <wsol:serviceOffering name="XXX" service= "xxx:xxxService" accountingParty="WSOL-SUPPLIERWS">
48 <wsol:constraint name="ResponseTime" xsi:type="qosConditionSchema:qosCondition"
49 service="xxx:xxxService" portOrPortType="xxx:xxxServicePortType" operation="xxx:yyyOperation">
50     <expressionSchema:booleanExpression>
51       <expressionSchema:arithmeticExpression>
52         <expressionSchema:arithmeticVariable avName="xxx:yyyRequest.ResponseTime"/>
53       </expressionSchema:arithmeticExpression>
54       <expressionSchema:arithmeticComparator type="<="/>
55       <expressionSchema:arithmeticExpression>
56         <wsol:numberWithUnitConstant>
57           <wsol:value>10</wsol:value>
58           <wsol:unit type="QoSMeasOntology:ms"/>
59         </wsol:numberWithUnitConstant>
60       </expressionSchema:arithmeticExpression>
61     </expressionSchema:booleanExpression>
62 </wsol:constraint>
63 <wsol:constraint name="FaultRecover" xsi:type="qosConditionSchema:qosCondition"
64 service="xxx:xxxService" portOrPortType="xxx:xxxServicePortType" operation="xxx:yyyOperation">
65     <expressionSchema:booleanExpression>
66       <expressionSchema:arithmeticExpression>
67         <expressionSchema:arithmeticVariable avName="xxx:yyyRequest.FaultRecover"/>
68       </expressionSchema:arithmeticExpression>
69       <expressionSchema:arithmeticComparator type="<="/>
70       <expressionSchema:arithmeticExpression>
71         <wsol:numberWithUnitConstant>
72           <wsol:value>500</wsol:value>
73           <wsol:unit type="QoSMeasOntology:ms"/>
74         </wsol:numberWithUnitConstant>
75       </expressionSchema:arithmeticExpression>
76     </expressionSchema:booleanExpression>
77 </wsol:constraint>
78 <wsol:constraint name="Reputation" ...>
79 ...
80 </wsol:constraint>
81 <wsol:constraint name="Availability" ...>
82 ...
83 </wsol:constraint>
84 <wsol:price name="Cost" service="xxx:xxxService"
85         portOrPortType="xxx:xxxServicePortType" operation="xxx:yyyOperation">
86     <wsol:numberWithUnitConstant>
87       <wsol:value>55</wsol:value>
88       <wsol:unit type="currencyOntology:yuan"/>
89     </wsol:numberWithUnitConstant>
90 </wsol:price>
91 ...
92 </wsol:serviceOffering>
```

Figure 4. Formal description for the QoS model in table1

IV. SLA MONITOR ARCHITECTURE

Service-Oriented Architecture (SOA) [11] is an emerging software architecture; systems based on this architecture consist in multiple services working together. Since web services may change their QoS in runtime due to environmental issues or changes made by the service provider, SOA systems need to be adaptable in runtime.

In this section, we propose a SOA-based tool for SLA monitoring called SLAMon. The monitor that is in charge of providing QoS information that is needed to detect SLA violations comprises a monitor, analyzer and arbiter. These three types of components are technology-independent and act as services running inside of SOSS. SLAMon is SOA-based which means service providers could substitute their services by others that possess the interfaces defined for SLAMon.

The architecture of SLAMon is shown in figure 5. It is composed of three types of services: Monitor Analyzer and Arbiter.

The Monitor service comprises Measurers; these components will bring the measures to the Monitor that has the responsibility to maintain this information updated. The update process is an iterative call to each Measurer in different intervals of time, saving the results in a database. Measurers are components instantiated in each monitored service to get all the basic performance data of the selected quality attributes. The interface and implementation of the Measurers is technology-independent because they are designed to support different kinds of services (web services, HTTP services, agent-based services et.al). They can be seen as plug-ins to support specific service technology.

The Analyzer manages Monitor and checks for SLA violations in SOSS. Performance data of the selected quality attributes collected by Monitor will be delivered to the Analyzer to compare with those metrics and indicators stored in a database. When a violation is detected, it is notified to the Arbiter.

The Arbiter service selects the best treatment to solve the SLA violations detected by the Analyzer in SOSS. The Arbiter could use a repository of treatments and alternative services and it will automatically select and execute the best treatment for the reported SLA violations.

The SLA can be configured manually with the interface provided by the Analyzer or automatically with a SLA standard document for each service. SLA can be understood as a set of conditions that must be true in some time interval. A condition is composed of the evaluated metric, a relational operator and a value for the comparison (i.e. "response time < =10ms" is a condition that must be true for the specified service during the specified time interval).

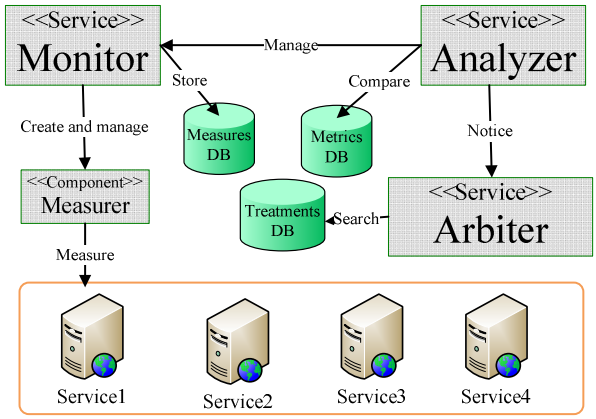

Figure 5. Architecture of SLAMon

V. CONCLUSION

Since SLA management is one of the key functions of SOSS, we study on the SLA management model and have done some work in this area. In this paper, we mainly focus on three aspects of SLA management: the lifecycle of SLA management for web services, SLA modeling and format description and SLA monitor architecture.

SLA management should support the SLA lifecycle. Base on study of the SLA lifecycle in the field of telecommunications and taking the characteristics of web services and specific needs of SOSS into consideration, we describe the SLA management lifecycle in the field of web services in section 2.

To implement SLA in a concrete system, we should first figure out what we want to monitor and how to describe the quality that we are interested in. We need to set up the model of SLA and describe it formally. After comparison among several related solutions, we present our SLA-based QoS model and its formal description based on WSOL.

SLA management system should have the functionalities of collecting the performance data of the selected quality attributes of services, comparing the contracted value with the collected data from the monitor, selecting the best treatment in case of the violation occur. We propose a SOA-based tool for SLA monitoring called SLAMon. The monitor comprises a monitor, analyzer and arbiter.

In the context of SOA systems, the dynamic changes are needed in order to make the services be in accordance with the QoS requirements stated in SLAs. SLAMon provides a method based on the current SLA standards and the monitored information to make self-adapting SOA systems. The SLAMon architecture can be used for different types of services due to its high technology independence.

Since SOA systems are always composed of services with different technologies, as in the future we plan to support monitoring of multiple types of services using the same monitor with different kinds of measurers, so we will be able to monitor an entire heterogeneous SOA system.

References

[1] Ward.C, Buco.M.J., Chang.R.N., Luan.L.Z., So.E., Chunqiang Tang, "Fresco: a Web services based framework for configuring extensible SLA management systems", 2005 IEEE International Conference on Web Services,11-15 July 2005 Page(s):237 - 245 vol.1.

[2] T. Bellwood, L. Clément, D. Ehnebuske, A. Hately, M.Hondo, Y.L. Husband, K. Januszewski, S. Lee, B. McKee, J.Munter, C. Riegen. 2002. "Universal Description Discovery& Integration (UDDI) Specification", http://www.oasisopen.org, February 2005.

[3] Gregory Bain, Jay Dia et al., "SLA Management Handbook – Volume 4: Enterprise Perspective", TM Forum, October 2004.

[4] Marilly.E., Martinot. O., Betge-Brezetz.S., Delegue.G., "Requirements for service level agreement management", 2002 IEEE Workshop on IP Operations and Management,2002 Page(s):57 – 62.

[5] A. Keller, H. Ludwig (IBM). "The WSLA Framework: Specifying and Monitoring of Service Level Agreements for Web Services", IBM research report RC22456, 2002.

[6] Akhil Sahai, Anna Durante, Vijay Machiraju, "Towards Automated SLA Management for Web Services", HPL-2001-310 (R.1), July 2002.

[7] V. Tosic, et al., "The Web Service Offerings Language (WSOL) and the Web Service Offerings Infrastructure (WSOI)," CASCON 2003 workshop Policy and Service Level Agreements: Their Role in Automated Management in Systems, 2003.

[8] E. Christensen et al., "Web Services Description Language (WSDL) 1.1," W3C Note, 15 Mar. 2001, http://www.w3.org/TR/wsdl (current June 2002).

[9] International Organization for Standardization. ISO Standard 8402: Quality management and quality assurance-Vocabulary, 1986.

[10] International Organization for Standardization. ISO/IEC Standard 9126: Software Engineering – Product Quality, part 1. 2001.

[11] M.P. Papazoglou. "Service-Oriented Computing: Concepts, Characteristics and Directions". 4th International Conference on Web Information Systems Engineering, 10-12 December 2003.