Analysis on Service Level Agreement of Web Services

Li-jie Jin, Vijay Machiraju and Akhil Sahai
HP Laboratories,
1501 Page Mill Road, CA 94304
{firstname_lastname}@hp.com

Abstract

The development of web technologies and standards such as HTTP, XML, SOAP, WSDL, and UDDI enables pervasive adoption and deployment of web services. In a highly competitive business environment, quality of service is one of the substantial aspects for differentiating between similar service providers. A Service Level Agreement (SLA) between a service provider and its customers will assure customers that they can get the service they pay for and will obligate the service provider to achieve its service promises. Failing to meet SLAs could result in serious financial consequences for a provider. Hence, service providers are interested in gaining a good understanding of the relationship between what they can promise in an SLA and what their IT infrastructure is capable of delivering. Similarly, consumers are interested in understanding the impact of the SLAs they sign on their own productivity. In this paper, we present a novel approach to model and understand these relationships. Our model captures composition relationships between providers and consumers, as well the SLA between them. Our approach is based on simulation of the model and sensitivity analysis.

Keywords: web services management, service level agreements, simulation, and sensitivity analysis

1. Background

1.1. Web Services

Web services are Internet based applications that communicate with other applications to offer business data or functional services programmatically. Businesses create web services by exposing specific business functions through Internet protocols and standards. Internally, these services are implemented by integrating legacy or mainframe-based applications or by using the services provided by other web services – internal or external. The development of web technologies and standards such as HTTP, XML, SOAP [SOAP], WSDL [WSDL], and UDDI [UDDI] enables pervasive adoption and deployment of web services.

Major companies and research institutions have been investing in developing web service platforms, tools and applications. IBM presented its Web Service Conceptual Architecture (WSCA) [WSCA] in 2001. Microsoft announces its .Net Framework in 2000. Through .Net, developers can build, deploy and execute XML Web Services and

applications [.NET]. Hewlett-Packard pioneered the Web Service platform development with its E-speak in 1999 [E-speak]. Other players include Sun Microsystems with Sun Open Net Environment (Sun One), Oracle with Orcle9i/Web Service framework, and BEA Systems with J2EE based Web Service Platform.

The first challenge that comes to mind in web services is that of interoperability. Web services are developed and deployed by various companies. So, how do they discovery and communicate with each other? Three standards have been proposed to address this problem – Web Services Definition Language (WSDL) for defining the business functions exposed by a web service, Universal Description, Discovery, and Integration (UDDI) for advertising and discovering services, and finally Simple Object Access Protocol (SOAP) for communication between web services in XML. The reader is pointed to references for a detailed description of these standards.

The next set of important questions to be addressed are: after discovering multiple similar web services, which one is the right one for a service customer in terms of availability, cost, response time, total duration etc., and how to make sure that the "right" service is always "right"? These questions have to be answered under the needs of business organizations and their business processes. Service Level Agreements provide an answer to this question.

1.2. Service Level Agreements

A service level agreement is an agreement regarding the guarantees of a web service. It defines mutual understandings and expectations of a service between the service provider and service consumers. The service guarantees are about what transactions need to be executed and how well they should be executed. An SLA may have the following components:

Purpose - describing the reasons behind the creation of the SLA
Parties - describes the parties involved in the SLA and their respective roles (provider and consumer).
Validity period - defines the period of time that the SLA will cover. This is delimited by start time and end time of the term.
Scope - defines the services covered in the agreement.
Restrictions - defines the necessary steps to be taken in order for the requested service levels to be provided.
Service-level objectives - the levels of service that both the users and the service providers agree on, and usually include a set of service level indicators, like availability, performance and reliability. Each aspect of the service level, such as availability, will have a target level to achieve.
Penalties - spells out what happens in case the service provider under-performs and is unable to meet the objectives in the SLA. If the agreement is with an external service provider, the option of terminating the contract in light of unacceptable service levels should be built in.

Optional services - provides for any services that are not normally required by the user, but might be required as an exception.
Exclusions - specifies what is not covered in the SLA.
Administration - describes the processes created in the SLA to meet and measure its objectives and defines organizational responsibility for overseeing each of those processes.

In a typical scenario, each web service interacts with many other web services, switching between roles of being a provider in some interactions and a consumer in others. Each of these interactions could potentially be governed by an SLA. Considering the legal and monetary implications in violating SLAs, providers need to design their SLAs only after understanding their capabilities. On the other hand, if there is too much leeway in the specification of SLAs, a web service may not be able to fully capitalize on its capabilities. Thus it is important to design SLAs that are able to balance between risk and benefit of all parties. This balance should be based on a good understanding of impact of various service levels on business processes in both the service provider and the customer.

In this paper, we describe an approach based on modeling and simulation that will help providers and consumers understand the relationship between their internal IT processes and SLAs. The model captures the composition relationships between web services as well as the SLAs between each of them. The simulation helps in answering "what-if" questions such as the following:

a. What is the impact of changing my suppliers and/or their SLAs on my capabilities to meet my SLAs?
b. To meet a particular level of service (SLA) for a class of customers, what kind of SLAs should I sign with my suppliers? In the case when my service does not depend on any suppliers, but is only dependent on internal applications, what kind of SLAs should I have with my IT department?

The rest of this paper is organized as follows: Section 2 describes our service composition and SLA model. Section 3 describes the simulation tool that we have developed. This tool would be helpful to a service provider in the SLA design stage. In section 4, we present a scenario and some results of simulation on that scenario. Section 5 concludes with some directions for future research.

2. Service Composition and SLA Model

In many cases, web services are exposed access points of business processes that are carried by service providers. Meanwhile, a business process could be composed of one or more web services that are provided by other business units or organizations. For example, in an E-procurement system, the purchase-order input interface can be published as an entry of a web service. The backend business process that enables this procurement service may invoke web services from delivering business, public catalog management service and financial services of involved business organizations.

So, our web service composition model has two abstractions – a web service and a business process. Every web service is modeled as a set of operations. Each operation in turn is implemented through a business process. The business process defines a flow (sequencing) of activities (or nodes). Each of the activities is in turn implemented by executing an operation on another web service. A business process can be made of decision points or branches, joining points, and loops.

When a web service is implemented by integrating a number of internal applications, the business process essentially captures the integration logic – how each operation is executed by invoking functionality on legacy applications. When a web service is implemented by composing other web services, the business process captures flow of logic from one provider to another. In a typical scenario, a business process is a combination of both.

In this composition model, it can be noted that the overall process represents an operation of one web service (to minimize confusion, we call this as offered web service), while each of the activities represents an operation on supplier's web services (with the additional note that a supplier can be the internal IT department). As a result, an SLA can be attached to each of the activities to represent supplier-side SLAs. Similarly, SLAs can be attached to the overall process to represent customer-side SLAs.

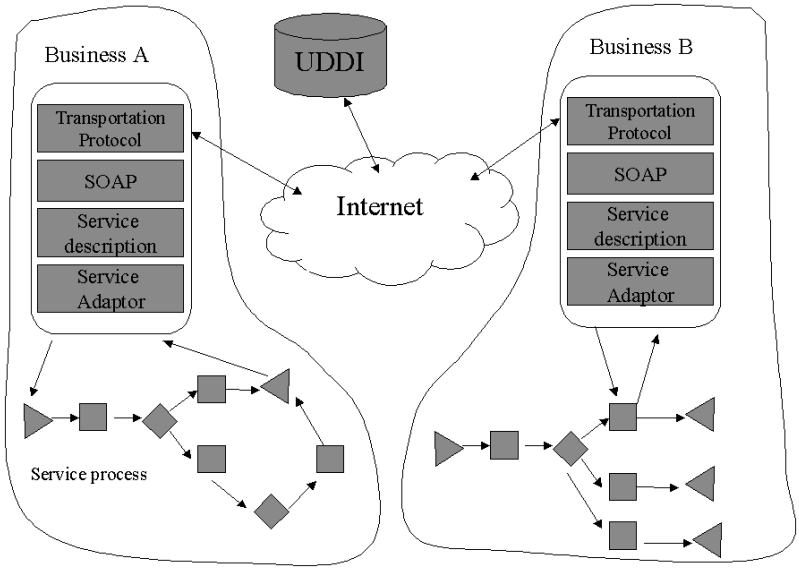

Figure 1. Web service and business process

An SLA itself is modeled as a simple distribution of a metric (for e.g., average response time). The distribution specifies the probability that the metric takes on a particular value. For example, if an SLA promises an average response time of 3 seconds for a certain operation with 95% probability, then the distribution is a normal distribution with a mean value of 3. Other distributions that are not normal can also be used to model SLAs. However, we have not captured more complex SLAs that cannot be expressed as distributions, or other surrounding factors such as penalties into the SLA model. This is a topic for future research.

3. SLA Simulation and Analysis Tool

Business Process Simulation Environment (BPSE) is an integrated environment that supports composite service modeling, simulation and SLA analysis. It supports developers to design, reengineer and verify their web services and related business processes. BPSE allows web service developers use historical execution data of resources, services and activities to lessen space of simulation input model [JCS02]. BPSE is composed by a business process modeling tool, a process simulator and an adaptor to a business process execution data warehouse. The process simulator uses discrete event simulation model to execute composite service. Figure 2 shows how BPSE components cooperate with each to support refining of a composite web service.

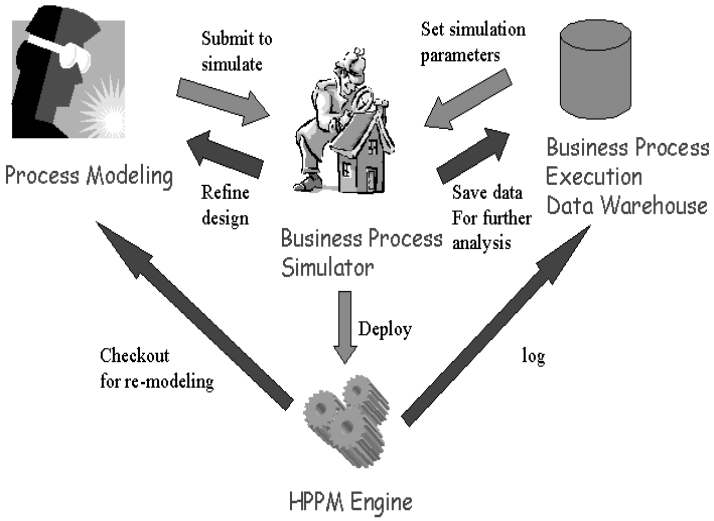

Figure 2 Business Process Simulation Environments

The applying scope of business process description for web services in this work intends not to be limited to one or few particular business process management systems. However, we will focus on business process model and execution mechanism that are supported by HP Process Manager (HPPM) for concreteness. In HPPM, a process is described by a directed graph that has several different kinds of nodes. Work nodes represent invoking of activities (also called services). They assign tasks to a human worker or an automated resource. Route nodes are decision points that route an execution flow among nodes based on associated routing rules. Event nodes denote points in the process where an event is notified to or requested from other processes. Start node stands for the entry point to processes. Complete nodes indicate termination points. Arcs in the graph denote execution dependencies among nodes. Resource in HPPM is defined as entity that executes an assigned activity of a process [HPPM]. Resource could be human workers, or application programs or web services.

4. Service Level Sensitivity Analysis

4.1 Technical issues of SLA automation

Nowaday, enterprises intend to outsource functionalities that other enterprises can provide effectively while concentrating on their own areas of expertise. It is an important decision thus to determine which service provider to choose for outsourcing purposes. When many web services with similar functionalities are avilable on the Internet, the quality of services and the performance/cost ratios will distinguish service providers from each other. Beside functionality, service quality information that is indicated by a set of service level indicators are important when a customer makes decision of picking up a suitable web service to create new business functions. A web service customer needs to evaluate these SLI information before selecting one web service vendor from a group of similar vendors. Web service providers need to publish SLOs for customers to review. These mechanisms are fundation of SLA automation that includes automatic SLA creation, SLA monitoring and control. Effective SLA automation gurentees automatic web service composition as a way to create mission critical business functions.

4.2 Simulation of SLI impact on business process

Before design and negotiating an SLA, service providers and customers should have knowledge of impact of various Service Level Objectives on its internal business processes. These knowledge is critical not only because that they can help managers comprehend the bottom line of their interestes but also know impact of a failure service level objective.

Work Node Name	Min. Duration	Max. Duration	Resource name
Get PR List	3	5	ServiceHost
Notify Rejection	1	5	ServiceHost
Send PR to Pur. Unit	1	5	ServiceHost
Purchase Unit	150	300	PurchaseUnit
Notify P.O. issued	1	5	ServiceHost
P.O. Ack.	20	40	Customer
Delivery service	100	150	DeliveryService
Inspection service	12	15	InspectionService
Delivery Confirm	20	50	Customer

Route Node Name	Yes (Null)	No	
Check Email	95	5	
Approved	95	5	
Pass Inspection	90	10	SLA term
Approved to Reorder	70	30	

Table 1 Parameters of the Purchase Process

In this section, we examine sensitivity of a purchase process, which is presented in Figure 3, to SLI changes. The process contains 12 work nodes that represent 12 activities in the process. It has 4 route nodes to set up control of branches and loops. It includes one start node that represent the entry and the first step of this process. It also has 3 end nodes that represent three different process exit conditions. Table 1 describes the simulation parameters that are synthesised from historical and experimental data. The duration time unit is in minutes. The branch rate of route nodes is in percentage.

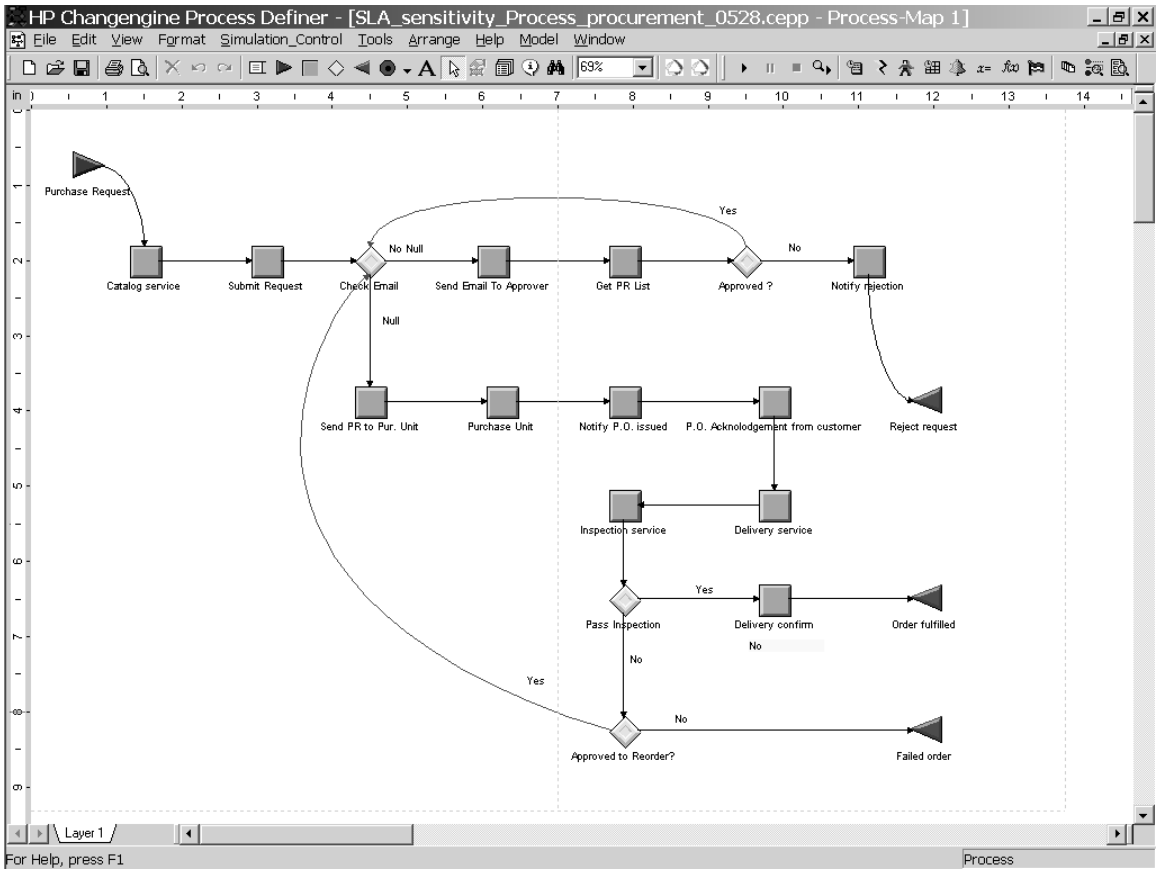

Figure 3 purchase process

The "Purchase Unit" is an outsourcing activity. It is carried through a web service that is hosted by another business unit. The ratio of qualified purchase, quality of service, is checked by a route node " Pass Inspection". In BPSE, the workload generator is set to generate 1000 transactions with random intervals between 5 to 50 minutes with a normal distribution. The "what-if" scenarios are set by combinations of two SLIs that describe service level of "Purchase Unit". There are "Purchase Unit" duration and qualified purchase rates. The duration range of activity "Purchase Unit" is from 300 minutes to 150 minutes with a 30 minutes interval. The qulified purchase rates that are investigated are in a form of percentage of "Yes" branches vs. persentage of "No" branches such as 70/30, 80/20, 85/15, 90/10, 95/5, 98/2, 99/1 and 100/0. The activity duration variety facilitates understanding of how the internal business process will behave if the process owner can negotiate a SLO with 10% less service duration than before. The assortment of qualified purchase rate reflects impact of different service quality on the sample

business process. The combination of various value of SLIs represents possible service level promises from different web service vendors.

Figure 4 demonstrates simulation results of total duration distribution of 1000 transactions under different scenarios. From this distribution chart, business managers get information of impact of SLI combination on its business process in terms of duration. The experimental results suggest that the throughput of the purchase process with its simulation parameter set up increases when at least one of the two SLIs, "Purchase Unit" duration and qualified purchase rate, decreases and the other one keeps still or decreases as well. In general, shorter response time and higher service quality (in terms of rate of qualified purchase) means higher cost of the service. Through balancing among local process execution cost, web service costs and value-adding that is expected from using the web service, business managers can design their SLOs in right SLI ranges and can compromise with web service hosts on top of a clear vision of potential yield.

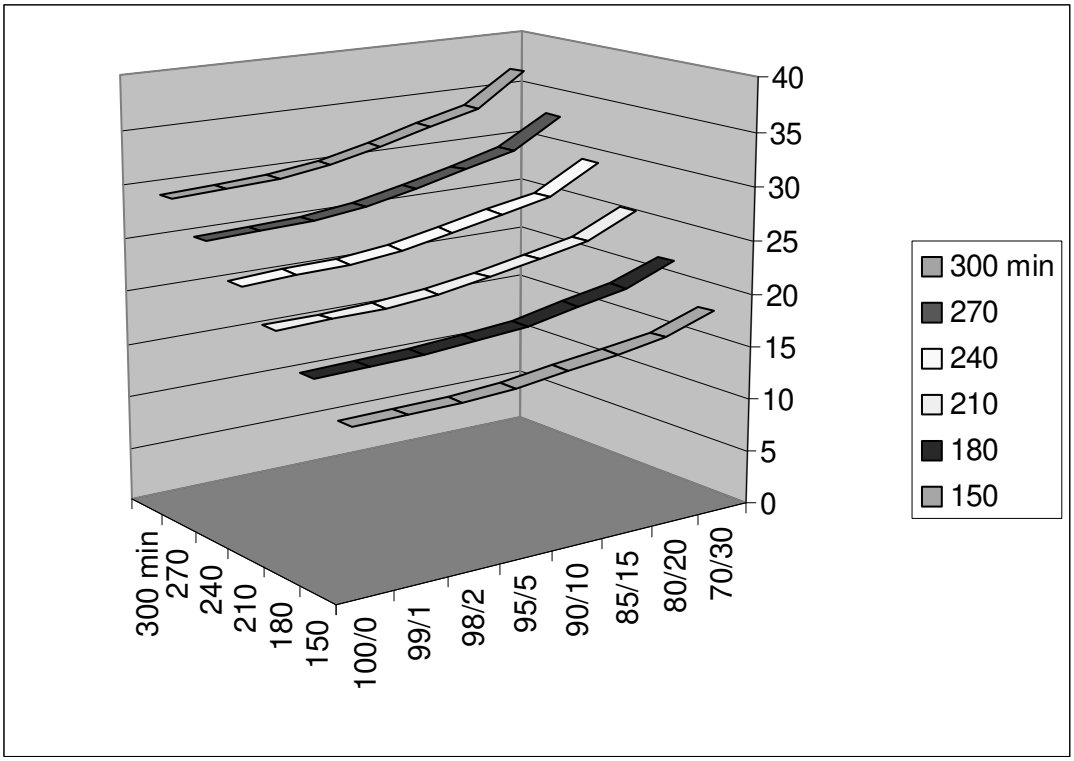

Figure 4. Duration distribution of 1000 transactions

Figure 5 shows the duration improve percentage when qualified purchase rate increaes. When the qualified purchase rate is improved from 70% to 80%, the 1000 transaction duration improve about 8%. The Reduced business process duration means higher transaction throughput, lower customer response time and lower process execution cost. The actual benefits of this duration reduction depends on the cost distribution of executing the purchase process. When the qualified purchase rate reaches 98%, the 1000 total transaction duration decreases 20% comparing to the duration when the qualified purchase rate is 70%.

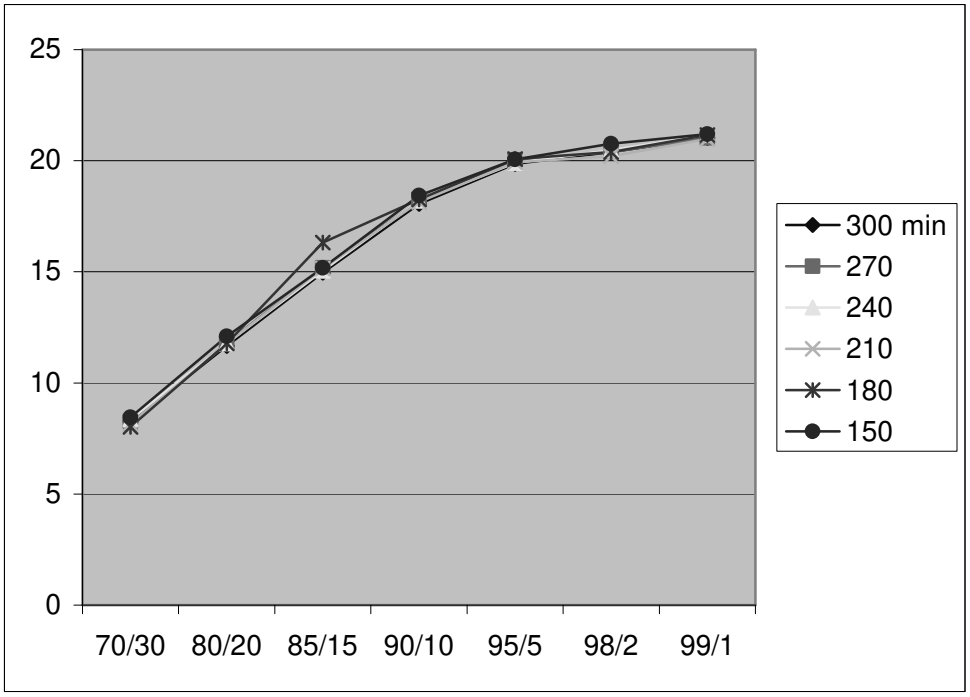

Figure 5 The duration improvement percentage when qualified purchase rate increases.

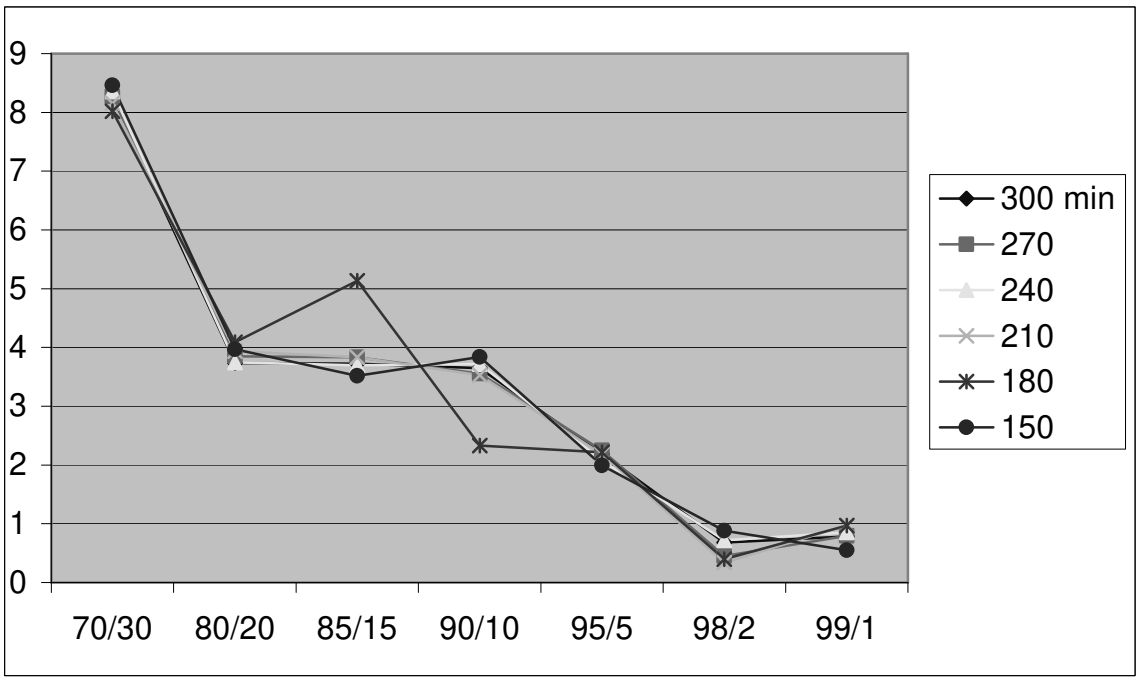

Figure 6 The duration improvement between two subsequent rate change.

Figure 6 gives out the duration improvement percentage when qualified purchase rate increases since last duration improvement. These kind of information will help business decision maker to balance the cost of requesting higher qualified purchase rate and the yield of having the rate. Similar simulation or analytical analysis on impact of related

SLIs, for example, process execution costs and average response time, on business processes will prepare rich information that support automatic SLA negotiation.

4.3 SLI information publishing and dynamic service ranking

Web services are sprouting up on the Internet in the form of portal and web sites. As they are numerous in number they need to discover each other if they are to collaborate with each other. Mechanisms of registration and discovery of service are thus needed. UDDI operator sites would be a set of sites where these web services will register themselves and will be discovered by other web services. The registrations will be done in certain vocabularies, ontologies, template models (tModels). These registrations however consist of static attributes that describe the service, for example, the name of the service, URL of the service, service type, protocols they support etc. These attributes do not change too much in real time and do not reflect service behavior, QoS they provide or their performance.

While looking up services it is essential to understand what are the static attributes and the dynamic attributes of the service. The static attributes of a service like, name, URL, service type, etc. are submitted in certain vocabulary or ontology at the registration time (possibly at UDDI operator site). However, the decision to choose a service also depends on a set of dynamic attributes. These dynamic attributes change in real time and are dependent on how these services are actually performing.

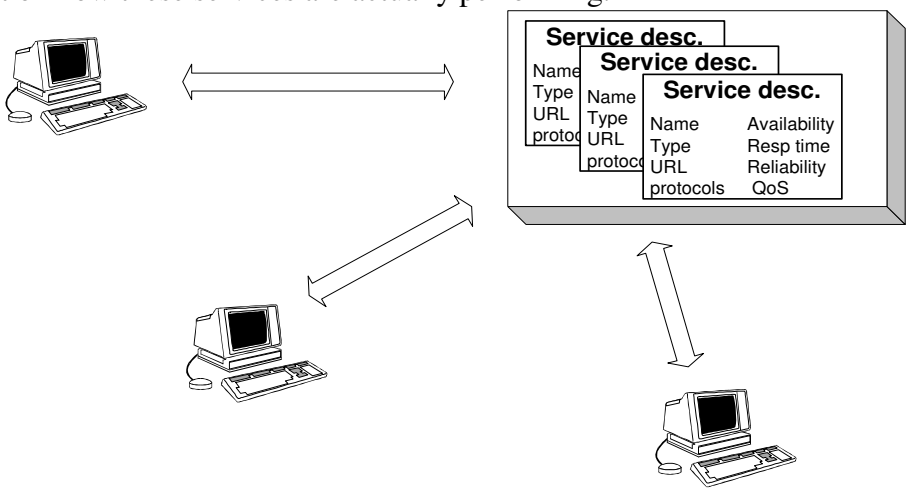

Figure 7. Service description

We assume that the services are looked in both static attributes as well as dynamic attributes based on service performance and QoS. This can be implemented in two different ways. In the first way, registration takes place in vocabularies that consist of static and dynamic attributes. Static attributes have values associated with them while dynamic attributes are resolved in run time. If a lookup is performed based on the dynamic attributes their values are fetched from the services by the repository at run time and displayed to the service/client performing lookup. The other option is for the services to publish the dynamic attribute values to the repository (e.g. UDDI operator site).

The other way could be a two-phase one too. In the first phase services are looked up using static attributes. The short listed services are then queried for dynamic attribute values directly. These dynamic attribute values are obtained dynamically from the short-listed services.

We assume that the services are being invasively or non-invasively instrumented for management purposes. The instrumentation is done through ARM, Mtrack, XARM, APIs or by monitoring HTP request response interactions. These instrumentation APIs collect the service (instance and type) level and transaction (instance and type) level information that can be used to calculate a set of high level metrics. These can be described as dynamic attributes in vocabularies that are resolved in run time at the service. A service is capable of providing this information as it already has the necessary information. An example set of dynamic attributes can be as follows:

- Current response time averages for service transactions
- Expected time to finish for the task
- SLA violation rate for the service
- Reliability
- Availability
- Levels of user differentiation
- Service rating
- Partner services it uses
- Cost of the service
- Level of control the service provides

The next question after getting dynamic attributes of a group of similar web services is ranking these services. Service ranking is a controversial problem in terms of both technical and social aspects. A self-ranking statement may not be reliable. A third party ranking service may not consider service quality aspects in a priority that matches customer's need. *Dynamic service ranking* is a concept about service ranking based on customer's real time needs and this ranking is done and available only for the party that needs to sort out a web service from similar service providers. When ranking a group of available web services, the customer gives out a priority of SLIs. It then accesses SLI information from registration service. With available SLIs, the customer use BPSE to evaluate possible combination of SLIs and get distribution charts of SLI impact on its business process. Base on these real time impact distribution charts, customer service level requests and available budget, the customer ranks those available web services when it needs to select one to fulfill a business function.

5. Conclusions

Internet affects modern businesses in many ways. Web-based applications and services are fast becoming the method of choice for these enterprises to make available their internal business processes to users.. Automatic accessing and composition of these applications and functions are an important issues researchers from many industrial and research institutions are working on. In this paper, we focus on information collection

and analysis at the creation stage of SLAs. Our experimental results suggest that having information of impact of various service levels on business process will give SLA negotiators, human managers or automatic components, a clear picture of pros and cons of various SLOs in an SLA. We also introduce the concept of *Dynamic service ranking*.

References

[.NET] Microsoft. http://www.microsoft.com/net/

[BSC98] Bhoj P, Singhal S, Chutani S. *SLA Management in a federated Environment.* HPL-98-203.

[CAH96] Campbell A, Aurrecoechea C., Hauw L .*QoS review Architectures,* Proceedings of the 4th International Workshop on Quality of Service (IWQoS)

[E-speak] Hewlett-Packard, http://www.e-speak.net/

[Forbath98] Forbath T. *Why and how of SLAs [service level agreements].* Business Communications Review, Vol 28. No. 2, Feb 1998

[HPPM01] Hewlett-Packard. *HP Process Manager Process Design Guide*. 2001. Available from www.ice.hp.com

[HR99] Hauck R, Reiser H. *Monitoring of Service Level Agreements with Flexible and Extensible Agents*. HP OpenView University Association (HP-OVUA) Plenary worhsop, Bologna, Italy, 1999.

[JCS02] Li-Jie Jin, Fabio Casati, and Ming-Chien Shan, *Business Process Simulation with HP Process Manager*, Proceedings of Collaborative Technologies Symposium. San Antonio, Texas, USA, 2002.

[KLB98] Katcgabaw M, Lutfiyya H, and Bauer M. *Driving Resource Management with Application-Level Quality of Service Specifications.* In the proceedings of ICE 98, USA.

[LJW01] Long T P, Jong W B, Woon HJ. *Management of service level agreements for multimedia Internet service using a utility model.* IEEE communications Managezine Vol 39, no.5, May 2001

[LR99] Lewis L, Ray P. *Service Level Management: Definition, Architecture, and Research Challenges*. In the proceedings of IEEE GlobeCom'99.

[SLAHandbook01] Tele Management Forum *SLA Management Handbook*, GB917, public evaluation version 1.5 , June 2001. http://www.tmfcentral.com/kc/repository/documents/GB917v1.5.pdf.

[SM01] Sahai A, Durante A, Machiraju V. *Towards Automated SLA Management*. HPL-2001-310

[SOAP] *Simple Object Access Protoc*ol 1.1

[UDDI] *Universal Description, Discovery, and Integration,* http://uddi.org/specification.html

[WSCA] *IBM Web Service Conceptual Architecture* 1.0

[WSDL] Web Services Description Language 1.1

[WV96] Wolter K, Van Moorsel A**.** *The Relationship between Quality of Service* and Business Metrics: Monitoring, Notification and optimization – HPL-2001-96.