WS-SLA: A Framework for Web Services Oriented Service Level Agreements

Wenhui Sun, Jinyu Zhang, Feng Liu
College of computer and information technology
Beijing Jiaotong University, Beijing,100044, China
wenhuisong@netone.net.cn

Abstract

Service level agreements (SLA) becomes the prevailing business model for delivery of a large number of products and services in telecommunication and IT arenas. We presents an integrated service level agreement model applied to operating supporting system (OSS), on the basis of WfMC workflow reference model and service oriented architecture (SOA). And an application scenario instance of SLA system architecture is described by a set of operations
models by Business Process Execution Language (BPEL).

Keywords: Service level agreements, BPEL, SOA , OSS

1. Introduction

A service level agreement(SLA)[1] is a contract or agreement that formalizes a business relationship, or part of the relationship, between two parties. Most often it takes the form of a negotiated contract made between a service provider and a customer and defines a price paid in exchange for an entitlement to a product or service to be delivered under certain terms, conditions, and with certain financial guarantees. Service level management is a method for a service provider to attract a customer and is applied to OSS application. So it is a hot research topic to design a framework of SLA driven OSS.

The Workflow Management Coalition (WfMC)[2] defines workflow as the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules. Workflow management system defines, create and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications. At present most work on workflow management focus on workflow management system itself and its software realization. It is a recent hot research topic to combine workflow management and service management.

Service oriented architecture (SOA) is a system architecture to integrate autonomous distributed components. The components are loosely coupled with each other by strictly-typed interfaces and standardized communication protocols. Service-oriented architecture (SOA) allows dynamic service discovery, binding, execution, monitoring, and evaluation. Web Services is an example of SOA and it uses a variety of standardized protocols such as UDDI, SOAP, WSDL. At present most research work on SOA focus on web service composition such as BPEL[5], BPEL4WS[6] and WS-Transaction[7], which are all XML-based flow languages that defines how business process interact within or between enterprises. The aim of this paper is to study how SOA is applied to implementation of a framework of SLA driven OSS.

The remainder of this paper is organized as follows. In Section 2, we propose implementation of a framework of service level agreements based on workflow management. And we discuss each component and its function of the framework of service level agreements. In Section 3, by using Business Process Execution Language (BPEL) of SOA, we present an application scenario instance of SLA system architecture. In Section 4, the conclusion and future research directions will be given.

2. A framework of service level agreements based on workflow management

At present almost all of SLA implementation are based on TMF's SLA handbook blueprint[1].There are two different approaches to implementing an SLA model, which stem from differences in company focus and from the product suites. One is customer-centric in the sense that it deals with customers, contracts, and billing cycles. For example, Amdoces's SLA blueprint[1] is such a product. The second is network-focused and deals with quality of service metrics, network fault tolerance such as Micromuse's Netcool suite and Orchestream's Resolve suite. These two approaches focus on customer-centric and network performance-oriented respectively. Merging these two approaches should be provided to implement a integrated SLA management. In addition, no matter what the model is customer-centric or network performance oriented, an effective business process management is put into implementation of SLA framework. We propose an integrated SLA model, which is based on the author's

1-4244-0165-8/06/$20.00 ©2006 IEEE.

belief that the only way to drive efficiency and ensure Quality of Service (QoS) commitments are consistently met is to drive the business processes directly from the SLA definitions. The integrated SLA framework is built on distributed computing technology, workflow technology, business intelligence and portal technology. The Integrated service level agreement framework is divided into five layers: a integration layer, an automation layer, a collaboration layer, an intelligence layer and a universal access layer.

The framework of the integrated SLA is shown in Figure1.

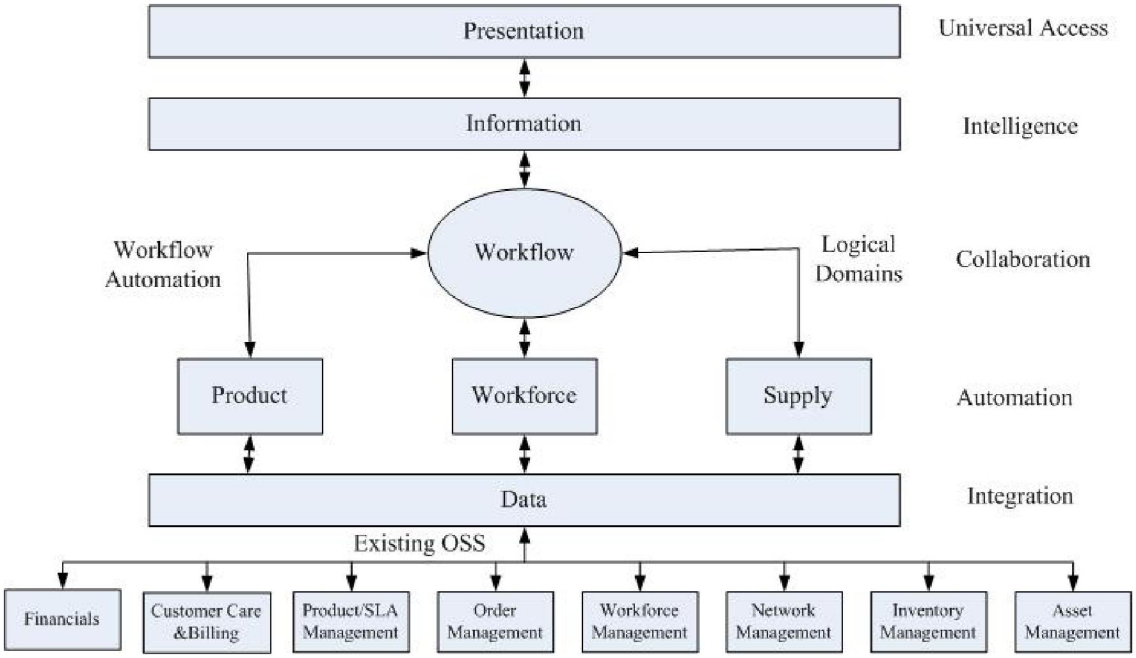

Figure 1. The Integrated Service Level Agreement framework

2.1 Integration layer

The environment must bring together participants, applications, processes, and interfaces seamlessly into a common management environment. Integration occurs at four levels: (1)organizational, (2)process, (3)data interchange, and (4)the collaborative level, which combines the integration of the other levels into a single work flow. Furthermore, the data integration is made up of both technical and semantic aspects.

In a distributed environment, each component or system functions in an autonomous manner need to interact with each other in order to satisfy the end-to-end processes handled by the OSS. Interaction between these systems normally follows one of three primary paradigms: an invocation paradigm, a messaging paradigm, or a publish-subscribe paradigm. The first common paradigm is a invocation paradigm. In a invocation paradigm one system needs a service that has been implemented within another system. The system that implements the service defines how calls are made to activate this service. Examples include CORBA invocations and XML-based invocations such as SOAP. The second common paradigm is message-oriented middleware(MOM). Products supporting this paradigm manage the transfer of messages from one system to another using a messaging bus. This mode is often called store-and-forward. The third common paradigm is publish-subscribe. This paradigm is based on the concept of a business event.

2.2 Automation layer

Workflow is the computerized facilitation or automation of a business process, in whole or part. While workflow may be manually organized, in practice most workflow is normally organized within the context of an IT system to provide computerized support for the procedural automation. A system that completely defines, manages and executes “workflows” through the execution of software whose order of execution is driven by a computer representation of the workflow logic, which is granted as Workflow Management System.

The Workflow Reference model has been developed from the generic workflow application structure by identifying the interfaces within this structure which enable products to interoperate at a variety of levels. All workflow systems contain a number of generic components which interact in a defined set of ways. These generic components are process definition tools, workflow enactment service, invoked applications, workflow client applications, and administration and monitoring.

2.3 Collaboration layer

The environment must support many different entities and parties working together in series, parallel, or in a combined manner. Collaboration too requires the ability to seamlessly support both human and electronic participation in collaborative efforts. Collaboration controls the execution of each individual system and manages dataflow among each individual system. Monitoring system of collaboration layer subscribes events produced by every individual system and produces high-level information. QoS management of collaboration layer deals with cost, performance, security, availability and scalability. Grid computing and Service oriented architecture is closely related to distributed application integration and collaboration domain. The aim of grid computing is to collaborate among a variety of resources to solve complicated scientific problems. Service oriented architecture is a new integration paradigm which focuses on service composition.

2.4 Intelligence layer

The environment must provide very high levels of appropriate SLA compliance and other business intelligence to community members. Intelligence may take the form of real-time or periodic workflow monitoring, key performance indicators, monitors, or reports. The systems within the environment must also function within the work flow and be able to recognize threshold violations and subsequently initiate task generation or trigger automation of predefined actions, such as notifications, escalations, queries, and so on. Business intelligence is actually a integrated application of data warehouse, OLAP and data mining.

2.5 Universal access layer (Portal layer)

The concept that all participants involved in any work flow, both human and electronic, must be able to securely access the environment in a convenient user-friendly manner. The portal provides the following functions which allow users to make these OSS applications efficiently by using our workflow-based portal.

User Authentication and Profile: This is basic components for a portal. A user is authenticated only once and provided all functions of our portal.

Graphic Workflow Editor: User can describe a workflow by graphical tool or text-based XML editor. First activities and its input/output data are defined. Second, User describes the interaction between them by making a workflow with them.

Workflow Execution: When workflow is running, this shows the state of executed workflow graphically. The execution of an activity, state of input/output data transfer, values of some variables in workflow, and standard output/error of a running activity are displayed.

Resource Information: Resource Information function enables us to find data and status about resources such as CPU, memory, network, OS.

3. Application scenario instance of SLA system architecture

The SLA life cycle consists of the phases: SLA development, negotiation and sales, implementation, execution, assessment. The Telemanagement Forum• s service level agreement life cycle is shown in Fig. 2.

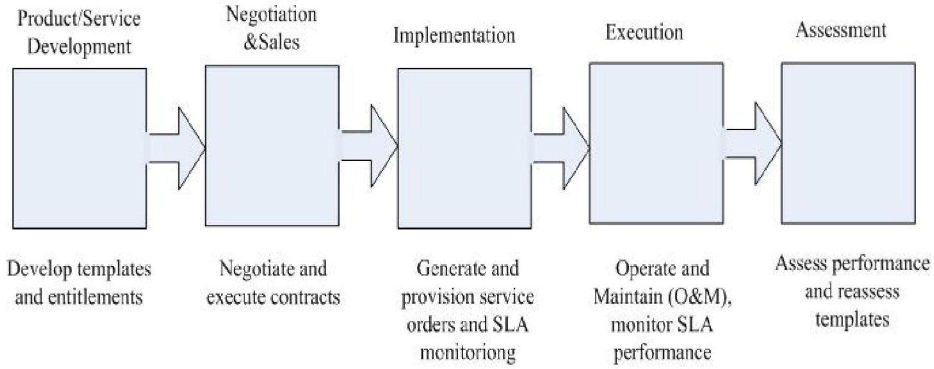
Product/Service Development
Negotiation &Sales
Implementation
Execution
Assessment
Develop templates and entitlements
Negotiate and execute contracts
Generate and provision service orders and SLA monitoriong
Operate and Maintain (O&M), monitor SLA performance
Assess performance and reassess templates

Figure 2. The Tele management Forum's service level agreements life cycle

The high-level flow of SLA-driven OSS application can be captured in BPEL of service oriented architecture shown in Fig. 3. The high-level flow of SLA life cycle starts from developing contracts, negotiating contracts, then moves to implementing contracts, monitoring contracts, and ends with assessing contracts.

```
<process name=”slaDrivenProcess”
   targetNamespace=http://ws-sla.com/ws-bp/ws-sla
   xmlns=http://schemas.xmlsoap.org/ws/2003/03/business-
process/
   xmlns:lns=http://ws-sla.org/wsdl/ws-sla>
<sequence>
<receive partnerLink=”Customers”
   portType=”Ins:ws-sla”
   operation=”sendServiceRequirements”
   variable=”ServiceRequirements”>
</receive>
<flow>
<sequence>
<invoke portType=”Ins:DevelopContract”
   operation=”requestDevelop”>
</invoke>
<invoke portType=”Ins:NegotiateContract”
   operation=”requestNegotiate”>
</invoke>
<invoke portType=”Ins:ImplementationContract”
   operation=”requestImplementation”>
</invoke>
<invoke portType=”Ins:MonitorContract”
   operation=”requestMonitor” >
</invoke>
<invoke portType=”Ins:AssessContract”
   operation=”requestAssess”>
</invoke>
</flow>
<reply partnerLink=”Customers”
   portType=”Ins:ws-sla”
   operation=”sendServiceRequirements”/>
</sequence>
```

Fig. 3: High-level SLA life cycle in BPEL

For more comprehensive discussion, we introduce example scenarios of SLA driven OSS. In this example we suppose a SLA framework consisting of two modules: SLA service management system and network management system support SLA service management system. SLA service management system is composed of three sub modules: contract creation, contract negotiation and contract implementation. We collect data from SNMP network management system to monitoring the execution of contracts. We wrap SNMP API into web services. Management interface can be obtained by applying the SNMP MIB to XML Translation algorithm[8]. In this case the produced interface is fine-grained. However we take a coarse-grained interface module. Given the SNMP MIB definition, we map each SMIv2 MIB table object into T-style interfaces and all remaining scalar objects into S-style interfaces. SNMP get operation for scalar objects can be implemented with WSDL. Interface of web services shown in Fig. 4.

```
<wsdl: definitions
   targetNamespace=http://ws-sla.com/getentry
   xmlns:wsdl=http://schemas.xmlsoap.org/wsdl/
   xmlns:xs=http://www.w3.org/2001/XMLSchema
   xmlns:tns=http://ws-sla.com/getentry
   …>
…
… <wsdl:types>
   <xs:schema>
   <xs:import
      namespace=http://ws-sla.com/getentry
      schemalocation=”…”/>
   </xs:schema>
   </wsdl:types>
      …
      … <wsdl:portType name=”getentry”>
           <operation name=”getentry” …
           …
           </wsdl:portType>
           …
           </wsdl:definitions>
```

Fig.4: SNMP get operation for scalar objects in WSDL

4. Conclusion and Future Work

In this paper, we propose a framework of service level agreements based on workflow management. We also present an application scenario instance of SLA system architecture by using Business Process Execution Language (BPEL) of service oriented architecture. The framework can accomplish automation of business process of SLA-driven OS and provide flexibility of and dynamics of service management. It can adapt to scalability of new services and make service management efficiently. At present the framework has bee applied to the implementation of the prototype of Hi-technology software incubator platform project (863 project).

What we do next is to take advantage of the design idea of service oriented architecture and BEA’s and IBM’s business process integration development tools and implement a product of SLA driven OSS.

References

[1] Lee John J, Ben-Natan Ron, *Integrating service level agreements with OSS,* Wiley Publshing, 2002

[2] WfMC TC00-1003, The Workflow Reference Model, 1995

[3]Watanabe T, “Agent-oriented model for managing long-lived transaction, based on workflow and task graph”, *Computational Intelligence and Multimedia Application*, New Delhi, India, 1999

[4]Cai XiaoLu, Liang YuQi, *Web services technology, architecture and application,* Electronic Industry Press, 2002

[5]Business Process Execution Language for Web Services Specification, version 1.1, available at: http://www106.ibm.com/developerworks/webservices/library/ws-bpel/

[6]Wang X.,Yue K.,Huang J.,Zhou A.,”Service selection for dynamic-driven Web services processing”, *Proceedings of the ICWS 2004*, San Diego, California, USA, 2004

[7]Yue Kun, Wang Xiaoling, Zhou Aoying, “An overview: Web services core technology”, *Journal of Software,* 2004, 15(3)

[8]Jeong-Hyuk Yoon, Hong-Taek Ju and James W. Hong, “Development of SNMP-XML translator and gateway for XML-based integrated network management”, *International journal of network management* 2003; 13