SERVICE LEVEL AGREEMENT FRAMEWORK FOR GRID-BASED SERVICES

Tarirai Chani
(200711453)
(B.Sc. Hons. Computer Science)
(Midlands State University, Zimbabwe)

A dissertation submitted in fulfillment of the requirements for the degree

of

Master of Science in Computer Science

Supervisor: Prof. M.O.Adigun

Department of Computer Science,
Faculty of Science and Agriculture,
University of Zululand

2008

Declaration

I declare that this dissertation is my own original work, conducted under the supervision of Prof. M.O. Adigun. It is submitted for the Degree of Master of Science in Computer Science in the Faculty of Science and Agriculture at the University of Zululand, KwaDlangezwa. No part of this research has been submitted in the past, or is being submitted, for a degree or examination at any other University. All sources used in the dissertation have been duly acknowledged.

Signature_____________

Tarirai Chani

DEDICATION

To my family

ACKNOWLEDGEMENTS

I would like to thank God Almighty for giving me the strength, courage, wisdom and faith to start and finish this dissertation. I would also like to thank Him for all He has done for me and my family. He has kept us safe and well at all times. Glory to you O Lord! I would also like to acknowledge and extend my heartfelt gratitude to the following persons who have made the completion of this dissertation possible:

My sincerest thanks go to my supervisor, Prof. M.O. Adigun, for providing me with this opportunity. I would like to thank him for all his excellent guidance, precious advice and endless support over the years during my postgraduate research at University of Zululand.

I am greatly indebted to Johnson Iyilade and Edgar Jembere for the support they extended to me throughout my work towards this dissertation, and for their informed guidance and advice. When I had only the vaguest idea of what research was, what Grid Computing and Web Services were but was curious to learn more, they offered me encouragement, and took time to explain the various concepts involved.

I also greatly appreciate the encouragement that Salah K. Kabanda gave me. She gave me the opportunity to acquire paper writing skills, and took the time to explain the various nuances of the different skills in conference and journal publications.

I wish to thank my many research colleagues at the Center for Mobile e-Services for providing a stimulating and fun environment in which to learn and grow. Divya Pillai, Ijeoma Mba, Yomi Otebolaku, Mcebo Buthelezi, Klaas Kabini, Sihle Sibiya and Sakhile Msane. I thank you all.

I am grateful to my pillar of strength, my best friend, Murimo Bethel Mutanga for being there to support me and love me throughout my research. And most of all for persuading me to enroll for M.Sc at the University of Zululand. It sure was a once in a life time experience, I'll be forever grateful to you.

I would like to thank my sponsors, Telkom for providing me with the opportunity to join one of their Centers of Excellence and funding me throughout my research.

TABLE OF CONTENTS

List Of Figures

LIST OF TABLES

List of Acronyms

BPEL4WS	Business Process Execution Language for Web Services
C-SLA	Consumer-initiated SLA creation framework
GRAAP-WG	Grid Resource Allocation Agreement Protocol Working Group
OGSA-WG	Open Grid Services Architecture-Working Group
P-SLA	Provider Initiated SLA creation framework
QoS	Quality Of Service
ROI	Return On Investment
SLA	Service Level Agreement
SLO	Service Level Objective
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOC	Service Oriented Computing
UDDI	Universal, Description , Discovery and Integration
WS	Web Service
WSDL	Web Service Description Language
XML	Extensible Mark-up Language

ABSTRACT

In this study, a consumer-initiated Service Level Agreement (SLA) framework was developed for Grid-based services in order to address the problems of rigidity and inability to meet consumer return-on-investment (ROI) of existing techniques that are provider initiated.

A model for SLA template selection that accepts consumer Service Level Objectives (SLOs) metrics such as availability, reliability and cost was proposed. The weighted Euclidean distance measure function was then used in the process of selecting SLA templates that meet consumer-initiated criteria from the pool of available service template directory. Thereafter, a performance comparison of the consumer-initiated (C-SLA) approach with the classical provider-initiated (P-SLA) approach using consumer satisfaction, scalability and precision was carried out. The performance model simulation was carried out using Java programming language.

The results obtained from the simulation showed that the C-SLA approach brings more satisfaction to a consumer as compared to the P-SLA approach. For instance, a statistical *z-test* analysis of difference of two proportions conducted on observed results for 900 sample requests returned 327 accepted SLA templates for the P-SLA scheme while 752 was accepted for C-SLA scheme. Also, the result showed that the C-SLA scheme is more scalable than the P-SLA scheme. For example, as the number of requests increased from 100 to 1000, template acceptance ratio increased from 78% to 82 % for the C-SLA scheme and declined from 38% to 33% for the P-SLA scheme. The result further showed that on average C-SLA is 83% precise in retrieving relevant SLA templates while the P-SLA is 60% precise.

It was concluded that C-SLA as proposed in this research provides more satisfaction to consumers, scales as the number of service request increases, and is more precise.

CHAPTER ONE

INTRODUCTION

1.1 Overview

As the move towards service-orientation within the software industry becomes a reality, Service Level Agreement (SLA) has been identified as an important process in the management of contractual agreement between service provider and consumer for Quality of Service (QoS) offered and required during Grid service provisioning and consumption. This research work proposes a framework that considers consumer-requirements during the selection of SLA templates.

Our approach selects and offers the most appropriate template to the consumer based on their QoS preferences. A consumer specifies both hard and soft QoS metric requirements elaborating much on the preferred weights and actual values together with the anticipated budget. This framework reduces overheads created during traditional iterative negotiations across different distributed administrative domains in the commercial Grid. Therefore, negotiation time is also reduced. It is also proposed that the resulting rate of failure to *successfully* complete an SLA creation process will be greatly reduced. In doing so, we wish to achieve two factors, flexibility and high consumer satisfaction. These factors would aid in the uptake of commercial Grids thereby enabling flexible adaptation to changing consumer requirements and also allowing efficient automation of SLA creation process. In this work we focus on the possibilities and impact of the creation of SLA templates being determined by the consumer of the service.

Chapter One begins by motivating the need for consumer-centric SLAs. It then continues by giving a brief background on commercial Grid SLAs with the challenges that call for a consumer-centric SLA creation process. The problem that this work addresses is described in Section 1.3. The rationale behind carrying out this research is presented in Section 1.4. The goals as well as the objectives to be met in order to address the stated problem are presented in Section 1.5. The methodology for addressing this problem is then discussed in Section 1.6 and finally the organization of the rest of the thesis is given in Section 1.7.

1.2 Background

The pervasive nature of the Internet coupled with its reliance on open standards has made it become the backbone of the world economy and has created a new global world of free exchange of ideas, interests, and transactions. Amongst the many benefits of the internet, electronic commerce (e-commerce) perhaps, seems the greatest. It offers businesses unique opportunity of a global connection of buyers, sellers and suppliers. Many businesses were able to take advantage of this opportunity by transforming their key business activities to e-business (Kabanda et al, 2007). However, advances in communication and networking technologies are currently facilitating collaboration and interactions among businesses dispersed in different physical locations. Thus, very large and rigid enterprise applications are being substituted by modular, distributed components offered as a service and are consumed on-demand (Masche et al., 2006).

One emerging technology facilitating business collaboration and enterprise application integration is Grid computing (Foster et al, 2001). Although Grid application in research and education has recorded tremendous success, its application in business is still at infancy stage. Grid adoption (in the commercial industries) depends on the ability of the technology to deliver increased business value (Joseph et al, 2004). The business issues relating to the adoption of Grid include key factors such as, reducing operational expenses, creating a scalable and flexible infrastructure, accelerating development time, improving time to market, increasing customer satisfaction and business productivity. In order to achieve these objectives, the utility computing model has been suggested for Grid deployment in the enterprise.

When the utility computing model is applied in a Grid environment, it means Grid services are accessed on-demand and consumers pay only for what is used. Whereas traditional model of Grid usage has focused on big corporations who can afford high performance computing infrastructure, the utility business model makes Grid technology accessible to all on pay-as-you-go, just like other public utilities (e.g water, electricity etc). In particular, this would be of benefit to the Small, Medium and Micro

Enterprises (SMMEs) who are constrained in accessing Information Technology (IT) services due to lack of suitable infrastructures.

In view of the foregoing, the Center for Mobile e-Services, Department of Computer Science, University of Zululand has proposed an architecture known as Grid-based Utility Infrastructure for SMME-enabling Technology (GUISET). GUISET is an e-infrastructure aimed at providing on-demand IT services to the infrastructure-constrained SMMEs. The GUISET architecture (Adigun et al, 2006; Buthelezi et al, 2008), leverages on emerging service-orientation of Grid and evolving utility computing business model for IT services provisioning. The GUISET infrastructure brings together an array of service providers and consumers through a generic middleware platform for service exchange.

Certainly, the key to the success of such an infrastructure is the efficient delivery of IT functionalities that meet consumer's need. As a result, Quality of Service (QoS) constitutes an important factor that must be given utmost consideration to achieve efficient service delivery. In fact, one of the basic requirements of any Grid system is the ability to provide high-level QoS needed for satisfactory consumer experience. Thus, QoS validation must exist as a basic feature in GUISET, as measured by the available resource or business metrics. These metrics include: response time measurements, aggregated event performance monitoring, resource scalability, availability, autonomic features and reliability.

To deal with this QoS requirement, there is the need to rely on a management framework in which the confidence of the consumer is established through a contract with the provider of the service. Such contracts, commonly known as Service Level Agreement (SLA) define the QoS as well as the terms and conditions that a consumer and a provider of a service have agreed upon (Michell et al, 2005). The SLA is an explicit statement of the expectations and obligations that exist in a business relationship between the service providers and the service consumers (Bhoj et al, 2001).

A lot of scholarly work has been proposed in the single service offering context covering areas such as SLA creation (Pichot et al, 2007), negotiation (Hung et al, 2004) and management (Kreger, 2003), together with SLA languages and specification (Andrieux et al, 2005), (Keller et al, 2003), (Tosic et al, 2003) or defining SLAs in commercial Grids (Leff et al, 2003), (Verma et al, 2001), (Sahai et al, 2003). However, despite all these efforts from the research community, very few business entities have practically taken up the use of SLAs as a way of managing expectations, clarifying responsibilities and facilitating communication among one another within the Grid environment (SPC, 1998). In fact, SLAs are by far any near from becoming an operational tool for establishing agreements between providers and consumers (Wieder et al., 2006: Peer Hasselmeyer et al., 2007).

Peer Hasselmeyer et al., (2006) identified low flexibility as the cause of resistance to e-business uptake by SMMEs. They also identified poor usability and high maintenance cost. An empirical study conducted by Aranda-Mena et al (2006) revealed the factors affecting up-take of e-business by SMMEs. These factors included attitudes, lack of drive for up-take, natural risk-aversiveness and also, the need for diffusion, awareness and skills development. Some other reasons include, fear of the unknown, SLA complexity or SLA rigidity. As a result, there still exists a gap between the development of SLAs as a *concept* and as *a widely accepted tool* for commercial Grid uptake. It is the latter that forms the driving force of this work.

The life-cycle of an SLA may broadly be classified into the following phases: *template creation*, *deployment and provisioning*, *enforcement and monitoring,* and *termination* (Dan et al., 2003). For two parties to have established an SLA between each other, they must have agreed on an SLA template. These SLA templates thus, form the basis of the negotiation process. After the signatory parties have agreed on the template, an SLA is established between them. The SLA would, explicitly define the service to be rendered by the provider, the cost of that service to the consumer and the penalties to be incurred by the provider the case of an SLA violation (Leff et al, 2003).

Most SLA templates are defined by the providers of services (Kaminski et al, 2006a; Vassiliadis et al, 2006). However, a closer look at the contents of the template reveals that it defines more of the provider's goals and objectives. As a result, providers offer a limited degree of customization of the QoS metrics and metric levels, endorsing rigidity and disregarding the involvement of consumers in order to ascertain their QoS needs. According to Vassiliadis et al, (2006) and Patel and Darlington, (2007), such forms of rigidity are no longer acceptable in a competitive commercial Grid as they lead to unachievable targets for the consumer and hence, the consumer might end up resisting the provider's service.

SPC Essentials! Newsletter, (SPC,1998) suggested that in order to reduce the risk of consumer resistance and gain competitive edge, it is important that service providers know, before hand, what metrics are most and least important to the consumer, so that they can then, aim toward satisfying the consumer's needs. To ensure a balance of risk and benefits for both the service consumer and the provider, it takes a good appreciation of the impact that various service levels may have on both parties (Trzec and Huljenic, 2003). The traditional way of offering provider-defined SLA templates is becoming less effective as it is rigid to the ever-changing consumer QoS preferences.

The degree of customization offered to the consumer is inadequate as it lacks the flexibilities they require. According to Vassiliadis et al (2006), such flexibilities are due to the fact that, because of proliferation of information through the Internet, consumers are becoming more aware of what they require from the provider, emphasizing more on metrics that improve Return-On-Investment (ROI), and that decisions are based on solid business metrics. In the past, SLAs aimed at exceeding a particular threshold of QoS metrics. For instance, a service with an availability metric of 99.9% would be favoured by a consumer but lately, consumers require QoS levels that are linked to tangible business productivity enhancements. Such a metric would probably be unnecessarily too high and thus, expensive (Kaminski and Perry, 2006a). Inefficient system behaviour of this nature from the provider in a highly competitive commercial Grid would not be desirable. When the provider tries to take control of the agreement creation process - which should include the participation of both parties, the service consumer might end up resisting the provider's service (SPC, 1998). As a result of this change, the type of SLAs sought in the past are now different from those sought presently.

In recognition of the significance of SLAs in service provisioning and the need for the SLA creation phase to be one that is inclusive of consumer requirements, we proposed a consumer-initiated SLA creation process for the GUISET infrastructure described in

(Adigun et al, 2006; Buthelezi et al, 2008). With this our work achieved increased consumer satisfaction, a factor that ultimately enhances and encourages uptake and lessens consumer resistance to commercial Grid uptake. Our work focused on the possibilities and impacts of an SLA template creation process that encompasses the service consumer and concluded that the resulting *rate of failure* to *successfully* complete the SLA creation process is kept minimal.

1.3 Statement of the Problem

Clearly, from the foregoing discussion, there is the need to reduce the rigidity often associated with the creation of the SLA template. Due to technological evolutions, consumers' QoS requirements are also responding to this change. As a result, SLAs being sought presently are different from those sought in the past with consumers no longer being attracted to the high threshold levels in QoS metrics. Rather, consumers are now more interested in threshold values that directly and positively improve on tangible productivity enhancements. Thus, the traditional SLA creation approach where providers solely create or formulate the SLA template is no longer suitable as it leads to unachievable targets for the consumer. Such SLA templates are characterized by limited degree of customization as well as reflecting mostly on provider's objectives. This study, therefore, aims at improving on traditional approaches to SLA creation by incorporating customization into the specification of SLA templates.

In view of the foregoing, this research proposes to investigate the following concepts:

(i) Formulation of flexible SLA templates that incorporates the interest of service providers and consumers.

(ii) Formulation of a method for the SLA template creation process that considers gradual consumer requirements changes.

As a result, the following main research question arises:

(i) How can SLA templates be flexibly selected to reflect individual consumer's requirements?

Sub questions that emanate from this main research question include:

a. How can an SLA framework that attempts to anticipate consumer QoS requirement changes be formulated?

b. How will the most appropriate template be selected among a list of alternatives?

1.4 Rationale of the Study

Service Level Agreements (SLAs) are an essential instrument for service providers to advertise their services' quality, as well as to manage their resources. Service consumers on the other hand use SLAs to formalize guarantees on service quality properties. However, with consumers being more aware of the QoS metrics they require from the provider, it is undeniable that consumer participation is vital to ensure each consumer's needs are satisfied. The success index of any software product is largely determined by consumer acceptance through communication with and getting the consumers requirements right (Marasco, 2006; Begic, 2005). Given this information, the envisaged consumer-centric SLA creation framework will enhance and encourage the uptake and seizure of commercial Grids, which are inevitably the future form of e-commerce. This work hopes that the targeted SMMEs, who are characterized by conservativeness and tight budget constraints, would most definitely see the need to move with the inevitable current trend towards e-services thereby, reducing marginalization by their already established large organization counterparts. Therefore, this research work was an effort at formulating an SLA framework that incorporates consumers in SLA template formulation, closing the gap between the development of SLAs as a *concept* and as a *widely accepted* tool for commercial grid uptake.

1.5 Research Goals and Objectives

1.5.1 Research Goal

The goal of this research was to develop a framework for flexible SLA template creation based on consumer's QoS requirements.

1.5.2 Research Objectives

The above goal was formulated as an equivalent of some lower-level objectives, which were:

(i) To investigate existing approaches to SLA creation and how service consumers were incorporated into the SLA creation process.

(ii) To formulate consumer-centric SLA architectural model based on (i) as a derivative of existing best practices.

(iii) To develop an algorithm capable of effectively selecting from a variety of Grid services the appropriate service based on consumer requirements and preferences and be able to anticipate gradual consumer requirements changes.

(iv) To evaluate the proposed architectural model and algorithm developed in (ii) and (iii) above, respectively against classical provider-centric SLA architectural models and algorithms.

1.6 Research Methodology

The achievement of the above mentioned objectives was possible through the employment of the following methodologies:

1.6.1 Literature search

This aspect of the research included conducting an extensive survey of existing approaches for SLA models in a service-oriented environment. During the survey, clarity was sought from existing works on accepted standards to implementing a commercial grid-based SLA using a Service Oriented Architecture (SOA) approach and also to what extent resource consumers were partaking in SLA creation in order to enable the selection of desired services. The survey exposed various languages used to specify QoS metrics in SLA and the selection algorithms and techniques used for grid or web services selection. Knowledge gained from this survey was utilized in the evaluation of existing provider-initiated SLA creation approaches, understanding of the functioning of web service selection algorithms and also for the formulation/identification of metrics to use during evaluation of the proposed model.

1.6.2 Framework development

Framework development required an analytical approach to evaluating what has already been achieved in SLA research. Then knowledge gained here provided the baseline wisdom needed for the target framework development. Finally, the architectural design of the SLA framework was done using SOA approach.

1.6.3 Proof of concept

As a proof of concept, a simulation of the proposed consumer-initiated SLA template creation framework was conducted. This was benchmarked with another simulation of classical SLA template creation frameworks based on the provider-initiated approach. Benchmarking was conducted using appropriate performance parameters as a way of evaluating the two approaches. These parameters included: *consumer satisfaction*, *scalability* and *precision*.

1.7 Organization of the Dissertation

The remainder of the dissertation is organized as follows: Chapter 2 outlines the background concepts in the area of SLAs, Grid and web services. Chapter 3 analyzes the related literature. In Chapter 4, we describe our proposed consumer-centric SLA creation and QoS-based SLA template selection framework. We present evaluations of the proposed framework in Chapter 5 which describes the simulation layout and then discusses sets of experiments conducted. The dissertation is concluded and some pointers on directions for future work are given in Chapter 6.

CHAPTER TWO

BACKGROUND

This Chapter presents a background of key concepts in Service Oriented Architecture and Grid computing. Grid computing has brought about the ability for consumers to utilize Grid services without having to own them. The Chapter then goes on to introduce SLAs as an approach to clearly state responsibilities and expectations of interacting parties in a Grid environment. SLAs have been introduced in Grid computing to ensure that consumers are guaranteed of the QoS they require for a charge avoiding the incompetence associated with service provision based on "best effort." Applying the SLA however, requires the co-operation from both service provider and consumer. While the process of SLA establishment varies among organizations, in order to realize success in the process, SPC Essentials! Newsletter, (SPC, 1998), proposed signatory parties to take heed of the following facts of establishing an SLA: (1) Use of the SLA as a weapon (2) Confusing the SLA document and the SLA process (3) Having unrealistic expectations from an agreement (4) Omitting the management elements of the agreement (5) Neglecting to manage the implemented agreement and, (6) Not creating the agreement unilaterally.

In Section 2.1, we present Service Oriented Architectures, while Section 2.2 discusses Web Services. Section 2.3 discusses Grid computing and the notion of Commercial

Grids. In section 2.4, we give an analysis of Service Level Agreements and Section 2.5 concludes this Chapter.

2.1 Service Oriented Architecture

The Service Oriented Architecture (SOA) is an architectural approach whereby an application is composed of independent, distributed and co-operating components called *services* (TNGC, 2006). A collection of such services constitutes the application. The services can be rendered within or outside of the organizational physical boundaries and security domains. Furthermore, the various service components can exist on varying platforms and can be implemented using different programming languages.

The key concept of SOA is that the functionality implemented by a service is exposed via a standard-based interface declaration. The implementation details are hidden from the consumers of the service; they only invoke the service based on the operations exposed by these interfaces. The SOA's basic components are elements and the operations messages they exchange with each other. There are three key elements of an SOA: Service Provider, Service Requestor and Service Registry (As shown in Figure 2.1).

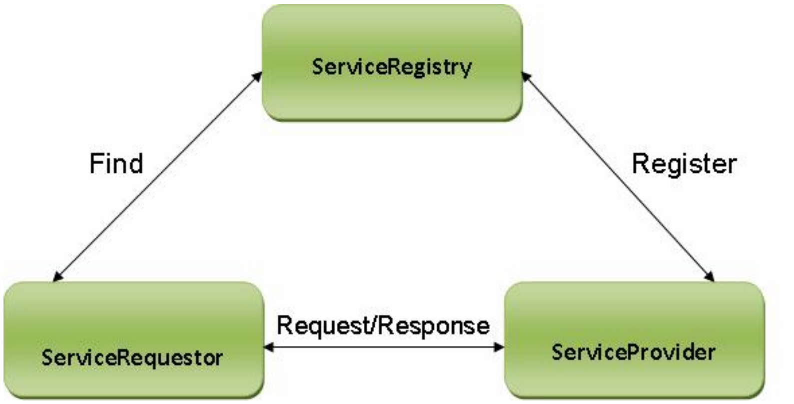

Figure 2. 1: Elements of the Service Oriented Architecture (SOA)

The *Service Provider's* role is to implement and expose a service, create a description for it, publish that service description to one or several service registries, and accept service invocation messages from one or several Service Requestors.

The *Service Requestor's* role is to find a service description that has been published by the service provider to one or more Service Registries, and for using service descriptions to bind to or invoke services hosted by Service Providers. Any consumer of a service can be considered a Service Requestor.

The *Service Registry's* role is to advertise service descriptions published to it by the Service Providers, and providing the means for Service Requestors to search from a collection of service descriptions contained within the Service Registry. Once the Service Registry provides a match between the Service Requestor and the Service Provider, the Service Registry is no longer needed for the interaction.

Application components can have any of the above responsibilities and can also have more than one of these responsibilities. The goal of SOA is to move away from monolithic inflexible applications towards flexible services that allow for massive reuse and on demand access. The SOA is a major advancement in the development, deployment and interconnection to e-business and e-government systems. One interesting way of implementing the SOA is to take advantage of Web services, which can be used in the process of service definition, discovery and execution as discussed in the next section.

2.2 Web Services

Web Services (WS) are application components that communicate using open protocols such as HyperText Transfer Protocol (HTTP), Extensible Markup Language (XML) and Simple Object Access Protocol (SOAP). The Web Services community has investigated requirements for WS architectures (Afzal et al., 2004) and provided a number of relevant standards proposing how Web Services can be described, registered, searched for, accessed or composed: Web Service Description Language, WSDL (Christensen, et al, 2001), UDDI (OASIS, 2003: Curbera et al., 2002), SOAP (Box et al., 2000), BPEL4WS (IBM, 2003) etc.

Web Services also support a more general trend towards Service Oriented Computing, promising to transform enterprise software systems to an orchestral of loosely coupled reusable service components. Providing a better connectivity among business partners, Web Services support not only e-commerce in its narrow scope (which by many is perceived as the electronic transactions among business partners), but also e-commerce in a wide scope (which many call e-Business) addressing all aspects of business operation and support. They are designed to support interoperable machine-to-machine interaction over a network. Web services are distributed, loosely coupled, autonomous software modules that offer specified functionality over web protocols (Tiropanis, 2003). The functionality of Web Services can be combined to offer composite services to human users or software applications. A concept often referred to as Web Service Orchestration.

As an emerging technology for application-to-application remote interactions over the internet, Web Services provide a standard means of communication among heterogeneous software applications regardless of their platforms. The standardization of Web Services and continued adoption by key industry players led to an alignment of the Grid vision with that of Web services and the formulation of the Open Grid Services Architecture (OGSA) by the Grid community. The OGSA is a Grid architecture that is formulated to take advantage of Web services technology especially in the area of messaging, discovery and invocation of services (OGSA-WG, 2003). With this convergence the Grid is now seen as a general-purpose service infrastructure that could meet various needs of the society at large.

2.3 Grid Computing

Grid computing is viewed as the next phase of distributed computing. Built on Internet standards, Grid computing enables organizations to share computing and information resources across departmental and organizational boundaries in a secure and highly efficient manner. Although, the idea of what constitutes a Grid is still a debate in the literature, but a definition widely adopted was the one by Ian Foster. He defined a Grid as a system that "coordinates resources that are not subject to centralized control using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of services" (Foster, 2002). In general, from the viewpoint of the application layer, a Grid can exist in one of three primary types namely: Computational Grid, Scavenging Grid and Data Grid (Jacob, 2003). A computational Grid is a Grid that focuses on setting aside resources particularly for enhancing computing power. A scavenging Grid usually includes large numbers of desktop machines which are scavenged for available CPU cycles and other resources. A data Grid is responsible for housing and providing access to data across multiple organizations.

Grid computing enables research-oriented organizations to solve problems that previously would not have been possible due to their computing and data requirements. Grids also reduce costs by means of automation and improving IT resource utilization. Finally, Grid computing can increase an organization's agility enabling more efficient business processes and greater responsiveness to change. Over time, Grid computing will enable a more flexible, efficient and utility-like global computing infrastructure. Organizations worldwide are making use of Grid computing where available in such diverse areas as collaborative scientific research, drug discovery, financial risk analysis, and product design (Quan, 2006).

2.3.1 Commercial Grids

While traditional Grids assume mutual cooperation between organizations, a commercially viable Grid needs to be governed to replace the assumption of cooperation. In view of the foregoing, SLAs have been proposed as the ideal tool for governing Grid service provisioning and consumption, guaranteeing QoS, and establishing consumer confidence (Mitchell and McKee, 2005).

The provision of services in a commercial Grid always implies a service consumer's consumption of a service that is purchased from a provider. In such a Grid computing environment, resource management systems need to provide mechanisms and tools that allow service consumers and service providers to specify their requirements and achieve their goals. The commercial Grid assumption is that there is a market where players such as service consumers and service providers participate with the aim to meet requirements and make profit (Leff et al., 2003). The use of SLAs can allow the provider to reduce costs and allow them to make more efficient use of their services by being able to plan ahead and, if necessary, commission or decommission new resources when there are spikes or troughs in demand. SLAs help consumers to evaluate their contracted service.

2.4 Service Level Agreements

Service-oriented computing (SOC) is an important focus area for industrial computer systems, as it highlights the extremely important interactions between service provider and service consumer. SLAs and service policies are key issues in SOC. An SLA typically incorporates a time bound and a probability bound on a particular path through the system (Clark and Gilmore, 2007). SLAs would clarify the metric against which the service is being judged, the process by which service provision would be measured, and the penalty to be incurred if the service is not delivered with the agreed level of QoS.

An SLA would define both functional and non-functional guarantees of a service provision. The latter plays a crucial role in service discovery, selection and substitution. It is these non-functional properties that enable a consumer to differentiate between services that provide the same functionality which can fulfill their particular need. Other tasks such as service negotiation, composition and monitoring are also based on non-functional properties of an SLA. The guarantee provided to the service consumer by an SLA would basically be that the services would be rendered in a fully operational and acceptable manner. This is stated in the SLA based on non-functional properties and QoS values. For service providers, SLA allow for ease in planning of resource allocation thereby averting the risks associated with unforeseen legal misunderstandings. The proliferating demand for e-business finds service providers eager to enforce SLAs allowing for autonomous service monitoring and management. The non-functional requirements and their QoS models play a great role in the widespread uptake of SLAs and all other service related tasks (NFPSLAM-SOC, 2008).

2.4.1 Structure of an SLA document

This sub-section describes the structure of SLAs as defined by the two leading SLA standards – Web Service Level Agreement (WSLA) and WS-Agreement. The general structure of an SLA, according to (Keller and Ludwig, 2002; Ludwig et al., 2003) shown in Figure 2.2, contains the following: parties, purpose, validity period, scope, restrictions, SLA parameters, Base metrics (resource and composite metrics) and Service Level Objectives (SLOs).

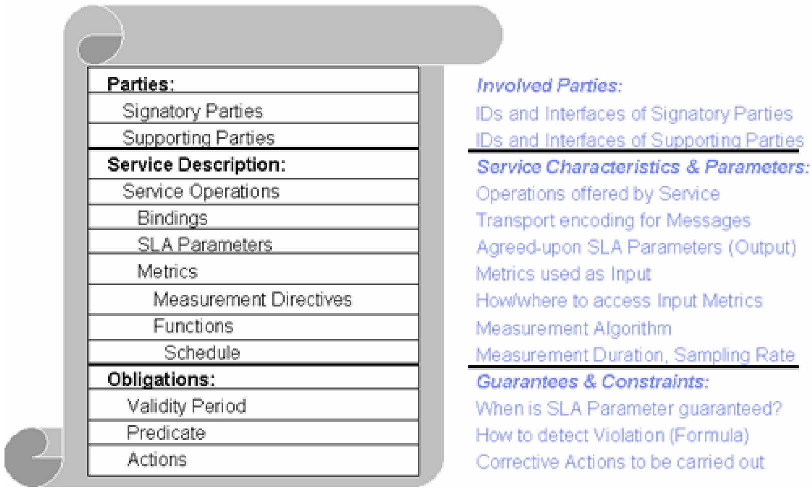

Figure 2. 2: Structure of WSLA (Ludwig et al., 2003)

i. ***Parties involved***: These constitute the signatory parties (the provider and the requestor) and third parties. Third parties may be introduced to act as unbiased judges in the case of violations. They may also be entrusted to measure the service performance and report violations.

ii. ***SLA Parameters*** – These are essential variants of the base metrics offered by the provider in a bid to cater for differences in the needs of requestors

iii. ***Base Metrics*** – A base metric is composed of resource and composite metrics. Base metrics are therefore used to compute the SLA Parameters. Resource metrics are retrieved from managed resources such as servers and routers that reside on the service provider's domain. A composite metrics is the result of combining various resource metrics by a given algorithm.

iv. ***Service level objective/Service guarantee*** – These act as a guarantee that a service would not change its state for a given time. Procedures for handling a violation of the SLA are stated in the SLA. A violation of the SLA occurs when objectives or guarantees are breached.

Figure 2.3 shows the relationship between the SLA Parameters, Composite Metrics and Resource Metrics.

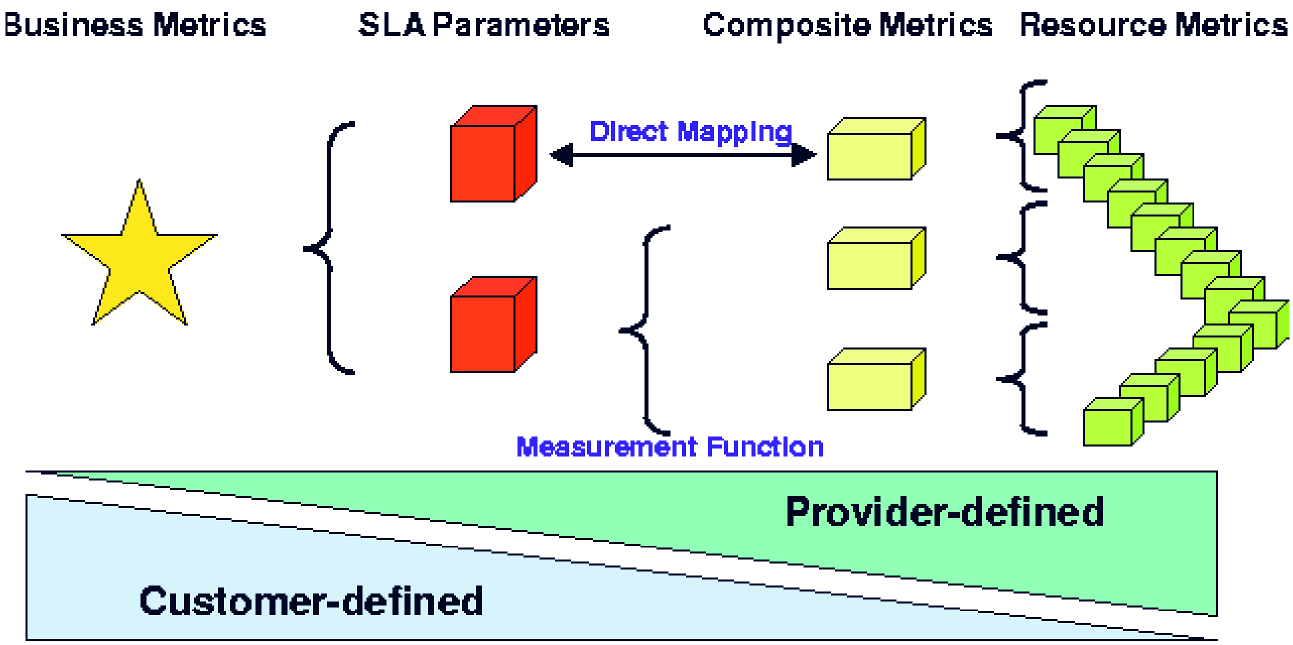

Figure 2. 3: Relationship between the SLA parameter types (Keller and Ludwig, 2002)

The figure shows that business metrics lie more in the domain of the consumers while resource metrics lie in that of the provider. This fact presented by Keller and Ludwig, (2002), best places the consumers not providers, as the key entities holding relevant knowledge about the best QoS metrics (business metrics) that directly impact on their individual businesses' ROI and productivity enhancement. The only knowledge the providers can gather may come, for example, from market surveys and is general to the consumers and not particular or specific to individual consumers.

It proves necessary for the consumers to be equipped with such knowledge about their business as service components like metering and billing are based on solid business metrics (Vassiliadis et. al., 2007). As these solid business metrics affect service cost, not knowing them will entail derailment from timely realizing ROI. This situation then makes it necessary for providers to know beforehand what QoS metric is most and least important to the consumer (SPC, 1998).

On the other hand, according to Andrieux et al, (2005), the structure of an SLA contains the following sections: ***Name***, ***Context***, ***Service Description*** and ***Guarantee Terms***. The "***Name***" section can be non-mandatory but "***Context***" is a mandatory part of the ***Agreement***. The context constitutes the name of signatories, duration and links to any other agreements associated with the main agreement. "***Service Description Terms***" and "***Guarantee Terms***" are the two types of Agreement types that WS-Agreement defines. Guarantee terms contain assurance that the terms of the agreement would be satisfied and *Service Description Terms* holds information of what a service offers.

Agreement Template: It has a similar structure to the Agreement, however, with an additional section called the "***Agreement Creation Constraint***". This constraint section states valid and acceptable values.

Figure 2.4 shows the generally accepted structure of an SLA taken from the WS-Agreement specification. WS-Agreement is a specification that has been defined and accepted by the Grid Resource Allocation Agreement Protocol Working Group (GRAAP-WG). The GRAAP-WG (2003), produces a set of specifications and supporting documents, which describe methods and means to establish SLAs between different entities in a distributed environment.

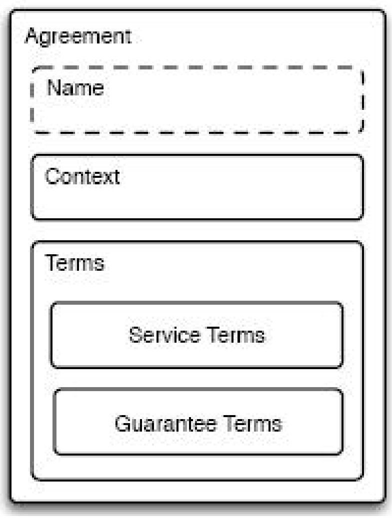

Figure 2. 4: A general Structure of an SLA (Andrieux et al, 2005)

2.5 Chapter Summary

In this chapter we have introduced and discussed basic concepts in Grid-based SLAs. Service Oriented Architecture, Web Services and Commercial Grid concepts have also been introduced. SLAs should define what the consumer wants and what the provider promises to supply through clearly described measurable standards of performance as well as to clearly state penalties encountered by both parties in cases of SLA violation.

Some SLA frameworks have been accepted and adopted by Grid consumers. For example, WS-Agreement has been adopted as a standard specification by the GRAAP-WG. According to (Wieder et al, 2006) the WS-Agreement specification is widely accepted since:

> "*(i) it is the result of the only active standardization effort for a framework supporting interoperable SLA specification, (ii) it is used or considered to be used in many other projects, (iii) it is extensible and adaptable to arbitrary domains due to pluggable term languages, and (iv) due to the possibility to define guarantee terms and business values it might be used in business or service oriented environments thus allowing a smooth migration from research application to business use.*"

In the next Chapter, we critically analyze this standard and other SLA frameworks. Particular attention will be focused on consumer involvement in the SLA template creation process.

CHAPTER THREE

LITERATURE REVIEW

The initial stages of SLA creation involve the creation of SLA templates that represents SLOs the provider might consider agreeing to. In most cases this SLA template creation procedure is solely done by the provider of services. Eventually this has lead to rigidity emanating from challenges related to lack of consumer involvement and expressiveness during this initial stage. In Chapter one, we identified the need to reduce the rigidity often associated with the creation of the SLA templates. Due to technological evolutions, consumers' QoS requirements are also responding to this change. An analysis of how existing scholarship has tried to solve this challenge is conducted in this Chapter.

Specifically, Section 3.1 gives an overview of SLA creation through an understanding of the SLA life cycle. Section 3.2 briefly discusses SLA management. Section 3.3 gives an analysis of how existing frameworks have involved consumers in the creation of the SLA template for negotiation and Section 3.4 concludes this chapter.

3.1 SLA Life Cycle

The life-cycle of an SLA may be broadly classified into the following phases: *creation*, *deployment and provisioning*, *enforcement and monitoring*, and *termination* (Dan et al., 2003). The problem we tackle in this research work concerns specifically the *creation* phase of the SLA life cycle. Figure 3.1 shows the popular model for describing an SLA's Life cycle as described by Telemanagement Forum (Telemanagement Forum, 2005).

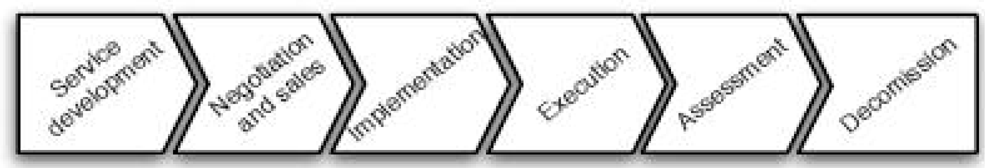

Figure 3. 1: SLA Life Cycle (Telemanagement Forum, 2005)

Taking a closer look at this model would reveal that the life cycle of an SLA also includes that of the service it applies to. For example, the stages of "service development" and "decommission" apply to the service and not the SLA. Due to this deficiency (in this SLA life cycle) and others such as omission of critical aspects of the life cycle, Parkin et al. (2008) proposed what they suggested was a better model shown in Figure 3.2.

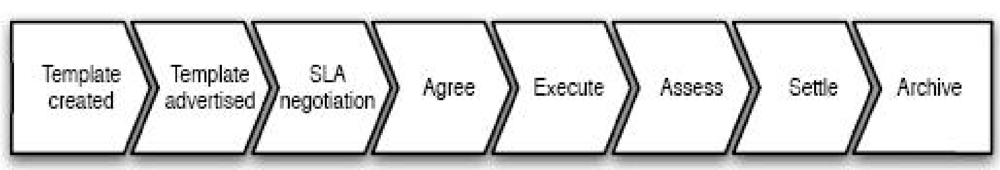

Figure 3. 2: SLA Life-Cycle proposed by Parkin et al., (2008)

They argued that their model is inclusive of all the necessary phases that are for and apply to SLA. Taking each of the stages in turn, the first stage of an SLAs life-cycle is when a general SLA template is formed then advertised by the provider. An individual SLA is negotiated and an agreement is reached on the basis of this template. However, both models depict that the SLA template is created solely by the provider. No involvement of the consumer is discussed at this phase. It is at the negotiation phase that there is some consumer involvement and expressiveness.

Vassiliadis et al. (2006) highlighted that QoS metrics like “99.999% availability of an application X” would no longer be acceptable or attractive to a consumer nowadays as it may not be linked to actual business productivity enhancements for that consumer. Business consumers are now, more aware of what they require from a provider, they emphasize more on Return on Investment (ROI) and are compelled to change their QoS requirements due to the fact that current SLA metering tools bill according to solid business metrics. These factors have thus contributed immensely to the current trend towards the need for a performance driven SLA template specification where QoS requirements are linked to tangible productivity improvements.

Vassiliadis et al., (2006) also reiterate on the need for SLAs to attempt to anticipate the changes in consumer service requirements. Quite a number of SLA frameworks (Dan et al., 2003; Hung et al., 2004) have attempted to address this problem by offering customized options on the SLA template. The degree of customization is, however, usually limiting to the consumer as the provider solely defines the SLA template offer. This has proved to be rigid to the consumer who is contemplating changing QoS metric or QoS level requirements. Due to continued evolution of business consumer QoS requirements, classic SLAs pose an inflexibility threat, which leads to unachievable targets for the consumers.

From the provider's perspective, it is difficult to determine the trend or shift in consumer service requirements due to the lack of insight and knowledge of individual business processes, targets and objectives. As a result, through activities such as market surveys, providers only offer in their SLA templates, QoS metrics and QoS levels that they anticipate would be generally favored by the consumer or by the target market. It is therefore, suggested that the increased lack of flexibility is a result of limited participation of the consumers in SLA template creation phase. Based on these facts, we conclude that there is the need for an additional phase in the SLA life-cycle that caters for consumers and accommodates their individual QoS metrics so that they have an influence in the creation of the SLA template and thus the resulting SLA itself.

Service consumers should play a major role in formulating their SLA template in order to ensure service selection is based on their requirements. As suggested by SPC

Essentials! Newsletter, (SPC, 1998), an agreement cannot be an agreement if both parties do not partake in its creation.

3.2 SLA Management

Active research on SLA management, mainly carried out in the context of single service offering, has covered various areas such as SLA specification and languages, SLA creation, operation, monitoring, termination, and so on. Hence, the importance of SLA management in Service Oriented Computing (SOC) is undeniable (Kreger, 2003). Service Level Management (SLM) is the integrated management of all functionalities in the SLA life cycle. When a consumer requests a service from a service provider, an SLA is negotiated and then a SLA is formed upon mutual agreement.

The service provider must perform SLA monitoring to verify whether the offered service is meeting the QoS parameters specified in the SLA. The SLA monitoring involves monitoring the performance status of the offered service and provides relevant information to the service level management system. In order for the service level management system to verify whether the specified QoS parameters are being met, the system must gather performance data from the underlying network performance monitoring system and map such data to the QoS parameters. The SLA management system is, therefore, expected to deliver functionalities such as:

(i) Offering SLA monitoring services and

(ii) Comparing service performance against the levels agreed in the SLA.

Although necessary in the SLA life-cycle, SLA Management is, however, beyond the scope of this work.

3.3 Consumer involvement in SLA Frameworks

In this section we analyze how existing SLA frameworks have involved the consumer during the SLA template creation phase of the SLA life cycle. To do this, for each framework, we first describe how it works then looking closely at the initial stages of the SLA creation process, we analyze the role played by consumers in the template creation phase.

The frameworks discussed in this chapter have been categorized into either *third party-based* negotiation frameworks or *direct service-interaction* negotiation frameworks. Within either of these two categories, each framework has also been classified as one that deals with or is specific to *single service offerings* or *composite service offerings:*

a. ***Third party based negotiation frameworks-*** This refers to frameworks whose interaction and communication of offers and counter offers between parties is conducted through either agents or brokers.

b. ***Direct service interaction -*** This refers to a framework that allows for direct communication between the consumer and provider without any agents or brokers involved, communicating on their behalf.

c. ***Single service offering -*** This refers to a framework whose SLA creation is based on simple tasks that require invocation of only a single service for them to be successfully executed.

d. ***Composite service offerings -*** This refers to a framework whose SLA creation is based on complex tasks that require invocation of more than one service in order to ensure successful completion of that task.

3.3.1 Third party-based single service offer frameworks

a. Autonomous Broker-based SLA Negotiation Framework – (Hasselmeyer et al., 2007).

The authors proposed the use of a third party negotiation broker which acts as a "go-between" in the whole negotiation process as shown in Figure 3.3. Their argument is that, with the introduction of the Negotiation Broker, both service providers and consumers would be eased off from incorporating and maintaining costly hardware and software within their organizations. The customer, would however, have to have some negotiation components on his side. The authors hold the notion that, a new market niche would be born for organizations that specialize in the provision of brokered services. However, this is likely to increase costs since the broker also charges for the services rendered to consumers. This might increase costs especially if one considers resource-constrained businesses like SMMEs.

The authors called this the NB-negotiation process. In general, the authors argued that the framework is intended to support all possible combinations like NN, BN, and BB. Every participant in an SLA negotiation is free to use a third party broker or to perform negotiation by himself. The business partner will not know the difference and is not affected by this choice. This, however, risks having two brokers in a single negotiation process hence increasing the cost of the negotiation process. The authors argued that the solution guarantees autonomy of individual business entities and allows for gradual migration towards brokered negotiation. How these aspects affect negotiation decisions is not covered.

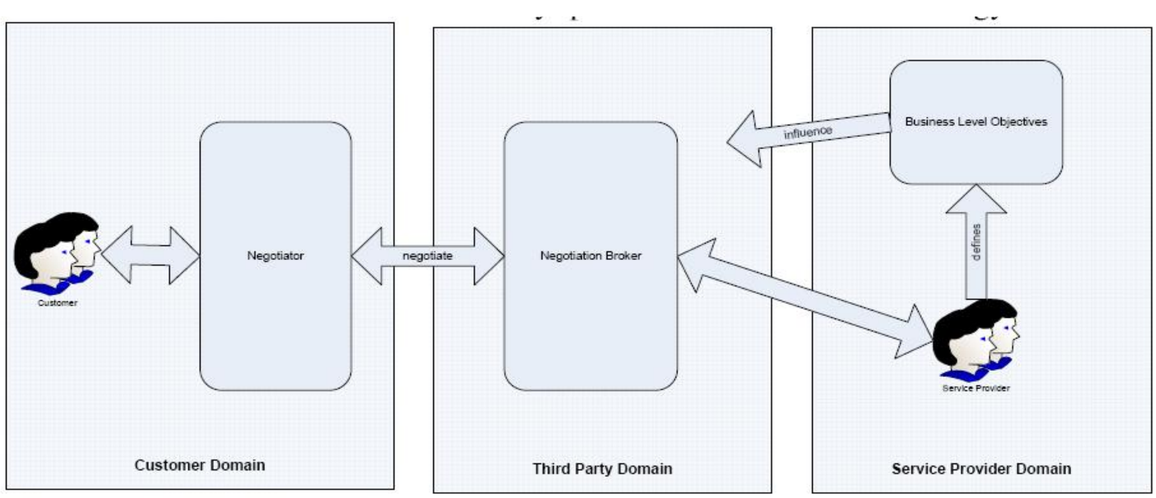

Figure 3. 3 : Broker-based SLA framework (Hasselmeyer et al., 2007)

Service consumers can transfer requirements terms within the SLA. The authors used WS-Agreement as their negotiation protocol. Although WS-Agreement defines an SLA creation data type that allows the specification of requirements (using XML data types) it does not cater for the preferences that the consumer might have on requirements. Requirements are usually stated as an open interval or a set of possible values. Minimum required availability might be 50% and the preferred value could be 95%. For useful negotiation, such preferences must be known. Although the authors claim that new structures were invented to cater for this, it is not clear in their paper how this was done.

The degree of consumer expressiveness in this proposal is therefore, very abstract. The user only specifies requirements without desired preferences and the broker is the one that makes the decision. Both parties (provider and consumer) must agree with the decision of the broker. This, however, might be problematic if either the consumer or the provider do not like the decision or if the decision disadvantages anyone of them. In fact, there is no interactive negotiation in the framework but there is a third party that makes the decision on behalf of the parties involved. Once the broker makes the decision, there is no further negotiations hence the framework is rigid and might not work well in the current business environment where businesses have different needs (hence the need for negotiation).

b. SLA Negotiation Manager – (Kaminski and Perry, 2006a).

Kaminski and Perry, (2006a) proposed to automate the creation of an SLA from a set of Service Level Objectives (SLOs), utilizing software agents and assuming a social order function through incorporation into the process of decision making. Their goal was to achieve automation of SLA development and creation through the use of intelligent agents. They developed a negotiating tool (SLA Negotiation Manager).

The Negotiation Manager is a truth-based system and has an overall system objective of calculating an efficient cost-gain relation. The negotiation system is interactive to aid the service provider to develop and assess an offer to the client (Figure 3.4). Each negotiation begins with the customer selecting one service offer from a pool of predefined service packages. These services are usually packaged based on service price, delivery, quality etc. The initial offers can be pre-defined and stored in a repository or they can be automatically generated by using existing SLOs and current system's state.

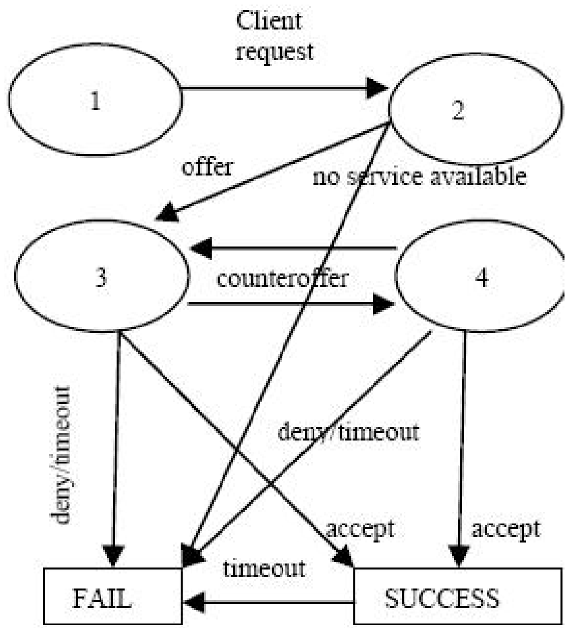

Figure 3. 4: Negotiation Process State Diagram (Kaminski and Perry, 2006a).

By adopting this system, the authors claim that the signatory parties can form SLAs and satisfy the need for fast and flexible agreements. However, the fact that their service offer is predefined renders their agreements rigid during creation. Their aim was to provide methods for dynamic, automated SLA creation.

The authors emphasize on the need for SLOs to be realistic, quantifiable (measurable), clear and meaningful, manageable cost effective and mutually acceptable. Their system is based on multiple agent framework where there is one agent per every instance that needs an agreement. Employing these agents has a tendency to impact on the cost which would in the long run affect consumers resulting in them shunning the use of this SLA framework.

Kaminski and Perry, (2006a) made use of the Web Service Level Agreement (WSLA) as a contract language. Their SLA Negotiation Manager makes use of templates proposed in WSLA. This framework offers predefined templates that have a very limited degree of customization. This, therefore, is rigid for the consumer. The authors concentrated mainly on a solution that benefits the provider of services through automating contract creation and then evaluating the success of the SLA. However, the resulting SLA is based on the SLOs of the provider and not much emphasis is made on having the consumer's requirements known.

3.3.2 Third party-based Composite service offerings

a. *Service Level Agreement Negotiation Framework for adaptive Service Composition – (Yan et al., 2006).*

The authors proposed a framework aimed at guaranteeing end-to end QoS requirements for service composition by supporting autonomous establishment and maintenance of service level agreements. Interconnected SLAs are grouped and maintained for a service composition, through autonomous agent negotiation. To enable this, they also made use of intelligent agents that operate as a unit on behalf of both the consumer and the provider to negotiate SLAs. In addition, this framework supports adaptive SLA re-negotiation in the dynamic and ever changing service environment. Their SLA negotiation framework involves two aspects. One aspect is the negotiation between the service consumer and one or many service providers for QoS constraints of a single service in the composition. The other aspect is the coordination of negotiation for multiple services to ensure end-to-end QoS. The authors' comprehensive framework that addresses these two functional aspects is as shown in Figure 3.5

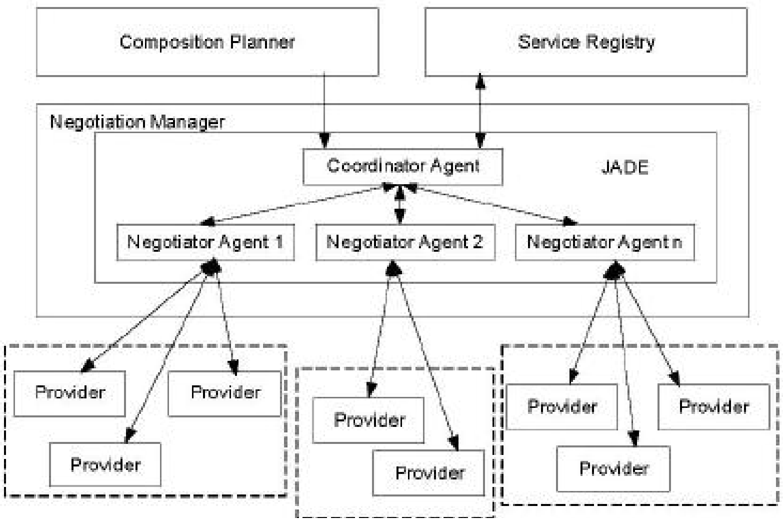

Figure 3. 5: SLA Negotiation Framework (Yan et al., 2006)

The negotiation capability of the service consumer is supported by a Multi-Agent System (MAS). This MAS consists of a Coordinator Agent (CA) and a set of Negotiator Agents (NA). The CA is responsible for the negotiation for the service composition as a whole. It interacts with the composition planner to receive the service composition definition. The use of intelligent agents can be costly as it will require powerful technology with high memory capacity. These costs could be propagated to the consumer through high price for service consumption. This situation is not desirable especially with infrastructure and budget constrained SMMEs in mind. This work is also centered on workflow management where task completion requires service composition and not single service offerings.

In their paper, the authors did not clarify to what extent consumers' preferences are captured. It is also not clear whether this framework allows for customization or flexibility when it comes to capturing consumer requirements. As their work encompasses re-negotiation this implies that consumers are presented with a proposal to which they make counter offers for negotiation and re-negotiation until an agreement is reached. This process can prove to be time-consuming hence costly for the consumer considering time is money in the business arena.

3.3.3 Direct Service interaction single service offerings

a. *The Web Service Level Agreements (WSLA) Framework - Keller and Ludwig (2003).*

The report introduced the novel WSLA framework for specifying and monitoring SLAs for Web Services. Although WSLA was designed for a Web Services environment, it is applicable as well to any inter-domain management scenario such as business process and service management or the management of networks, systems and applications in general. The WSLA framework consists of a flexible and extensible language based on XML Schema and a runtime architecture consisting of various SLA monitoring services, which may be outsourced to third parties to ensure maximum objectivity. Although outsourcing third parties is used to ensure maximum objectivity it also poses as an additional cost that could be prohibitive to SMMEs. Hasselmeyer et al. (2006) acknowledge that high maintenance costs is one of the many reasons SMMEs do not take up proposed Grid SLA solutions in e-Business.

Figure 3.6 shows the WSLA framework and its atomic building blocks, namely the elementary services needed to enable the management of an SLA throughout the stages of its lifecycle. In an effort to avoid ambiguity during SLA creation, WSLA enables service customers and providers to specify the SLA parameters and the way they are measured, and relate them to managed resource instrumentations.

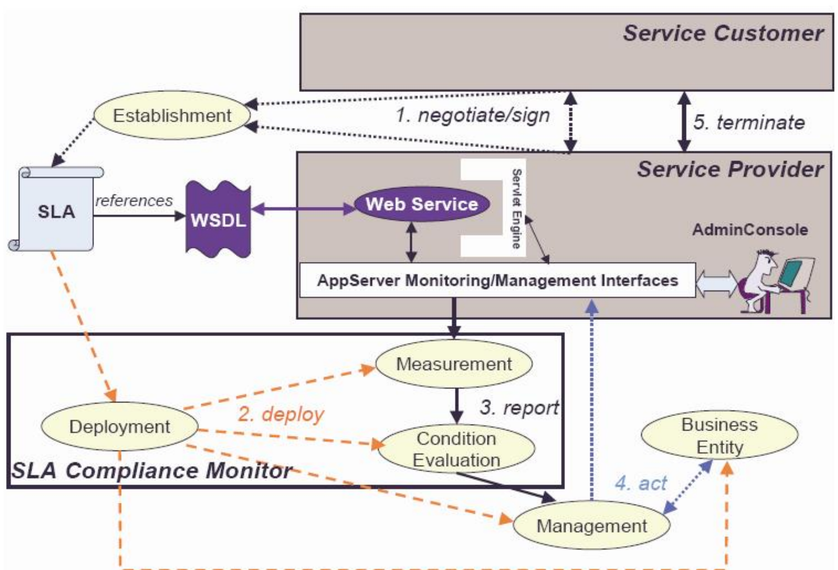

Figure 3. 6: WSLA Services and their interactions (Keller and Ludwig 2003)

This process is rather quite complex and/or time consuming for the service consumers who might not be technically sound enough to know how they want SLA parameters to be measured and then go on to further tie these parameters to managed resources. The consumer is also required to define the data collection algorithm which can prove to be a complex activity as well. In order to ensure an automated process, they propose the use of SLA templates that include the use of several automatically processed fields in an otherwise natural language written SLA. The main concepts of WSLA are *parties*, *service definition* and *obligations*. These are utilized in WSLA templates and contracts. WSLA contracts contain the SLA parameters and SLOs formed based on the WSLA template offered to the consumer. WSLA template consists of two parts: first part provides a partially filled contract that defines basic characteristics (e.g. who the parties are).

The second part extends the first one with an “offering document”, which defines constraints for the template. Constraints for SLA parameters can define a range of or list of acceptable values to limit negotiation. Our work acknowledges this IBM effort, but participation of the consumer is only considered at a later stage in the SLA creation. The consumer is restricted to what the provider perceives is needed by the consumer. Also the customization process is very complex and tedious as the consumer is loaded with the burden of defining data collection algorithms as well as arbitrary input parameters. Such unnecessary activities affect the usability of the system resulting in lack of consumer uptake.

The flexibility of this approach is limited and only suitable for a small set of variants of the same type of service using the same QoS parameters and a service offering that is not likely to undergo changes over time.

b. *Web Service Agreement Specification (WS-Agreement) – (Andrieux et al., 2005).*

The Global Grid Forum (GGF) published the Web Service Agreement specification (WS-Agreement), which is an XML variant for specifying an agreement between a service provider and a consumer, and a protocol for creation of an agreement using agreement templates.

The objective of the WS-Agreement draft specification is to provide an organization independent and standard way to establish and monitor SLAs. Another objective of WS-Agreement is that it should be interoperable with other negotiation protocols. The specification comprises three major elements: (i) a description format for agreement templates and agreements, (ii) a basic protocol for establishing agreements, and (iii) an interface specification to monitor agreements at runtime (Wieder et al., 2006).

In the WS-Agreement specification, an agreement between a service consumer and a service provider specifies one or more Service Level Objectives (SLOs) which state the requirements and capabilities of each party on the availability of resources and service qualities. For example, an agreement may provide assurances on the bounds of service response time, service availability, or service reliability.

WS-Agreement also specifies a very simple negotiation protocol. More of a "take it or leave it" protocol, that does not allow offer refinement (see Figure 3.8 for the Interaction Model). This is a very limited interaction model as a result it limits consumer expressiveness immensely.

WS-Agreement defines three types of documents: *Agreement templates*, *Offer* and *Agreement*. A service provider publishes an agreement template for the services provided. The template may contain service characteristics, guarantees, as well as creating constraints, which would specify the desirable range of some values. A consumer would retrieve and "fill in the template." The filled template is sent as an offer to the provider. The provider checks the offer against the template to make sure that there is compliance and decides whether to accept or reject depending on his capacity.

i. WS-Agreement Interaction Model

Figure 3.7 depicts the WS-Agreement interaction model which defines that consumers  can request agreements from service providers  by issuing an agreement request based on available agreements templates, which, if accepted, results in new agreements.

The interaction protocol as specified in the WS-Agreement specification only allows for a single "request, accept" interaction, in which the requesting party receives either an accept or reject message from the providing party as a response to an agreement request. This is a very limited interaction model.

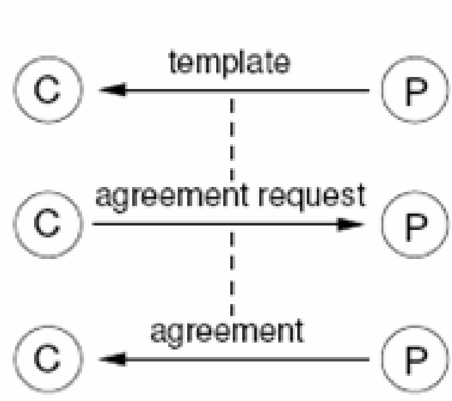

Figure 3. 7: Interaction of the WS-Agreement Protocol (Mobach et al., 2006)

Consumers can only fill in some customizable parts of the template that is already prepared for by the provider. As QoS parameters are now playing a key role in selecting Grid resources in order to ensure alignment to consumer needs as well as realizing optimized resource usage efficiently, (Truong et al., 2006) an SLA language like WS-Agreement should allow the specification of these QoS needs in order to derive more satisfactory SLA templates and subsequently make use of the template to select the satisfactory Grid service.

This is essential in order to specify tradeoffs between the parameters. For instance, a consumer could be interested in making tradeoffs between response time and availability, or between size and quality of the data. Another interesting possibility for relating QoS parameters is for consumers to give them a weight or level of importance, allowing customers to establish their priorities. This could prove useful to the SLA provisioning system as it asserts the priority or level of importance attached to that QoS parameter by the service consumer.

a. *Dynamic SLA-negotiation Framework based on WS-Agreement – (Pichot et al., 2007)*

As SLAs are basic building blocks for Grid resource orchestration and distributed resource management the authors showed how a bilateral WS-Agreement based negotiation process can be used to dynamically negotiate SLA templates. For this, they propose a simple extension of the WS-Agreement (Andrieux et al., 2005) protocol in order to support a simple offer or counter-offer model. They did not address extensions that enable support for auctions-based negotiation in WS-Agreement. The second relevant part they addressed is the creation of distributed SLAs. They discussed two different strategies to co-allocate SLAs in the Grid which are the one- and two-phase-commit protocols in a distributed resource management domain.

The authors made a minimal extension to WS-Agreement by not negotiating SLAs but by negotiating and refining the templates used for creating an SLA. Their focus was on the bilateral negotiation of agreement templates. To implement their model in the WS-Agreement protocol, they proposed a new function called *negotiateTemplate*. This function takes one template as input (offer), and returns zero or more templates (counter offer). The negotiation itself is an iterative process. Initialization of the negotiation process begins by the negotiation initiator (Grid scheduler) querying a set of SLA templates. From these templates, the initiator chooses the most suitable one for the negotiation process. As this process requires iterative negotiation before an SLA can be reached, it can be very costly or slow. The cost of negotiation not only entails monetary costs but also resources usage (i.e. resources needed to facilitate negotiation between disparate administrative domains e.g. network), time spent during negotiation.

The authors' work still does not take heed of the importance of knowing the initiators QoS requirements. This is probably the reason why this framework relies much on iterative negotiation. The fact that the initiator wishes as much as possible for the resulting SLA template selected to be one which reflects his requirements and the service provider doing the same brings about a negotiation indefinite loop.

No clear indication is given on the extent to which the degree of customization is catered for. It is evident, however, that the initiator's starting SLA template is one that is already predefined and hence rigid therefore instituting the need for costly iterative negotiations.

3.3.4 Direct Service interaction composite service offerings

a. *WS-SLA: The integrated Service Level Agreement Framework- (Sun et al., 2006).*

Sun et al, (2006) propose a framework of Service Level Agreements based on workflow management. An application scenario instance of SLA system architecture by using Business Process Execution Language (BPEL) of service oriented architecture was presented. The framework accomplished automation of business process of SLA-driven operating supporting system (OSS) and provide flexibility of and dynamics of service management. The framework of integrated SLA is shown in figure 3.8

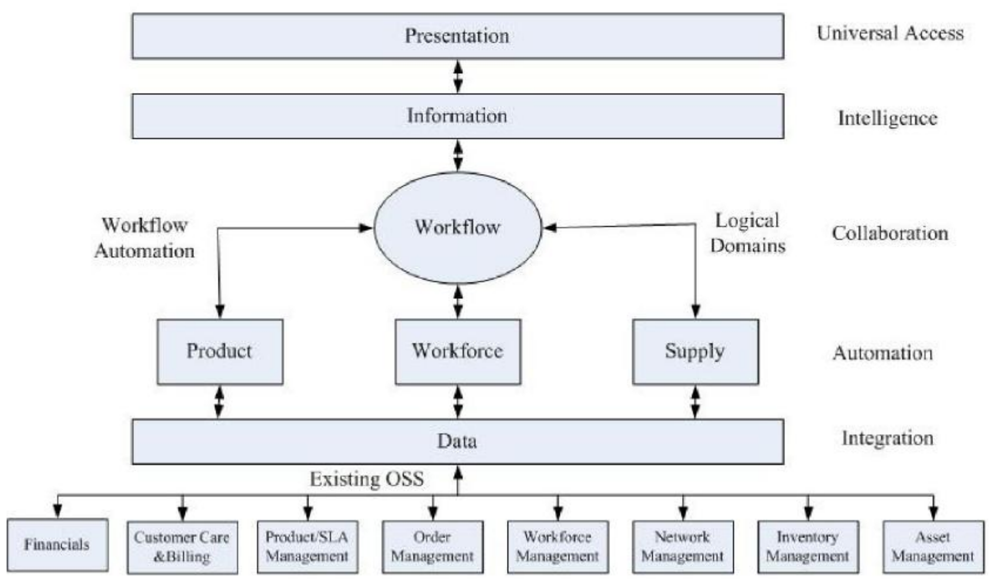

Figure 3. 8: The integrated Service Level Agreement Framework (Sun et al., 2006).

Below we describe the different layers of the framework as described by the authors as follows:

The *integration layer* brings together participants, applications, processes, and interfaces seamlessly into a common management environment. Integration occurs at four levels namely organizational, process, data interchange, and the collaborative level, which combines the integration of the other levels into a single work flow.

The *automation layer* provides computerized support for the procedural automation of workflows.

The *Collaboration Layer* supports many different entities and parties working together in series, parallel, or in a combined manner. The layer also controls the execution of each individual system and manages dataflow among each individual system and ensures the ability to seamlessly support both human and electronic participation in collaborative efforts.

The *Intelligence Layer* provides 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.

The *Universal access layer (Portal Layer)* ensures that both human and electronic workflow participants, can securely access the environment in a convenient user-friendly manner. The portal provides the functions such as User and Profile Authentication, Graphic Workflow Editor, Workflow Execution and Resource Information.

This framework takes into consideration the creation of SLAs for service composition and does not consider single service requests. As a result consumer input is in the form of descriptive workflow through graphical or text based XML editor. The use of such graphic tools may require expensive hardware and software that SMMEs cannot afford.

No mention is made on how consumer QoS requirements as well as preferences are captured and handled. However, reference is made to the infamous Tele management Forum's SLA life cycle (discussed in Chapter 3.1). This life cycle is rigid as it emphasizes on templates that are solely created by the provider.

2.6 Chapter Summary

The increased significance of SLAs reflects on the changes taking place in the commercial Grid environment. SLAs provide one means of attracting consumers and can contribute to establishing the credibility of service providers by committing to provide guaranteed levels of support with compensation if such guarantees are not met.

Future Grid services would have to meet a number of QoS requirements resulting from rapidly changing markets and technologies. Within this open market of services, the aspects of their customization and instant provision are of fundamental importance (Triec and Huljenic, 2003) and influencing the development of emerging technologies, such as the negotiation protocol WS Agreement (Andrieux. et al, 2005), which is being defined by the Global Grid Forum (GGF) GRAAP Working Group.

The possibility of service consumers originating SLAs in commercial Grid markets has not been fully explored in literature but can result in a decrease in overheads created during a series of negotiations conducted across different distributed administrative domains. These iterative negotiations are usually an attempt for the SLA template offered to the consumer (by the provider) to reflect more on what the consumer requires of a service.

Naturally, the provider's aim is to ensure that the SLAs reflects organizational goals and as a result a lot of time is consumed during SLA creation as the consumer tries to negotiate (by making a series of counter offers) and ensure his own business goals are achieved. This process proves to be time consuming for the business consumer. It is also essential in any grid market for the provider to ensure increased customer satisfaction for the purposes of acquiring a competitive edge. Therefore, issues such as non-performance and failure to meet QoS requirements needs should be avoided where possible.

In essence, it is crucial for the service provider to realize SLA creation as a vital step in the business process. An SLAs focus should be inclusive of the consumers business objectives (Masche et al., 2006). This can only be achieved if the provider values the importance of knowing beforehand what QoS metric is most and least important to the consumer (SPC, 1998).

CHAPTER FOUR

MODEL DESIGN

We have identified the need to achieve a flexible consumer-centric process of SLA creation. This chapter describes the design of a consumer-initiated SLA model proposed in this work. We propose a consumer-initiated SLA life-cycle as well as a QoS-based selection framework that incorporates the consumer, selects and offers the most appropriate SLA template based on the provider's capabilities. Furthermore, this Chapter will address our attempt to respond to the challenges raised in Chapter One, which are to ensure flexible SLA template creation and consider gradual QoS requirements changes of the consumer. In the process of this SLA template creation, it is also the goal of this work to guarantee that an appropriate SLA template is offered to the consumer from a list of alternatives.

Section 4.1 discusses the design criteria, while Section 4.2 reveals the proposed SLA life cycle and Section 4.3 gives the proposed framework based on the design criteria and proposed SLA life cycle. A description of each of the framework's components is presented in Section 4.4, while section 4.5 harnesses these components together to reveal their interaction. Section 4.6 details service selection for creating appropriate SLA template offers and Section 4.7 gives the concluding remark.

4.1 Design Criteria for a Consumer-centric SLA Creation Framework

From our review of literature, we identified the design criteria to take into consideration when designing a consumer-centric SLA creation framework as: flexibility, customization, differentiation and selection.

i. ***Flexibility:*** The dynamic nature of commercial Grid markets prompts the need for the consumers to always want to change their QoS requirements to meet their business objectives and achieve fast realization of ROI. It calls for the service provider to be agile enough to always anticipate these changes in order to ensure consumers are satisfied in any initial formalization of agreements. Doing so ensures that at any point in time the consumer's needs are always factored-in when it comes to the resulting SLA and subsequently the service selection. Flexibility results in the consumer being able to state any desired QoS Level as an SLO. And for both to exploit the benefits of outsourcing, the provider needs to have flexibility during service provisioning and management (Masche et al., 2006). The provider is, therefore, assured of increased consumer satisfaction.

ii. ***Customization:*** In any Service Oriented Computing (SOC) environment, it is desirable for service consumers and providers to obtain guarantees on the services that they require and offer respectively. It is essential for the consumer to be able to request service levels for the service in such a way that they are not predefined on the SLA template but can be custom-made. A commercial Grid environment should thus, ensure service management systems provide mechanisms and tools that allow service consumers and providers to specify their requirements and achieve their goals (Patel and Darlington, 2007). Services have to be acceptable to the consumer. A few years back this meant that the metrics for the services should be above a specific threshold (e.g. availability of 99.999%). Service acceptance is nowadays related to parameters such as ROI. As a result service providers should allow for a wider degree of customization that allows the consumer to express the exact QoS level required. QoS requirements of consumers should, therefore, be the primary criteria for initiating the template creation and selection process.

iii. ***Differentiation*** – Ideally, for providers to achieve the criterion stated above, there should be many levels for the same service and the levels would differ in QoS and cost. Levels of service can be pre-defined for the services of the same type and the same level of service can be used by many consumers. The existence of a number of service levels and performance metrics for each service results in multiple SLOs for every service. Each company should set SLOs that support business needs in order to automate the preparation of an effective SLAs (Kaminski and Perry, 2006b).

iv. ***Selection*** – In order to ensure that business consumers meet their targets, it is ideal for the provider to take an interest in knowing consumers' specific requirements. These being used as the basis of SLA template selection would eventually result in the flexible, less tedious and precise selection of the appropriate SLA. Inefficiency is also greatly reduced as consumers are not flooded with irrelevant SLA templates and information during this process. As a result usability problems are eliminated. The adaptation to the specific consumer's notion of utility will result in higher satisfaction and thus in higher quality of service.

It is believed, in our work, that considering flexibility, customization and differentiation during drafting of an SLA template ensures appropriate selection of SLA templates, enhancing accuracy, consumer satisfaction, and acceptance of SLA creation. This would ultimately enhance uptake of SLAs within commercial Grids.

4.2 Proposed SLA Life Cycle

One of the concerns raised in Chapter One was "how a method for SLA template creation that considers consumer's gradual QoS requirements changes can be formulated?" To address this concern, we propose that the initial step to the SLA life cycle be the specification of consumer requests. From the work reported in Parkin et al., (2008) and discussed in Section 3.1 of this work, we propose a more inclusive model for the life cycle of an SLA shown in Fig. 4.1:

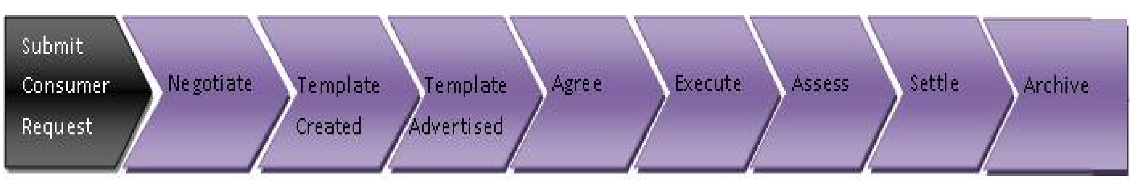

Figure 4. 1: A more inclusive SLA Life-cycle with consumer request

In this work we still acknowledge the fact that the first stage of an SLAs life-cycle is when an SLA template is formed. However, we suggest that, the formulation of this SLA template should not be done solely by the provider, but should also involve the consumers of the services. The first step to achieve this would then be for the consumer to specify desired QoS metrics and their levels as well as the weights attached to that metric. We also refer to the weight as the consumer's "*preference*" towards a particular SLO. As a result the SLA template is no longer general but specific to the consumer. In this case, it is no longer an individual SLA that is being negotiated and agreed upon but, the SLA template is being negotiated, agreed upon which then results in an SLA.

4.3 The Consumer-initiated SLA (C-SLA) Creation Framework

To reduce rigidity, we have to address the need for SLA template creation that considers the impact of gradual quality of service requirement changes. Service consumers should play a major role in formulating their SLA template. Our approach is to support customization and incorporating consumer input and aid automation in the creation of the SLA. Having identified the need to involve consumers in the creation of their SLA template the framework depicted in C-SLA was proposed as shown in Figure 4.2.

4.3.1 Description of the Framework Components

i. ***Service*** – this refers to the registered resources the consumers wish to utilize from the grid resource or service pool. Examples include: online credit card checks and online stock quotes.

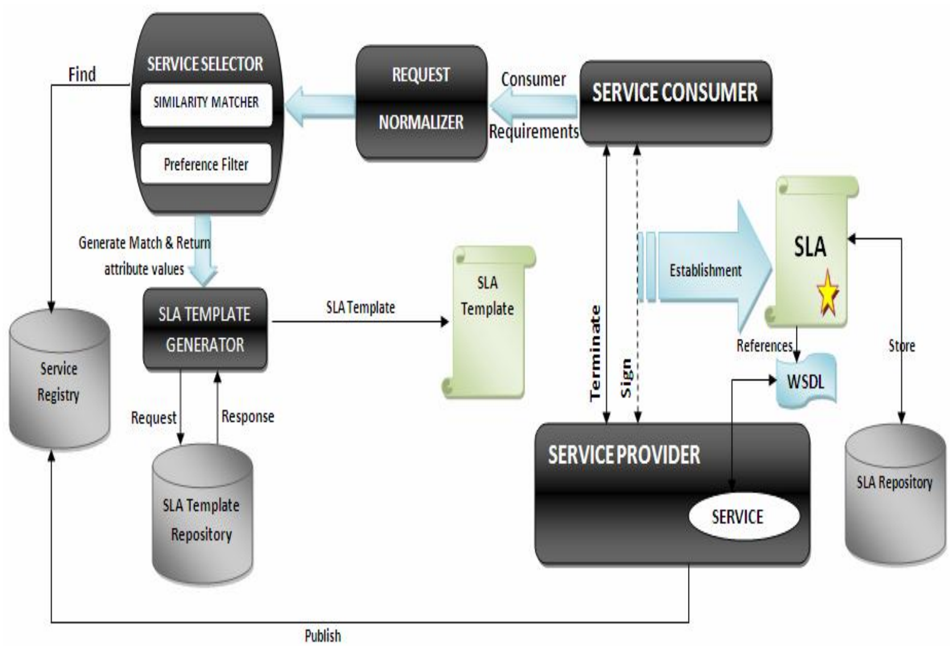

Figure 4. 2: Consumer-initiated SLA creation framework (C-SLA)

ii. ***Service Registry*** - This is a repository where service providers register their services. We assume that the providers would also supply details of the QoS metrics and that they will not publish false QoS values for one reason or the other.

iii. ***Service Consumer*** – A service consumer could either be an application or another service that wishes to utilize services in the commercial grid for the purpose of executing, completing and achieving a particular task. The consumer initiates the SLA creation process by first submitting the QoS metric levels or SLOs to be met for our already assumed QoS metrics. In our case, for simplicity purposes, we assumed 4 (four) metrics including *availability*, *reliability*, *response time*, and *cost*. However, other QoS metrics could be easily added to our framework without changing the methodology. We have also assumed that SLOs are numeric in nature. For metrics like *availability* and *reliability* assumed here, the consumer would have to clearly state in percentage what level is required while response time requirement would be expressed in seconds and *cost*, in Grid dollars (G$). We therefore, assumed that the consumer is economically rational and is aware of its required SLO for the successful and satisfactory completion of its task. For these SLOs specified, a *weight* should be attached to each of them. This weight determines whether that particular SLO is a hard or soft requirement. It shows the preference the consumer has towards a particular SLO.

In addition, the consumer is required to state (as input together with desired QoS levels) its individual "*Satisfaction Threshold*" for each request i.e. to what extent does the resultant match have to be precise? *Satisfaction threshold* is expressed as a percentage. This metric delineates the point at which the consumer derives satisfaction from a given SLA template. By comparing the *Satisfaction Threshold* and the *consumer utility* of the resultant selected SLA template, this should assist in determining the satisfaction consumers derive from the selected SLA template. Appendix I shows how this comparison was done. When the provider selects the best service it is presented to the consumer as an SLA offer. The acceptance of the SLA template offer lies with the consumer. Since the acceptance of the template is given by the consumer, the provider's service might suffer a denial-of-service attack as the consumer might never send back the accept or reject offer. Services will be "locked" or "reserved" and unable to be offered elsewhere. To avoid this, an expiration time was set from the time the template offer is sent to the consumer. At this point of expiration, the SLA creation process is aborted and the reserved services are released back into the Grid service pool.

iv. ***Service Provider -*** The service provider registers its service in the service registry. Along with all the other information about that service, the provider should also include the QoS metrics and metric levels the service offers. With that we assume that the service providers will publish true values of QoS for each service variant and these values are to be numeric in nature. We also assume that, in order to allow for service differentiation, each service has service variants that differ from each other by the QoS attributes and attribute values that they posses. The number of QoS metrics or attributes a service offers as well as the QoS levels it offers differentiates the service variants.

v. ***Request Normalizer*** – Due to the fact that QoS metrics do not fall in the same range or scale and are not using the same measurement units, this component compensates for this and normalizes consumer input as well as provider capability metrics using min-max normalization. Min-max normalization performs a linear transformation on the original data (Han and Kamber, 2006). When a request is sent, the *Request Normalizer* creates a QoS matrix (shown in Figure 4.3) to determine the appropriate available service variants determining also provider capabilities. This process, therefore, sets the constraints of the available service variants of a particular provider.

$$S_{QoS} = \begin{bmatrix} S_{11} & S_{12} & S_{13} & S_{n} \\ S_{12} & . & . & . \\ S_{13} & . & . & . \\ S_{1m} & . & . & S_{nm} \end{bmatrix}$$

Figure 4. 3: QoS matrix

Where **S** = *{all services in a grid system}*

S = $\{S_{ij}\}$

where $i = 1,..,n$ is the service index,

$j = 1,.., m$ is the variant index within a service.

Each QoS metric is normalized by scaling its values so that they fall within a small specific range, such as 0 and 1. Normalization, therefore, eliminates biasness towards metrics that fall within a larger range allowing for a more precise service selection process. A QoS metric that optimizes an economically rational consumer's utility when its value is minimal (eg. cost, response time) would be normalized by minimization whilst those that optimizes an economically rational consumer's utility when its value is maximal (eg. reliability, availability) would be normalized by maximization. A more elaborate normalization procedure is illustrated in Section 4.3.5

vi. ***Service Selector*** – This component utilizes the normalized request to search from the service registry the appropriate service to meet the consumer's requirements. Since all our values are numeric in nature, there exists an analogy between the specifications of distance functions and the nature of the problem raised by our work. Therefore, the classical Euclidean distance measure function (Han and Kamber, 2006: Dunham, 2003) is used for the matchmaking process. A distance measure function is a number that is assigned to a pair of points in a space which indicates dissimilarity between the points. However, since our work encompasses the expression of preferences towards the particular metrics, the Weighted Euclidean distance measure becomes the more appropriate solution to the problem.

The selection algorithm procedure is detailed in section 4.3.5 and represented in Figure 4.5. For each service variant, the **Similarity Matcher** uses a technique based on similarity distance measures to find the relative distance between the consumers request and the provider's capability**. Preference Filter** ensures that the hard and soft constraints on the preference of the service requested for by the consumer are adhered to during the selection of the service. The processes involved in the service selector are detailed section 4.4

vii. ***SLA template*** - the SLA template comes to being if and only if the appropriate service has been selected and the corresponding QoS values or SLOs have been appended on to a generic SLA (gSLA) template. The difference between a gSLA template that we speak of here and an SLA template as defined by Keller and Ludwig, (2003) (discussed in Chapter 2 section 2.4.1) is that the gSLA does not contain service guarantee i.e. the SLOs. These can only be obtained, after an appropriate consumer service has been selected. Once the SLOs have been obtained, they are then appended to the gSLA template which will then be displayed to the consumer as an SLA template for acceptance or rejection.

viii. ***SLA template repository*** – this component contains generic SLA (gSLA) templates. However, unlike the SLA template described by Keller and Ludwig (2003), gSLAs are without the service level specifications (SLSs) of the service to be rendered. These would later on be appended by the *SLA Template Generator* after an appropriate service has been identified and selected.

ix. ***SLA Template Generator*** – this component inserts or appends the service level specifications (SLSs) of the selected service as well as the information about the signatory parties onto a generic template. This generic template contains all the components of an SLA template as described by Keller and Ludwig, (2003) with the exception of the SLSs and the signatory parties. These would later on be appended by the SLA Template Generator. The SLA template becomes whole only after the SLSs have been appended. This SLA template is then displayed to the consumer as an offer for approval, rejection or termination.

x. ***SLA*** – An SLA results from the signing of the SLA template by both parties. By offering the SLA template the provider is signing and acknowledging its capabilities on the template and by accepting the template, the consumer is also signing and hence, agreeing to the offer presented to him by the provider. These events result in an SLA. When an SLA template is accepted, its Web Service Description Language (WSDL) interface can then be exposed and it binds to the particular reserved service. Hence, service deployment and provisioning commences.

xi. ***WSDL*** - Web Service Description Language (WSDL) (W3C, 2001) is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate.

xii. ***SLA Repository*** - Every accepted SLA template is stored here as an SLA. The details of its transaction are also logged here for future use and audit trails.

4.3.2 Assumptions

The operation of the framework is conditioned on the following general assumptions:

ii. QoS metric levels or values are numeric in nature.

iii. Consumers are economically rational.

iv. Service providers are trustworthy and will publish true values of QoS metrics.
v. Each service has variants that are differentiated by the QoS attributes and attribute values they possess.

4.3.3 Operations of the Framework

The Consumer Initiated SLA Template (C-SLA) selection framework works as follows: The service consumer application submits QoS requirements of the desired service to the *service selector* component contained within the generic commercial Grid middleware. This component will select the appropriate service whose QoS attributes best matches or are similar to those requested for by the consumer. The consumer requirements information specification consists of the QoS metric levels and the weights attached to these QoS metrics. These weights are an indication of the consumer's individual preferences on each of these metrics.

Once the appropriate service has been identified and selected its attribute values and other information are captured and inserted into a gSLA template to form one that is particular to that consumer. The gSLA template will contain all the necessary components (Keller and Ludwig, 2003) which include names of the signatory parties, guarantees, obligations, except for the Service Level Objectives (SLOs) which would be obtained from the appropriate service selected by the *service selector* for that given consumer. The resulting SLA template is then displayed to the consumer for approval.

If the consumer chooses to reject the SLA template, it would be requested to change its initial QoS requirement submissions and the selection process is redone or it can choose to discontinue by terminating the process. The consumer can also terminate the selection process if it takes longer than desired. However, if the consumer is satisfied with the selected SLA template it would have to accept it. By accepting the SLA template an SLA is established between the two parties and hence service provisioning commences.

During service deployment conformance to the SLA is monitored and policy enforcers are employed to avert any predicted violations of the SLA. Given that a violation does occur, compensation procedures are taken as stipulated by that particular SLA. After service provisioning, the SLA is stored for record purposes as well as the audit trail for the whole transaction. Figure 4.4 depicts the interaction of the components of the C-SLA template selection framework.

4.3.4 Selection of Service Variants

The service variant selection is the step taken by the service provider of deciding on the best variant offer to the particular consumer's request to be reflected on the recommended SLA template. Services should be selected to match the consumer's requirements, to be reflected and offered on the SLA template. According to Lamparter et al., (2007) for selection to seamlessly flow, there is need for a means of communicating offers and requests between parties and an appropriate algorithm for ranking the service offers according to the requests. The agreement creation starts off with a set of consumer QoS requirements rather than provider offers. The requests include QoS metrics level requirements and their weights together with an anticipated budget.

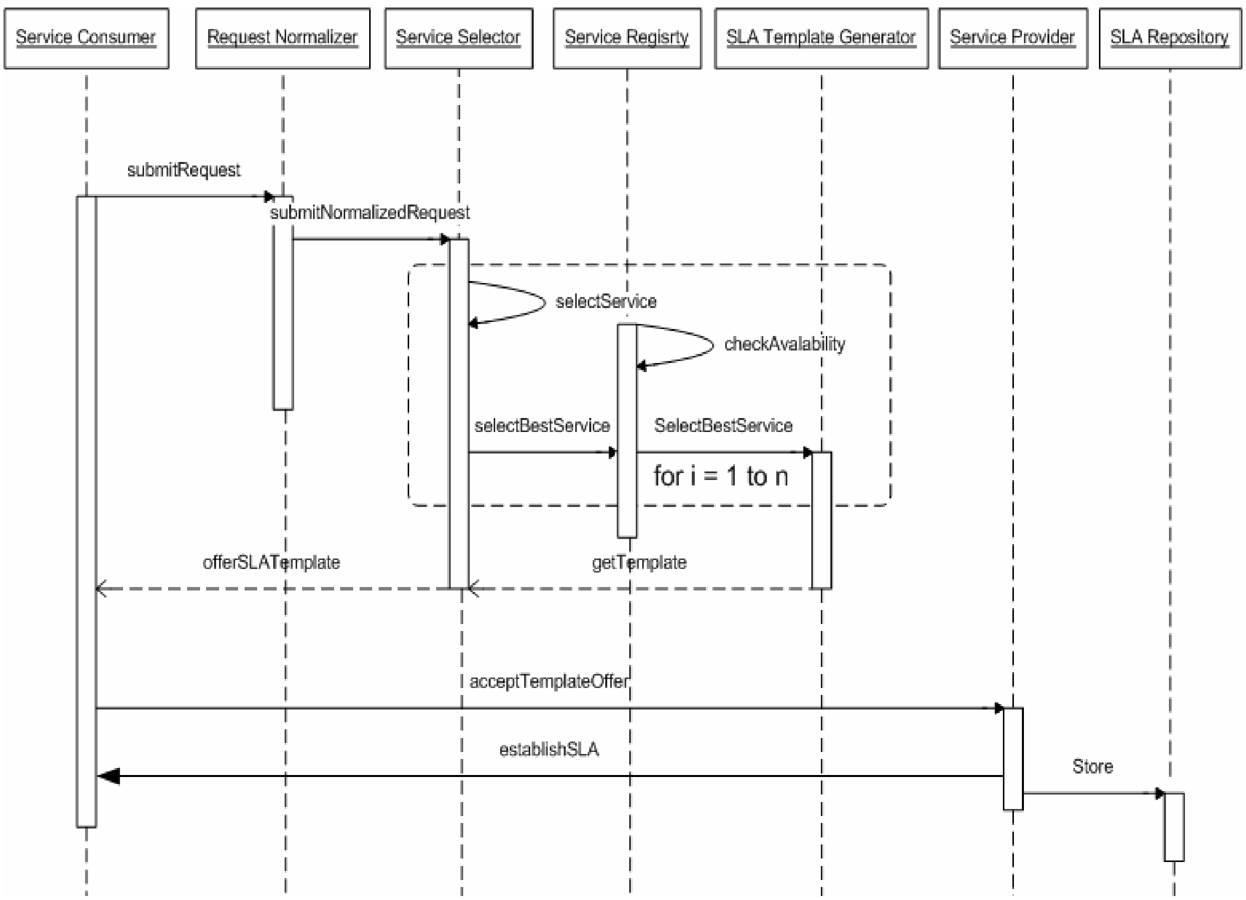

Figure 4. 4: Framework Component Interaction

It is based on these requests that matching and selection of the appropriate service variants will be done. This enables the provider to offer an appropriate SLA template that will maximize the consumer's satisfaction thereby minimizing the rate of failed attempts to create an SLA between the two parties. Communication of the two parties will be by means of an SLA template from which an SLA would be established upon agreement to the template by the signatory parties. This communication shall consist of consumer service requests and provider service offers.

4.3.5 SLA Template Selection Approach

In general, a SLA may include a large set of parameters, referring to different kinds of QoS metrics (e.g., *response time*, *availability*, *reliability* and *cost*) and different ways of measuring them (e.g., averaged over some time interval, individual) (Dan et. al, 2004). In our work, we consider SLAs concerning the average values of QoS attributes and, for the sake of simplicity, we restrict our attention to the following four attributes of which other attributes could be easily added to our framework without changing the methodology:

i. ***Availability:*** the probability that the service is available when invoked;
ii. ***Reliability:*** offers available and properly working components that can get an accurate outcome for a specified amount of time.
iii. ***Response time:*** the interval of time elapsed from the invocation to the completion of a service.
iv. ***Cost:*** the price charged for each invocation of a service.

Based on the analogy that exists between the problem that we solved and the concept of distance measure between two objects, we have chosen the classical Euclidean distance measure for our algorithm. Matchmaking problem meets the question of distance measure between objects, there are many approaches to measure dissimilarity between any two objects based on their numerical or semantic closeness. However, to incorporate the issue of using weights with QoS properties in QoS matchmaking algorithm to present consumer's preferences towards specific QoS properties we chose the Weighted Euclidean distance measure for our algorithm. We hereby describe a mathematical model of our proposed solution as follows (Figure 4.5 gives a summary of this model):

Let **S** = {all services in a grid system}

Then **S** = {S_{ij}} where i = 1,.., **n** is the *service index* and j = 1,.., **m** is the *variant index* within a service.

A consumer requirement information specification (**R**) is given by,

$$\mathbf{R} = f(\mathbf{X},\mathbf{W}) \qquad (1)$$

where, **X** (X_1, X_2,..., X_n) is a tuple representing consumers preference values for a requested service.

And, **W** (W_1, W_2,...,W_n) is a tuple representing the weights attached to service attributes. Service attributes refer to the QoS metrics associated with each service variant. We use service attribute and QoS metric interchangeably.

Due to the difference in the measurement units of each of the QoS metric values, there is a need to normalize (Taher et al., 2005) them to be in the range [0, 1].We use the following equations:

Normalize (X) = $$\frac{X_{v\max} - X_{v}}{X_{v\max} - X_{v\min}} \qquad (2)$$

Normalize (X) = $$\frac{X_{v} - X_{v\min}}{X_{v\max} - X_{v\min}} \qquad (3)$$

Where X_v = the QoS metric to be normalized by minimization using equation (2) and maximized using equation (3). A metric, which optimizes user utility when its value is minimal, like response time, would be normalized by minimization using equation (2) while those, which optimizes user utility when its value is maximal, like reliability would be normalized by maximization using equation (3).

And $X_{v\max}$ = the maximum value for the same QoS metric for each relevant grid service variant returned by the system.

And $X_{v\min}$ = the minimum value for the same QoS metric for each relevant grid service variant returned by the system.

We defined an array A = {1, 1, 0, 0}. 0 indicates that the QoS metric in question should be normalized using equation (2) and 1 indicates that QoS metric in question should be normalized using equation (3).

Let $\mathbf{Q}_j$ = {attribute values of a service variant j } where j = 1,...**m** and **m** = number of variants of a particular service.

If **n** = number of attributes used to differentiate variants of a particular service. *(In our case* ***n*** *= 4)*

and **d** = the Euclidean distance between a particular variant and a consumer QoS request.

then $\mathbf{d}_{min}$ = the minimum Euclidean distance between a particular variant and a consumer QoS request

An SLA template is selected by considering the preference values **(X)** and weights **(W)** expressed on the service by the consumer.

Therefore, the weighted Euclidean distance function is used to measure the closeness of the consumer-defined preference values to the available variant values $\mathbf{Q}_j$.

This is given by:

$$\mathbf{d}_{min}(\mathbf{X}, \mathbf{Q}_j) = \sqrt{\sum_{k=1}^{n} W_k \left(X_k - Q_{jk} \right)^2} \qquad (4)$$

The SLA template with the least dissimilarity, $\mathbf{d}_{min}(\mathbf{X}, \mathbf{Q}_j)$ from the consumer's requirements represents a selection that is close to consumer preference and is, therefore, recommended. A summary of the flow of events is shown in Figure 4.6

Figure 4.5 illustrates a procedure for the SLA template selection.

1. Get Input QoS requirements (R) where **R= f(X,W)**
2. Get QoS levels (Q) of available service variants and constraints by constructing a QoS matrix
3. for each QoS metric

   {
   Normalize consumer input (R) using equation-2 or equation-3
   }

4. for each QoS metric

   {
   Normalize Q using equation-2 or equation-3 and array A
   }

5. Compute Euclidian distance between X and Q using equation-4
6. Find **d** with the minimum distance ($\mathbf{d}_{min}$)
7. Output ($\mathbf{d}_{min}$)

Figure 4. 5: Selection Algorithm Procedure

Figure 4.6 illustrates the flow of events during the service selection process.

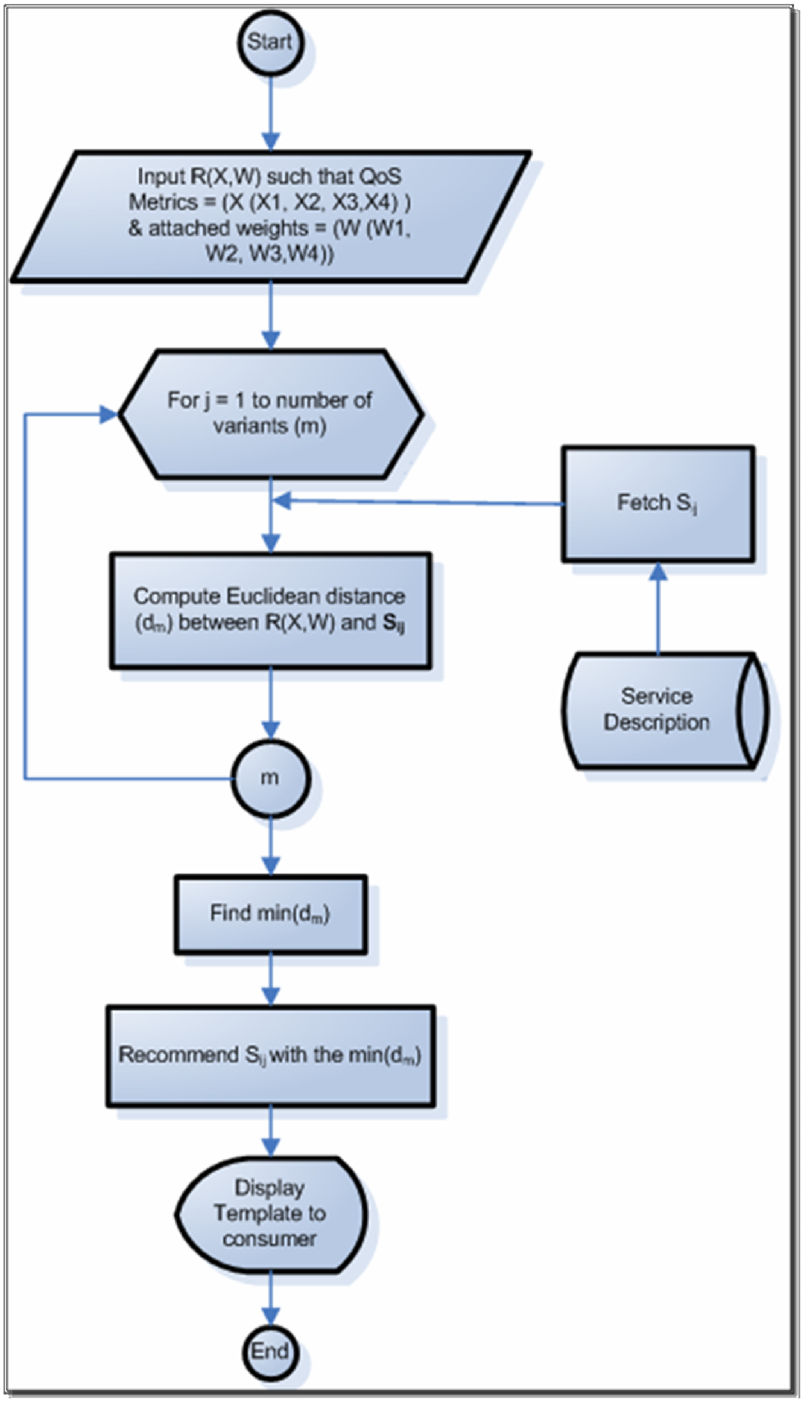

Figure 4. 6: Flow chart for service selection

4.4 Chapter summary

In this chapter, we have described the design of a consumer initiated SLA template creation framework for a grid-based environment. We also suggested the modification of the SLA life cycle by ensuring that its initial stages are not solely provider dominated but also inclusive of the consumer's preferences on QoS. As stated in section 1.5, the goal of this work was to develop a framework for flexible SLA template creation based on consumer QoS requirements. This has been partly achieved in this chapter with the detailed description of the model design and algorithm.

CHAPTER FIVE

MODEL SIMULATION AND PERFORMANCE EVALUATION

In Chapter 4 we presented the Consumer-initiated SLA (C-SLA) creation framework. This framework allows for the selection of the best services in terms of QoS for the consumer based on the consumer's initial QoS request. In order to prove the concept being discussed in this dissertation, this chapter presents the simulation experiments and analyzes the results obtained.

First, the assumptions of the simulation are described in section 5.1 while 5.2 present the simulation experiment setup. Section 5.3 presents our test cases and results that were obtained. Section 5.5 concludes this chapter.

5.1 Simulation Assumptions

The following general assumptions were made during the simulation experiment:

1. In order to mitigate the effect of network delay, the simulation experiment was conducted on a single stand alone computer, therefore, it is assumed that network delay is zero.
2. The providers of the P-SLA approach lack information about a specific consumer's business need and only delivered its service based on general knowledge of the market. This is realistic because the provider normally does not

consult each consumer in packaging its service and so would not know their individual needs.

5.2 Simulation Design

The simulation was conducted using Java with Development Kit Version 6 (JDK6). We simulated both the Consumer Initiated and the Provider Initiated SLA template creation framework. Figure 5.1 illustrates the simulation setup.

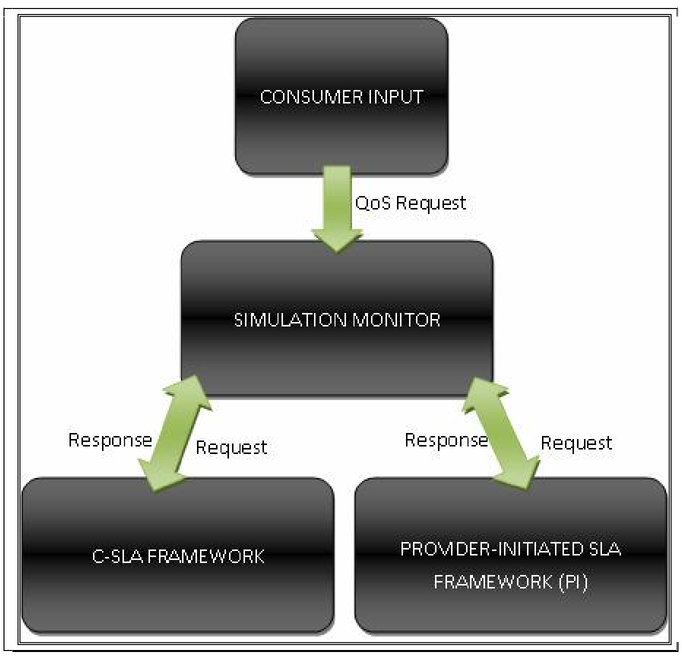

Figure 5. 1: The simulation setup

The *simulation monitor* component acts as an entry point to pass parameters to the two schemes as well as to collect the results. The system generates service requests and both schemes would have to select and recommend the most appropriate service variant. Table 5.1 summarizes the parameters used in the simulation and their range of values.

Table 5. 1: Parameters and their default values for the simulation studies

Parameter Class	Parameter Type	Descriptions	Value
QoS Metrics	Reliability	A non-functional QoS metric which denotes the probability that an available service is properly working and is rendering accurate outcome for a specified amount of time	[50,...,100%]
	Availability	A non-functional QoS metric that denotes the probability that a service is available when invoked.	[50,...,100%]
	Response Time	A non-functional QoS metric that denotes the interval of time elapsed from invocation to the completion of a service.	[1,..,6 sec]
	Cost	A non-functional QoS metric that denotes the price charged for each invocation of a service	[G$150...,400]
QoS Constraints	Weight	The weight is an indication of the consumer's individual preferences on each of the QoS metrics. It ranges from 1 to 3 where 1 depicts low, 2 medium and 3 high priority.	[1,2,3]
	Satisfaction Threshold	This metric delineates the point at which the consumer derives satisfaction from a given SLA template. By comparing the *Satisfaction Threshold* and the *consumer utility* of the resultant selected SLA template, this should assist in determining the satisfaction consumers derive from the selected service.	[50,...,100%]
Service Specifications	No. of services		[1,...,4]
	No. of templates or Service variants		[1,...10000]

5.2.1 The Consumer-Initiated SLA Creation framework (C-SLA)

This framework was defined in chapter 4. A simulation of the proposed architecture was developed, in order to observe if the Consumer-Initiated (C-SLA) SLA creation framework offers performance benefit against P-SLA for dynamic selection of appropriate SLA templates.

5.2.2 The provider-initiated SLA (P-SLA) Creation framework

Since the provider solely creates and provides the SLA template, the Provider Initiated SLA framework (P-SLA) is based on a random selection technique that does not require any information about the consumer's QoS request. As a result, the SLA selection decision is done in a random manner.

5.3 Simulation Parameters Discussion

This section describes the performance analysis between the proposed C-SLA template selection framework and the classical Provider-initiated SLA (P-SLA) creation framework. For the comparison, the following performance metrics were used:

i. Consumer Satisfaction
ii. Scalability
iii. Precision

5.3.1 Consumer Satisfaction

Consumer satisfaction is achieved when the selected SLA template is similar to or an exact match of the consumer request i.e. the selected SLA template derives increased *consumer utility*. Increased *Consumer utility* comes about when the Euclidean distance between the consumer request and the selected SLA template (d_{min}) falls below the consumer's *Satisfaction Threshold utility* (d_u) (See Appendix II). Given d_{min} is above the consumer's *Satisfaction Threshold utility* this implies that the selected SLA template derives decreased *consumer utility*.

a) Analysis of Consumer Satisfaction for C-SLA and P-SLA Framework

Pass *n* number of requests to *simulation monitor*

Compare d_{min} of resultant SLA template selected by each scheme with *Satisfaction Threshold.*

    If $\mathbf{d}_{min} \leq$ *Satisfaction Threshold utility*

        then accept SLA template

        Get Template Acceptance Time (TAT)

    else reject SLA template

Observe all accepted SLA templates (*p*) for each scheme

Analyze using *z-test* of difference of two proportions

5.3.2 Scalability

b) *Analysis of SLA Template Acceptance Ratio Vs. No. of Requests for C-SLA and P-SLA framework*

Pass a batch of requests to the *simulation monitor* in intervals Record the accepted SLA templates for each interval Compute template acceptance proportions at each interval

c) *Analysis of Scalability of the C-SLA framework with increases in the number of service variants*

Increment the number of service variants in intervals For each interval record the time taken to complete SLA template selection

5.3.3 Precision

d) *Analysis of Precision of C-SLA and P-SLA framework*

Precision refers to the relevance of the retrieved service variants out of the actual number of retrieved service variants.

$$\text{Precision} = \frac{\{relevant \cap retrieved\}}{retrieved}$$

5.4 Experimental Results Analysis

This section describes in detail all the experiments and analysis and discusses the results that were obtained.

5.4.1 Experiment I: Consumer Satisfaction Level for P-SLA and C-SLA

An experiment was conducted to determine the consumer satisfaction derived from each approach. The experiment was set up as follows: 900 consumer requests were randomly generated with specified QoS parameters. These requests were made to a database with 502 simulated services. The selection of the appropriate services was then done by the two schemes. Both the requests and the corresponding results generated by each scheme were observed and analyzed. To test the performance of the two approaches, a *z-test* of difference of two proportions was conducted upon the observed results. The hypothesis was that the C-SLA template selection framework better satisfies the consumer than the P-SLA selection framework.

i. *The Hypothesis*

Null hypothesis: H_0 :There is no difference in consumer satisfaction of the C-SLA and P-SLA template selection process

Alternative Hypothesis: H_1 :Consumer initiated SLA template creation better satisfies the consumer than the Provider Initiated one.

ii. *The Decision Criteria*

The critical value of z = $Z_{tab}(0.05)=1.6449$

If $Z_{calc} > Z_{tab}$ then we reject the null hypothesis H_0

iii. Data Gathered

Table 5. 2: Data Gathered during the experiment

	P-SLA	C-SLA	$\hat{p}$
Observations (p)	327	752	0.5994
Sample size (n)	900	900	

iv. The Hypothesis Evaluation

The Test statistic is given by:

$$Z_{calc} = \frac{P_{ci} - P_{pi}}{\sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{n_{ci}} + \frac{1}{n_{pi}}\right)}}$$

where $\hat{p} = \frac{p_{pi} + P_{ci}}{n_{pi} + n_{ci}}$

If $Z_{calc} > Z_{tab}$ then we reject the null hypothesis H_0

where $Z_{tab}(0.05)=1.6449$

thus Z_{calc}= 20.4431, therefore, reject H_0 and conclude that the Consumer Initiated SLA template creation process performs better than the Provider Initiated one. This is due to the fact that the C-SLA framework has knowledge of the consumer requirements before it can select an SLA template. It is based on this knowledge that the framework recommends a satisfactory SLA template.

5.4.2 Experiment II: SLA Template Acceptance Ratio against No. of Requests.

Having proved that the C-SLA framework offers better consumer satisfaction as compared to the P-SLA framework, the following experiment was carried out in order to ascertain whether this performance is affected by the increase of SLA template requests. In this experiment, a series of experiments were conducted by varying consumer template requests for each of 10 sets. For each set, the numbers of accepted templates were recorded against the number of requests submitted in each set. To eliminate some biasness in the experiment, proportions of the number of accepted templates were used for all the 10 sets. Table 5.3 shows the simulation results observed and Figure 5.2 the behavior of the two frameworks.

Table 5. 3: Observation of Accepted SLA templates

No. Of Request		100	200	300	400	500	600	700	800	900	1000
Proportions of Accepted Templates	P-SLA	0.38	0.34	0.39	0.34	0.33	0.34	0.36	0.35	0.36	0.33
	C-SLA	0.78	0.82	0.83	0.79	0.8	0.80	0.80	0.8	0.84	0.82

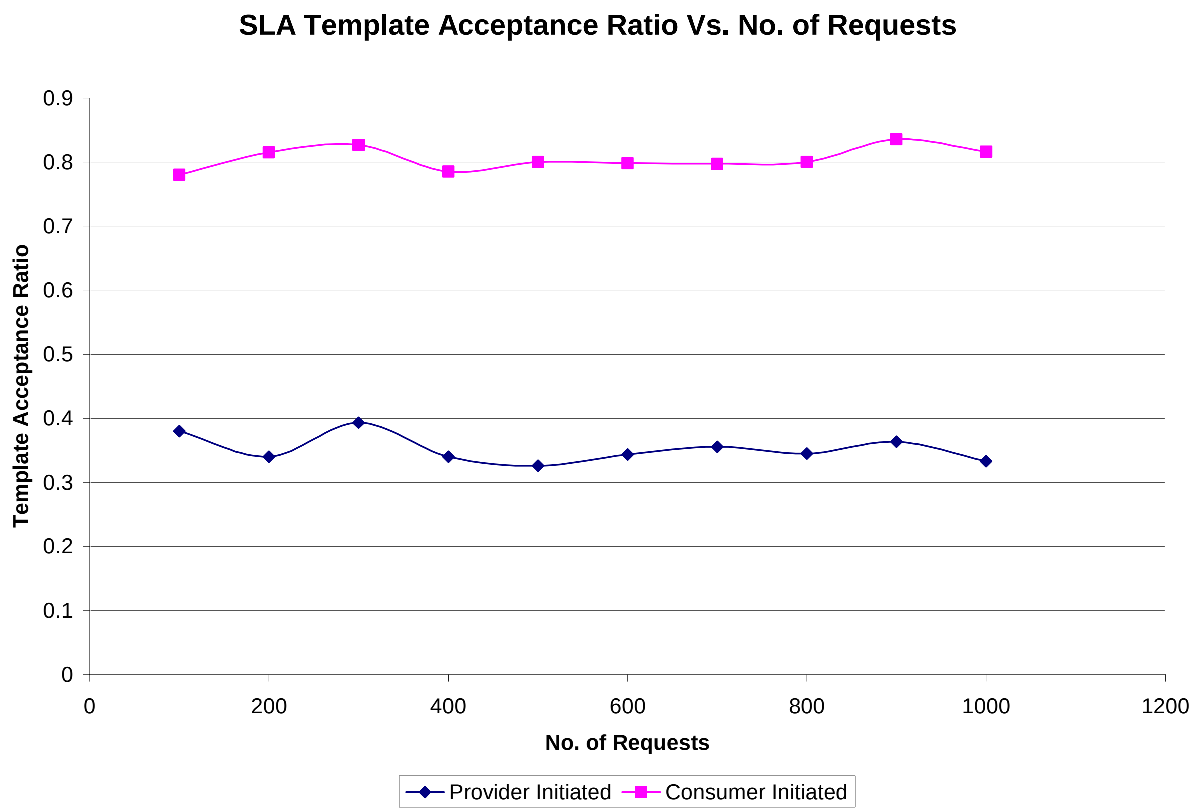

Figure 5. 2: Graph of SLA TAR against No. of Requests.

5.4.2.1 *Experiment Discussion*

It was observed that the increase of SLA template requests does not in any way affect the C-SLA framework's ability to recommend a satisfactory SLA template to the consumer. For instance, an increase in the number of requests from 500 to 800 maintained a steady template acceptance ratio of 0.8 for C-SLA whilst that of P-SLA fluctuated from 0.33 to 0.34 to 0.36 and then dropped to 0.35. The huge gap between the performances of the two frameworks is attributed to the C-SLA's ability to select and recommend an SLA template based on prior knowledge of the consumers QoS preferences. The C-SLA, therefore, scales well with an increase in number of request and maintains high levels of Template acceptance ratio.

5.4.3 Experiment III: Average RRT Vs. Number of Service Variants

This experiment was carried out in order to determine the effect of an increase in the numbers of service variants on the RRT (i.e the time it takes to retrieve an SLA template after a request has been made. A particular service variant was searched for as numbers of service variants were being increased. The average time taken for that particular service variant to be retrieved was recorded. Table 5.4 shows the simulation results observed.

Table 5. 4: Number of service variants and the corresponding RRT

No. of service variants	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
Average RRT (sec)	0.402	0.635	0.841	1.125	1.524	1.954	2.356	2.634	3.214	3.540

The scatter plot presented in Figure 5.3 shows a linear relationship between RRT and Number of service variants. This implies that the C-SLA framework is scalable. An exponential relationship of this nature would imply poor performance as grid service variants increase hence, an inability to scale well.

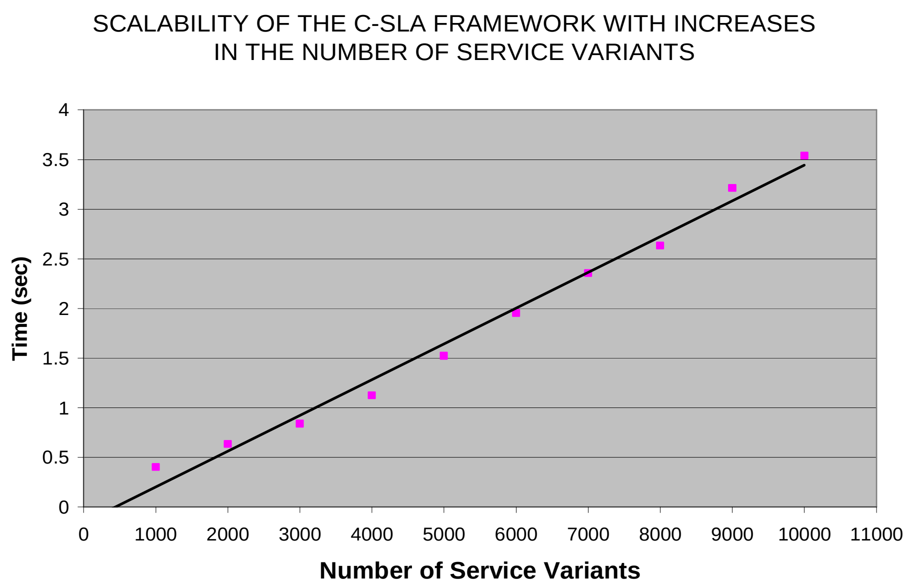

Figure 5. 3: Graph of Request Response Time against Number of Grid Services

5.4.4 Experiment IV: Precision of C-SLA and P-SLA framework

This experiment measures the success of the SLA template selection based on the relevance of the selected SLA templates. It may be possible that some SLA templates which are recommended might not be relevant to the consumer's QoS request.

In this experiment, 502 service variants were generated. A hierarchical clustering of these variants was conducted based on the QoS parameters. It was identified that these variants could best be clustered in two groups. For each of the test runs conducted in this experiment, 20 requests were forwarded to the service database. The returned services for both the C-SLA and the P-SLA framework were observed. i.e the request was processed by both frameworks. Each request was classifieds into one of the two clusters based on their similarity to these two clusters. Euclidean distance was used as a dissimilarity measure. If a request returns a service in its cluster, it is said to have returned a relevant service otherwise the returned service is not relevant. Using the 20 requests from each run, the precision of each scheme was calculated. A total of 8 runs were conducted for this experiment.

5.4.4.1 Results

Table 5.5 shows the precision observed for each framework.

Table 5. 5 Analysis of precision of the C-SLA and the P-SLA framework

Test No.		*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	AVERAGE
Precision	*C-SLA*	0.75	0.85	0.8	0.75	0.8	0.95	0.9	0.85	0.83125
	P-SLA	0.5	0.55	0.55	0.55	0.6	0.65	0.7	0.75	0.60625

Hypothesis

H_0: the precision of the C-SLA approach is not different from that of the P-SLA approach

H_1: the precision of the C-SLA approach is different from that of the P-SLA approach

Table 5. 6: Difference in precision between the C-SLA approach and the P-SLA approach

		Mean	t	df	Sig. (2-tailed)
Pair 1	Consumer Initiated - Provider Initiated	.22500	9.721	7	.000

Table 5. 7: Paired Sample Statistics

Paired Samples Statistics					
		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	Consumer Initiated	.8312	8	.07039	.02489
	Producer Initiated	.6063	8	.08634	.03053

P-value = 0.00 <0.05.

Reject H_0 and conclude that the precisions of the two approaches are significantly different.

These results indicated that the mean of C-SLA precision is significantly different from the P-SLA precision (t = 9.721, p = .000).

C-SLA has higher average precision as compared to the P-SLA approach (See Figure 5.4). On average C-SLA is approximately 80% precise in retrieving relevant SLA templates as compared to 60% precision for the P-SLA approach. This goes to show the difference in precision of the two approaches. C-SLA, therefore, recommends more relevant SLA templates as compared to P-SLA approach.

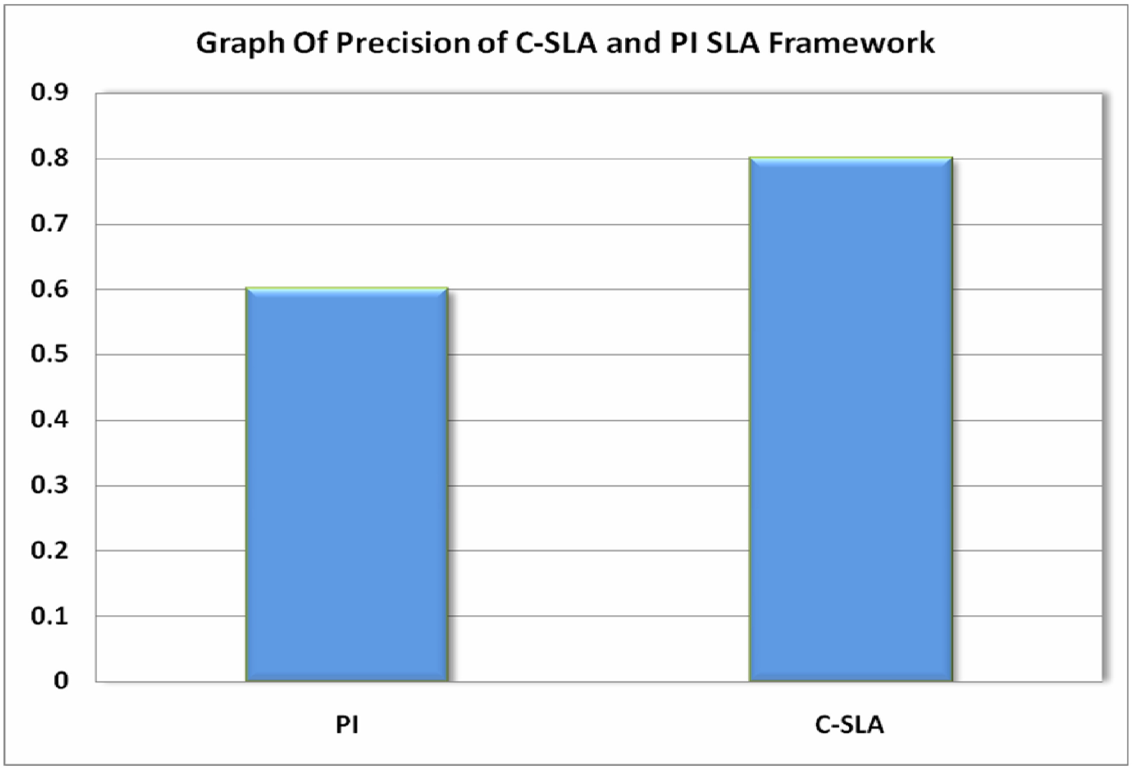

Figure 5. 4: Graph Of Precision of C-SLA and P-SLA template selection Framework

5.5 Chapter summary

In this chapter we presented the simulation results of C-SLA template selection framework. We compared the C-SLA with the classical Provider initiated (P-SLA) SLA template selection framework. Experiments on consumer satisfaction, scalability and precision were conducted. The goal was to achieve increased consumer satisfaction through flexibility and customization of the SLA template creation process. This goal was achieved by incorporating consumer input at the very initial stage of the SLA lifecycle and then utilizing that input to select the most appropriate SLA template for the consumer.

CHAPTER SIX

CONCLUSION AND FUTURE WORK

In this dissertation, we have tried to address the challenges faced by both consumers and providers of Grid services as a result of rigid SLA templates. These challenges brought about the need for a more flexible SLA template creation process. This work specifically focused on the selection of the most appropriate template for the consumer based on their QoS requirements and obviously, on the provider's capability as the solution to this rigidity problem. The overall goal of this research was to develop a framework for flexible SLA template creation based on consumer QoS requirements. The framework is what we referred to as a Consumer-initiated SLA creation framework abbreviated as C-SLA.

This Chapter concludes the dissertation and summarizes it in Section 6.1. In Section 6.2, we conclude this chapter by suggesting future works and defining the limitations of our work.

6.1 Conclusion

From the existing literature, we realized that the consumer's requirements are only considered at the later stages (i.e during negotiation) of the SLA life cycle, at which stage the parties need to indulge in intense negotiations before an agreement can be reached. The degree of customization in SLA templates offered by service providers is limited. Consumers find it difficult to express their individual QoS preferences that allow them to realize ROI timely, during the classical SLA creation phase.

We argue that the initial phases of the SLA life cycle should encompass the consumer requirements before the SLA template can be formulated and advertised. We, therefore, proposed a consumer-initiated SLA creation framework that factors in the consumers request to select the most appropriate SLA template for that consumer. The consumer- initiated approach to SLA template selection is more appropriate in delivering flexible creation of SLAs and promotes increased consumer satisfaction as compared to the provider initiated approaches.

We simulated our proposed consumer initiated approach to SLA template creation as well as the provider initiated approaches. The proposed approach (C-SLA) was compared to existing ones that are based on the classical provider initiation approach to SLA template creation. We used this simulation to conduct performance evaluation experiments between the two approaches. In the simulation, the following evaluation metrics were used:

1. Consumer Satisfaction
2. Scalability
3. Precision

Analysis of the results obtained from the simulation revealed that our C-SLA framework brings more satisfaction to a consumer as compared to the P-SLA approach. The result also showed that the C-SLA framework is more scalable than the P-SLA approach and finally that C-SLA is more precise in retrieving relevant SLA templates in comparison to P-SLA approach.

In essence, the evaluations concluded that C-SLA as proposed in this research could provide the needed flexibility in SLA template creation that would improve the uptake of commercial Grid.

6.2 Limitations and future Enhancements

Although C-SLA has been proved to be an applicable approach to dynamic selection of the appropriate SLA template, it has some limitations which could be recommended for future enhancements. For instance, in practice network factors would have a direct effect on many QoS properties such as response time, negotiation time etc. In this work, these factors were not considered. It would be interesting to see how these network-metrics influence the behavior of C-SLA.

We believe that if the provider allows for consumer input during SLA template creation, through studying this input, the provider will be able to "learn" the trends of the consumer requirements and hence, if need be, the provider would be able to "adjust" its services to suit the current trend resulting in an increase in consumer satisfaction. It would also be desirable for C-SLA be able to learn the behavior patterns of each consumer and should at least keep profiles of all its consumers. This would also aid in it being able to increase its precision during SLA template selection.

Also in our work, only limited QoS metrics were used for the experiments and evaluation. In future we intend to investigate the impact of other QoS metrics in the SLA template selection.

Overall, considering that the results of this work are based on a simulation experiment, in future, we intend to deploy our model on an actual grid infrastructure (preferably GUISET) together with existing SMMEs as case studies and hope to report the findings in subsequent peer-reviewed publications of this work.

BIBLIOGRAPHY

2nd Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop - NFPSLA-SOC'08. Available at: http://events.sti2.at/nfpsla-soc08/. Last accessed 21 November 2008

Adigun, M.O. Emuoyibofarhe, O.J. Migiro, S.O. (2006). "Challenges to Access and Opportunity to use SMME enabling Technologies in Africa", a presentation at 1st All Africa Technology Diffusion Conference, June 14 – 16, Johannesburg, South Africa.

Afzal, A., Mayer, A., Young, L. (2004). "Predictable Workflow Deployment Services", Proceedings of Grid Services Workshop, GGF11. IEEE, June 2004.

Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M. (2005). "Web Services Agreement Specification (WS-Agreement)". GWD-R (Proposed Recommendation), Open Grid Forum. Available at: http://www.ogf.org/Public_Comment_Docs/Documents/Oct-2005/WS-AgreementSpecificationDraft050920.pdf. Last accessed 21 November 2008.

Aranda-Mena, G., Wakefield, R., Lombardo, R. (2006). "A Diffusion Theoretic Approach To Analysing E-Business Up-Take In Small Building Enterprises", ITcon Vol. 11, Special Issue e-Commerce in construction , pg. 149-159, http://www.itcon.org/2006/11. Last accessed 21 November 2008.

Balke, W.T. and Wagner, M. (2003). "Towards personalized selection of web services". Proceedings of the 12th International World Wide Web Conference, Budapest, Hungary, 2003. Available at: http://www.l3s.de/apis/paper/www03.pdf. Last accessed 21 November 2008.

Begic, G. (2005). "Strategies for software development project success: A personal perspective", IBM developerWorks, 15 Nov 2005. Available:

http://www.ibm.com/developerworks/rational/library/nov05/begic/index.html.Last accessed 26 November 2008.

Bhoj, P., Singhal, S. and Chutani, S. (2001). "SLA management in Federated Environments", Computer Networks, Vol 35(1). pp5-24.
Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H.F., Thatte, S., Winer, D. (2000). "Simple Object Access Protocol (SOAP) 1.1" W3C Note 08, World Wide Web Consortium, May 2000, www.w3.org/TR/SOAP/.

Buthelezi, M.E., Adigun, M.O., Ekabua, O.O., and Iyilade, J.S., (2008). "Accounting, Pricing and Charging Service Models for a GUISET Grid-Based Service Provisioning Environment", proceedings of The 2008 International conference on E-learning, E-business, Enterprise Information system, and E-government, pp. 350 - 355.

Christensen, E., Curbera, F., Meredith, G., Weerawarana, S. (2001). "Web Services Description Language (WSDL) 1.1". W3C Note 15, World Wide Web Consortium, March 2001, http://www.w3c.org/TR/wsdl.

Clark, A. and Gilmore, S. (2007). "Evaluating Quality of Service for Service Level Agreements". Available at : http://www.pst.ifi.lmu.de/projekte/Sensoria/del_24/D4.3.a.pdf . Last accessed 21 November 2008

Condor (2008). High Throughput Computing. Available at: http://www.cs.wisc.edu/condor/description.html. Last accessed 21 November 2008.

Cubera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi N., Weerawarana, S. (2002)."Unraveling the Web Services Web. An Introduction to SOAP, WSDL, and UDDI", IEEE Internet Computing, pp. 86-93, 6(2), March, 2002.

Dan, A., Ludwig, H., Pacifici, G. (2003). "Web Service Differentiation with Service Level Agreements", White Paper, IBM Corporation, Mar. 2003. Available at: http://www.ibm.com/developerworks/library/ws-slafram/. Last accessed 21 November 2008.

Darko-Ampem, S. and Katsoufi, M. (2006). "Towards a secure Negotiation Protocol for Virtual Organisations", Unpublished Masters Thesis, Royal Institute of Technology, Computer Science Department , Sweden, March 2006.

Demirkan, H., Goul, M., Soper, D.S. (2005). "Service Level Agreement negotiation: A Theory-Based Exploratory Study As A Staring Point For Identifying Negotiation Support System Requirements", Proceedings Of the 38th Hawaii International Conference on System sciences, Hawaii, USA, January 2005. Available: http://www2.computer.org/plugins/dl/pdf/proceedings/hicss/2005/2268/01/22680037b.pdf?template=1&loginState=0&userData=madigun1227252938274. Last accessed 21 November 2008.

Dunham. M.H., (2003). "Data Mining, Introductory and Advanced Topics". Prentice Hall. Upper Saddle River, New Jersey (2003)

Foster I., (2002). "What is the Grid? A Three Point Checklist", Daily News and Information for the Global Grid Community, July 22, 2002. Available: http://www.gridtoday.com/02/0722/100136.html. Last accessed May 31, 2008.

Foster, I., Kesselman, C., Tuecke, S. (2001). The anatomy of the grid: Enabling scalable virtual organizations. International Journal of High Performance Computing Applications, 15(3), pp. 200-222, www.globus.org/research/papers/anatomy.pdf.

GRAAP-WG (2003). Grid Resource Allocation Agreement Protocol Working Group. Online at: <http://forge.gridforum.org/sf/projects/graap-wg/>. Last assessed 26 November 2008

GridForge. OGSA-WG (2003). The Open Grid Services Architecture-Working Group. Available: http://forge.gridforum.org/projects/ogsa-wg Last accessed November 2008.

Han, J. and Kamber, M. (2006). "Data Mining: Concepts and Techniques". Morgan Kaufmann Publishers, San Francisco, CA, 2006.

Hasselmeyer, P., Mersch, H., Koller, B., Quyen H.-N., Schubert, L., Wieder, P. (2007). "Implementing an SLA Negotiation Framework", Exploiting the Knowledge Economy: Issues, Applications, Case Studies (eChallenges 2007), The Hague, The Netherlands, pp. 154-161, October 2007.

Hasselmeyer, P., Qu, C., Koller, B., Schubert, L., Wieder, P. (2006). "Towards Autonomous Brokered SLA Negotiation" Exploiting the Knowledge Economy: Issues, Applications, Case Studies (eChallenges 2006), Barcelona, Spain, October 2006. IOS Press, ISBN 978-1-58603-682-9, pp. 44-51.

Hey, A. J. G. and Trefethen, A. E. (2002). The UK e-Science Programme and the Grid. *Lecture Notes in Computer Science*, 2329 . pp. 3-21.

Huhns, M. N. and Singh, M. P. (2005). "Service-Oriented Computing: Key Concepts and Principles". IEEE Internet Computing, vol. 9: pp. 75–81, Jan./Feb. 2005.

Hung, P.C.K., Li, H., Jeng, J. (2004). "WS-Negotiation: An overview of research issues". Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS'04), Big Island, Hawaii, USA, January 2004, pp. 33–42.
IBM. "Business Process Execution Language for Web Services Version 1.1." Available: http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel.pdf

Jacob, B. (2003). "Grid computing: What are the key components?". Available: http://www-128.ibm.com/developerworks/grid/library/gr-overview/. Last accessed 12 March 2007.

Joseph, J., Earnest, M., Fellenstein C. (2004). "Evolution of grid computing architecture and grid adoption models. IBM Systems Journal. 43(4). 624-645.

Kabanda, S.K., Iyilade, J.S., Adigun, M.O. (2007). "Knowledge Resources in a Grid-enabled Infrastructure – The case of the deep rural SMMEs in South Africa". In Proceedings of the 4th International Conference on Intellectual Capital and

Knowledge Management (ICICKM, 2007), 15 - 16 Oct 2007, Cape Town, South Africa, pp 199-208.

Kaminski H. and Perry, M. (2006a). "Employing Intelligent Agents to Automate SLA Creation", Emerging Web Services Technologies workshop Conference Proceedings ECOWS, December 2006, Zurich, Switzerland, Springer-Birkhauser, "Emerging Web Services Technology" pp. 33-46

Kaminski, H. and Perry, M. (2006b). "SLA Automated Negotiation Manager for Computing Services", Proceedings of the 8th IEEE International Conference on E-Commerce Technology and the 3rd IEEE international conference on Enterprise Computing , E-Commerce, and E-Services (CEC/EEE'06), June 2006 pp. 347-350.

Keller, A. and Ludwig, H. (2003). "The WSLA Framework: Specifying and monitoring service level agreements for web services", E-business management journal of Networks and Systems management 11, 2003.

Keller, A. and Ludwig, H., (2002). "Defining and monitoring service level agreements for dynamic e-business", in: Proceedings of the 16th System Administration conference, LISA 2002, Philadelphia, USA, November,2002, pp. 189-204.

Kreger, H. (2003). "Fulfilling the web services promise", Communication of the ACM VOL. 46 (6) pp. 29-34, June 2003.

Lamparter, S., Ankolekar, A., Studer, R., and Grimm, S. (2007). "Preference-based selection of highly configurable Web Services." Proceedings of the 16th international conference on World Wide Web, 2007 pp. 1013-1022

Leff, A., Rayfield, J.T., Dias, D. M. (2003). "Service-Level Agreements and Commercial Grids," IEEE Internet Computing, vol. 7, no. 4, pp. 44-50, Jul/Aug, 2003.

Ludwig, H., Keller, A., Dan, A., King, R.P., Franck, R. (2003). "Web Service Level Agreement (WSLA) Language Specification". IBM. USA. July 2002 Available:

http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf . Last accessed 21 April 2008.

Marasco, J. (2006). "Software development productivity and project success rates: Are we attacking the right problem?" IBM developerWorks, 15 Feb 2006. Available: http://www.ibm.com/developerworks/rational/library/feb06/marasco/index.html. Last accessed 26 November 2008.

Masche, P., McKee, P., Mitchell, B. (2006). "The Increasing Role of Service Level Agreements in B2B Systems". In J. A Moinhos Cordeiro, V. Pedrosa, B Encarnac¸˜ao, and J. Filipe, editors, Proceedings of the 2nd International Conference on Web Information Systems and Technologies (WEBIST), pp. 123–126, 20 April 2008.

Maximilien, E. M. and Singh, M. P. (2004b). "Toward Autonomic Web Services Trust and Selection", Proceedings of 2nd International Conference on Service Oriented Computing (IC-SOC 2004), New York, pp. 212-221, November 2004.

Maximilien, E.M. and Singh, M.P. (2004a). "A Framework and Ontology for Dynamic Web Services Selection". IEEE Internet Computing, 8(5): pp. 84-93, 2004.

Mitchell, B. and Mckee, P., (2005). "SLAs – A key Commercial Tool", eChallenges e-2005, Ljubljava, Slovenia. Available: http://www.nextgrid.org/download/publications/echallenges.pdf . Last accessed 20 May 2008

Mobach, D. G. A., Overeinder, B. J., and Brazier F. M. T. (2006). A WS-Agreement based resource negotiation framework for mobile agents. Scalable Computing: Practice and Experience, 7(1):23–36.

NGS. (2007). National Grid Service. Online at: http://www.grid-support.ac.uk/. Last accessed 26 November 2008.

OASIS. (2003). "UDDI Version 3.0.1. UDDI Spec Technical Committee Specification". http://uddi.org/pubs/uddi_v3.htm. Last accessed 26 November 2008.

Parkin M., Badia R.M., Martrat, J. (2008). "A Comparison of SLA Use in Six of the European Commissions FP6 Projects" Technical report, TR-0129, Institute on Resource Management and Scheduling, CoreGRID - Network of Excellence, April 2008. Available: http://www.coregrid.net/mambo/images/stories/TechnicalReports/tr-0129.pdf. Last accessed 11 September 2008.

Patel, Y. and Darlington, J. (2007). "Novel Stochastic Profitable Techniques For Brokers In A Web-Service Based Grid Market" Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence: pp. 132-140.

Pichot, A., Wieder, P., Wäldrich, O. and Ziegler, W. (2007). "Dynamic SLA-Negotiation based on WS- Agreement", CoreGRID - Network of Excellence, 2007, CoreGrid Technical Report; 82, TR-0082.

Quan, D.M. (2006). "A Framework For Sla-Aware Execution Of Grid-Based Workflows", unpublished PhD Thesis, November 2006.

Ran, S. (2004). "A Model for Web Services Discovery with QoS". SIGEcom Exchanges, vol. 4, no. 1, 2004, pp. 1–10.

Sahai, A., Graupner, A., Machiraju, V. and van Moorsel, A. (2003). "Specifying and monitoring Guarantees in Commercial Grids through SLA", proceedings of 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid. (Tokyo, Japan, IEEE Computer Society Press.), pp. 292-299.

Skene, J, Lamanna, D. D., Emmerich, W. (2004). "Precise Service Level Agreements", International Conference on Software Engineering, Proceedings of the 26th International Conference on Software Engineering, pp. 179 – 188, 2004

SPC (1998). E-ssentials! E-newsletter on Management and practical software development. Available: www.spcusa.com/essentials/sept2199.htm [Last accessed March, 2008]

Sturm, R. (2005). “Service Level Objectives”, Network Word Fusion, 2002 Enterprise Management Associates, Inc. Retrieved from: http://www.slminfo.com/articles/slobjectives.htm on Dec 12, 2005.

Sun, W., Zhang, J., Liu F. (2006). “WS-SLA: A Framework for Web Services Oriented Service Level Agreements”, Proceedings of CSCWD'2006. pp.714-717.

Telemanagement Forum. SLA Management Handbook. Vol 2, Concepts and Principles, 2005.

Teragrid (2008). The US TeraGrid. Online at: http://www.teragrid.org/

TNGC. Thai National Grid Centre. TGCC (2006) “Grid Services”. Available Online: www.med.cmu.ac.th/secret/mis/eiu/informatics/GridComp/TGCC2006/Grid%20Computing/Grid%20Service.pdf . Last accessed 12 January 2009.

Tiropanis, T., (2003).“Business Roles and Negotiation Models for Web Service Based Provision”, Proceedings of the second Workshop on e-Business, Seattle, US, December 2003. ISBN 957-01-5801-8. Available: http://eprints.soton.ac.uk/50059/01/Web2003_Tiropanis_CR2.pdf . Last Accessed 11 September 2008.

Tosic V., Pagurek, B., Patel, K. (2003). “WSOL - A Language for the Formal Specification of Various Constraints and Classes of Service for Web Services”, In The International Conference On Web Services, ICWS’03, pp. 375-381, 2003.

Truong, H. L., Samborski, R., Fahringer, T. (2006). “Towards a Framework for Monitoring and Analyzing QoS Metrics of Grid Services”, Proceedings of the Second IEEE International Conference on e-Science and Grid Computing, Amsterdam, Netherlands, Dec. 4- 6, 2006, pp. 65.

Trzec, K., Huljenic, D. (2003). "Modeling agent-based framework for the automation of SLA management lifecycle." Telecommunications, 2003. ConTEL 2003. Proceedings of the 7th International Conference on Volume 1, Issue, 2003, pp. 69 – 76.

Vassiliadis, B., Stefania, A., Tsaknakisb, J. and Tsakalidisb, A. (2006). "From application service provision to service-oriented computing: a study of the IT outsourcing evolution." Telematics & Informatics, 23:4, pp. 271-293.

Verma, D., Beigi, M. and Jennings, R. (2001). "Policy based SLA Management in Enterprise Networks", In Policies for distributed Systems and Networks. Springer-Verlag, Berlin, Germany, pp. 137-152.

W3C (2007). Web Services Policy Framework (WS-Policy), available: http://www.w3.org/TR/ws-policy/

Web Services Description Language (WSDL) 1.1 (2001).Available: http://www.w3.org/TR/wsdl

Wieder, P., W¨aldrich, O., Yahyapour, R., and Ziegler, W. (2006). Improving Workflow Execution through SLA-based Advance Reservation. In S. Gorlach, M. Bubak, and T. Priol, editors, Integrated Research in Grid Computing, CoreGRID Integration Workshop 2006, Krakow, Poland, pages 333–344. Academic Computer Centre CYFRONET AGH, 2006. ISBN: 83-915141-6-1

Yan, J., Zhang, J., Lin, J., Chhetri, M., Goh, S., Lowalczyk, R. (2006). "Towards autonomous service level agreement negotiation for adaptive service composition", Proceedings of the 10th International Conference on CSCW in Design, CSCWD2006, Nanjing, China, 3–5 May, 2006. Available online: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4019289&isnumber=4019032. Last accessed 31 December 2008.

APPENDIX I

Calculating the *Satisfaction Threshold utility (d_u)*

The consumer specifies a percentage between 50 and 100% as the desired *Satisfaction Threshold.*

Let ($\mathbf{d_u}$) be the *utility* represented by this *satisfaction threshold*,

$$\mathbf{d}_u(\mathbf{X}, \mathbf{Q}_j) = \sqrt{\sum_{k=1}^{n} W_k \left(X_k - Q_{jk}\right)^2}$$

Let *consumer utility* = U

For U = 100% Then $\mathbf{d}_u(\mathbf{X}, \mathbf{Q}_j) = 0$

For U = 0% Then $\mathbf{d_u}(\mathbf{X}, \mathbf{Q}_j) = \sqrt{\sum W(1-0)^2}$

where $W_{max} = 3$ and the total QoS metrics = 4

$$= \sqrt{3*4} = \sqrt{12}$$

Let **P** be the consumer defined *satisfaction threshold*,

Therefore $\mathbf{P} = \dfrac{\sqrt{12 - \mathbf{d}_u}}{\sqrt{12}}$

Hence $\mathbf{d_u} = \sqrt{12}(1 - \mathrm{P})$

If $\mathbf{d}_{min} \leq \mathbf{d}_u$ then accept SLA template Get Template Acceptance Time (TAT) else reject SLA template