School of Information Systems
Curtin Business School

A Methodology for Trust and Risk Mash up for Enhanced Business Intelligence in Cloud Environment

Adil Mumtaz Hammadi

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

April 2014

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgment has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature: …….

Date: ……...

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

With the advancement of open computing paradigms such as cloud computing, enterprises are moving their IT infrastructure to clouds. Lower cost, more computing resources and flexible on demand architecture is an attractive choice for most large organizations. However, in adopting cloud services, enterprises have Quality of Service (QoS) concerns and one of these concerns is Service Level Agreement (SLA) management. SLA management is the management of services based on a Service Level Agreement (SLA). Various researchers have published work on how a service provider can manage SLAs. But SLA management from a consumer's point of view, which is important for giving the consumer control over the management of services, such as selecting a trustworthy service provider before the start of an interaction, monitoring the performance of a service provider and determining the financial risk after the start of an interaction, is largely missing in the existing literature. In the approaches proposed in the literature, the importance of trust and risk in cloud computing has widely been ignored and these have not been acknowledged as important factors when selecting a service provider and monitoring its performance. This thesis is an effort to conduct research in this area for the effective management of SLAs from a consumer's point of view.

In order to achieve the aforesaid objective in this thesis, an SLA management framework is proposed. SLA management from a consumer's point of view is proposed as a two-step process: first, selecting a trustworthy service provider who is capable of providing the requested service; and second, monitoring the performance of service delivery to ensure the quality of service. This is achieved by dividing the interaction time space into pre- and post-interaction time phases. Service provider selection is carried out in the pre-interaction time phase on the basis of his trustworthiness and the performance assessment of a service provider along with the financial risk due to service deviation is carried out in the post-interaction time phase. The result of financial risk is then combined with the risk attitude of the consumer to assist him in decision making as to whether to proceed or not to proceed in the interaction with the service provider. This decision making system provides effective SLA management for the consumer.

This thesis also validates the proposed SLA management approaches by building a software prototype to carry out a trustworthiness evaluation of a service provider before the start of an interaction and service continuation decisions on the basis of risk in the interaction after the start of an interaction. The prototype system simulates the real world interaction scenario of a selected service provider and his performance assessment. The service consumer can make an informed decision about the selection of a service provider and about the continuation of a service on the basis of the proposed system.

ACKNOWLEDGEMENTS

First and foremost, I thank God Almighty for giving me the strength to complete this thesis. "*He (God) has taught man that which he knew not*" (Al-Quran). After this, I acknowledge my parents whose continuous prayers are a source of protection for me. I acknowledge the continuous support of my wife throughout the years of my thesis. I acknowledge her sacrifices and her patience. I also express my thanks to my children to whom I could not give much time during my thesis. I am also thankful to my brothers and sisters who live overseas and whom I miss the most.

My thanks go to my supervisor, Dr Omar Khadeer Hussain for his tireless efforts towards the completion of my thesis. I learnt qualities from Dr Omar such as honesty, dedication and attention to detail. These skills are important assets. I am thankful to Professor Tharam Dillon, Professor Elizabeth Chang and Dr Farookh Khadeer Hussain for their initial support at the start of my research. I am especially inspired by Professor Tharam Dillon for his command in scientific research. At the end, I dedicate this thesis to my father-in-law whom I wish to be in this world to celebrate this moment of achievement.

LIST OF PUBLICATIONS

Refereed Journal Article

1. A. Hammadi, O. K. Hussain, T. Dillon, and F. K. Hussain, "A framework for SLA management in cloud computing for informed decision making," *Cluster Computing,* Vol. 16, Issue 4, pp. 961 - 997, 2013.

Refereed Conference Articles

2. Adil M. Hammadi and Omar Hussain, "A Framework for SLA Assurance in Cloud Computing", 26th IEEE International Conference on Advanced Information Networking and Applications (AINA-2012), Fukuoka Japan, March 26 – 29, 2012, pp. 393 – 398.

3. Adil M. Hammadi and Omar Hussain, "Transactional Risk Assessment-based Approach for Service Degradability Management", IEEE International Conference on e-Business Engineering (ICEBE), Oct 19 – 21, 2011, Beijing China, pp. 145- 152.

4. Adil M. Hammadi, Tharam S. Dillon, Elizabeth Chang, "Business Service Composability on the Basis of Trust", IEEE Digital Ecosystems and Technologies Conference, 1 – 3 June, 2009, Istanbul Turkey, pp. 437 - 440.

5. Adil M. Hammadi, Farookh Khadeer Hussain, Elizabeth Chang, Tharam S. Dillon and Saqib Ali, "Ontological Framework for Trust & Reputation for DBE", IEEE Digital Ecosystems and Technologies Conference, 26 – 29 February, 2008, Phitsanulok Thailand, pp. 650 - 653.

CHAPTER 1

INTRODUCTION

1.1 Introduction

In today's modern technical world, the internet and web technologies play a vital role in daily life. Online shopping, internet banking, entertainment and online communication are some of the commonly used applications. Web-based applications such as e-Commerce, web services and cloud computing have changed the way we interact with each other and acquire the services which we need. The online presence of businesses promotes competition among competitors which results in lower prices and better quality products for the consumer. For example, we have witnessed the formation and rapid growth of Amazon.com, the largest online retailer, which has revolutionized the traditional buying experience into a virtual shopping experience. This has been made possible due to open standard paradigms such as Service-Oriented Architecture (SOA) and cloud computing where the *service* is the key concept. In these paradigms, the functionality of components is wrapped in the form of *services* and exposed using a public interface. The services are selected and composed on demand. In implementations of SOA such as Web Services, the *services* are considered as *software services* whereas in cloud computing, services can be platform services, infrastructure services or software services. The management of these services from a consumer's point of view in cloud computing, particularly for *software services,* is a complex task and this issue needs to be addressed to give consumers more control over service management. Such problems will be discussed in this thesis. This chapter is divided into the following sections. A brief introduction of cloud computing is given in Section 1.2. In Section 1.3, the important challenges in cloud computing are discussed. The importance and the role of Service Level Agreements (SLAs) in business interactions is described in Section 1.4. In Section 1.5, the role of Quality of Service (QoS) parameter trust and risk is discussed. Section 1.6 presents the objectives of this thesis. Section 1.7 overviews the scope of the thesis. The significance of the thesis is presented in Section 1.8. Section 1.9 provides the plan of the thesis and Section 1.10 concludes the chapter.

1.2 Cloud Computing

Cloud computing is defined as *a parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements* [1]. Cloud computing is the paradigm in which the pool of computing resources (hardware and software) from service providers is shared with users over the internet, based on a pay-as-you-go mechanism [2]. The cloud computing architecture is depicted in Figure 1.1.

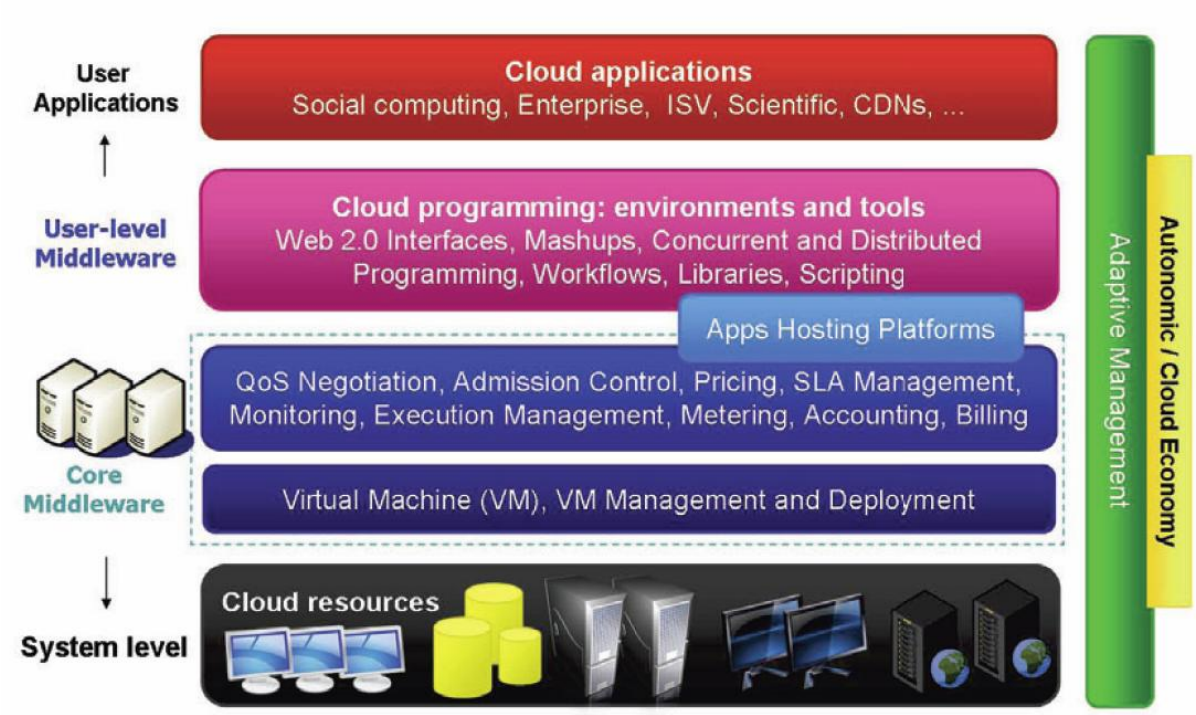

Figure 1.1: Cloud computing reference model (adopted from [1])

Understanding of cloud architecture depicted in Figure 1.1 helps to identify the layer on which the work in this thesis will be carried out. *System level layer* consists of physical cloud resources on which the cloud services run. *Core middleware layer* is the virtualization layer with SLA and QoS aspects. *User-level middleware layer* consists of programming environments for the developers to create cloud services and application hosting environments which run on virtual layer. *User applications layer* is the cloud applications layers which consists of software as a service.

In cloud computing, based on service-oriented computing, everything is a service. In terms of computing resources, hardware, software, applications, infrastructure and platform are considered a *service*. Cloud computing is based on a grid computing and service-oriented computing paradigm. It shares common features such as service discovery, service description, service composition and service management, with service-oriented computing. It is based on the architectural principles and software specifications of service-oriented computing to connect computers and devices using standardized protocols across the internet [3]. However, cloud computing is different from service-oriented computing in that it provides flexible on-demand services with flexible cost models for the services [4].

Based on virtualization and a pay-as-you-go mechanism, cloud computing promises the quick development and deployment of IT infrastructure, low maintenance cost, on-demand services, and fast computation to its users. Similar to services in SOA, cloud computing has a self-serve autonomous mechanism for the provision of services on a consumer's demand [4]. The economic benefit of cloud computing makes it an attractive choice for many real-world businesses. In cloud computing, since services are requested on demand, the service charges are applied only to the service that has been consumed. There is no upfront cost for the service infrastructure and there are no charges for service subscription.

International Data Corporation (IDC) estimates that by 2015, nearly 20% of information will be touched by cloud computing service providers and about 10% of data will be maintained in a cloud [5]. Today, around 10% of information runs through virtual servers and it is expected that this number will grow to more than 20% in 2015. Figure 1.2 depicts the adaptation of public clouds as predicted by IDC.

Figure 1.2: The digital universe and public cloud, 2015 (reproduced from: [5])

However, to realise the benefits of cloud computing, several challenges must be overcome, one of these challenges being the *service management* of software services. In this thesis, informed decision making techniques in cloud computing are proposed which help the service user to make decisions about the service selection and the continuity of the service with the service provider through service management. This challenge is discussed further in the next section.

1.3 Challenges of Cloud Computing

As discussed in the previous section, by using cloud computing architecture, users can access massive computing resources for short time periods without having to build their own infrastructure. This is achieved in the form of services. Also, by using cloud computing, service providers will be able to provide cost effective and on-demand IT services to multiple-service users on a multi-tenancy basis for their various needs. Cloud computing architecture works on the vision of the services (required by the service users and provided by the service providers) being delivered as promised at the required capacity, time and need. However, the two main challenges that need to be addressed in order to ensure that this vision is achieved are:

(a) the service user must select a capable service provider (from the possible ones) from whom they can achieve the desired outcomes;

(b) the service provider has to ensure that they have the capability and necessary resources to deliver on the agreed service to the service user. This will ensure the achievement of financial gains to the service providers as an outcome of the service.

The above challenges relate to *service management*. In addition to service management, other issues that are considered important in cloud computing are *privacy* and *security* issues.

Privacy refers to the fact that a consumer doesn't want to reveal his proprietary information to his competitors, whereas security refers to the provision of a safe computing environment in which to carry out transactions or to consume services. In addition to privacy and security as important issues, the current literature also emphasizes performance of a service provider [1]. The challenge lies in assessing the performance of the service provider through service management and assessing it from the consumer's point of view.

In the cloud computing environment, service management starts with a contract between a service consumer and a service provider, called the Service Level Agreement (SLA). An SLA is a formal contract between a service provider and a consumer, guaranteeing Quality of Service (QoS). A service provider usually establishes a threshold against which a service is expected to be delivered, based on their capabilities. The performance of service provider is measured against these thresholds, called Service Level Objectives (SLOs). Structurally, an SLA may consist of a number of SLOs. These SLOs define QoS properties for the agreed-upon service, such as availability of service, throughput and response time. Effective service management can be achieved by effective SLA management, therefore in the next section, SLA management is discussed.

1.4 SLA Management

Since on-demand IT services are provided to multiple-service users on a multi-tenancy basis, SLA management in cloud computing is an emerging and important issue [6]. In the current literature, a great deal of work has been done on SLA

management, which comprises SLA negotiation and formation; service provider selection and SLA monitoring in the cloud computing environment from the service provider's perspective [7]. In the digital world of computing, since there is no physical contact in the online market place, the consumer has to rely on the promises of service providers. However, it is difficult to judge the intention of a service provider. An experience with a service provider may be good or bad. Therefore, it is important to implement SLA management from the consumer's point of view where a consumer is able to select a service provider and monitor the performance of the service provider. Most *SLA management* techniques in the literature have been proposed from the service provider's perspective through the service provider's dashboard. From the consumer's point of view, performance monitoring through the service provider's dashboard may not be satisfying since the level of service shown by the dashboards may be different from the actual level of service that a consumer receives, due to factors such as the internet connection and the location of the consumer [8]. SLA management, in this case, should involve a mechanism through which the consumer has control over SLA management. SLA management consists of different phases, three of these phases being: SLA establishment, service provider selection and SLA monitoring. These phases are discussed the following in sub-sections.

1.4.1 SLA Establishment

SLA establishment is a three-step process: the identification of a potential service provider, the negotiation and the formation. After this process, a capable service provider is selected. Since SLA is a contract between the consumer and the service provider, it contains terms and conditions on which both the service provider and consumer agree. It is important for both the service provider and consumer to clearly define the terms and conditions of the agreement to avoid any conflict in the future [9] which is one phase of *SLA management*. Forming SLAs is beneficial to both groups of users. For a service provider, an SLA provides performance metrics against which he checks his performance to improve quality of his service and for a consumer, it provides assurance of QoS.

For the SLA establishment process, a consumer submits his requirements to the service provider [10]. The service provider maps the consumer's requirements to his capability. If the consumer's requirements match the service provider's capability to offer the service, the SLA is established. Otherwise, negotiation starts between the consumer and the service provider. Multi-round negotiation may consist of offers. This may take place by adjusting some of the terms and conditions in the request. After successful negotiation, SLA formation starts by mapping the service requirements to a standard service description. The process of SLA negotiation and formation is shown in Figure 1.3.

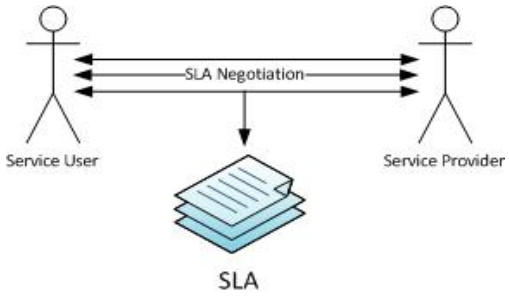

Figure 1.3: SLA negotiation and formation

After SLA establishment, the service provider needs to be selected on the basis of his capability to deliver the requested service and this capability of a service provider needs to be assessed. Once a consumer selects a service provider based on his best capability to fulfil consumer’s request, the service provider’s performance needs to be monitored.

It can be concluded from above discussion that SLA management broadly is a two-step process that includes service provider selection and monitoring his performance after the SLA establishment phase. In the next section, these two steps are discussed further.

1.4.2 Service Provider Selection and SLA Monitoring

The SLA establishment process between the consumer and service provider is carried out before the start of an interaction. After SLA establishment and before the start of an interaction, the consumer has to select a service provider. Once the consumer enters into an interaction with the selected service provider, he then needs to monitor

the performance of the service provider. For SLA management, the time duration of an interaction can be divided into two sections, as depicted in Figure 1.4.

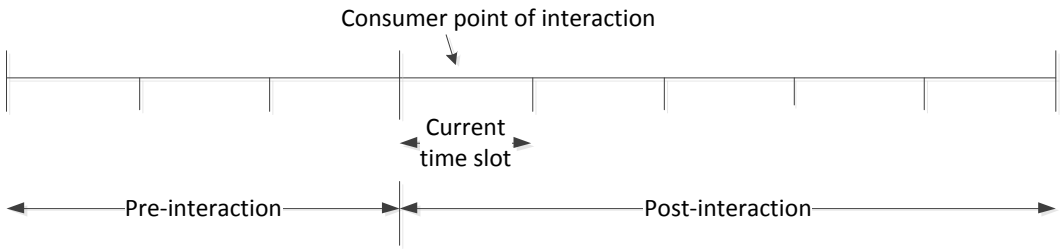

Figure 1.4: Time space of an interaction

In Figure 1.4, the total duration of an interaction is represented by time space. The pre-interaction time phase represents the portion of time space before the start of an interaction and the post-interaction time phase represents the portion of time space after the start of an interaction. Each time space is then divided into 'time slots' which are non-overlapping interval of time in the time space. 'Consumer point of interaction' in the time space is the point of time on the time space in which a consumer interacts with the service provider. It is suggested in this thesis that service provider selection be done in pre-interaction time phase and service provider performance monitoring be done in post-interaction time phase.

For a service provider selection, *trust* has been used in the existing literature [12]. A service provider may offer different service levels for the same service on different rates. For example, a video conferencing service may be offered by a service provider with different features. A consumer may subscribe to different service levels with the service provider. Since selection of cloud service provider on the basis of service levels is a challenging task, the concept of *trust* can be used for this purpose in cloud computing.

For service performance assessment, a consumer has to monitor the service during its delivery to ensure the quality of service according to the SLA [11]. In literature *risk* has been used to assess the performance of a service provider [16]. As highlighted earlier in this section, after a consumer enters into an interaction with a service provider, he can monitor the performance of the service provider through the

performance dashboards provided by the service provider. It was also discussed that due to the internet connections and location of the consumer, the actual service level received by the consumer may be different from the level of service shown on service provider's dashboard, leaving the consumer with limited control over the performance of the service provider. There is a need for an independent methodology for the consumer to monitor the performance of the service provider and to make informed decisions about the continuity of the service [8].

Since service provider performance assessment in cloud is crucial for SLA management, the concept of *risk* can be used for this purpose in cloud computing. Risk analysis of the performance of a service provider provides an insight to the consumer about possible financial losses in the interaction due to SLA violation.

The overall objective is to improve QoS for the consumer through SLA management in a cloud computing environment through a flexible model and this flexibility can be obtained using trust and risk analysis. In next section, these two important concepts are discussed.

1.5 The Concepts of Trust and Risk

Before the concepts of trust and risk are discussed, it is important to elude misconceptions about trust and security as well as security or privacy risk and performance or financial risk. Trust and security are treated as a single concept which is misleading. *Trust* refers to a service requester's belief that a service provider will deliver a mutually agreed service according to a promise whereas *security* refers to the provision of a secure computing environment for business activities. Similarly, when discussing the concept of *risk*, it is often associated with *security* or *privacy risk* which does not represent financial loss due to the poor performance of the service provider in the interaction. It important to clarify these concepts because the concept of trust used in this thesis is not associated with security and the concept of risk in this thesis is associated with performance and financial risk not the security or privacy risk. In the next sub-sections, the concepts of *trust* and *risk* are discussed.

1.5.1 Trust

Trust plays an important role in our daily life. Most particularly, trust plays a crucial role in business dealings. On the basis of trust, one can decide whether to enter into a transaction or to continue an interaction with the business partner. For online systems' users, opinions about the service or the service provider and an assessment in the form of service ratings are common tools to determine the trustworthiness of a service provider. Since in a virtual environment, the consumer has no physical contact with service provider, the aim of the trust evaluation is to give consumers a feeling of 'try before you buy' before they interact with the service provider.

Trust has benefits for both the consumer and the service provider. It boosts consumer confidence in the service provider's capability to provide the requested service according to the consumer's specifications. By providing a good service and abiding by the SLA's terms and conditions, a service provider gains a good reputation, which results in the growth of their business. It is only through trust evaluation systems that service providers are able to collect consumer feedback to improve their service.

Trust is a way to evaluate service quality in business interactions. Thus, a service user 'A' evaluates the service quality of service provider 'B' for a given timeslot and a given context. For example, a service user 'MyBusiness' evaluates a service provider 'GoodInternet' for a video conferencing service which he used for a month and rates the service provider as either trustworthy or untrustworthy. Trust values for different timeslots but in the same context can be accumulated to form the *reputation* of the service provider over a time period. The reputation value for a specific service provider is useful in business and can be shared when asked by other service users. The aforementioned concept of trust has been defined by Chang et al. [12] as *"the belief that the trusting agent has in the trusted agent's willingness and capability to deliver a mutually agreed service in a given context and in a given timeslot"*. In this definition, the trusting agent represents the consumer or service user and the trusted agent represents the service provider. In this thesis this definition of trust has been adopted. Since cloud is based on service-oriented architecture and trust has been used successfully for service provider assessment [12], the same notion of trust should also hold true for cloud environments.

Since SLA management is an important issue in cloud computing, one way to address this problem is to use the trust evaluation of the business partner to ensure the QoS. As discussed earlier in this section, as a human we evaluate trust consciously, sub-consciously and unconsciously and base our decisions on this. This is part of our daily life. When building e-Commerce systems using open standard computing such as cloud computing on the basis of trust, this human behaviour should be replicated. Decisions about service quality in virtual environments such as cloud computing before the start of an interaction can be based on trust evaluation. Therefore, it is important to look at trust evaluation methodologies in these environments.

1.5.2 Risk

Another important concept related to SLA management is risk. Risk, in simple words, is the possibility of the occurrence of an undesirable event that leads to failure or loss. Risk provides a way to determine the potential failure or loss and helps preventive measures to be taken against it. This is part of daily life. As humans, we evaluate risk in most decisions of our life. For example, we evaluate risk while driving a car, operating electrical equipment, making online shopping purchases etc. In every situation of life, either consciously, unconsciously or subconsciously, we evaluate risk. This is part of human nature. We identify a threat or hazard in a particular situation and we evaluate the possible consequence of this hazard. In cases when consequences are high, we take preventive measures.

Events in our daily life can be identified as either low risk or high risk. Low risk means that the impact or consequence of the event may be low, whereas high risk means that the impact or consequence of the event may be high.

Risk assessment of the service provider is a valuable tool for SLA management in cloud computing. Risk highlights the threats and the level of loss in the interaction. In the business domain, risk is the possibility that the expectations of a service requester will not be met due to service degradation or service failure. It helps in determining the financial loss due to expectations not being met. Like trust, risk determination is also a very important part of business partner evaluation in a virtual environment, such as cloud computing, since the service requester or consumer may

not have physical contact with the service provider. In cloud the notions of performance and financial risk can be used to determine the impact of service degradation on consumer's expectations.

It should be noted that risk will be analysed differently, according to the viewpoint in which it is being analysed. As previously discussed, the level of risk in security and privacy is not the same as the level of risk in a business interaction. The level of risk in security and privacy ascertain that a business transaction can safely occur but it does not guarantee that the service provider will act according to the service requester's expectations.

Risk management is a well-developed discipline, particularly in engineering, which includes risk identification, risk assessment, risk prioritization and risk control. As a formal risk assessment process, risk identification is important. The risk identification process determines the threat or hazard in an environment in which risk is being assessed. The outcome of the risk identification process is a list of undesirable events (threats). The risk assessment process determines the possibility of the occurrence of these undesirable events and their consequences. Risk prioritization is the process of listing risk according to its importance. Depending on the priority and risk assessment, risk control is put in place to avoid the impact of an undesirable event.

1.5.3 Relationship of Trust and Risk in Informed Decision Making

As discussed in the previous subsection, both trust and risk play key roles in determining the service quality. These two aspects of quality of service complement each other. Together, these concepts are helpful in detecting the fraudulent and deceptive behaviour of a service provider in a virtual environment where it is more difficult to judge a service provider's behaviour due to physical absence. It is not uncommon in such environments that service providers act unscrupulously. If a consumer is not aware of malicious service providers and commits resources such as money to the service of a service provider, they may suffer financial loss.

In other cases, a service provider may not exhibit malicious or fraudulent behaviour but they may not provide the service according to the commitments they made to the

consumer. The concepts of trust and risk, in this case, help to determine the level of service provided by the service provider.

It can be seen from the above discussion that trust and risk have a relationship with the behaviour of the service provider. There are three types of relationships between trust, risk and behaviour proposed by [13]. These relationships are discussed in order to understand the importance of both trust and risk in online transactions in the following:

1) When trust and risk are considered independently to behaviour

In this type of relationship between trust, risk and behaviour, it is considered that trust and risk are not related through the cause-effect relationship and are independent of each other but both simultaneously affect behaviour. In some cases [14], both trust and risk are used without specifying what type of relationship exists between the two. In other cases [15], trust is used alone to determine the level of trust, along with uncertainty.

2) Mediating relationships

In this type of relationship, risk is seen as a mediator between trust and behaviour. In other words, trust affects perceived risk, which affects behaviour. For example, the perceived risk of buying a book from a trusted seller is lower than buying a book from a stranger. In e-Commerce, a large number of researchers agree with the mediating role of risk in the relationship between trust and behaviour.

3) Moderating relationships

In this type of relationship, it is considered that the level of trust is different when the level of risk is low versus when the level of risk is high. In other words, when the perceived risk is high, trust has more importance; when the level of risk is low, trust may not have much importance. For example, consider the sale of low cost items versus high cost items on the web. The likelihood of a negative outcome may be the same for both low cost items versus high cost items, but the magnitude of loss will be higher for high cost items. Therefore, trust can be used to alleviate the fear of an online buyer depending on the likelihood and magnitude of loss.

From the discussion in the previous sections, it can be concluded that, to the best of our knowledge, no such *SLA management* technique exists in the literature that allows the consumer to select the best service provider on the basis of trust and then allows the consumer to monitor the performance of a service provider to make informed decisions about the continuity of service. The objectives of this thesis are presented in the following section.

1.6 Objectives of this Thesis

In the previous sections, the importance of SLA management in the cloud environment was discussed. Also, two important QoS concepts, namely trust and risk, which can be used for effective SLA management were discussed. In this thesis, an SLA management framework that incorporates trust and risk to make an informed decision on the selection of a capable service provider and to ascertain the financial outcomes to a service consumer in the event of service degradation by the service provider in the cloud environment is proposed. In other words, the overall objective of this thesis is to propose an SLA management framework by which a consumer can evaluate a service provider's performance before and after the interaction. This objective can be broken down into the following sub-objectives:

1. To propose a trust-based approach to determine the capability of a service provider to deliver the requested service before the start of an interaction.
2. To propose a risk-based approach to determine the financial consequence of the performance degradation of a service provider after the interaction.
3. To propose a risk-based decision making system to assist the service user in the decision making process about the continuity of the service.
4. To propose an evaluation framework to evaluate the SLA management framework.

1.7 Scope of the Thesis

This thesis focuses on Quality of Service (QoS) assessment using trust and risk concepts in cloud computing. Initially, trust helps to establish the relationship with the service provider. Then, business interaction is judged on the basis of the risk involved due to service degradation. The QoS assessment in this thesis is limited to the concepts defined above and does not consider the privacy or security concepts

associated with business interaction. Moreover, trust and risk are purely defined in the business interaction context. The meaning of trust or risk may be different for different domains.

1.8 Significance of the Thesis

To the best of my knowledge, in the present literature, SLA management to ascertain the financial outcomes of a consumer in the cloud environment have not been addressed before. In this thesis, such an SLA management framework that will enable a consumer to make informed decisions about the selection and continuation of a service is proposed. Hence, the significance of the thesis can be summarized as follows:

1. This thesis proposes SLA management from a consumer's point of view by introducing trust- and risk-based assessment methodologies.
2. Assessment before the start of an interaction consists of the selection of a capable service provider using trust or reputation analysis. This thesis proposes trust and reputation determination methodologies.
3. Assessment after the interaction consists of determining the probability of service degradation using risk analysis. This thesis proposes a performance risk assessment methodology for this purpose.
4. Performance risk assessment follows the determination of the impact of service degradation. This thesis proposes a financial risk assessment methodology for this purpose.
5. This thesis also proposes a methodology through which the financial loss in an interaction can be determined based on financial risk analysis.
6. This thesis proposes an SLA management framework by which a service user is able to make informed decisions about the selection and continuity of a service with a service provider.

This thesis proposes that, on the basis of the assessments in the pre- and post-interaction time phases a consumer should be able to select a trustworthy service before the start of the interaction and should be able to decide on the continuation of a service after the interaction.

1.9 Plan of the Thesis

As previously mentioned, in this thesis, an SLA management framework is proposed by which the consumer can ensure his business objectives are realised while interacting with the service provider. In order to achieve the above objective, the organization of this thesis is as follows:

Chapter 2: In Chapter 2, an extensive review of the existing methods of SLA management in SOA, grid computing and cloud computing, including trust and risk assessment methodologies in the business domain, is provided. The shortcomings of the current literature are highlighted, which lead to a discussion of the issues for which a solution is proposed in this thesis.

Chapter 3: In Chapter 3, the background of the problem and the problem definition is formally discussed. The problem definition is then broken into different research issues. The concepts and terminologies that will be used while solving each issue are defined. Further, in this chapter, the various research methodologies are discussed and the most pertinent is chosen, which will be utilized in this research.

Chapter 4: In Chapter 4, the first step towards proposing a framework for SLA management in cloud computing is taken. Each part of the definition is explained in detail. A brief solution overview for each of the research issues mentioned in the last chapter is proposed.

Chapter 5: In Chapter 5, a methodology for the consumer to determine the trustworthiness of the service provider before the start of an interaction is proposed. This chapter is divided into two main parts: determining the trustworthiness of the service provider for the direct interaction of a consumer with the service provider; and the trustworthiness of the service provider for the indirect interaction of the consumer with the service provider.

Chapter 6: In Chapter 6, the methodology for the performance analysis of the service provider is proposed. Two scenarios for performance evaluation are considered: the interaction of the consumer with the service provider in the current timeslot and the interaction of the consumer with the service provider in the future timeslot.

Chapter 7: In Chapter 7, a methodology is proposed for the financial risk assessment of the resources in which the consumer invests in an interaction and that are at stake for the duration of the interaction. In addition, the performance risk that is determined in Chapter 6 is utilized to assess financial risk from which financial loss is determined.

Chapter 8: In Chapter 8, a methodology is proposed for decision making on the basis of level of loss in the interaction that is determined in Chapter 7 and the consumer's risk attitude. This methodology assists the consumer to make decisions on the future course of the interaction with the service provider.

Chapter 9: In Chapter 9, the implementation and the operation of the prototype system that is engineered in this thesis is presented, in order to validate the methodologies proposed in this thesis.

Chapter 10: In Chapter 10, the thesis is concluded by summarizing the work which is developed in this thesis, after which the potential for further work is identified.

1.10 Conclusion

The adoption of web technologies, such as cloud computing, can be facilitated by SLA management using trust and risk analysis. The importance of SLA management in the cloud computing paradigm was discussed, as was how trust and risk can be used for the evaluation of the SLA before and after the interaction. The relationship between trust and risk was also discussed. The objectives of the thesis were outlined along with the scope and significance of the thesis. Finally, the plan of the thesis was presented.

1.11 References

[1] Rajkumar Buyya, Suraj Pandey, and Christian Vecchiola, "Market-Oriented Cloud Computing and the Cloudbus Toolkit", in Cloud Computing. vol. 5931, M. Jaatun, G. Zhao, and C. Rong, Eds., ed: Springer Berlin Heidelberg, 2009, pp. 24-44.

[2] B. Furht and A. Escalante, *Handbook of Cloud Computing*, Springer Publishing Company, Incorporated, 2010.

[3] M. N. Huhns, "Service-oriented computing: Key concepts and principles", *IEEE internet computing,* vol. 9, p. 75, 2005.

[4] T. Dillon, C. Wu, and E. Chang, *Cloud Computing: Issues and Challenges, 24th IEEE International Conference on Advanced Information Networking and Applications*, April 20 - 23, 2010, pp. 27-33.

[5] D. R. John Gantz. (2011, June 5, 2011). Extracting Value from Chaos. Available: http://idcdocserv.com/1142

[6] J. Morin, J. Aubert, and B. Gateau, "Towards Cloud Computing SLA Risk Management: Issues and Challenges", *45th Hawaii International Conference on System Science (HICSS)*, 2012, pp. 5509-5514.

[7] S. Bose, A. Pasala, D. Ramanujam A, S. Murthy, and G. Malaiyandisamy, "SLA Management in Cloud Computing: A Service Provider's Perspective", in *Cloud Computing*, ed: John Wiley & Sons, Inc., 2011, pp. 413-436.

[8] Z. Zheng, X. Wu, Y. Zhang, M. Lyu, and J. Wang, "QoS Ranking Prediction for Cloud Services", IEEE Transactions on Parallel and Distributed Systems, Vol. 24, Issue 6, 2012.

[9] T. B. Quillinan, K. P. Clark, M. Warnier, F. M. T. Brazier, and O. Rana, "Negotiation and Monitoring of Service Level Agreements", in *Grids and Service-Oriented Architectures for Service Level Agreements*, P. Wieder, R. Yahyapour, and W. Ziegler, Eds., ed: Springer US, 2010, pp. 167-176.

[10] Y. L. Sun, R. Perrott, T. J. Harmer, C. Cunningham, P. Wright, J. Kennedy, A. Edmonds, V. Bayon, J. Maza, G. Berginc, and P. Hadalin, "SLA-aware Resource Management", in *Grids and Service-Oriented Architectures for Service Level Agreements*, P. Wieder, R. Yahyapour, and W. Ziegler, Eds., ed: Springer US, 2010, pp. 35-44.

[11] D. Khader, J. Padget, and M. Warnier, "Reactive Monitoring of Service Level Agreements", in *Grids and Service-Oriented Architectures for Service Level Agreements*, P. Wieder, R. Yahyapour, and W. Ziegler, Eds., ed: Springer US, 2010, pp. 13-22.

[12] E. Chang, F. Hussian, and T. Dillon, *Trust and Reputation for Service-Oriented Environments: Technologies For Building Business Intelligence And Consumer Confidence*: John Wiley & Sons, 2005.

[13] D. Gefen, V. Srinivasan Rao, and N. Tractinsky, "The conceptualization of trust, risk and their electronic commerce: the need for clarifications", *36th Annual Hawaii International Conference on System Sciences*, Jan 6-9, 2003.

[14] K. Kim and B. Prabhakar, "Initial trust, perceived risk, and the adoption of internet banking", presented at the Proceedings of the twenty first international conference on Information systems, Brisbane, Queensland, Australia, 2000, pp. 537 - 543.

[15] J.-Y. Son, S. Narasimhan, and F. J. Riggins, "Factors affecting the extent of electronic cooperation between firms: economic and sociological perspectives", presented at the Proceedings of the 20th international

conference on Information Systems, Charlotte, North Carolina, United States, 1999, pp. 556-560.

[16] O. K. Hussain, " Risk measurement, prediction and management in e-business and service oriented environment ", Ph.D., Curtin Business School, Curtin University, Perth, 2008.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the previous chapter, the importance of SLA management in cloud computing has been highlighted to ensure the achievement of desired outcomes by employing trust and risk concepts. In this chapter, an overview of existing literature on SLA management in cloud computing has been studied and presented. The relationships between trust and risk concepts is studied for the assessment of SLA management in cloud computing. The findings of this study are summarized considering by evaluating existing SLA management techniques in cloud computing from consumer's point of view in an interaction with the service provider.

To gain a better understanding of problem discussed in previous chapter, a literature review is undertaken which examines SLA management in cloud computing in section 2.2. The importance of trust and reputation concepts for SLA management is discussed in section 2.3. In section 2.4, the importance of transactional risk assessment for SLA management is presented. In section 2.5, an integrative view of all approaches for SLA management in cloud computing before and after the interaction is given. Finally, section 2.6 is the conclusion of the chapter.

2.2 SLA Management in Cloud Computing

SLA management in cloud computing has been formally defined and discussed by Bose et al. [1]. They defined five phases for SLA management namely, feasibility, on-boarding, pre-production, production and termination. After receiving a consumer's request, the service provider performs a feasibility analysis. After the feasibility analysis, which includes technical, infrastructure and financial feasibility, a service provider moves to the on-boarding of the application and the pre-production phases, where the service provider performs SLA negotiation and formation. In the production phase, the performance is monitored by the service provider. Although the work done by Bose et al. is significant in defining SLA management, their work is from a service provider's point of view.

In this section, the SLA management approaches from user's perspective in cloud computing are discussed. This section is divided into two sub-sections. In the first sub-section, the literature that discusses role of SLA in cloud computing is explored. In the second sub-section, the SLA evaluation approaches in cloud computing are discussed.

2.2.1 SLA in Cloud Computing

In previous chapter, the importance of SLA in ascertaining and maintaining the QoS in cloud computing was discussed. In this section, the literature that discusses the role of SLA in ascertaining QoS is discussed which leads to the discussion of SLA monitoring and evaluation in Section 2.2.2. In reviewing the existing literature for SLA in cloud computing the evaluation criteria such as *the concepts of risk and trust in cloud computing, service provider performance assessment according to SLA, the financial impact of performance degradation,* and *financial cost of service migration* are considered.

Buyya et al. [2] presented high-level market-oriented cloud architecture. They discussed the SLA resource allocator that acts as an interface between the cloud service provider and the external user. For resource management, the SLA resource manager requires a service request examiner and admission control and pricing mechanisms. The service request examiner and the admission control mechanisms check the submitted request against the QoS parameters and accepts or rejects the request. To do this, it needs the latest status of the resource usage and the status of workload processing. The pricing mechanism decides how service requests are charged on the basis of the resources assigned to them. In addition to these mechanisms, they also introduced the VM monitor and service request monitor mechanisms. They discuss risk with respect to security but they do not discuss the service provider's performance according to the SLA.

Furht [3] discussed the concept of virtualization for cloud computing. Virtualization is a very important part of cloud computing. It is a way of sharing resources among different applications with the objective of better server utilization. Two important components of virtualization are virtual machines and virtual networks. The virtual machines provide IT infrastructure on-demand, while virtual networks support users

with a customized network to access cloud resources. Furht discussed issues such as performance, security and privacy, control, bandwidth costs and reliability. Performance is one of the major issues. In some data-intensive and transactional-oriented applications, cloud computing may lack adequate performance. He also mentioned that bandwidth cost is high for data-intensive applications. Although he discussed performance issues, he did not discuss the impact of performance on business outcomes.

Dillon et al. [4] discussed several issues and challenges of cloud computing. They mentioned security as a top issue of clouds, followed by performance and availability. The multi-tenancy model and resource pooling introduces new challenges. Two issues related to multi-tenancy are shared resources the on same physical machine and reputation fate-sharing, where a notorious user with a criminal mind may affect many good cloud 'citizens'. The costing model is an important issue as well. There is a trade-off between computation, communication and integration. Migration can raise the communication cost while reducing the infrastructure cost. Moreover, a cloud is suitable for CPU-intensive jobs compared to data-intensive jobs unless cost-saving offsets extra data transfer costs. The splitting and integration of data requires different data adaptors for different cloud service providers due to their proprietary protocols. However, splitting and integration adds substantial financial cost. The Service Level Agreement (SLA) is another issue discussed by the authors. The client needs assurance of quality, availability, reliability and performance from the cloud service providers. This assurance is provided through the SLA. There are several issues associated with SLAs. The first issue is defining the SLA specifications. With an appropriate level of granularity, most of the consumer's expectations should be covered in the SLA and at the same time, these expectations are simple to weight, verify, evaluate and enforce using the resource allocation mechanism. In addition, the real-time evaluation and adjustment of an SLA is also needed which raises a number of problems for the cloud providers from an implementation point of view. This means that resource managers should be able to access precise and updated information on the resource usage within the cloud at any particular time. This needs to be carried out in an automatic fashion due to the "self-service" feature of cloud computing. Also, advanced SLA mechanisms are needed to

constantly incorporate user feedback and customization features into the SLA evaluation framework. Dillon et al. discussed performance issues and SLA evaluation in detail, but they did not discuss its financial impact in their work.

Zhang et al. [5] discussed several issues/challenges of cloud computing, such as automated service provisioning, virtual machine migration, server consolidation, energy management, data security, novel cloud architectures etc. In relation to automatic service provisioning, the authors mentioned that it is not easy to map Service Level Objectives (SLOs), such as QoS requirements, to low-level resource requirements, such as CPU and memory requirements. They mentioned that a performance model should predict future demand and resource requirements. They proposed the Queuing theory, Control theory and Statistical Machine Learning to build the performance model. They also discussed pro-active and re-active approaches for resource control. Both of these approaches should be used in a dynamic operating environment, such as a cloud. The authors also discussed the benefits of virtual machine migration, especially 'live migration'. However, they did not discuss the financial cost associated with migration.

Voorsluys et al. [6] discussed the cost of 'live migration'. Through an experimental setup, they defined the SLA parameters against which SLA violation is recorded. They noticed the duration of the migration effects and downtime experienced by the application. They concluded that, provided the migrations are done at correct times, there is no cost associated with them. On the basis of this experiment, the authors proposed future work to develop smarter and more efficient SLA-based resource allocation systems. They used response time as a criterion to evaluate migration cost. This work provides valuable insight into live migration. But the issue to determine migration cost in terms of finances to meet SLOs was not discussed in this work.

In other work, Voorsluys et al. [7] discussed the issues of security, privacy and trust. In addition to this, data lock-in and standardization is also an issue. It is a concern of cloud computing users that their data is locked in by a certain provider. If a user wants to move to another service provider, due to a lack of interoperability between cloud service providers, the user is unable to do so. So work is in progress for cloud computing standardization. The authors also discussed QoS requirements by

implementing availability and performance guarantees in SLA. But they did not discuss the service provider's performance and its impact on the business outcomes of a consumer.

Buyya et al. [8] discussed the importance of SLA in their work for QoS guarantees. They discussed the open challenges in clouds. Since virtualization is one of the key concepts in clouds, they discussed issues related to virtualization. The major issue with virtualization is that when a large number of VMs are created, an effective management mechanism is needed to meet the expectations of the users. To provide this guarantee, VMs need to migrate to suitable servers to ensure QoS. They discussed computational risk management in their work which identifies, assesses and manages risks in the execution of applications. They also discussed performance risk associated with not fulfilling an SLA. But they did not discuss the impact of performance risk on business outcomes.

2.2.2 Service Evaluation in Cloud Computing

As highlighted in the previous chapter, service evaluation is very important for a service consumer and one way of undertaking service evaluation is SLA evaluation. To do this, good SLA management techniques are needed that not only evaluate an SLA before the start of the interaction but also evaluate the SLA after the interaction to ensure the initiation of the desired outcomes of a consumer. In this section, the existing literature on SLA management is evaluated on the basis of following criteria: *SLA evaluation from a service consumer's point of view, SLA decomposition, data collection intervals for SLA evaluation* and *SLA evaluation for transactional risk assessment*. Moreover, SLA evaluation in Service-Oriented Architecture, Grid computing and Cloud Computing is reviewed in the next sub-sections.

2.2.2.1 SLA Evaluation in Service-Oriented Architecture (SOA)

He et al. [9] discussed the SLA formation and negotiation process. The SLA process can be undertaken using agreement protocols such as a WS-Agreement. In short, a service requester submits their request through a web portal. The client submits their requirements to a Business Rule Consultant which activates the agreement. The service provider's service templates are already available through a service registry.

The Business Rule Consultant maps the user requirements to the available templates of the service providers. After completing this phase, negotiation starts through agreement negotiator agents. If negotiation is successful, a final agreement between agents is reached through the agreement factory. After this, to enforce the agreement, SLA monitoring begins. At this stage, for reasons of credibility, the authors propose commissioning a third-party agent to monitor the SLA.

Although He et al. proposed that SLA evaluation in SOAs should be done using performance metrics, which is SLA decomposition, they did not propose an approach that captures and assures the performance of the service provider for SLA evaluation. Moreover, their proposed approach for SLA evaluation is not from a service consumer's point of view.

Foster et al. [10] proposed SLA monitoring in SOAs by a method of decomposing SLAs into expressions suitable for monitoring. For monitoring, they discussed either intrusive [11-12] or event-based [13-14] monitoring in service-based systems. However, they did not consider SLA monitoring intervals.

Park et al. [15] introduced an SLA monitoring module, called S-Mon, which enhances the security capability of the billing system called THEMIS. In this model, a monitoring report is generated for the user after the end of one session or on the user's request to monitor data. Details on the system status are then detected. However, Park et al. failed to discuss the mapping of data to SLA parameters.

2.2.2.2 SLA Evaluation in Grid Computing

Most of the work on SLA monitoring has been done from a service provider’s point of view. However, a service consumer is uncertain about a service provider’s ability to meet service expectations. Smith et al. [16] addressed this uncertainty using statistical methods through which they can predict a service provider’s performance. However, they did not discuss the impact of the service provider’s performance on the financial outcome of the consumer to assess the transactional risk.

Quillinan et al. [17] mention that the Quality of Service (QoS) provided by a service provider can be assessed on the basis of the SLA. When it comes to monitoring of the SLA, penalties can be imposed as a result of violation by service provider. Online

monitoring is one way of monitoring service violations. Online monitoring means that continuous monitoring of service is done based on monitoring intervals. These monitoring intervals depend upon agreement terms and can be in seconds, minutes, hours or days. A second monitoring approach discussed here is reactive monitoring and a third monitoring approach is offline monitoring. In terms of penalties for service violations, two types are defined: reputation-based penalty and monetary-based penalty. Although Quillinan et al. discussed SLA monitoring techniques that are based on different data collection intervals, their proposed approach is for both service consumer and service provider. However it does not assess the transaction risk in an interaction.

Khader et al. [9] discussed the reactive monitoring technique for SLAs. After discussing online and offline monitoring, they stated that the reactive monitoring approach balances the trade-off between online and offline monitoring. The authors suggested that the monitoring technique should be chosen according to the SLA monitoring requirements. For the evaluation of some contracts, reactive monitoring is suitable while for others, online or offline monitoring is required. Khader et al. introduced the reactive monitoring approach but they did not discuss the outcome of not achieving the required performance from a consumer's point of view.

Oliveira et al. [10] provided a benchmark to monitor SLAs using a grid system. They argued that since most of the monitoring techniques are embedded in a particular SLA specification, they rely on specific protocols which impede the wide-spread adoption of grid infrastructures. They introduced a benchmark, called Jawari, to monitor the SLA. Jawari is an external entity that can be used by end users to validate adherence to grid services. Oliveira et al. introduced a few benchmarks (metrics) on the basis of which the SLA evaluation is done. However, these metrics are not fine grained, that is, the SLA cannot be decomposed into measurable metrics. Moreover, the objective of SLA evaluation is not to assess transactional risk.

2.2.2.3 SLA Evaluation in Cloud Computing

In a work carried out by Betgé-Brezetz et al. [18] they first discussed the importance of SLA management, which they described as a key differentiator between the selection of service providers. In other words, for a consumer, SLA provider

differentiation is the key factor between the service provider's offers. The reliability of SLA management and monitoring by SLA providers provides a way for the client to judge the service quality. In the cloud environment, due to its dynamic nature, the SLA between a service user and a service provider should be formalized as much as possible to provide QoS. After discussing the importance of SLA management, they introduced pro-active SLA assurance architecture which consists of pro-active Service Level Specification (SLS) violation detection and management. However, this assessment is from a service provider's point of view.

Morin et al. [19] discussed SLA management through risk management. They proposed that, due to the nature of the cloud environment, traditional parameters for risk management are not effective for cloud computing and therefore, SLA management in the cloud should be done through a flexible risk management model for which they proposed dynamic risk management coupled with SLA and exception management. However, they discussed that risk management is a security risk not a transactional risk.

In their work, Zheng et al. [20] proposed a cloud ranking architecture through predictive QoS ranking which assists the consumer in decision making for the selection of a service provider. They provided an example of a cloud service provider which promised to provide the level of service requested by the client and agreed in the SLA but the service provider received different levels of service due to factors such as internet connection and the location of consumer. It is therefore important that each service user should be able to evaluate the performance of the service provider instead of using performance dashboards provided by the service providers. One way of achieving this is by using the past performance of service providers based on the experience of other users. According to them, a client-side performance measure is a more realistic experience of user usage experience.

A monitoring system, called Sandpiper's, was introduced by Wood et al. [21]. This system detects hotspots automatically in addition to the remapping or reconfiguration of VMs. It works by proactively preventing SLA violation by setting strict thresholds. However, the system proposed by Wood et al. fails to consider

monitoring intervals and SLA decomposition. Furthermore, it cannot provide the necessary monitoring information to the service user.

An SLA monitoring framework, called LoM2HiS, was proposed by Emeakaroha et al. [22]. In this framework, there is a mapping of rules between resource metrics and user-defined SLAs by implementing SLA management that monitors low-level infrastructure parameters. Later on, they used this framework as an architecture called DeSVi for the early detection of SLA violations through strict thresholds [23]. For the autonomic monitoring of SLAs, they use FOSII infrastructure. Although they propose architecture for monitoring and detecting SLA violations, they do not evaluate transaction risk for SLA management.

Cardellini et al. [24] proposed heuristic policies for ASP (Application Service Provider) resource management. This policy is based on Recursive Least Square (RLS) prediction algorithm to assess the workload in the next timeslot. However, this assessment is in the current timeslot only and is not able to prevent SLA violations. Moreover, the technique proposed by Cardellini et al. is from a service provider's point of view and does not assess transaction risk in the interaction.

Jun-Yan et al. [25] proposed a service-assurance-oriented platform called Cloud BOSS to manage and guarantee the service quality level in the cloud. This platform maps KPIs (Key Performance Indicators) to KQIs (Key Quality Indicators) to meet the requirements in SLAs by measuring QoE (Quality of Experience) instead of measuring QoS. The monitoring approach adopted in this framework is proactive which predicts SLA violation by setting a waning threshold. The authors undertook a quality test from the customer's perspective. Their platform monitors, controls and manages the underlying cloud infrastructure consisting of servers, storage, network elements and the delivered services, such as virtual machines. However, they do not consider transaction risk assessment in their framework.

Daniel et al. [26] proposed an SLA-based scheduling architecture which also includes service monitoring. One important component of this architecture is the trust monitor which acts as a third party and monitors the cloud. In the case of service level violation by the service provider, the trust monitor sends a notification

to the user and provider. Although the authors proposed this architecture, they do not explain how this architecture is implemented.

Ciciani et al. [27] proposed key design choices underlying the development of the Workload Analyzer (WA), a crucial component of the Cloud-TM platform which is a European Union (EU) project that provides a self-optimizing transactional data platform for the cloud. The job of WA is to monitor and categorize the resource consumption data. Then time-series-based analysis is performed on the basis of data to forecast future trends of the workload fluctuations. The prediction of SLA violation can be achieved by this analysis by enabling the users to have control over what they want. Although the proposed model can predict service violations in transactional applications, it does not take transactional risk assessment into account.

In this section, the SLA management approaches in cloud computing are discussed to assess service provider's performance. It is highlighted in previous chapter that the aim of this thesis is to assess a service provider's performance before and after the interaction. In next section, the discussion of service provider performance before the start of an interaction using trust is carried out. Therefore, the current approaches of trust and reputation assessment for business interactions are discussed.

2.3 Discussion of Determining Trust and Reputation in SLA Management in Cloud Computing

To examine the literature on trust and reputation and to highlight the importance of trust and reputation in business interactions, this section is divided into three sub-sections. The first sub-section discusses the concept of trust in literature. This discussion is important to understand the nature of trust and form of trust that can be used in trust computation. The second sub-section discusses the concept of reputation and the form of reputation that can be used in reputation computation. The third sub-section, discusses the current approaches to determine trust or reputation.

2.3.1 The Concept of 'Trust' in Literature

In this sub-section, the concept of trust in business interactions is discussed. For this, the current literature is evaluated on basis of following criteria. Trust is context dependent and trust changes over the time as defined by Chang et al. [28] and in this

thesis this definition of trust is adopted. These are important characteristics and without these the trust calculation is meaningless.

Deutch et al. [29-30] in their work, stated that the result of the trusting behaviour of a person where they are forced to trust another person in an interaction can be either good or bad. They stated that an entity is forced to trust another entity whenever it encounters an uncertain situation. However, they did not define trust.

Sztompka [31] defined trust as "a bet about the future contingent actions of others" and discusses the probabilistic nature of trust. According to him, trust is a probability with which an entity A is likely to perform an action. However, from a service-oriented point of view, this definition has the following drawbacks:

The definition does not consider that trust is a belief that an entity A has in entity B and this belief is interpreted as probability. Moreover, the definition fails to consider that trust is dependent on the context and time of the interaction. Since trust is defined by Chang et al. [28] in a service-oriented environment as the belief that an entity has in the willingness and capability of another entity to provide a specific service in a specific context and in a given point in time, the definition of Sztompka does not consider that trust is created due to the willingness and capability of an entity to provide a particular service. Hence, Sztompka's definition does not apply to a service-oriented environment.

Luhmann [32] defined trust as "a solution for specific problems of risk". This definition of trust can be used to mitigate the risk that an entity A has in dealing with another entity B. Similar to the definition of Deutch et al., Luhmann's definition describes a scenario where trust can be applied. However, no definition of trust is given.

Gambetta et al. [33] give a detailed definition of trust as "trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent will perform a particular action, both before [we] can monitor such action (or independently of his capacity of ever to be able to monitor it) and in context in which it affects [our] own action". This defines that trust has a probabilistic nature and therefore, it is similar to the definition of Sztompka. However, in contrast to

Sztompka, Gambetta defines that trust is context specific. The drawback of this definition is that Gambetta does not address the dynamic nature of trust. Moreover, it fails to address the point that trust is created due to the willingness and capability of an agent to provide a mutually agreed service.

Josang [34] defined two types of trust: trust in passionate entities (human beings) and in rational entities (software agents). Trust in passionate entities is defined as "the belief that it will behave without any malicious intent". Trust in rational entities is defined as "the belief that it will resist malicious manipulation by a passionate entity". Josang's definition is applicable in Information Technology. The first definition of trust, trust in passionate entities, is applicable to human-to-human interaction. However, this definition is too broad and does not consider the context in which trust is being evaluated. It does not consider the encounter where a human entity has behaved maliciously with another human entity. In other words, a human entity is still considered trustworthy if they acted maliciously in an interaction with another entity. This definition also fails to consider the willingness and capability of a human agent as important factors for trust.

Dasgupta [35] defined trust as "the expectation of one person about the actions of others that affects the first person's choice, when an action must be taken before the actions of others are known". According to this definition, if an entity X is interacting with entity Y, entity X will not expect entity Y to act in a malicious way. This definition does not consider the context specific and dynamic nature of trust.

Golbeck [36] defined trust as "trust in a person is a commitment to an action based on a belief that the future actions of that person will lead to a good outcome". According to this definition, if an entity X is interacting with an entity Y, entity X has the confidence that future actions with entity Y will always result in a positive outcome. The problem with this definition is that it considers positive outcomes only as a result of interaction. Moreover, this definition fails to consider the context specific and dynamic nature of trust. This definition also fails to consider willingness and capability as important factors for trust.

Buist [37] defined trust as "To trust is to willingly relinquish control, making yourself vulnerable to someone else for a certain outcome or consequence. Trust

grows as a result of positive experiences accumulated over time". This definition clearly indicates that an entity hands over control to another entity and it does not have much power in an interaction. Buist's definition is more towards control or power than trust.

Mui et al. [38-40] defined trust as "a subjective belief an agent has about another's future behaviour based on the history of their encounters". In this definition, the subjective belief of one entity in another entity is derived from their past experiences. In the literature, researchers have used this definition extensively. However, the definition of Mui et al. does not consider the context specific and dynamic nature of trust and how it is created in another entity.

Wang et al. [41-42] defined trust as "a peer's belief in another peer's capabilities, honesty and reliability based on its own direct experiences". This definition applies to peer-to-peer networks where trust is the belief that one peer has in another peer. Peer X has trust in peer Y on the basis of direct past experiences. Trust in a peer originates as a result of the capability, honesty and reliability of that peer. However, Wang et al. do not define the terms: capability, honesty and reliability. This definition also fails to consider factors such as the context of interaction and the dynamic nature of trust.

Mezzetti et al. [43-44] defined trust as "a measure of how much reliance a trustor can justifiably place on the dependability of the trustee's behaviour with in a specific context". Mezzetti et al. defined trust in terms of the behaviour of an entity providing a service to another entity in a given context. They consider the context for trust but they fail to consider the dynamic nature of trust. Moreover, they also fail to define fuzzy terms, such as reliance and dependability.

2.3.2 The Concept of 'Reputation' in Literature

In this section, the concept of reputation in business interaction is discussed. Most of the researchers consider trust and reputation as the same concepts and therefor, they use these concepts interchangeably. However, Change et al. [28] highlighted that reputation is a distinct concept and should be treated separately when evaluating a service provider's performance in a certain period of time. Similar to trust, reputation

is context dependent and changes over time. In this thesis the definition of reputation provided by Change et al. is adopted. These characteristics of reputation to evaluate current literature are used.

Sabater et al. [45-48] defined reputation as "an opinion or view of one about something". We infer from this definition that Sabater et al. consider context for reputation when they mention 'something' in their definition. However, their definition does not consider the dynamic nature of reputation. They also fail to address how the reputation of an entity originates.

Adul Rahman et al. [49-52] defined reputation as "an expectation about an agent's behaviour based on information about its past behaviour". According to this definition, the reputation that one entity has on another entity is based on past experiences. However, the term 'expectation' in this definition is vaguely defined. This definition does not consider the context specific and dynamic nature of reputation.

Mui et al. [38-40] defined reputation as "a perception that an agent creates through past actions about its intentions and norms". In this definition, perception can be thought of as an expectation that an entity has about another entity on the basis of its past experiences. This definition resembles the definition of Adul Rahman et al. but it fails to consider context and time as important factors for reputation.

Misztal et al. [53] stated that "reputation helps to manage the complexity of social life by singling out trustworthy people in whose interest it is to meet promises". This definition is from a social science perspective in which reputation is a way of selecting trustworthy partners who fulfil their promises. However, this definition fails to address the context specific and dynamic nature of reputation.

Grishchenko et al. [54-55] defined reputation as "an expectation about compliance of an expected event to be near to an average compliance level of a set of past events". The average level of compliance means averaging the degree of compliance of all past events. This definition takes the average of the degree of all past events without considering the context of the transaction. This definition also does not consider the dynamic nature of reputation.

Wang et al. [41-42] described reputation as “a peer’s belief in another peer’s capabilities, honesty and reliability based on recommendations received from other peers”. Their definition of reputation is consistent with their definition of trust. In their definition of reputation, reputation is calculated on the basis of the recommendations received by peers. But Wang et al. also do not consider the context specific and dynamic nature of reputation.

Kerps et al. [56] defined reputation as “a characteristic or attribute ascribed to one person by another person (or community)”. By this definition of the reputation of an entity is a characteristic that is attributed to that entity by others. This definition is from a social sciences' point of view. However, Kerps et al. fail to consider the context specific and dynamic nature of reputation.

2.3.3 Computational Approaches for Trust and Reputation

There are different computational approaches for trust and reputation calculations. These approaches are used to synthesize the trustworthiness value from the recommendation agents’ opinions. These approaches include the deterministic approach, probability-based approach, and fuzzy logic-based approach. Each of these approaches has an application and benefits for trust and reputation calculation.

The deterministic approach has an application in the e-Business environment. It is based on average-based aggregation methodologies which capably allow evaluating factors affecting trust or reputation on the basis of third-party opinion or on the basis of past interaction with service provider. However, with the deterministic approach, it is difficult to model the dynamic behaviour (changes in behaviour) of a service provider in a short period of time.

In the probability-based approach, Bayesian inference or Dampster-Shafer theory is used to evaluate the trust and reputation of a service provider. But to obtain the best result from these approaches, large datasets are needed which is not always possible and may be time consuming. Moreover, using probability theory, a service provider can be classified only as either trustworthy or not, which is not a rational view. Alternatively, a socio-cognitive approach takes into account the social environment of an agent. Through the use of a cognitive approach, it is possible to model the

subjective beliefs of a trusted agent by considering the willingness, capability and credibility of trusted agent. However, it is difficult to convert a socio-cognitive-based trust or reputation model to a computational model because it is not possible to quantify and aggregate subjective beliefs.

A fuzzy logic-based approach, however, enables us not only to model trust and reputation concepts considering their fuzzy and dynamic nature, it can also quantify and aggregate these values. In other words, it is a way to model the human behaviour of approximation and decision making. Therefore, fuzzy logic is a perfect choice for our trust and reputation evaluation model.

In the following sections, the existing literature for trust and reputation computation technologies is examined using deterministic approach, probabilistic approach, and fuzzy logic-based approach.

2.3.3.1 Deterministic Approach

In this section, we discuss different deterministic approaches. Marsh [57], [58] investigated trust in computing science. According to him, trust can take three forms; *basic trust*, *general trust* and *situational trust*. Marsh uses a value range of [-1, +1) to represent the trustworthiness of an agent in his calculations. Further, he divides this value range into 12 sub-ranges with semantic labels for each range (-1="*complete distrust*"; <-0.9="*very high distrust*", …, +1="*blind trust*"). Marsh considers the time factor in his model by introducing a temporal window for which information is collected. He also considers the context of transaction for trust calculations. To calculate the overall trust value for the selection of an interaction partner, Marsh proposes three statistical methods depending on the nature of the agent. A *pessimistic agent* uses the minimum trust value from its interaction record, while a *realistic agent* uses the mean trust value and an *optimistic agent* takes the maximum trust into account.

Marsh's trust assessment methodology is simple and transparent and it considers the context and time dependent nature of trust. However, his notion of *basic trust* also depends on the environment of the agent, which could be interpreted as a notion that includes social information about domains.

Abdul Rahman et al. [49-52] proposed a distributed trust model, which is based on Marsh's initial model, to support the decision-making process of agents. In this model, an agent can calculate the trust-based recommendations provided by peer agents. This model also provides a mechanism through which agents maintain records of direct interactions and experiences to evaluate the quality of opinions provided by recommender agents and adjust them if needed. If agent A is interacting with agent B in a given context c, agent A maintains a 4-tuple {"*very trustworthy*", "*trustworthy*", "*untrustworthy*", "*very untrustworthy*"} through direct experiences. For example, a tuple of {1, 3, 1, 2} would result in a direct trust level of "*trustworthy*" as this is the largest element in the tuple. A final trustworthiness value for agent B can be reached by aggregating the weighted opinions (reputation of opinions) provided by the recommender agents. This weight is calculated by the semantic distance between the agent's own direct experience and the opinion provided by the recommender agent and adjustments are made if necessary to the weighted opinions.

Although Abdul Rahman et al. used recommender agents' opinions and calculated trust and credibility, they failed to address the dynamic nature of trust, such as the decay of trust over time.

Golbeck et al. [59-64] proposed a methodology that allows the exchange of trust and distrust information about known users of the Friend-of-a-Friend (FOAF) ontology. They proposed a model through which trust and reputation measures can be integrated into semantically enriched social networks such as FOAF on the Internet. For a trust measure, they introduced nine levels {1="Distrusts absolutely";…;9= "Trusts absolutely"}. The final trust value is reached by a weighted average. The trust value is represented by a directed, weighted graph, where an edge between two agents, (say) A and B, is annotated by a value which indicates the trust level. For the weighted average, the distance between two agents on the graph specifies the weight of the opinion which is then rounded to its nearest integer representation of either 0 or 1 which indicates distrust or trust.

Although the model proposed by Golbeck et al. takes the context of a trust relationship into account, they do not consider the dynamic nature of trust (decay of trust over time).

The trust management model proposed by Aberer et al. [65-67] is specifically designed to work in a peer-to-peer (P2P) environment called *P-Grid* [68]. To support the decision-making process, Aberer et al. used P2P-specific integration and the management of trust and reputation measures. They derive reputation information from a third party for trust assessment. *P-Grid* distributed data about interacting parties which is collected in the form of complaints to database stores. For the trustworthiness assessment of a peer, Aberer et al. propose to query the *P-Grid* database to obtain the rounded values of -1 and +1 which indicates complaints and positive reports, respectively. All information is then aggregated and rounded again to determine if an agent is dishonest or honest.

Although the trust evaluation approach proposed by Aberer et al. provides the desired scalability for P2P environments through *P-Grid* in terms of data management, they do not consider the context and timeslot of the interaction when storing interaction records such as complaints. Information obtained from the system ,therefore, lacks contextual information and predictions about the future behaviour of a reputation-queried peer cannot be made. Moreover, without time stamps, the information about the reputation-queried peer may be very old and a correct judgement about their trustworthiness cannot be made.

Xiong and Liu [69-71] proposed a trust evaluation model for P2P-based eCommerce environments called PeerTrust. There are five factors in their model to calculate trust. Peer feedback is the first factor which is the reputation opinions provided by witness agents. Feedback scope is the second factor that is calculated by the number of interactions that the requesting agent had and the number of witness opinions it has received. Credibility of the opinions of the witness agent is the third factor. Transaction context is the fourth factor. Community context is the fifth factor which provides a measure of the social environment of the agent. Overall, trustworthiness is calculated by aggregating the first four factors and then multiplying it with a

normalised weighted factor. The rating of the transaction is given in a binary factor where 0 denotes failure and 1 denotes the success of the transaction.

In Xiong and Liu's model, reputation values are stored in Distributed Hash Hables (DHT) which represent aggregated global reputation values. But the problem with such a global view of reputation is that it does not allow an agent to calculate the reputation as a function of time. Moreover, global reputation values do not represent the context-specific behaviour of the target agent.

2.3.3.2 Probability-based Approach:

This section covers the probability based approaches for modeling trust and reputation. The two sub-categories of this approach namely, Bayesian inference approach and Dempster-Shafer theory are discussed.

a) Bayesian Inference Approach

Mui et al. [72], [39-40] proposed statistical methods to calculate a reliable estimate of the future behavior of a target agent. The trustworthiness of a target agent is calculated by assessing the probability that a satisfactory outcome is going to occur. For such a calculation, statistical samples are needed. Mui et al. uses an estimator to calculate the probability that the expected event will occur. To address the problem of calculating the level of confidence and the error of the estimator, they used the Beta probability density function [73]. The expected trustworthiness is then calculated by combining the trustworthiness values given by the estimator, based on the commonly used Beta distribution.

Although the model proposed by Mui et al. determines trustworthiness using commonly used statistical methods, it has a number of shortcomings. Trust should be expressed as a set of qualitative terms accompanied by their semantics, not as a binary concept, as discussed by Mui et al. Mui et al. does not take into account the valuable information from opinions provided by third parties, rather they used direct interactions for the trustworthiness evaluation of the target agent. Also, the binary view of trust, as implied by Bayesian theory, is against human thinking.

Wang and Vassileva [41-42], [74-76] proposed a trust assessment model for file exchange in P2P environments. For trust evaluation, they use three criteria. These criteria are trust in the agent's capability to provide a fast download, trust in the agent's capability to provide good quality files and trust in the agent's capability to deliver the file in the expected format. For the evaluation, they use the naïve Bayesian network. The assessing agent can make an informed decision on whether it wants to interact with the target agent or not on the basis of the trust values of the different criteria studied. The assessing agent uses direct experiences to evaluate the trust of the target agent. However, if direct experiences are insufficient, the assessing agent requests opinions from peers. The weighted average is then used to aggregate the recommending opinions, where trustworthy agents are assigned a higher weight than unknown agents. The review process is carried out after the completion of a transaction in which the agents review each criteria with respect to their own expectations. The Bayesian network is then updated on the basis of the review results.

Although Wang and Vassileva proposed a trust-based decision support model which is context aware, they do not consider the decay of trust over time in their model. Furthermore, the binary view of trust does not correspond with the human view of trust and reputation. Trust and reputation assessments should be assessed using more qualitative and quantitative degrees of such fuzzy concepts.

b) Dempster-Shafer Approach

Yu and Singh [77-79] proposed a reputation assessment model which is based on the direct interaction of the assessing agent with the target agent. In case of the absence of direct interaction, witness information is taken into account. They proposed referral chains, called *TrustNet,* which are used to systematically incorporate witness evidences. *TrustNet* is a network where, if a queried agent cannot provide information due a lack of interaction with the agent in question, they refer to other witnesses who can provide information about their direct interactions with the agent in question.

The mathematical model of Yu and Singh is based on two independent hypothesises; that a target agent is trustworthy *T* (expressed as belief function $m(\{T\})$ and that the same agent is untrustworthy ($\neg T$) (expressed as belief function $m(\{\neg T\})$). They also introduce a third hypothesis $\Theta = \{T, \neg T\}$ (expressed as belief function $m(\{T, \neg T\})$) which represents all evidential information that does not support *T* or $\neg T$. The user of the system needs to define two thresholds as boundaries for the grouping of evidential information ω and Ω so that $0 \leq \omega \leq \Omega \leq 1$ so that

$m(\{T\}) = \sum_{x_k=\Omega}^{1} f(x_k)$; $m(\{\neg T\}) = \sum_{0}^{x_k=\omega} f(x_k)$; and $m(\{T, \neg T\}) = \sum_{x_k=\omega}^{x_k=\Omega} f(x_k)$, where $m(\{T\}) + m(\{\neg T\}) + m(\{T, \neg T\}) = 1$.

The function $f(x_k)$ denotes the probability that the outcome of an interaction is of quality x_k, where $x_k \in \{0.0, \ldots 1.0\}$. To calculate overall trustworthiness, Yu and Singh use Dempster-Shafer theory which allows the combination of evidence while separating the information sources. An agent will be trustworthy if $m(\{T\}) - m(\{\neg T\}) \leq \rho$ where ρ is the user-defined trustworthiness threshold of the agent.

Apart from the advantages of Yu and Singh's model, it also has some drawbacks. One of the major drawbacks is that they strongly differentiate between direct and witness information. Witness information is not considered if there is evidence from direct past interactions between the two partners. If the assessing agent has old evidence from a direct interaction, the trustworthiness calculation of the target agent should consider the witness' information which may contain more recent information. Furthermore, they do not consider the decay of the trust value over time.

After discussing probabilistic approaches, in next section the fuzzy logic-based approach for trust/reputation calculation is examined.

2.3.3.3 Fuzzy Logic-based Approach

The first approach that is discussed in this section is proposed by Castelfranchi et al. [80-82] which is based on Fuzzy Cognitive Maps (FCM) [83]. Using FCMs, they create tree structures to derive fuzzy rules. The sample tree consists of a trustfulness value as a root element. *Internal* and *external* factors are child elements. *Internal* factors have further child elements *ability*, *availability* and *harmfulness* and *external*

factors have child elements *opportunity* and *danger*. They use the universe of discourse [-1,1] to model credibility and therefore the weighting of belief sources in the fuzzy model. But they do not show the specific results of their work.

The model proposed by Castelfranchi et al. is a major improvement over the model proposed by Faclone et al. [84-85]. Fuzzy logic makes it possible to express imprecise inputs such as the belief discussed by Castelfranchi et al. However, they fail to indicate that how the necessary information is collected and how the review results are integrated after the interactions.

Sabater and Sierra [48], [86], [45] proposed a reputation-based decision support model called *ReGreT*. In their model, reputation is represented as dimensions. For example, direct trust is influenced by individual dimensions. In the absence of direct trust, reputation is determined by the social dimension. For the social dimension, Sabater and Sierra recognize three contributing factors: *witness reputation*, *neighbourhood reputation* and *system reputation*.

In the case of information from witness agents, their credibility needs to be evaluated for which Sabater and Sierra use fuzzy rules [46-47]. Fuzzy rule provides a better way to encode information such as low reputation [87]. The final dimension in Sabater and Sierra's model is the ontological dimension which allows the combination of reputation values related to different contexts. For the review approach, they compare the vector of actual attributes with the vector of expected attributes. If both vectors have a low matching value, a reputation value of -1 is assigned and if both vectors match to a high degree, a reputation value of +1 is assigned. On the basis of the review, the results model is adjusted to increase the accuracy of the reputation assessment.

Sabater and Sierra overcame the issues of the context dependent and time dependent nature of trust. However, in their model, they use reputation and trust concepts interchangeably without providing a clear distinction between these two. In their model, credibility represents trustworthiness and the value of opinions provided by third parties. Furthermore, the review results are represented as a binary view (-1 or +1) which does not reflect the human perception of the reputation concept. Instead of

using flexible and user-friendly fuzzy rules, Sabater and Sierra use comlex mathematical functions which are difficult to comprehend.

Ramchurn et al. [88-90] proposed a computational model for trust that helps software agents to make decisions about interaction partner selection. According to them, trust is a complex issue which is influenced by factors such as context, reputation, direct trust, risk and confidence. They consider the modeling of the confidence value as a very important factor which represents a measure of certainty for an agent's performance over time. They represented the confidence as a fuzzy concept due to the imprecise and fuzzy nature of the concept. The reputation model of Ramchuran et al. is based on the *REGRET* model introduced by Sabater and Sierra [48], [86], [45].

Song et al. [91-93] proposed a trust-based security model to complement the current public key infrastructure (PKI)-based security approaches for computational grids. They introduce a trust index which is based on the defense capability of a remote site, as well as the reputation of the remote site within the grid to overcome the weakness of the PKI security model. The defense capabilities include secure job allocation capabilities, firewall capabilities, intrusion detection capabilities and antivirus capabilities. The reputation assessment of the resource site is based on the evaluation of the historic performance. For performance determination, the contributing factors are job turnaround times, job execution success rates, overall site utilization and job slowdown ratios. These factors are aggregated using fuzzy logic instead of using the deterministic approach. Song et al. argued that evaluating defense capability and performance factors using fuzzy logic better represents evaluation attributes because of their inherent fuzziness and uncertainty.

Song et al. do not consider the context dependent nature of trust. Moreover, they do not model decay of trust and reputation over time.

2.3.3.4 Issues with existing trust and reputation assessment approaches for SLA management in Cloud Computing

As discussed in Chapter 1, trust has been used to assess the performance of the service provider. In this section, the existing computational approaches for trust and reputation were discussed and their main short comings in the application for service

management in cloud computing were highlighted, particularly the lack of a trust and reputation approach for selecting a trustworthy service provider in the pre-interaction time phase and the monitoring of the service provider's performance in the post-interaction time phase. These short comings have to be addressed in order to have a trust or reputation determination approach that can be used for the selection of a service provider in the pre-interaction time phase in a cloud environment. In the next section, the existing literature on risk management in cloud computing is discussed.

2.4 Discussion of Risk Management for SLA Management in Cloud Computing

As discussed in the previous chapter, risk management in cloud computing is a way to ensure the financial outcome of the consumer in the post-interaction time phase of the interaction. In this section, the work for the risk assessment approaches, in particular, approaches for the sub-categories of *performance risk* and *financial risk* in a business interaction is analysed. The risk-based decision making approaches to assist informed decision making in the interaction are also analysed. For this, the criteria to evaluate the literature work include the explicit representation of risk (performance risk and financial risk) not the risk associated with security or privacy, the representation of the level of risk along with the magnitude of the risk and the time of the transaction (current or future timeslot of the interaction). This section is divided into two sub-sections, Section 2.4.1 discusses the risk assessment approaches in business interactions and Section 2.4.2 discusses the risk-based decision making approaches.

2.4.1 Risk Assessment Approaches

Different approaches have been proposed by several researchers to identify, measure and analyse the financial-based risk in an interaction to measure the overall interaction success factors. A few such approaches are discussed in this section and their impact on decision making is highlighted.

Lam et al. [94-95] proposed an approach in which agents participate in the interaction. The interaction represents a business transaction in which the buying agent selects an agent to buy a requested product “p” based on its maximum amount. While their approach utilizes risk to maximise effectiveness, the trust and risk

associated in the interaction is considered. The buying agent computes the net value of the transaction as the difference between the proposed maximum pay amount and the individual price offered by each seller. The utility function of each seller is computed by determining the impact of their risk attitude on the net value in the interaction. This helps in deciding the appropriate seller for product "p". While their approach is considered to be an effective one, it has the several following drawbacks: in this approach, the authors do not propose a method to measure the exact value of the maximum pay amount at risk in the transaction; hence its net value and utility function cannot be calculated.

In other work, Josang et al. [96-97] adopted a different approach of specifying the range values to determine the risk in the interaction. The identified risk is perceived to be high if the value involved in the transaction is very high. Additionally, they linked the high transaction value to be the "stake" of the transaction. The risk in an transaction is specified to be in the range of [-1,0] and the "gain factor" variable is enumerated with a 'successful' or 'failed' state along with "expected gain" which represents the final result of the transaction. The authors defined "decision trust' in the range of [-1,1] based on reliability trust. If the value is between [-1, 0] then, according to the authors, the trusting agent does not reply to the other agent in the transaction.

One of the limitations in this work by Josang et al. is that they represented 'decision trust' in the continuous range of [-1, 1] which means there are an unlimited number of possible levels in this range, hence, it becomes challenging if not impossible to associate an interpretation to each possible interval. Additionally, they did not present any method or approach for obtaining the financial value at stake for a given agent.

In the peer-to-peer environment, Wang and Lin [98-99] proposed trust negotiation before the transaction at the pre-interaction stage. In their work, the requesting peer determines the possible risk in the transaction by using a level of trust with the target peer before initiating the interaction. They divided the transaction amount into different categories and obtained the variances between the previous and new transaction to determine the impact factor. Then, the impact factor is compared with

the trust value of the previous interaction to determine the new one. Noticeably, the authors determined the risk in the transaction by using it to complement and consider the risk and trust in the transaction in a continuous range of [0, 1]. In addition to focusing on peer-to-peer architecture, a few limitations in their work are observable. The authors considered the temporal characteristics of time from the past trust value, while for the new trust value, they did not consider context specific characteristics. Likewise, the temporal characteristics of time are ignored while determining the impact factor. Their proposed approach does not provide any insight to determine the behaviour of the peer in an interaction in future.

Salam et al. [100-101] analysed the situation in which the user interacts with the online service spontaneously, based on the trust level. They identified factors such as web presence, user intuition and external variables to determine the trust between the vendor and the potential user/buyer in an e-commerce setting. While the evaluation is based on an empirical experiment, their work has certain limitations. For example, the authors did not present any formal mechanism to quantify the degree and level of trust while the consumer is interacting with the website using the above mentioned factors. Their approach is not tailored to support the determination of trust in future transactions and they do not offer any mechanism by which a user can compare different vendors, based on trust values.

English et al. presented a solution for trust-based collaboration in a global environment [102-103]. Their approach considered trust and risk as the core components for enabling trust-based collaboration. They defined risk as the *uncertainty and the cost of benefit of the outcome in the interaction* and linked trust with security. The classification of an agent is based on the risk profile of each agent which reflects the observed behaviour. They highlighted the importance of risk and trust in decision making and calculated the level of trust according to behaviour. The limitations of their work include the absence of an approach to determine the level of trust and risk in the transaction. Also, they didn't explain how risk exposure is reduced in transactions. They did not define how to obtain a possible risk profile in the transaction based on the trust value. Similar to Josang et. al. [96-97], they did not mention how to determine the level of trust and risk in the transaction and what are the semantics for these ranges.

Hussain [104] proposed that transactional risk should be assessed by sub-categories of risk namely, *performance risk* and *financial risk*. Performance risk represents the level of failure in an interaction due to the occurrence of undesired outcomes; whereas, financial risk represents the financial consequences that could be experienced as a result of the failure of the interaction. He proposed transactional risk assessment methodology that consider the semantics associated with the concept of risk as well as the quantitative representation of risk. To evaluate the perceived risk before the start of an interaction, a risk assessing agent uses the past trust values of the risk assessed agent. These trust values can be obtained either by the direct experience of the risk assessing agent with the risk assessed agent or by asking for trust values from third party agents. The author emphasized that to evaluate transactional risk, both sub-categories should be taken into account to determine the financial outcome of a service user because this is the most important outcome in a business interaction.

Other work done by Hussain et al. [105] uses transactional risk to determine the financial cost based on dependable and non-dependable criteria.

2.4.2 Risk-based Decision Making Approaches

In previous section, it has been discussed that existing literature does not propose a risk management approach through which a consumer assess the transactional risk in the interaction and based on that, make an informed decision about the continuity of the service with the service provider. Based on the assessed risk, some of the approaches for risk management have been discussed in literature, for example Xu et al. [106], Lin and Varadharajan [107], Schläger [108-109] etc. but these approaches lack of a method by which a consumer makes an informed decision for the continuation for the service with the service provider.

In this section, two noticeable approaches for decision making are discussed in Service-Oriented and cloud computing environments.

Martens et al. [110] proposed a risk-based decision support system in the cloud computing environment. In particular, their work defines risk as the product of a negative outcome and importance of loss caused by the occurrence of this outcome.

For risk assessment, they consider the following factors: service failure and its probability, loss due to service failure and the risk attitude of the decision maker. Loss is expressed as loss in sales per output quantity. The type of risk they consider is security risk divided into three categories: confidentiality, integrity and availability. Total risk in the interaction is then calculated as the combination of all three categories of risk.

Although the work done by Martens et al. is in the domain of cloud computing, they don't consider the performance risk independent of security risk. Moreover, the risk determined by them is not the combination of sub-categories of risk, such as performance risk and financial risk. Although they consider the risk attitude of the consumer for decision making, they don't consider the level of risk to determine the impact of risk attitude on the level of risk of the consumer.

Hussain [104] considers the risk-based decision making approach which not only considers the risk attitude of the consumer as an important factor, but also considers the level of perceived risk before the start of the interaction. It is quite possible in the business interaction that levels of perceived risk are associated with magnitude of risk. A risk assessing agent must consider these levels while making an informed decision. The outputs of this decision support system are the recommendations for the service consumer to proceed in the interaction or not to proceed.

Although the work done by Hussain et al. is significant, their work applies to the determination of transactional risk in a peer-to-peer based service-oriented environment which is different from determining transactional risk in open standard-based cloud computing. Moreover, risk assessment in their approach is based on historical data whereas, as discussed in Section 2.2, risk assessment in the cloud needs near-to-real-time data.

2.4.3 Issues with existing risk assessment approaches for SLA Management in Cloud Computing

As discussed in Chapter 1, risk has been used to assess the performance of a service provider. In this section, the existing approaches for risk assessment have been discussed. It is concluded that no such risk assessment approach which considers

both performance risk and performance risk for the assessment of a service provider in a cloud environment exists in the literature. In the next section, a critical evaluation of the existing approaches is discussed.

2.5 Critical Evaluation of Existing Approaches for SLA Management in Cloud Computing: An Integrative View

In this section, the issues that were identified in the critical evaluation of the related research work in the preceding sections of this chapter are summarised. This critical evaluation serves the purpose of outlining the research issues that need to be addressed to develop an SLA management framework that evaluates the performance of a service provider before and after the interaction to ascertain the financial outcome of a consumer. In the previous sections, we reviewed the SLA management techniques, the role of trust and risk in SLA evaluation and trust and risk-based computational models.

The major shortcomings in literature review related to SLA management in cloud computing has been identified. The issues related to trust and risk while assessing SLA has also been identified. The detail shortcomings identified in literature review are as follows:

- ***Lack of SLA management from consumer's point of view:*** It can be seen from the discussion in section 2.2 that most of the SLA management techniques are proposed for service providers to monitor their performance. But there is a need for an SLA management technique which provides more control to the consumer over the performance of a service provider and which enables them to make informed decision about the continuity of the service. More control refers to an approach that assists the consumer in making informed decisions for the management of their service with the service provider. As discussed in Section 2.2, there is a lack of a proper framework for SLA management in the cloud. To address this issue, an SLA management framework in cloud computing should be introduced which takes into account SLA formation and the selection of a service provider on the basis of their past performance before the start of the service and monitoring of the performance of the service provider after the start of an

interaction. As highlighted in the previous paragraph, this could be done from a consumer's point of view.

- ***Suitable trust/reputation determination approach to select service provider in pre-interaction time phase:*** Reputation determination approaches were discussed in section 2.3 for service provider selection and/or evaluation. It can be concluded from this discussion that the Bayesian approach does not allow fine gradations in satisfaction, since satisfaction/non-satisfaction of an Agent A with Agent B is modelled only using [0,1] approach. Although the deterministic approach models the the reputation system well, the weights which apply to input parameters are based on human judgement which needs human intervention. Moreover, weight functions may vary (adjustable parameters) according to the domain/context of the system. Although the fuzzy-based reputation systems discussed above suit the fuzzy nature of our input variables, the adjustment of input parameters is based on a heuristic approach which is time consuming. By using a tuning algorithm with fuzzy logic, we can optimize input and output parameters much better, which results in a self-adaptive system.
- ***Suitable assessment approach to determine risk in post-interaction time phase:*** In Section 2.4, transactional risk assessment approaches are discussed. It has been discussed that most of the researchers consider the concept of risk inferior than concept of security or trust. Moreover, only few of these approaches attempt to consider *performance risk* and *financial risk* while making decision. But to the best of my knowledge, none of these approaches consider assessing performance and financial risk in an open standard-based system such as cloud computing.
- ***Lack of Decision Support System for SLA management in cloud computing:*** As discussed in Section 2.4.2, in open system-based environment, the services needs to be evaluated continuously on regular interval. There is a need of decision support system that is based on regular (continuous) SLA evaluation of the service in post-interaction timeslot and for decision making it can take advantage of the risk-based decision support methodology introduced by Hussain [104].

In summary, to the best of my knowledge, no notable research work has been published on a SLA management framework that takes a service provider's reputation into account before the start of an interaction and then monitors the service provider's performance after the interaction to ensure financial outcome of the consumer.

2.6 Conclusion

In this chapter, the issues related to SLA management in cloud computing with respect to achieving desired outcomes of the consumer have been discussed. Trust and reputation evaluation techniques for the service selection before the start of an interaction and risk assessment techniques for evaluation of service provider's performance have been studied. Based on identified problems in the literature, in the next chapter problem is formalized with the issues that will be addressed in this thesis.

2.7 References

[1] S. Bose, A. Pasala, D. Ramanujam, S. Murthy, and G. Malaiyandisamy, "SLA management in Cloud Computing: A service provider's perspective", *Cloud Computing: Principles and paradigms, Rajkumar Buyya, James Broberg, and Andrzej Goscinski, Eds. New Jersey, USA: John Wiley & Sons,* pp. 413-436, 2011.

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", *Future Gener. Comput. Syst.,* vol. 25, pp. 599-616, 2009.

[3] B. Furht and A. Escalante, *Handbook of Cloud Computing*, Springer Publishing Company, Incorporated, 2010.

[4] T. Dillon, C. Wu, and E. Chang, *Cloud Computing: Issues and Challenges, 24th IEEE International Conference on Advanced Information Networking and Applications*, April 20 - 23, 2010, pp. 27-33.

[5] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges", *Journal of Internet Services and Applications,* vol. 1, pp. 7-18, 2010/05/01.

[6] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, *Cost of virtual machine live migration in clouds: A performance evaluation,* 1st International Conference on Cloud Computing, 2009, pp.254 - 265.

[7] W. Voorsluys, J. Broberg, and R. Buyya, "Introduction to Cloud Computing", in *Cloud Computing*, ed: John Wiley & Sons, Inc., 2011, pp. 1-41.

[8] Rajkumar Buyya, Suraj Pandey, and Christian Vecchiola, "Market-Oriented Cloud Computing and the Cloudbus Toolkit," in Cloud Computing. vol. 5931, M. Jaatun, G. Zhao, and C. Rong, Eds., ed: Springer Berlin Heidelberg, 2009, pp. 24-44.

[9] Q. He, J. Yan, R. Kowalczyk, H. Jin, and Y. Yang, "Lifetime service level agreement management with autonomous agents for services provision", *Information Sciences,* vol. 179, pp. 2591-2605, 2009.

[10] H. Foster and G. Spanoudakis, "Advanced service monitoring configurations with SLA decomposition and selection", presented at the Proceedings of the 2011 ACM Symposium on Applied Computing, TaiChung, Taiwan, 2011, pp. 1582 - 1589.

[11] D. Bianculli and C. Ghezzi, "Monitoring conversational web services", presented at the 2nd international workshop on Service oriented software engineering: in conjunction with the 6th ESEC/FSE joint meeting, Dubrovnik, Croatia, 2007, pp. 15 - 21.

[12] L. Baresi, D. Bianculli, C. Ghezzi, S. Guinea, and P. Spoletini, "Validation of web service compositions", *Software, IET,* vol. 1, pp. 219-232, 2007.

[13] K. M. Spanoudakis G., "Non-intrusive monitoring of service-based systems", vol. 15, pp. 325-358, 2006.

[14] W. M. P. v. d. Aalst, M. Dumas, C. Ouyang, A. Rozinat, and E. Verbeek, "Conformance checking of service behavior", *ACM Trans. Internet Technol.,* vol. 8, pp. 1-30, 2008.

[15] K. Park, J. Han, and J. Chung, "THEMIS: A Mutually Verifiable Billing System for the Cloud Computing Environment", *IEEE Transactions on Services Computing,* vol. 6, Issue 3, 2013, pp. 300 - 313.

[16] C. Smith and A. Moorsel, "Mitigating Provider Uncertainty in Service Provision Contracts Economic Models and Algorithms for Distributed Systems", D. Neumann, M. Baker, J. Altmann, and O. Rana, Eds., ed: Birkhäuser Basel, 2010, pp. 143-159.

[17] T. B. Quillinan, K. P. Clark, M. Warnier, F. M. T. Brazier, and O. Rana, "Negotiation and Monitoring of Service Level Agreements", in *Grids and Service-Oriented Architectures for Service Level Agreements*, P. Wieder, R. Yahyapour, and W. Ziegler, Eds., ed: Springer US, 2010, pp. 167-176.

[18] S. Betgé-Brezetz, O. Martinot, G. Delègue, and E. Marilly, "Pro-Active SLA Assurance for Next Generation Network", *WTC Paris, France,* 2002.

[19] J. Morin, J. Aubert, and B. Gateau, "Towards Cloud Computing SLA Risk Management: Issues and Challenges", in *System Science (HICSS), 45th Hawaii International Conference on*, 2012, pp. 5509-5514.

[20] Z. Zheng, X. Wu, Y. Zhang, M. Lyu, and J. Wang, "QoS Ranking Prediction for Cloud Services", IEEE Transactions on Parallel and Distributed Systems, Vol. 24, Issue 6, 2012, pp. 1213 - 1222.

[21] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Sandpiper: Black-box and gray-box resource management for virtual machines", *Comput. Netw.*, vol. 53, pp. 2923-2938, 2009.

[22] V. C. Emeakaroha, I. Brandic, M. Maurer, and S. Dustdar, "Low level Metrics to High level SLAs - LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments", in *High Performance Computing and Simulation (HPCS), 2010 International Conference on*, 2010, pp. 48-54.

[23] Vincent C. Emeakaroha, Rodrigo N. Calheiros, Marco A. S. Netto, Ivona Brandic, and Cesar A. F. De Rose, "DeSVi: An architecture for detecting SLA violations in cloud computing infrastructures", presented at the 2nd International ICST Conference on Cloud Computing (CloudComp 2010), Barcelona, Spain, 2010.

[24] V. Cardellini, E. Casalicchio, F. L. Presti, and L. Silvestri, "SLA-aware Resource Management for Application Service Providers in the Cloud", presented at the Proceedings of the 2011 First International Symposium on Network Cloud Computing and Applications, 2011, pp. 20 - 27.

[25] H. Jun-Yan, W. Chun-Hung, C. Chia-Chen, L. Kuan-Hsiung, Y. Hey-Chyi, H. Yung-Yi, H. Chung-Hua, and L. Huan-Guo, "Constructing a Cloud-Centric Service Assurance Platform for Computing as a Service", in *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2011 International Conference on*, 2011, pp. 139-145.

[26] Daniel D. and S. Jeno Lovesum, "A novel approach for scheduling service request in cloud with trust monitor", in *International Conference on Signal Processing Communication, Computing and Networking Technologies*, July 21 - 22, 2011, pp. 509 - 513.

[27] Bruno Ciciani, Diego Didona, Pierangelo Di Sanzo, Roberto Palmieri, Sebastiano Peluso, Francesco Quaglia, Paolo Romano, "Automated Workload Characterization in Cloud-based Transactional Data Grids", presented at the 17th IEEE Workshop on Dependable Parallel, Distributed and Network-Centric Systems, Shanghai, China, 2012, pp. 1525 - 1533.

[28] E. Chang, F. Hussain, and T. Dillon, *Trust and Reputation for Service-Oriented Environments: Technologies For Building Business Intelligence And Consumer Confidence*: John Wiley & Sons, 2005.

[29] M. Deutsch, *Distributive justice: A social psychological perspective*: Yale University Press New Haven, USA, 1985.

[30] M. Deutsch, *The resolution of conflict: Constructive and destructive processes*, Yale University Press, 1977.

[31] P. Sztompka, *Trust : a sociological theory*. Cambridge [u.a.]: Cambridge Univ. Press, 2006.

[32] N. Luhmann, "Familliarity, Confidence, Trust: Problems and Alternatives", *Trust: Making and Breaking of Cooperative Relations, Basil Blackwell, New York,* pp. 94-107, 1988.

[33] D. Gambetta, "Can we trust trust?", *Trust: Making and Breaking Cooperative Relations, Basil Blackwell, New York,* pp. 213-237.

[34] A. Josang, "The right type of trust for distributed systems", presented at the Proceedings of the 1996 workshop on New security paradigms, Lake Arrowhead, California, United States, 1996, pp. 119 - 131.

[35] P. Dasgupta, "Trust as Commodity", *Trust: Making and Breaking Cooperative Relations, Basil Blackwell, New York,* pp. 49-72, 1988.

[36] J. Golbeck, Ed., *Computing with Social Trust*. Springer London, 2009.

[37] K. Buist. (July 17, 2012). *Definition of Trust*. Available: http://www.teamtechnology.co.uk/trustworthiness/definition-of-trust.html

[38] L. Mui, M. Mohtashemi, and A. Halberstadt, "Notions of reputation in multi-agents systems: a review", presented at the Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1, Bologna, Italy, 2002, pp. 280 - 287.

[39] L. Mui, M. Mohtashemi, and A. Halberstadt, "A Computational Model of Trust and Reputation for E-businesses", presented at the Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 7, 2002.

[40] L. Mui, A. Halberstadt, and M. Mohtashemi, "Evaluating reputation in multi-agents systems", presented at the Proceedings of the 2002 international conference on Trust, reputation, and security: theories and practice, Bologna, Italy, 2003, pp. 123 - 137.

[41] Y. Wang and J. Vassileva, "Trust and Reputation Model in Peer-to-Peer Networks", presented at the Proceedings of the 3rd International Conference on Peer-to-Peer Computing, 2003, pp. 150 - 157.

[42] Y. Wang and J. Vassileva, "Bayesian Network-Based Trust Model", presented at the Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence, 2003, pp. 372 - 378.

[43] N. Mezzetti, "Towards a Model for Trust Relationships in Virtual Enterprises", presented at the Proceedings of the 14th International Workshop on Database and Expert Systems Applications, 2003, pp. 420 - 424.

[44] N. Mezzetti, "A Socially Inspired Reputation Model Public Key Infrastructure", vol. 3093, S. Katsikas, S. Gritzalis, and J. López, Eds., ed: Springer Berlin / Heidelberg, 2004, pp. 605-605.

[45] J. Sabater and C. Sierra, "REGRET: reputation in gregarious societies", presented at the Proceedings of the fifth international conference on Autonomous agents, Montreal, Quebec, Canada, 2001, pp. 194 - 195.

[46] J. Sabater and C. Sierra, "Reputation and social network analysis in multi-agent systems", presented at the Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1, Bologna, Italy, 2002, pp. 475 - 482.

[47] S. Jordi, "EVALUATING THE ReGreT SYSTEM", *Applied Artificial Intelligence,* vol. 18, pp. 797-813, 2004.

[48] J. Sabater, "Formalising Trust and Reputation for Agent Societies", PhD Thesis, Institut d' Investigaci´o en Intel-ligencia Artificial, Bellaterra, Catalonia, Spain, 1998.

[49] A. Abdul-Rahman and S. Hailes, "Supporting trust in virtual communities", *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, 2000.

[50] Alfarez Abdul-Rahman and Stephen Hailes, "Relying on Trust to Find Reliable Information", in *Proceedings of the International Symposium on Database, Web and Cooperative Systems (DANTE 1999)*, Baden-Baden, Germany, 1999.

[51] S. H. Alfarez Abdul-Rahman, "Using Recommendations for Managing Trust in Distributed Systems", in *Proceedings of the IEEE Malaysia International Conference on Communication (MICC 1997)*, Kuala Lumpur, 1997.

[52] A. Abdul-Rahman and S. Hailes, "A distributed trust model", presented at the Proceedings of the 1997 workshop on New security paradigms, Langdale, Cumbria, United Kingdom, 1997, pp. 48 - 60.

[53] B. A. Misztal, *Trust in Modern Societies: The Search for the Bases of Social Order*: Polity Press, USA, 1996.

[54] V. S. Grishchenko, "Redefining Web-of-Trust: reputation, recommendations, responsibility and trust among peers", in *Proceedings of the First Workshop on Friend of a Friend, Social Networking, and the Semantic Web*.

[55] V. S. Grishchenko, "A fuzzy model for context-dependent reputation", in *ISWC 2004 Workshop on Trust, Security, and Reputation on the Semantic Web*, Hiroshima, Japan, 2004.

[56] D. M. Kreps and R. Wilson, "Reputation and imperfect information", *Journal of Economic Theory*, vol. 27, pp. 253-279, 1982.

[57] S. Marsh, "Trust in distributed artificial intelligence Artificial Social Systems", vol. 830, C. Castelfranchi and E. Werner, Eds., ed: Springer Berlin / Heidelberg, 1994, pp. 94-112.

[58] S. P. Marsh, "Formalising Trust as a Computational Concept", PhD, Computing Science and Mathematics, University of Stirling, 1995.

[59] J. Golbeck, "Personalizing applications through integration of inferred trust values in semantic web-based social networks", presented at the Semantic Network Analysis Workshop at the 4th International Semantic Web Conference, Galway, Ireland, 2005.

[60] J. Golbeck, "Semantic Web Interaction through Trust Network Recommender Systems", presented at the End User Semantic Web Interaction Workshop at the 4th International Semantic Web Conference, 2005.

[61] J. Golbeck and J. Hendler, "Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-Based Social Networks Engineering Knowledge in the Age of the Semantic Web", vol. 3257, E. Motta, N. Shadbolt, A. Stutt, and N. Gibbins, Eds., ed: Springer Berlin / Heidelberg, 2004, pp. 116-131.

[62] J. Golbeck and A. Mannes, "Using Trust and Provenance for Content Filtering on the Semantic Web", in *Models of Trust for the Web Workshop*, May 22-26, 2006, Edinburgh, UK.

[63] J. Golbeck and B. Parsia, "Trusting Claims from Trusted Sources:Trust Network Based Filtering of Aggregated Claims".

[64] J. Golbeck and J. Hendler, "Inferring binary trust relationships in Web-based social networks", *ACM Trans. Internet Technol.,* vol. 6, pp. 497-529, 2006.

[65] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system", presented at the Proceedings of the tenth international conference on Information and knowledge management, Atlanta, Georgia, USA, 2001, pp. 310 -317.

[66] K. Aberer, P. Cudré-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, and R. Schmidt, "P-Grid: a self-organizing structured P2P system", *SIGMOD Rec.,* vol. 32, pp. 29-33, 2003.

[67] K. Aberer, P. Cudré-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, R. Schmidt, and J. Wu, "Advanced Peer-to-Peer Networking: The P-Grid System and its Applications", *PIK Journal-Praxis der Informationsverarbeitung und Kommunikation, Special Issue on P2P Systems,* 2007, Vol. 26, Issue 2.

[68] K. Aberer, "P-Grid: A Self-Organizing Access Structure for P2P Information Systems Cooperative Information Systems", vol. 2172, C. Batini, F. Giunchiglia, P. Giorgini, and M. Mecella, Eds., ed: Springer Berlin / Heidelberg, 2001, pp. 179-194.

[69] X. Li, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities", *IEEE Transactions on Knowledge and Data Engineering,* vol. 16, pp. 843-857, 2004.

[70] L. Xiong and L. Liu, "Building trust in decentralized peer-to-peer electronic communities", presented at the Fifth International Conference on Electronic Commerce Research (ICECR-5), 2002.

[71] L. Xiong and L. Liu, "A reputation-based trust model for peer-to-peer ecommerce communities", presented at the Proceedings of the 4th ACM conference on Electronic commerce, San Diego, CA, USA, 2003, pp. 275 - 284.

[72] L. Mui, M. Mohtashemi, C. Ang, P. Szolovits, and A. Halberstadt, "Ratings in Distributed Systems: A Bayesian Approach", in *Workshop on Information Technologies and Systems (WITS)*, 2001.

[73] D. Heckerman, "A Tutorial on Learning with Bayesian Networks Innovations in Bayesian Networks", vol. 156, D. Holmes and L. Jain, Eds., ed: Springer Berlin / Heidelberg, 2008, pp. 33-82.

[74] Y. Wang and J. Vassileva, "Trust and reputation model in peer-to-peer networks", in *Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings. Third International Conference on*, 2003, pp. 150-157.

[75] Y. Wang and J. Vassileva, "Bayesian network-based trust model in peer-to-peer networks", in *Workshop on Deception, Fraud and Trust in Agent Societies, AAMAS'03*, 2003, pp. 57–68.

[76] Y. Wang and J. Vassileva, "Trust-Based Community Formation in Peer-to-Peer File Sharing Networks", presented at the Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, 2004, pp. 341 - 348.

[77] B. Yu and M. P. Singh, "Distributed Reputation Management for Electronic Commerce", *Computational Intelligence,* vol. 18, pp. 535-549, 2002.

[78] B. Yu and M. P. Singh, "An evidential model of distributed reputation management", presented at the Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1, Bologna, Italy, 2002, pp. 294 - 301.

[79] B. Yu and M. P. Singh, "Detecting deception in reputation management", presented at the Proceedings of the second international joint conference on Autonomous agents and multiagent systems, Melbourne, Australia, 2003, pp. 73 - 80.

[80] C. Castelfranchi, R. Falcone, and G. Pezzulo, "Trust in information sources as a source for trust: a fuzzy approach", presented at the Proceedings of the second international joint conference on Autonomous agents and multiagent systems, Melbourne, Australia, 2003, pp. 89 - 96.

[81] C. Castelfranchi, R. Falcone, and G. Pezzulo, "Integrating trustfulness and decision using fuzzy cognitive maps", presented at the Proceedings of the 1st international conference on Trust management, Heraklion, Crete, Greece, 2003, pp. 195 - 210.

[82] R. Falcone, G. Pezzulo, and C. Castelfranchi, "A fuzzy approach to a belief-based trust computation", presented at the Proceedings of the 2002 international conference on Trust, reputation, and security: theories and practice, Bologna, Italy, 2003, pp. 73 - 86.

[83] B. Kosko, "Fuzzy cognitive maps", *International Journal of Man-Machine Studies,* vol. 24, pp. 65-75, 1986.

[84] C. Castelfranchi and R. Falcone, "The dynamics of trust: From beliefs to action", in *Second Workshop on Deception, Fraud and Trust in Agent Societies*, 1999, pp. 41–54.

[85] R. Falcone and C. Castelfranchi, "Social trust: a cognitive approach", in *Trust and deception in virtual societies*, ed: Kluwer Academic Publishers, 2001, pp. 55-90.

[86] J. Sabater and C. Sierra, "Social ReGreT, a reputation model based on social relations", *SIGecom Exch.,* vol. 3, pp. 44-56, 2001.

[87] D. Dubois and H. Prade, "What are fuzzy rules and how to use them", *Fuzzy Sets and Systems,* vol. 84, pp. 169-185, 1996.

[88] S. D. Ramchurn, N. R. Jennings, C. Sierra, and L. Godo, "DEVISING A TRUST MODEL FOR MULTI-AGENT INTERACTIONS USING CONFIDENCE AND REPUTATION", *Applied Artificial Intelligence,* vol. 18, pp. 833-852, 2004/10/01.

[89] S. Ramchurn, "Multi-Agent Negotiation using Trust and Persuasion", Doctoral, ECS, University of Southampton, 2004.

[90] S. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings, "A computational trust model for multi-agent interactions based on confidence and reputation", presented at the 6th International Workshop of Deception, Fraud and Trust in Agent Societies, 2003, pp. 69 - 75.

[91] S. Song, K. Hwang, and M. Macwan, "Fuzzy Trust Integration for Security Enforcement in Grid Computing Network and Parallel Computing", vol. 3222, H. Jin, G. Gao, Z. Xu, and H. Chen, Eds., ed: Springer Berlin / Heidelberg, 2004, pp. 9-21.

[92] S. Song, K. Hwang, and Y.-K. Kwok, "Trusted Grid Computing with Security Binding and Trust Integration", *Journal of Grid Computing,* vol. 3, pp. 53-73, 2005.

[93] K. H. S. Song, "Trusted Grid Computing with Security Assurance and Resource Optimization", in *17 thInternational Conference on Parallel and Distributed Computing Systems (PDCS-2004)*, 2004, pp. 15-17.

[94] Y.-H. Lam, Z. Zhang, and K.-L. Ong, "Buying and selling with insurance in open multi-agent marketplace", presented at the Proceedings of the 9th Pacific Rim international conference on Artificial intelligence, Guilin, China, 2006, pp. 945 - 949.

[95] Y.-H. Lam, Z. Zhang, and K.-L. Ong, "Trading in Open Marketplace Using Trust and Risk", presented at the Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2005, pp. 471 - 474.

[96] Audun Jøsang, Daniel Bradley, and S. J. Knapskogsang, "Belief-based risk analysis", presented at the Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation - Volume 32, Dunedin, New Zealand, 2004, pp. 63 - 68.

[97] A. Jøsang and S. Presti, "Analysing the Relationship between Risk and Trust: Trust Management", vol. 2995, C. Jensen, S. Poslad, and T. Dimitrakos, Eds., ed: Springer Berlin / Heidelberg, 2004, pp. 135-145.

[98] W. Yan and L. Fu-ren, "Trust and Risk Evaluation of Transactions with Different Amounts in Peer-to-Peer E-commerce Environments", in *e-Business Engineering, 2006. ICEBE '06. IEEE International Conference on*, 2006, pp. 102-109.

[99] Yan Wang, Duncan S. Wong, Kwei-Jay Lin, and V. Varadharajan, "Evaluating transaction trust and risk levels in peer-to-peer e-commerce environments", *Information Systems and E-Business Management,* pp. 1-24, 2007.

[100] A. F. Salam, L. Iyer, P. Palvia, and R. Singh, "Trust in e-commerce", *Commun. ACM,* vol. 48, pp. 72-77, 2005.

[101] A.F. Salam, Lakshmi Iyer, Prashant Palvia, and R. Singh, "Understanding Trust in Electronic Commerce Relationships", in *National Decision Sciences Institute*, 2002.

[102] C. English, S. Terzis, and W. Wagealla, "Engineering Trust Based Collaborations in a Global Computing Environment:Trust Management", vol. 2995, C. Jensen, S. Poslad, and T. Dimitrakos, Eds., ed: Springer Berlin / Heidelberg, 2004, pp. 120-134.

[103] C. English, S. Terzis, W. Wagealla, H. Lowe, P. Nixon, and A. McGettrick, "Trust Dynamics for Collaborative Global Computing", presented at the Proceedings of the Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003, pp. 283 - 288.

[104] O. K. Hussain, " Risk measurement, prediction and management in e-business and service oriented environment ", Ph.D., Curtin Business School, Curtin University, Perth, 2008.

[105] O. Hussain, T. Dillon, F. K. Hussain, and E. Chang, "Probabilistic assessment of financial risk in e-business associations", *Simulation Modelling Practice and Theory,* vol. 19, pp. 704-717, 2011.

[106] J. Xu, Z. Liu, and Y. Li, "Integrating processes of logistics outsourcing risk management in e-business", in *Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, 2006, pp. 544-547.

[107] C. Lin and V. Varadharajan, "Trust based risk management for distributed system security-a new approach", *First International Conference on Availability, Reliability and Security ARES*, 2006.

[108] C. Schläger, M. Sojer, B. Muschall, and G. Pernul, "Attribute-Based authentication and authorisation infrastructures for e-commerce providers", *E-Commerce and Web Technologies,* pp. 132-141, 2006.

[109] C. Schläger and T. Nowey, "Towards a risk management perspective on AAIs", *Trust and Privacy in Digital Business,* pp. 41-50, 2006.

[110] B. Martens and F. Teuteberg, "Decision-making in cloud computing environments: A cost and risk based approach", *Information Systems Frontiers,* pp. 1-23, 2012.

CHAPTER 3

PROBLEM DEFINITION

3.1 Introduction

One of the key requirements for the success of cloud computing is the provision of services being delivered as promised at the required capacity, time and need. This can be achieved by the service user by first establishing the SLA with the service provider who is capable of providing the required service, and then monitoring the performance of the service delivery to ensure that the quality of the service is according to the defined parameters as stated in the SLA. In this regard, the role of SLA management is discussed in Chapter 1. SLA management is important to ensure and assure that the defined service meets certain threshold criteria that are established in the agreement. However, as discussed in Chapter 1, SLA management from a consumer's point of view in cloud computing is a challenging task and service provider selection before the start of interaction and service monitoring after the start of interaction are important steps. To address the issue of SLA management various approaches to service provider selection and SLA monitoring were discussed in the previous chapter.

In this chapter, the problem to be addressed in this thesis is defined and discussed. Firstly, the concepts and terms that will be used are defined in Section 3.2. Then the formal definitions of the problems that will be addressed in the various research issues are given in Sections 3.3 and 3.4. Finally, the research methodology which is adopted to solve the problems is presented in Section 3.5. The conclusion of the chapter is given in Section 3.6.

3.2 Key Concepts

In this section, the terms and definitions that will be used throughout this thesis are presented. These terms then can be used to explain the concepts in this chapter and in subsequent chapters.

3.2.1 Service User

A *service user* is an entity who requests, selects and consumes the service. A *service user* can therefore be a *service requester*, a *client* or a *customer*. In this thesis, a service user is referred to as a *service consumer* and the terms *service user*, *service consumer* and *consumer* are used interchangeably.

3.2.2 Service Provider

A *service provider* is an entity who can fulfil the service requests of a service consumer.

3.2.3 Context

Context is defined as the base or the purpose for which the service consumer forms a service with a service provider. In other words, context represents the high level nature of the service user's interaction with the service provider and gives the description of the interaction. For example, a service consumer may request a video conferencing service from a cloud service provider. In this example, *Video Conferencing* is the context of the interaction.

3.2.4 Criteria

Criteria is defined as the related functionalities or activities that come under a particular context. In other words, the context of the interaction can be decomposed into several detailed aspects which are regarded as its criteria. Continuing the above example of *Video Conferencing*, a service user may have the following requirements: instant access from a browser with no software installation requirement and a high quality video.

3.2.5 Assessment Criteria or Desired Outcomes

Assessment Criteria is defined as the criteria from the particular context of the interaction which the service user wants to fulfil in its interaction with the service provider. *Desired outcomes* are defined as the collection of the assessment criteria which the service user wants to achieve while interacting with the service provider in a particular context, suitably weighed by their importance. In other words, the

desired outcomes of the interaction are the demand or set of factors which show specifically what the service user wants in its interaction with the service provider in the particular context. The assessment criteria in above example are access speed and quality of video.

3.2.6 Expected Behaviour

Expected Behaviour is defined as the behaviour to which the service user expects the service provider to adhere in the interaction, before the negotiation phase. This behaviour is formed by the service user and is according to the desired outcomes that it wants to achieve while interacting with the service provider. Continuing the above example, a service user expects a high speed access through a browser and a high quality of service.

3.2.7 Expectations

Expectations are defined as the behaviour which both the service user and service provider negotiate and decide to act on in the interaction. This behaviour is a step further from expected behaviour and comes into effect when both the service user and service provider negotiate and decide on how they will behave in the interaction. Based on these expectations, a service level agreement is formed between the interacting business partners, and subsequently, based on this behaviour, the service user should analyse the various factors to manage his service with the service provider.

3.2.8 Actual Behaviour

Actual Behaviour is defined as the behaviour of the service provider when it interacts with the service user. In other words, actual behaviour is the quantitatively expressed set of functionalities or activities delivered by the service provider in its interaction with the servicer user.

3.2.9 Time Space

Time Space is defined as the total period of time which the service user takes into consideration to ascertain the factors for service management in interacting with the service provider. The range of the time space might vary according to the duration of

time considered by the service user to determine the reputation and transactional risk in interacting with the service provider.

3.2.10 Timeslot

Timeslot is defined as a non-overlapping interval of time in the time space of the interaction. The timeslot is obtained by dividing the time space into different equal non-overlapping parts of time as depicted in Figure 1.4.

3.2.11 Trustworthiness

Trustworthiness is defined as the capability of a service provider to complete a certain task. The trustworthiness of a service provider is determined from a service user's own past interaction history.

3.2.12 Recommending Users (RUs)

In the case of indirect interaction between a consumer and a service provider, the consumer solicits the trustworthiness of the service provider from a group of users who interacted with the service provider in the same context in the past. This group of users provide their opinions about the service provider in the form of trustworthiness values. This group is referred to as *Recommending Users (RUs).* Each RU stores past experiences in the form of trust values assigned to a service provider in an information repository.

3.2.13 Recommendations

Recommendations or *recommendation opinions* are the feedback provided by RUs to the consumer on their query about the reputation of a service provider.

3.2.14 Reputation

Reputation is defined as the aggregation of all the recommendations (trustworthiness values) from the RUs about a service provider. Reputation is based on recommendations, time of interaction of RU with the service provider and credibility of RU in providing such opinions.

3.2.15 Credibility of the Recommendations

Credibility is the trustworthiness of opinions provided by RUs and it defines how trustworthy an RU is in providing opinions.

3.2.16 Time Delay

Time delay is the time variability of the reputation opinion of the RU which can be represented by the rate of decay. Time delay is the time which has elapsed between the previous interactions of the RU with the service provider and the current time. The RU may have had more interactions in the intervening period with the service provider and therefore, can give more accurate opinions.

3.2.17 Resources

Resources is that item in which the consumer invests in the interaction to achieve the desired outcomes while interacting with a service provider. The resources might vary according to the context of the interaction.

3.2.18 Financial Resources

In the domain of e-commerce business interaction, the resources in which the consumer invests in the interaction are defined as *financial resources* or *monetary value* while interacting with a service provider to achieve the desired outcomes.

3.2.19 Service Degradability

Service degradability is defined as an event which causes the reduction in quality of a service that is promised to be delivered by the service provider according to terms given in SLA.

3.2.20 Probability of Failure of Service Delivery

A service consumer has to determine the probability of failure of service delivery against promised SLOs parameters. In other words, how much a service has been degraded is determined using probability of failure of service delivery.

3.2.21 Consequences of Service Degradation

A consumer also needs to determine the consequences of a service degradation. In other words, a consumer needs to determine the impact of service degradation on his financial outcomes.

3.3 Problem Definition

In Chapter 1, the importance of SLA management in cloud computing was discussed, including service provider selection after SLA establishment and SLA monitoring after the initiation of the service with the service provider. In Chapter 1, it was also discussed that one of the techniques to evaluate the capability of the service provider before the start of interaction is *trust*. Trust plays an important role in defining the level of belief in the service provider's ability to commit to the SLAs. Since trust plays important role in e-Commerce, which is a self-serve business model, it is reasonable to assume that similar perceptions of trust would apply to self-service cloud providers [1].

There are two ways by which a service consumer evaluates the capability of the service provider to commit to the SLAs. First, a service consumer may have a direct past interaction with the service provider. In this case, the consumer evaluates the level of trust on the basis of his past experiences with the service provider. However, if the consumer has not had direct interaction with the service provider, he may have to rely on recommending users to provide recommendations (reputation value) on the basis of their past experiences with the service provider in same context. In the scenario where there are both direct and indirect interactions, a consumer may give preference to his personal experience to ascertain the capability of a service provider and uses the recommendations of the recommending users only if he has no personal interaction with the service provider or if his interaction with the service provider is too old after a period of time. In both cases, for trust evaluation, the context of the service should be taken into account along with the dynamic nature of time. For indirect interaction, additional factors should be considered for reputation assessment of the service provider. These factors may include the credibility of the recommending users in providing the recommendations.

As discussed above, the problem of trust evaluation of a service provider for service selection consists of two parts:

(a) to calculate the capability of the service provider by using its trustworthiness in the case of direct interaction and;
(b) to calculate the capability of the service provider by using its reputation.

Several computational approaches for trust and reputation were discussed in Chapter 2. Each of these approaches has its benefits and drawbacks. For cloud computing, there is a need for a reputation determination methodology that is self-adaptive and which fits the automatic QoS control model for cloud computing. Such a methodology will increase the robustness of the service provider selection process in cloud computing. To the best of the researcher's knowledge, no such methodology exits in literature that has the aforementioned features. Such a methodology will be proposed in this thesis. The overview of the solution to this problem is presented in the next chapter and the detailed solution is given in Chapter 5.

After the service provider selection phase of SLA management, the consumer enters into an interaction with the service provider. The next phase of SLA management is SLA monitoring in the post-interaction time phase in which the performance of the service provider is monitored against the performance criteria defined in the SLA. As discussed in previous chapters, one way to evaluate the performance of the service provider is risk. There are two parts of assessing the performance of a service provider: performance risk, which is the probability of failure of not performing according to agreed SLA; and financial risk which is the impact of the probability of failure on the outcomes of what a consumer wants to achieve through the interaction. Although risk assessment approaches in cloud computing have been discussed in the literature, these approaches mainly focus on the privacy and security of data. The type of risk that assesses the performance of the service provider according to SLA has not been discussed. This can be achieved by considering both performance risk and financial risk.

Performance risk deals with identifying and quantifying undesirable events due to the non-compliance of the service provider to the SLA. In business transactions, the quantification of performance risk can be achieved by determining the level of risk of

the service provider in committing to his desired outcome or the probability of not achieving the desired outcome as a result of the non-fulfilment of his expectations in interacting with the service provider.

However, while determining performance risk, the consumer should consider different timeslot scenarios. In addition to determining performance risk in the current point of time, the time period of interaction might extend to a point in time in the future. For the current point of time, it may be the case that the consumer wants to interact with the service provider and hence wants to make a risk-based decision in that timeslot. In this situation, it is important that run-time parameter values should be obtained in the cloud environment which can be obtained by utilising the service provider's measurement interface. But it may be the case that the consumer wants to make a risk-based decision in a point of time which extends into the future. In this scenario, the consumer should use prediction methods which consider the past performance risk values of the service provider in the same context.

Financial risk deals with determining the extra amount of resources that a consumer needs to keep at stake due to performance risk. For financial risk, the consumer should first ascertain the financial resources that he needs to invest in an interaction in the same timeslots which he considered for the performance risk. Then, the impact of performance risk should be determined on invested cumulative financial resources in order to determine the extra resource investment. To determine the extra resource investment, a consumer should consider two types of assessment criteria: assessment criteria which depend on the performance of the service provider (dependable criteria) and assessment criteria which do not depend on the performance of service provider (non-dependable criteria). From extra resource investment from both of the assessment criteria, the consumer should determine the financial loss.

Once the consumer determines the financial loss in an interaction, he should then utilize it to determine the risk based recommendation based on his current risk attitude. In the previous chapter, the existing risk management approaches in the literature which consider the risk attitude of the consumer as an important factor were highlighted, these approaches do not consider the impact of risk attitude on the determined financial loss.

The problems associated with SLA monitoring in the post-interaction time phase in the cloud from the consumer's point of view are as follows:

(a) To determine the performance of the service provider according to the defined SLA in the given time scenarios namely, in the current point in time or in a future timeslot.
(b) To determine the impact of service degradation on financial outcomes of the consumer which results in the extra investment that a consumer has to make due to the non-conformance of the service provider to the SLA to achieve the desired outcome.
(c) To determine the magnitude and level of loss in an interaction because there may be different levels of loss and the consumer may want to determine the impact of each of these levels on their decision making.
(d) To determine the impact of the risk attitude of a consumer on the levels of financial loss to assist the consumer with informed decision making for the continuation of the service with the service provider.

To address these issues, the solution overview of the post-interaction assessment is presented in the next chapter and the detailed solution is provided in Chapters 5, 6, 7 and 8.

To solve the problems associated with service provider selection in pre-interaction time phase and the SLA monitoring in post-interaction time phase, as discussed above, a conjoint approach is needed that uses trust or reputation assessment in the pre-interaction time phase and risk analysis in the post-interaction time phase to enable the consumer to make an informed decision. It is concluded that an SLA management framework for cloud computing is needed to solve the above mentioned problems.

Having discussed the importance of SLA management in cloud computing using trust and risk based approaches, the problem that will be addressed in this thesis is broadly defined as follows:

To develop an SLA management framework that enables consumers to make an informed decision about selecting a service provider in the pre-interaction time

phase and on the continuity of the service in the post-interaction time phase to achieve the desired financial outcome in the cloud computing environment.

This problem statement can be broken down into the following sub-problems:

1. to propose an SLA management framework in general and an SLA monitoring framework in particular to monitor the financial outcome of a consumer
2. to propose a methodology for service provider selection in the pre-interaction time phase
3. to propose a methodology for service monitoring in the post-interaction time phase
4. to propose a methodology by which the consumer can make an informed risk-based decision by considering his risk attitude with the level of financial loss in an interaction
5. to measure the effectiveness of the SLA framework in the cloud environment and to propose an evaluation framework representing real world scenarios.

3.4 Research Issues

The following research issues were identified from the discussion in the previous section:

Issue 1: Propose an SLA management framework

As discussed in the previous section, an SLA management framework is needed from the consumer's point of view that implements SLA management as a conjoint approach to evaluate the trust or reputation of a service provider for the selection of services and to evaluate the financial risk to determine whether to continue with the service provider or not. This framework is discussed in the next chapter.

Issue 2: Propose a methodology for service provider selection in the pre-interaction time phase

The concept of trust or reputation can be used to assist consumer with the selection of service provider. As discussed in the previous section, there are two scenarios when determining the trustworthiness of a service provider, namely the direct interaction scenario and the indirect interaction scenario. The *trust* of a service

provider can be utilised to determine his trustworthiness in the case of direct interaction and the *reputation* can be utilised to determine the trustworthiness of a service provider in the case of indirect interaction. Both these scenarios are considered in Chapter 5.

Issue 3: Propose a methodology for service monitoring in the post-interaction time phase

As discussed in the previous section, from a consumer's point of view, a methodology is needed for effective service monitoring which can be achieved by determining the probability of failure of not performing according to SLA and probability of not achieving the financial objectives in an interaction with the service provider. As discussed in previous section that performance risk deals with identifying and quantifying undesirable events and financial risk deals with determining extra resource investment to determine the financial loss or levels of loss in the interaction, both performance risk and financial risk should be used together to achieve the assurance of service quality from a service provider. Chapter 6, the methodology of performance risk is proposed and in Chapter 7, the methodology of financial risk is discussed.

Issue 4: Propose a methodology by which the consumer can make an informed risk-based decision.

From the discussion in the previous section, it can be concluded that a methodology is needed which assists the consumer in the decision-making process for the continuity of the service by utilizing the risk attitude of the consumer and then determining its impact on the levels of loss with the service provider. This methodology is discussed in Chapter 8.

Issue 5: Validate the proposed methodologies by simulation experiments to prove their effectiveness

To validate the proposed methodologies, given issues 1 – 4, it is necessary to build a prototype system that simulates a business interaction in the cloud environment. By simulation, the effectiveness of the SLA management framework will be modelled and verified, demonstrating that the proposed methodologies are suitable for SLA

management in real world business interactions. Details on the simulation and validation are presented in Chapter 9.

3.5 Research Approach to Problem Solving

In addressing the stated problem, this thesis focuses on the development and testing and validation of a framework for SLA management in cloud environment. To address the research issues that have been raised in previous sections, a solution should follow methodical scientific approach for methdology development. For this purpose scientifically-based research methods and the choice a particular research method for the given problem are discussed in this section.

3.5.1 Research Methods

In this sub-section two broad categories of research methods in information systems are discussed which are:

(a) the science and engineering approach, and

(b) the social science approach.

Using the science and engineering approach, the theoretical predictions can be confirmed. According to Gallier [2] to make something work, three phases of model should be followed which are explained below:

- Conceptual phase (level one): this phase concerns with defining new ideas and concepts through problem analysis.
- Perceptual phase (level two): this phase concerns with modelling of new system based on ideas and concepts of conceptual phase. This phase includes design and development.
- Practical phase (level three): this phase verifies the model developed in last phase through testing and validation.

This research approach may result in new theoretical framework along with new methodologies or new techniques. Moreover, it may result in solution of problem along with defining problem itself.

The second research approach, the social science research can be either qualitative or quantitative. Most common tool to carry out this research is survey or interview. For qualitative research interviews are devised to address or investigate a particular issue. Qualitative research usually does not result in collection or analysis of statistical data. Quantitative research on the other hand uses surveys to collect extensive data and then it uses statistical analysis of this data to verify the hypothesis. In a typical social science research approach, the problem is identified by use of questionnaire or surveys which are then followed formulating of hypothesis. Social science research approach helps in verifying a formulated hypothesis [3-5]. This approach helps researchers to understand social issues within the area of research. However, there are debates in literature about the usefulness of social science research approach. One such argument is since the survey data (usually textual data) is quantified, it may fortify the phenomenon which a researcher is trying to understand in social or cultural context [6]. This kind of research helps in determining a methodology that may or may not work for a given problem and the reason for it. However, this research approach does not propose what methodology should be used and for a particular problem how to device a new methodology. The social research is based on testing and evaluation of a method that has already been produced from science and engineering research.

This thesis deals with the development of a framework for SLA management in cloud computing, and hence, it clearly falls into the domain of science and engineering research.

3.5.2 Choice of a Science and Engineering-based Research Method

In this thesis, for the solution of the proposed problem, a science and engineering-based research is selected and the phases of this method are depicted in Figure 3.1.

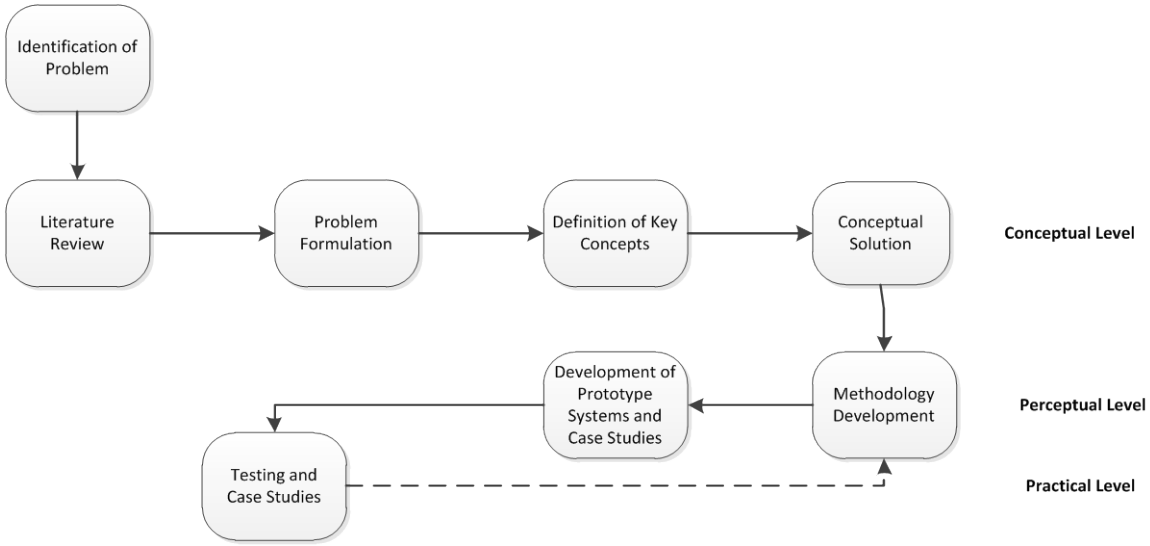

Figure 3.1: Phases of science and engineering-based research method

Firstly, the research problems were identified. In conceptual phase, comprehensive literature review was carried out to identify the problem. Problem formulation defines the problem based on the hypothesis. The key concepts were then defined that are needed to address the problem. Finally, in conceptual phase, conceptual solution in the form of conceptual framework was formulated. In perceptual phase, methodologies for trust and reputation determination, performance and financial risk were developed for pre- and post-interaction assessment of a service provider in cloud environment. Prototypes were developed for proposed methodologies and case studies were presented to test proposed methodologies. In practical phase, based on the prototype in perceptual phase, the methodologies were tested and verified. On the basis of testing results the methodologies can be adjusted if needed.

In this thesis the research method proposed by Nunamaker et al. [5] has been adopted which consists of problem definition, conceptual solution and system prototypes processes with the evaluation and validation of research output. First, the analysis is performed on the problem and the solutions and proofs are developed as a result of analysis. This result of analysis is then used for evaluation of research outcomes. In this thesis, the research method proposed by Nunamaker et al. [5] is used for the verification and validation of research output, through proof of concept.

3.6 Conclusion

In this chapter, the issues in the literature were summarized and the problem definition that will be addressed in this thesis was formulated. The defined problem was then decomposed into different research sub-problems, which when solved individually, leads to the solution to the defined problem of making an informed decision in a business interaction. Further, the different research approaches available were discussed and the chapter concluded with the science and engineering research methodology which uses system development, similar to the one used in this research. In the next chapter, the solution overview of the research issues discussed in this chapter is proposed.

3.7 References

[1] B. Kaliski, "Multi-tenant Cloud Computing: From Cruise Liners to Container Ships", in *Trusted Infrastructure Technologies Conference, 2008. APTC '08. Third Asia-Pacific*, 2008, pp. 4-4.

[2] R. D. Galliers, *Information Systems Research: Issues, Methods and Practical Guidelines*: Blackwell Scientific Publications, 1992.

[3] Donald G. McTavish and H. J. Loether, *Social Research*: Allyn & Bacon, 1999.

[4] F. Burstein and S. Gregor, "The systems development or engineering approach to research in information systems: an action research perspective", in *Proceedings of the 10th Australasian Conference on Information Systems*, 1999, pp. 122-134.

[5] Jay F. Nunamaker, Minder Chen and Titus D. M. Purdin, "Systems Development in Information Systems Research", *Journal of Management Information Systems,* vol. 7, pp. 89-106, 1991.

[6] Bonnie Kaplan and Joseph A. Maxwell, "Qualitative Research Methods for Evaluating Computer Information Systems", *Evaluating Health Care Information Systems: Methods and Applications,* pp. 45-68, 1994.

CHAPTER 4

SOLUTION OVERVIEW

4.1 Introduction

As discussed in the previous chapters, a large body of scholarly research in the literature has attempted to implement SLA management by specifically SLA monitoring in cloud computing. But none of these approaches takes into account factors such as service provider selection in the pre-interaction time phase and monitoring the service provider's performance in the post-interaction time phase from the consumer's perspective. To address this, in the previous chapter, four different research issues were discussed, based on which the consumer can select a capable service provider from the possible ones and then analyse its performance in the post-interaction time phase of the interaction to utilize it to make an informed decision about continuing its service. In this chapter, an overview of the solution for each of the defined research issues is proposed.

4.2 Broad Solution Overview

In order to implement the solution in a pragmatic setting, the architectural foundation to make the solution scalable and modular is described. Modularity has been achieved by introducing a multi-layered architecture [1] which implements the separation of concerns to allow each layer to encapsulate the core functionality and communicate with other layers to invoke the required services. Based on the multi-layered architecture principles, the proposed solution is depicted in Figure 4.1. The overall solution architecture comprises of three layers, each containing a dedicated set of services in addition to interacting with other layers. Each layer accepts inputs and performs its operations by invoking different services and then generates an output. This output is then used as an input for the invocation of services in the next layer. In the following, the role, responsibility, input and output parameters of each layer, service and interaction are described.

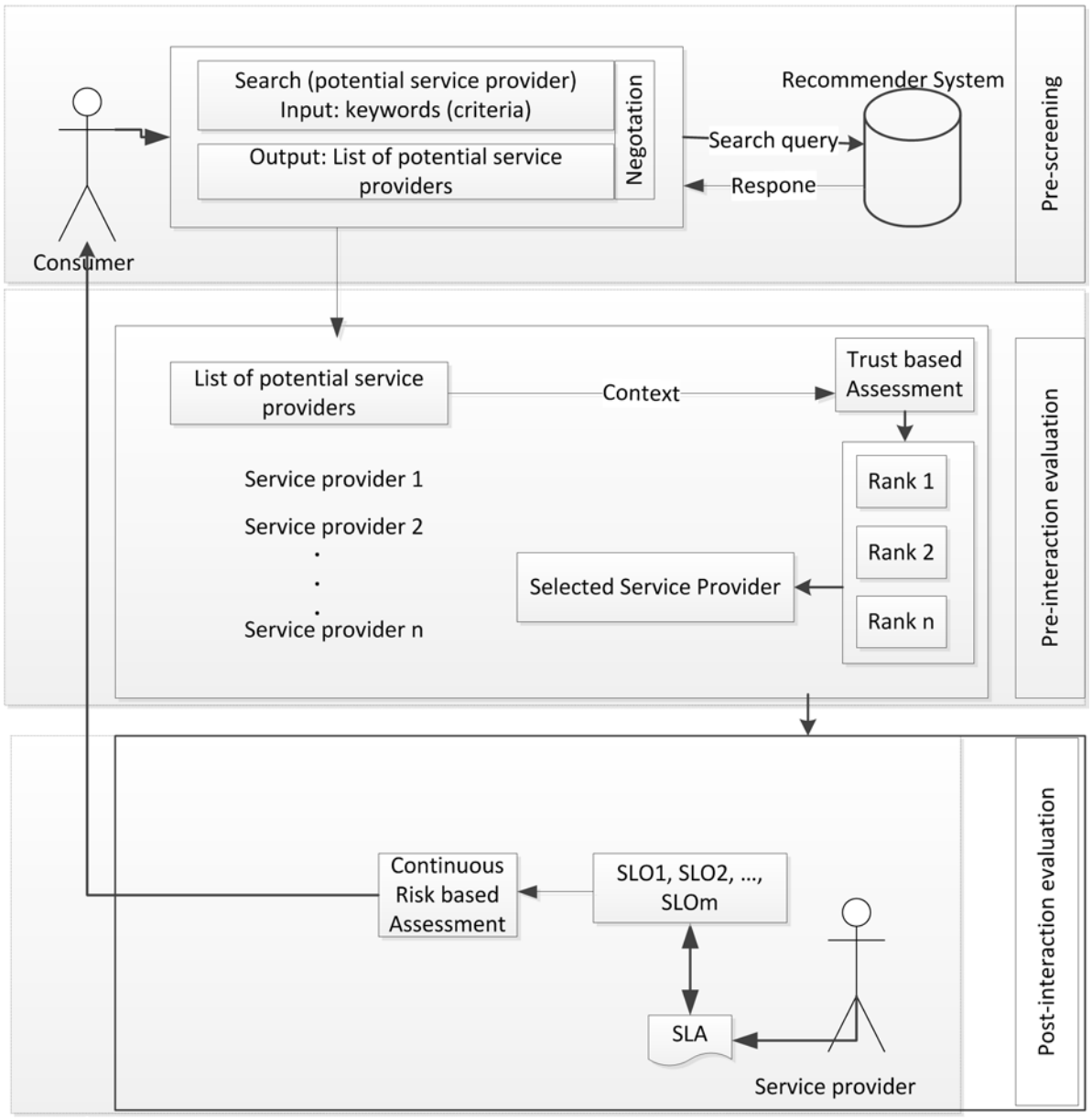

Figure 4.1: Conceptual framework for SLA management in cloud computing

In Figure 4.1, the consumer interacts with all three layers. The consumer interaction starts with pre-screening layer where he enters search criteria. In pre-interaction evaluation layer, the consumer proceeds to third layer when service selection results are presented to him. In post-interaction evaluation layer, the consumer enters several risk assessment criteria such as time phase, time slots and financial expectation values. The service selection result, SLA and SLOs are stored in internal database of the system. The implementation details of this framework are presented in Chapter 9.

The first layer of the proposed solution is known as the *pre-screening* layer. The role of this layer is to shortlist the potential service provider candidates. Since the recommender system merely provides the service provider's profile [3], in this layer, the recommender system is accessed to pre-screen the potential service provider candidates. This layer interacts with the recommender system by passing keywords (search parameters), against which the recommender system provides the list of matching service providers. In order to resolve the ambiguities that arise during the interaction with the recommender system and the pre-screening layer, a negotiation service is used. The negotiation service helps in clarifying the keywords by suggesting alternating synonyms to the recommender system. The output of this layer is the list of potential service providers. Further detail of pre-screening layer is discussed in Section 4.3.

The second layer is known as the *pre-interaction* layer. The role of this layer is to select a service provider based on its trustworthiness and reputation in the pre-interaction time phase. A list of potential service providers is used in the pre-interaction layer as input. The trustworthiness and/or reputation of each service provider on the list are then assessed using the *trust* or *reputation assessment* process in the pre-interaction layer. The reputation or trustworthiness values of the service provider are then aggregated. After calculating the reputation or trustworthiness of each service provider on the list, they are then ranked. The service provider with the highest rank (highest reputation values) is selected to fulfil the request of the consumer. The output of this layer is the selected service provider. This output is passed to a third layer to assess the continuous performance of the service provider. Further detail of pre-interaction layer is discussed in Section 4.4.

The third and last layer of this proposed solution is known as the *post-interaction* layer. The role of this layer is to assess the performance of the service provider in the post-interaction time phase. The SLOs for the requested service are obtained from the SLA provided by the service provider. A process called *continuous risk-based assessment* is used to determine the performance of the service provider and to determine the impact of this performance on the business outcomes of the consumer. The final outcome of this assessment process is risk based recommendation to consumer based on his current risk attitude. Based on risk based recommendation, a

consumer may decide on its future course of interaction with the service provider. Further detail of post-interaction layer is discussed in Section 4.5.

In proposed SLA management framework, I suggest a trusted third-party service provider that carries out pre- and post-interaction assessments on the behalf of consumer. In cloud computing, it has been suggested that QoS assessment can be implemented by a trusted third-party service provider that provides unbiased assessment to both the consumer and service provider. In other words, SLA monitoring can be delegated to third-party service providers [2] by both consumer and service provider. In Chapter 1, it was highlighted that most service providers provide consumers with dash boards for performance monitoring. However, the consumer needs more control over the performance of a service provider which is possible through an independent service provider who performs QoS assessment on the consumer's request.

In my solution, I use a third-party service provider for SLA management called the Third-Party Service Level Agreement Manager (TP SLA Manager) for service provider selection and service level monitoring. The role of the TP SLA Manager is discussed in the next sections.

4.3 Pre-screening for Identifying a List of Possible Service Providers

The service search process consists of a service knowledge base which stores the domain specific service ontologies and the service description entity (SDE) metadata [3]. The architecture of the semantic service search system is shown in Figure 4.2.

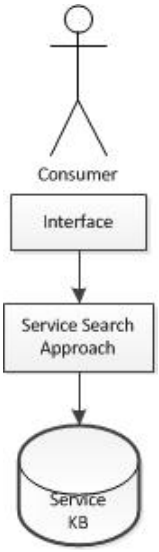

Figure 4.2: Semantic search engine architecture

The workflow of the system is as follows:

The consumer uses the search interface to enter a set of key terms. The service search approach retrieves service metadata from the service knowledge base. In this system, SDE elements are annotated and mapped with the service ontology to allow querying the knowledge base. The service ontology also enables mapping with upper ontologies to create a relationship between entities residing in different systems. However, in this thesis, a service ontology to semantically describe the data describing the service descriptions is implemented. In the following section, the service search module is discussed.

Service Search Approach

The system architecture of the system search module is shown in Figure 4.3.

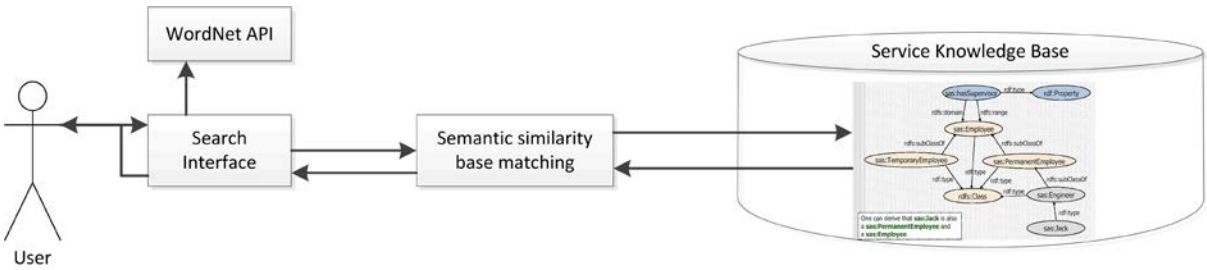

Figure 4.3: Architecture for the Service Search Approach

The workflow for this approach is as follows: A user enters a set of query terms into the search interface. Each query term will be send to the WordNet API[1] by the search interface. Synonyms for each query term will be retrieved from the API, if it exists. The search interface will then send the query terms and their synonyms to the semantic similarity base matching. A matching algorithm is run by the semantic similarity base matching to compute the similarity values between the service ontology concepts stored in the service knowledge base and the query terms.

4.4 Solution Overview of Pre-interaction Service Provider Selection

In the previous chapter, the need for the trust and reputation assessment methodologies to determine the trustworthiness or reputation of a service provider was highlighted. These methodologies help the consumer in the decision making process regarding the service selection process. In Section 4.2, an SLA management framework was proposed in which layer 2 is responsible for the pre-interaction assessment on the basis of trust/reputation of potential service providers. In this section, the solution overview of the trust/reputation assessment methodologies that are used to determine the trustworthiness or reputation of a service provider are presented.

For pre-interaction service provider selection, to solve the problem of direct or indirect interaction scenario which is highlighted in last chapter the solution proposed in this chapter consists of two parts: first is a solution for a direct interaction scenario and second is a solution for an indirect interaction scenario of a service consumer with a service provider. Direct interaction is based on direct past experience of a consumer with a service provider. For the direct interaction, the relative importance of each timeslot in the pre-interaction time phase is given by assigning an appropriate weight to each timeslot and then determining the trustworthiness value according to the rate of decay. Indirect interaction is based on recommendations provided by *recommending users* (RUs) which are discussed in last chapter. For the indirect interaction, reputation is determined by considering

[1] https://wordnet.princeton.edu/wordnet/related-projects/ (accessed on: April 12, 2014)

three factors namely, the recommendation opinion given by RUs, the credibility of RUs and time decay function which are highlighted in last chapter.

For the direct interaction scenario, to determine the time decay for adjusting the weight given to the trust value in each timeslot, the use of the logistic decay function is proposed. For the indirect interaction scenario, the use of the fuzzy logic-based approach [4] to determine the reputation of a service provider is proposed. An overview of the approach to determine the capability of the service provider is shown in Figure 4.4.

Steps in Figure 4.4 are explained as follows:

Forms the context and divides the time space into pre- and post-interaction phases: First thing that a consumer needs to establish before the start of interaction is the context of interaction. Then a consumer needs to divide the time space of interaction into pre- and post-interaction time phases.

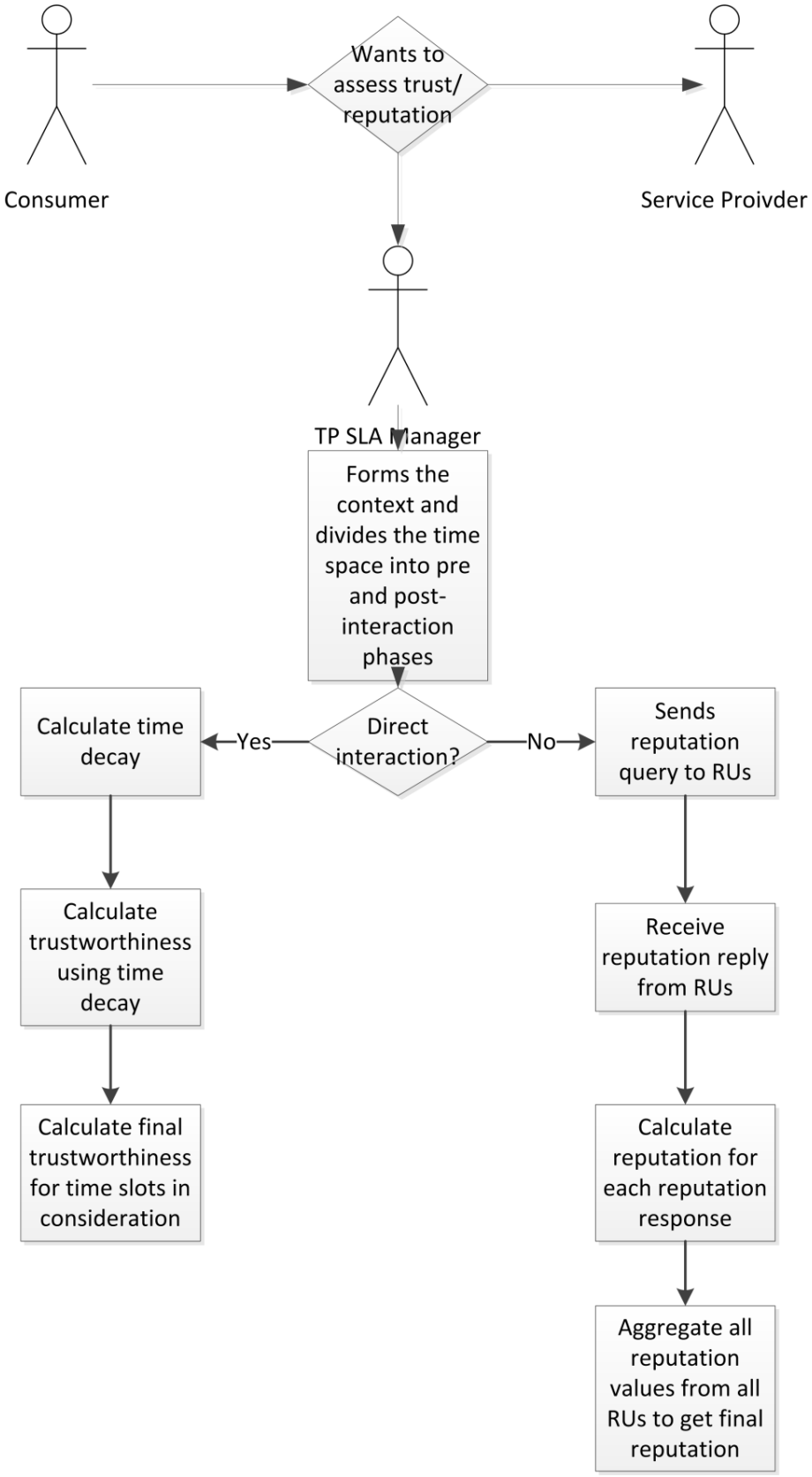

Figure 4.4: Solution overview of trust and reputation methodologies

Direct interaction? Consumer starts pre-interaction assessment in pre-interaction time phase. That is, before the start of an interaction, a consumer needs to determine the trustworthiness of a service provider. Trustworthiness can be assessed by two possible ways: through direct or indirect interaction of a consumer with the service provider. Based on these interaction scenarios two different pathways are followed which are discussed below.

Calculate time decay: In case of direct interaction, a consumer first needs to determine the time decay of an interaction according to the current point of interaction. In other words, time decay is the passage of time since the last interaction of consumer with the service provider.

Calculate trustworthiness using time decay: Trust values that have been assigned to a service provider are weighted according to the time decay. This is done for each past timeslot that a consumer considers for trustworthiness determination.

Calculate final trustworthiness for timeslots in consideration: The trustworthiness determined for each timeslot using the time decay weightage needs to be aggregated to give final trustworthiness of a service provider. Weighted average is proposed for this purpose. Further detail about the service provider selection using the direct interaction scenario is given in Section 4.4.1.

Sends reputation query to RUs: In case of indirect interaction of a consumer with the service provider, *Recommending Users* (RUs) are consulted by TP SLA Manager on behalf of the consumer.

Receive reputation reply from RUs: RUs provide their opinions about the reputation of a service provider about whom the query is made.

Calculate reputation response for each reputation response: Reputation is calculated for each reputation response because reputation response consists of recommendation opinion of a RU and time delay. These factors along with the credibility of the RU are synthesized together to determine the reputation of the service provider about whom the query is made.

Aggregate all reputation values from all RUs to get final reputation: The final step is the aggregate all reputations from all RUs to get one final reputation value for the service provider. Further detail of service provider selection using indirect interaction scenario is discussed in Section 4.4.2.

Based on the steps discussed above, in the following sub-sections, the solution overview of the *direct* and *indirect interaction* scenarios is discussed.

4.4.1 Solution Overview for Direct Interaction Scenario

In order to determine the trustworthiness of the service provider in the direct interaction scenario, the following are the important features for the proposed methodology:

- To determine the trustworthiness in current timeslot, firstly the context of the interaction is determined and the pre-interaction time phase is divided into timeslots.
- Then the weightage assigned to the timeslots is determined, using the time decay function.
- For each timeslot, the time decay function assigns the weight which ranges from 0 to 1.
- The next step is to determine the trust value in each timeslot based on the assigned weights. To do this, the trust value from the trust database is obtained and multiplied by the weight for each timeslot.
- The overall trustworthiness is then obtained by dividing the sum of weighted trust values by the total number of timeslots under consideration.

4.4.2 Solution Overview for Indirect Interaction Scenario

Since I am using fuzzy logic-based approach to determine the reputation of a service provider in the direct interaction scenario, the following are the important features for the proposed methodology:

- Firstly, the input variables, namely Recommendation Opinion, Credibility and Time Delay, are fuzzified.

- Next, a fuzzy inference method called the Takagi-Sugeno method [5] is used to generate the trained Fuzzy Inference System (FIS).
- This trained system is then utilized by the TP SLA Manager in the SLA framework which sends a reputation query request to a network of recommending users (RUs).
- Recommending users who have interacted with the service provider in the same context reply to this query by sending the reputation query reply.
- For each reputation query response received from an RU, the TP SLA Manager obtains the reputation value using the trained FIS.
- The TP SLA Manager then aggregates all reputations by all RUs involved in the reputation query. It will give the final reputation for a service provider.
- Finally, the TP SLA Manager compares the reputation values of all potential service providers and selects the service provider with the best reputation.

Details of proposed methodologies for direct and indirect interaction assessments are discussed in next chapter. After selection of trustworthy service provider a consumer has to monitor the performance of a service provider, assess the financial risk, determine the financial loss and make an informed decision in post-interaction time phase whether to continue with the service provider or not as discussed in next section.

4.5 Solution Overview of Post-interaction Assessment

In last section the proposed solution for assessment in pre-interaction time phase has been highlighted. In this section, the proposed solution for assessment in post-interaction time phase is discussed. In Section 4.2, an SLA management framework was proposed in which layer 3 is responsible for the post-interaction assessment on the basis of risk assessment methodologies that are used to determine the performance risk and financial risk.

For post-interaction assessment, the performance of the service provider is first determined considering the two scenarios: consumer's point of interaction is in current timeslot and consumer time of interaction is future timeslot. After determining the performance risk in the given interaction scenario, financial risk of a consumer is then determined. From determined financial risk, financial loss is

computed. Financial loss along with consumer's risk attitude is used in risk based decision support system to give risk based recommendation.

Figure 4.5 depicts the solution overview of the post-interaction assessment.

Steps in Figure 4.5 are explained as follows:

Considers the post-interaction time phase: When a consumer wants to assess the performance of the service provider and its financial impact on the interaction, the TP SLA Manager carries out this job on the behalf of the consumer. The first thing that TP SLA Manager has to determine is the time spot of interaction in post-interaction time phase.

Is the time spot in current timeslot? For performance assessment, the time spot of interaction may be in current timeslot or in future timeslot. Based on the scenario in which the time spot of interaction is, two different paths are followed for performance assessment. Next, the pathway for current timeslot is discussed first.

Obtain run-time service parameters: For performance assessment in current timeslot, this step consists of process to obtain run-time service parameters for the performance criteria set as SLOs.

Compare with SLOs: Run-time performance parameter values obtained in last step are compared with threshold set as SLOs.

Performance assessment in future timeslot: For performance assessment in future timeslot, past service deviation levels are needed to predict future service deviation levels.

Calculate performance risk: Based on the pathways for current or future timeslots as discussed above, the service deviation levels are determined using performance risk process. Further discussion of performance risk is given in Section 4.5.1.

Calculate financial risk: Based on service deviation levels, extra resource investment is determined through financial risk process. It also includes determining extra resource investment due to criterion like service migration. Further discussion of financial risk is given in Section 4.5.2.

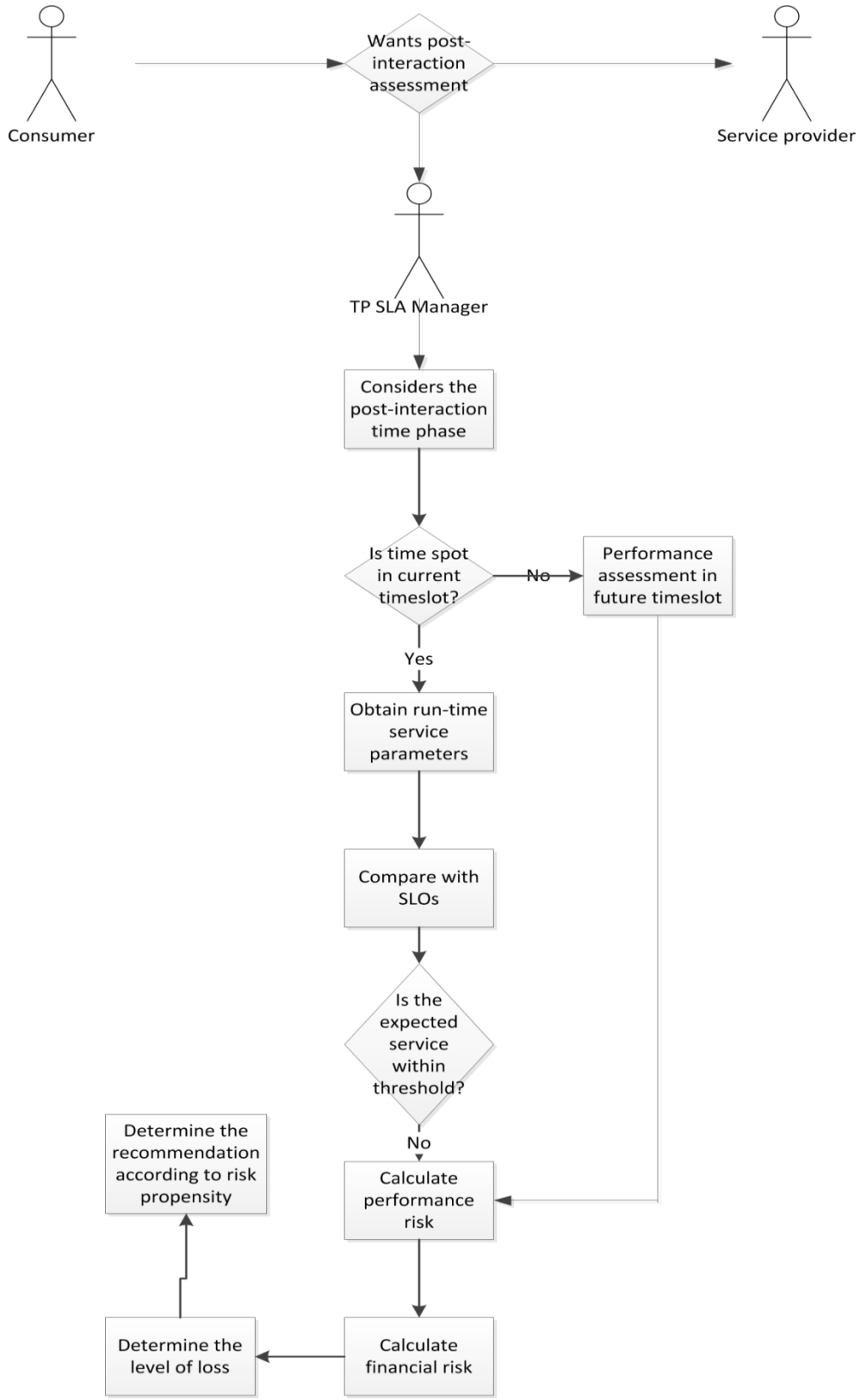

Figure 4.5: Solution overview of the assessment in the post-interaction time phase

Determine the level of loss: Based on what the consumer's expected investment in the interaction was and the extra resources that he needs to invest, financial loss is determined in this step. Further discussion of how to determine level of loss is given in Section 4.5.3.

Determine the recommendation according to risk propensity: Based on the levels of financial loss and risk attitude of the consumer, recommendation is determined. This recommendation assist consumer to make decision whether to proceed or not to proceed in the interaction. Further discussion of risk based recommendation is given in Section 4.5.4.

Based on the steps discussed above, in the following sub-sections, the solution overviews for *performance risk*, *financial risk*, *financial loss* and *risk based decision making* are discussed.

4.5.1 Solution Overview for Determining Performance Risk in an Interaction

As discussed in the previous chapter, performance risk represents the level of failure in an interaction due to the occurrence of undesired outcomes. In the domain of businesses in cloud computing, it translates to the probability of undesirable business outcomes as a result of the non-fulfilment of a consumer's expectations in an interaction. Steps to determine performance risk are shown in Figure 4.5. Following are the important features of the proposed performance risk assessment methodology to determine the probability of undesirable outcomes:

- The performance risk assessment will take place in the same context that has been set in pre-interaction time phase.
- Consumer expectations are already set in the SLA as Service Level Objectives (SLOs) against which the performance of the service provider is to be assessed. The service provider sets these SLOs as thresholds of the service.
- To measure the performance of the service provider, I propose a term *service deviation level* as the level of the service that deviates from the threshold defined in the SLA. In other words, the service deviation levels indicate the non-compliance of the service provider with the SLA. It is important to measure the

service deviation levels in order to measure the performance of the service provider.

- Interaction time space is already divided into finite number of timeslots as discussed in Section 4.4.
- The consumer should then determine the time spot of the interaction, which is the time of the interaction in the time space at which the consumer wants to make a risk-based decision.
- Based on the time spot, there are two possible scenarios:
  - The consumer point of interaction is in the current timeslot: in this case, the consumer needs to assess the near-to-real time performance of the service provider by comparing the run-time SLA parameters against SLOs given in the SLA to determine the service deviation level.
  - The consumer point of interaction is in the future timeslot: in this case, if the duration of the interaction extends to a future point in time, the consumer has to determine the service deviation level in these timeslots by employing prediction methods based on past service deviation levels. Past service deviation levels are stored in a central information repository. These methods are discussed in chapter 6.
- After determining the service deviation levels in the timeslots of the interaction, it is proposed that the consumer should then determine the service deviation level curve of the interaction, which quantifies the different levels of performance risk in the interaction. This curve represents the different service deviation levels along with their probability of occurrence.

The detail of the methodology by which the consumer carries out the aforementioned steps to determine the performance risk of the service provider is presented and consequently, determines the service deviation level curve of the interaction in the various time scenarios.

4.5.2 Solution Overview for Determining Financial Risk in an Interaction

After determining the performance risk, the financial risk has to be ascertained. As discussed in the previous chapter, financial risk represents the financial consequences that could be experienced by the service user as a result of the failure of the

interaction. In the cloud computing environment, it represents the financial consequences that could be experienced by the consumer as a result of the service deviation levels in the interaction. The important features of the proposed financial risk assessment methodology to determine the financial risk are as follows and depicted in Figure 4.6:

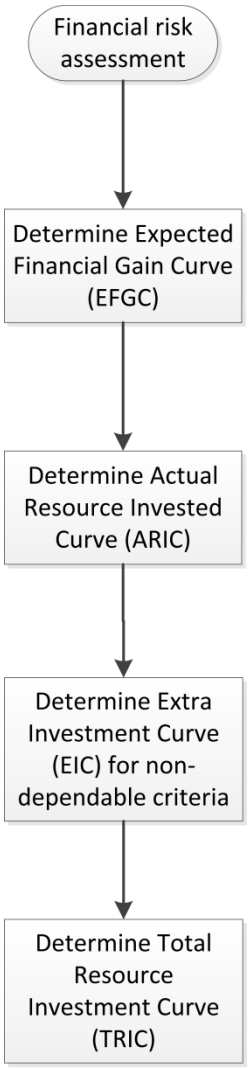

Figure 4.6: Financial risk assessment process

- The financial resources of the consumer are at stake in the timeslots of the time space in which the consumer is interacting or wants to interact with the service provider.
- For financial risk, the consumer should then determine the worth of the financial resources that he is going to invest in the interaction. While doing this, he should

consider the nature of the investment in the various time scenarios of his interaction. These resources will be at stake for the duration of the interaction.

- To achieve this, the consumer determines the Expected Financial Gain Curve (EFGC) which shows the probability of an amount at stake from the consumers' invested resources for the duration of an interaction.
- The consumer then determines the Actual Resource Invested Curve (ARIC) which indicates the required probability of an amount to be kept at stake in the interaction by considering the performance risk of the service provider.
- In addition to the extra resources that a consumer needs to keep at stake in an interaction due to performance risk, he may also set aside the extra amount that he needs to invest for non-dependable assessment. This additional amount is shown by the Extra Investment Curve (EIC). To determine the total financial loss in the interaction due to migration cost, the consumer needs to plot the Extra Investment Curve (EIC).
- The consumer obtains the Total Resource Investment Curve (TRIC) which indicates the total resources including the extra investment amount and the extra migration cost that a consumer needs to keep at stake in the interaction.
- Each consumer has his maximum investment capacity which indicates the amount of loss a consumer can tolerate in an interaction due to performance risk. This amount is indicated as a point on TRIC. The portion of TRIC after that point indicates total loss in the interaction.

  The detail of the methodology by which the consumer carries out the aforementioned steps to determine the financial risk in the interaction is presented in Chapter 7.

4.5.3 Solution Overview for Determining the Magnitude and Level of Loss in an Interaction

As discussed in the previous chapter, it is quite possible there is more than one level of loss in an interaction. A consumer needs to take into account each loss level and its magnitude for risk-based decision making. To achieve this, the consumer needs to determine both level and magnitude of loss. The features of the proposed risk assessment methodology to determine the level and magnitude of loss are as follows:

- The consumer determines the level of loss in an interaction numerically and linguistically.
- For this, the consumer uses the possibility theory to determine the level of loss quantitatively in the interaction.
- The determined level of loss is then transformed to possibility distributions.
- The numerical representation of loss represents the possibility of occurrence of different levels of loss in its defined universe of discourse that could be present in interacting with the consumer.
- To determine the level of loss linguistically and semantically, the consumer utilizes a fuzzy inference model. The universe of discourse of *loss* in which levels are ascertained in linguistic terms are defined semantically. These levels are then utilized in the next step of risk-based decision making.

The detail of the methodology by which the consumer carries out the aforementioned steps to determine the level and magnitude of loss in the interaction are presented in Chapter 7.

4.5.4 Solution Overview of informed Risk-based Decision Making

As discussed in Chapter 2, for risk based decision making, consumer's risk attitude plays an important role and this also holds true for cloud computing environment. Since a consumer may have different risk attitudes and his current risk attitude needs to be considered along with current level of financial loss to determine the risk based recommendation. The features of proposed risk based decision support system are as follows as shown in Figure and as shown in Figure 4.7:

- Consumer needs to obtain the level of loss in the interaction as discussed in last section.
- Once the consumer identifies the level of loss in the interaction, he then combines his risk attitude with the loss level to determine the risk-based recommendation.

For this, a fuzzy logic-based approach with two input variables is proposed, namely, *risk attitude* or *risk propensity* and *loss* and one output variable *recommendation*

which represents the risk-based recommendation to proceed or not to proceed in the interaction. The detail of this methodology is discussed in Chapter 8.

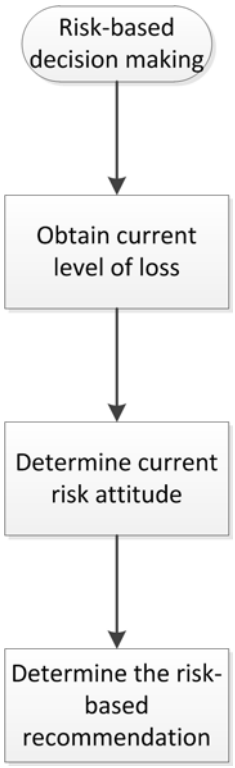

Figure 4.7: Risk-based decision making process

4.6 Conclusion

In this chapter, the solution for the problem that is addressed in this thesis was proposed. The architectural framework that was used to address the research issues that were raised in previous chapter was introduced, which will lead us to solve the defined problem of the thesis. In the next chapters, the detailed solution for each research issue discussed in this chapter is given.

4.7 References

[1] Daniel A. Menascé and Virgilio A. F. Almeida, *Scaling for E-Business: Technologies, Models, Performance, and Capacity Planning*: Prentice Hall, 2000.

[2] A. Dan, D. Davis, R. Kearney, A. Keller, R. King, D. Kuebler, H. Ludwig, M. Polan, M. Spreitzer, and A. Youssef, "Web services on demand: WSLA-driven automated management", *IBM Systems Journal,* vol. 43, pp. 136-158, 2004.

[3] D. Hai, F. K. Hussain, and E. Chang, "A Service Search Engine for the Industrial Digital Ecosystems", *Industrial Electronics, IEEE Transactions on,* vol. 58, pp. 2183-2196, 2011.

[4] M. R. Berthold and D. J. Hand, Ed., *Intelligent data analysis: an introduction*. Springer, 2003.

[5] T. Takagi and M Sugeno, "Fuzzy identification of systems and its applications to modeling and control", *IEEE transactions on Systems, Man, and Cybernetics,* vol. 15, pp. 116-132, 1985.

CHAPTER 5

TRUST AND REPUTATION ASSESSMENT IN PRE-INTERACTION TIME PHASE

5.1 Introduction

In the last chapter, the solution overview of the research issues that were identified in Chapter 3 was highlighted. The broad overview of the proposed solution consists of pre-interaction and post-interaction time phase assessments to solve the issues of SLA management. In this chapter, a pre-interaction assessment solution is discussed in detail, which is the trust or reputation evaluation of a service provider. As mentioned in Chapter 4, the purpose of providing a solution to the issue of pre-interaction service provider selection is to enable the consumer to select a cloud service provider on the basis of its trustworthiness or reputation.

The solution for the pre-interaction assessment consists of two parts, namely a direct interaction scenario and an indirect interaction scenario between a service consumer and a service provider. Figure 5.1 depicts the scenarios for the pre-interaction time phase.

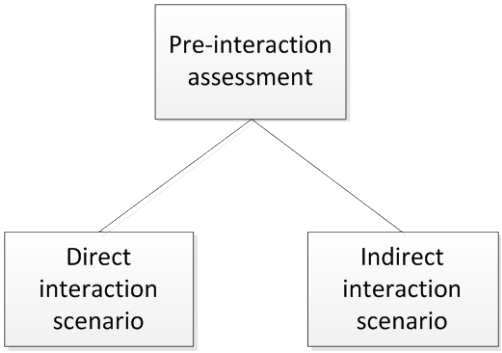

Figure 5.1: Different scenarios of pre-interaction time phase assessment

The main aim of presenting a solution in this chapter is to develop a decision making system that enables a consumer to make an informed decision about the selection of a service provider. This decision making system consists of two sub-systems: a system to determine the trustworthiness of a service provider when there is direct past-interaction history and a system to determine the reputation of a service provider when there is no direct past interaction history. Both these systems are developed in

three phases, namely the *input phase*, *computation phase* and *analysis phase*. In the *input phase,* the input factors required to calculate either the trust or reputation value are defined. In the *computation phase,* the algorithms to compute trustworthiness and reputation are explained. In the *analysis phase,* the systems are analysed for their validity using different numerical analysis techniques. These phases are depicted in Figure 5.2.

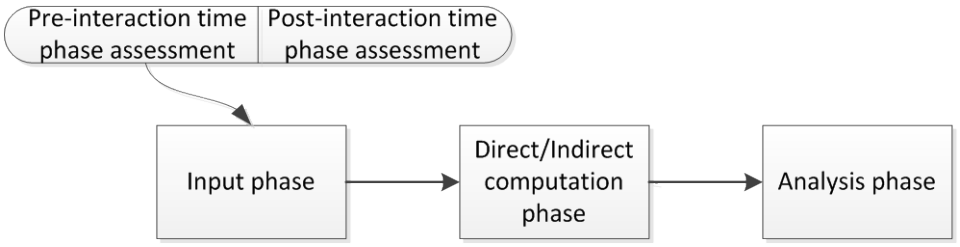

Figure 5.2: Phases in developing trust and reputation systems

This chapter is organized as follows: Section 5.2 provides information on the components for the trustworthiness evaluation methodologies. Section 5.3 discusses the trustworthiness evaluation methodology for direct interaction scenarios. Section 5.4 discusses the factors affecting reputation for indirect interaction. Section 5.5 discusses the methodology for reputation evaluation for indirect interaction scenarios. Section 5.6 details the fuzzy inference system for reputation determination. Section 5.7 discusses the results. Section 5.8 concludes the chapter.

5.2 Overview of Direct and Indirect Pre-interaction Scenarios Assessment

In this section, a brief overview of pre-interaction assessment in both these scenarios is discussed.

5.2.1 Direct Interaction Scenario Assessment

As discussed in Chapter 4, direct interaction is based on direct past experience of a consumer with a service provider. In this case, a consumer, using this past experience, determines the *trustworthiness* of a service provider. Let us recall from Chapter 3 that *trustworthiness* is defined as the capability of a service provider (in terms of belief) to complete a certain task. While determining trustworthiness, the past *timeslots* of a consumer's interaction with the service provider are considered to determine its trust value.

- ***Inputs for a trustworthiness determination system:*** As discussed in Chapter 3, in order to calculate trustworthiness, both context and time of interaction should also be considered. The trustworthiness of a service provider varies with time. Therefore, while determining the trustworthiness of a service provider in the case of direct interaction, the time of interaction should be given due importance. Two input factors that need to be considered for trustworthiness determination are *trust values* and the *timeslot* of an interaction.
- ***Information repository:*** In open-standard-based environments, such as the cloud service environment, the trust values assigned to a service provider along with their context and time of interaction should be retained so that they can be used or shared later to determine trustworthiness or reputation. Trust databases have been suggested for this purpose [1]. In the case of direct interaction, a consumer's trust database is explained with an example.

  Example: Consider a consumer 'A' has direct interaction with service providers 'B', 'C' and 'D' in different contexts. On the basis of experience of interacting with service providers 'B', 'C' and 'D', the consumer 'A' records the timeslot of interaction and trustworthiness values that are assigned to the service provider for each interaction in the trust database, as shown in Table 5.1. Trustworthiness values are assigned on the basis of how good or bad the experience of consumer is with the service provider and in thesis trustworthiness is represented on the scale of 0 to 5 where 0 represents not trustworthiness at all and 5 represents very trustworthy. The data in Table 5.1 from the trust database of consumer 'A' is helpful if he wants to interact with the service providers in the future. Based on the context and timeslot of interaction, consumer 'A' can assign weights to the timeslots of service providers 'B', 'C' or 'D' using the logistic decay function that is discussed in Section 5.5.1.

Table 5.1: An example of the trust database of Consumer 'A' (direct interaction)

Service provider	Context	Timeslot	Trustworthiness Value
B	Video conferencing	2010	4
B	Video conferencing	2011	3.5
C	VoIP	2010	5
D	Video conferencing and VoIP	2011	3.5

These values are used by the TP SLA Manager where the trustworthiness determination method is employed. Using the first tuple data from Table 5.1, the scenario of determining the trustworthiness of service provider 'B' in 2013 is depicted in Figure 5.3.

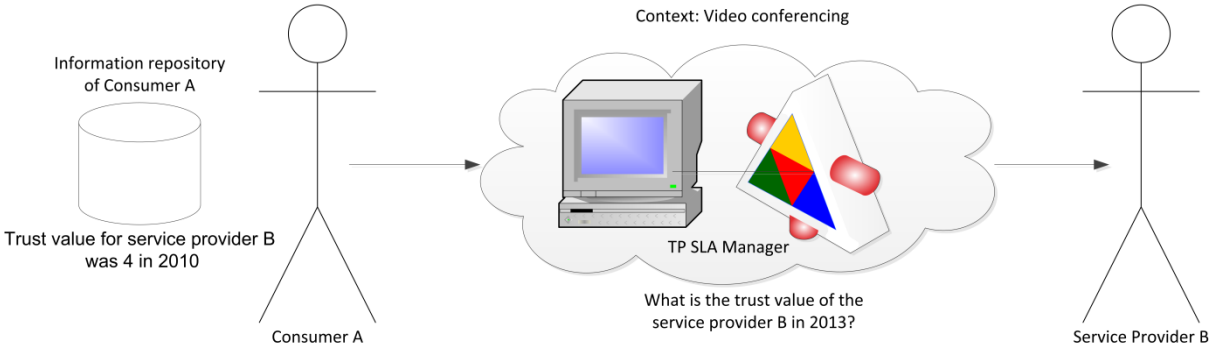

Figure 5.3: Trustworthiness determination example

Figure 5.3 shows the TP SLA Manager in proposed pre-interaction time assessment uses the information repository (trust database) of consumer A to determine the trustworthiness of service provider B in 2013. To determine the weight of timeslot 2010, logistic decay function has been used in this thesis which was discussed as a proposed solution for pre-interaction time assessment in the previous chapter.

5.2.2 Indirect Interaction Scenario

As discussed in Chapter 4, indirect interaction is based on the indirect experience of a consumer with a service provider. The consumer determines the *reputation* of a

service provider in this case. Let us recall from Chapter 3 that *reputation* is defined as the aggregation of all the recommendations (trustworthiness values) from the Recommending Users (RUs) about a service provider. Similar to trustworthiness, reputation is also determined in past timeslots of the pre-interaction phase of the time space of an interaction.

- ***Inputs for a reputation determination system:*** As discussed in Chapter 3, to calculate reputation, the recommendation opinions of the recommending user along with the time when the recommendation was given and the credibility of this recommendation opinion should be taken into account. The time factor is important because the recommendation opinion varies with time. Three input factors that are considered for reputation determination are the *recommendation opinion,* the *credibility* of the RU in given recommendations and the *time delay.* These input factors were defined in Chapter 3. In this thesis, the recommendations for the service provider from other users are represented on a scale of 0 to 5, where 0 indicates the lowest (worst) recommendation and 5 represents the highest (best) recommendation. The representation of the input variable credibility is similar to the representation of the recommendation opinion. The representation of time delay is the same as in the previous section.

- ***Information repository:*** For indirect interaction, a consumer, through the TP SLA Manager, sends a reputation query about a service provider's reputation to a network of RUs. Only those *recommending users* who have interacted with the service provider in the same context in which the consumer wants to interact with the service provider reply to reputation query. The reputation query reply from each RU contains the trust value assigned to the service provider on the basis of its past interaction with that service provider (recommendation opinion) and the time when this interaction has occurred (time delay). The RU makes use of its trust database to retrieve and share this information. When the TP SLA Manager receives the reputation reply from each RU, it stores these values in a database called *reputation database*. Assuming that the consumer has consulted these RUs in the past, the

consumer has the *credibility* value of each RU. The format of the reputation database is then composed of the following fields:

RU – recommending user who participated in the reputation query
RO – recommendation opinion given by an RU about the service provider
TD – time elapsed since the interaction of the RU with the service provider with respect to the current timeslot
CR – the credibility of the opinion of an RU.

Example: Consider that a consumer 'A' wants to select service provider 'B' for his business and he hasn't had any previous interaction with these service providers. Consumer 'A' seeks the opinions of the Recommending Users in this case. The reputation database in this case consists of the following values:

Table 5.2: An example of the reputation database of Consumer 'A' for the RUs (indirect interaction)

RU	Context	Time delay	Recommendation Opinion (trust value)	Credibility
RU1	Video conferencing	1	2.3	5
RU2	Video conferencing	1.8	5	4.3
RU3	Video conferencing and VoIP	3.4	4.5	2.5
RU1	Video conferencing and VoIP	1.3	2	3.5

Figure 5.4 depicts the reputation determination where consumer 'A' wants to determine the reputation of an unknown service provider 'B' through RU1. The credibility, recommendation opinion and time delay values have been used from the respective information repository. TP SLA Manager uses these values to determine the reputation of service provider 'B', using the reputation determination method which is discussed in detail in the next sections.

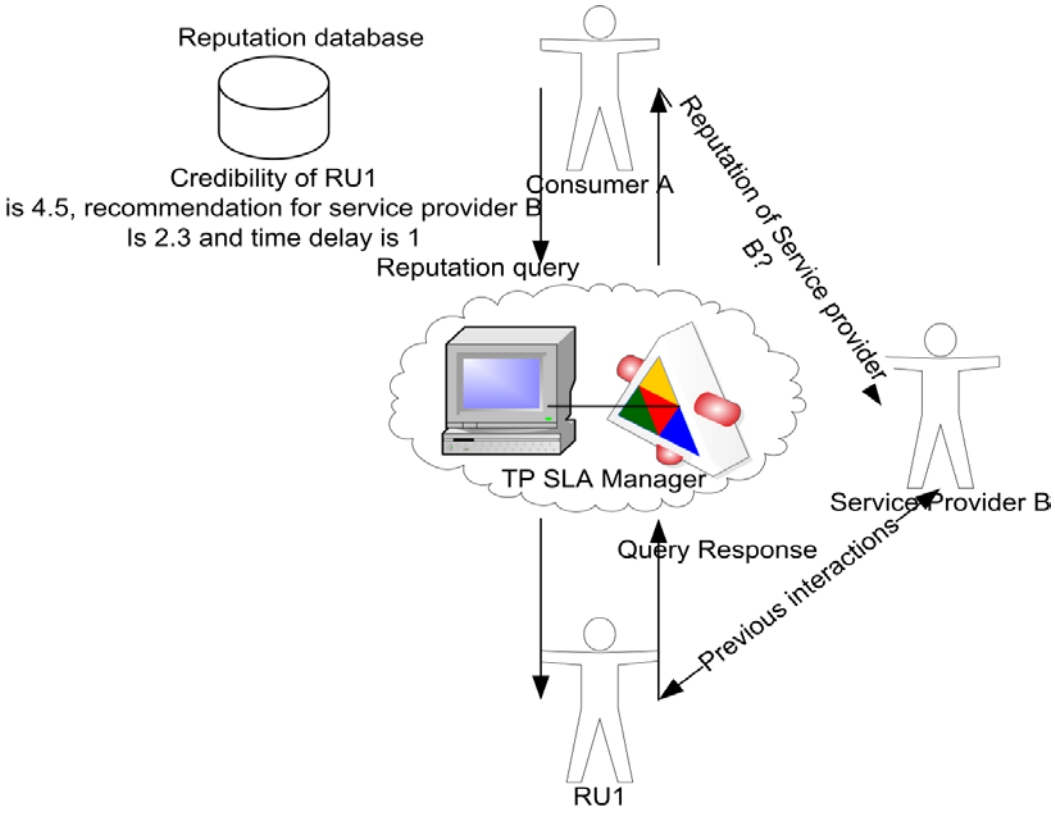

Figure 5.4: Reputation determination example

In the next section, the conceptual framework for trustworthiness and reputation determination is discussed.

5.3 Conceptual Framework

Direct and indirect interaction scenarios were discussed in the previous section. These scenarios led to development of two systems: a *trustworthiness determination system* and a *reputation determination system*. The trustworthiness determination system is based on a deterministic approach which is discussed in the next sub-section and the reputation determination system is based on a fuzzy logic-based approach which is discussed in Section 5.6. To carry out the pre-interaction assessment, a computational framework is depicted in Figure 5.5.

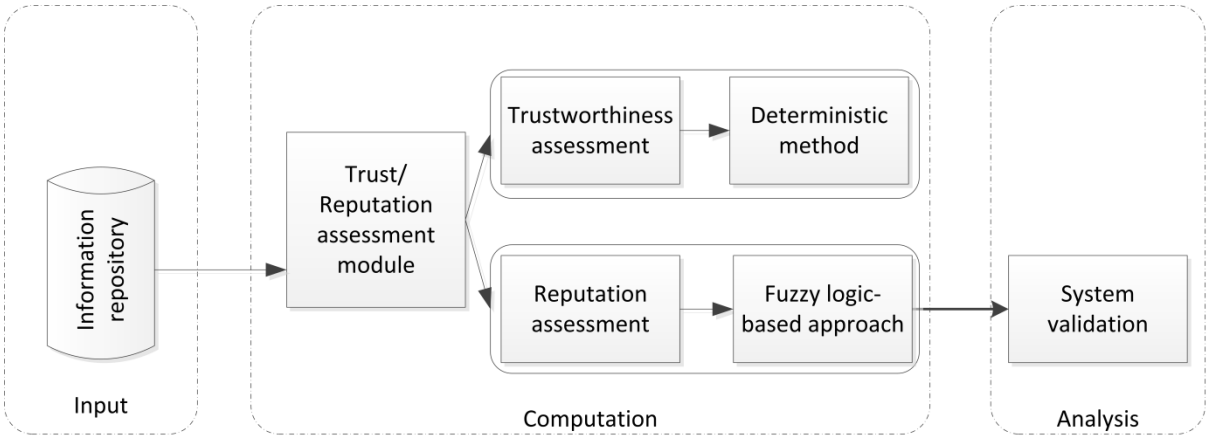

Figure 5.5: Conceptual framework for pre-interaction time phase assessment

The conceptual framework consists of three phases, namely the *input phase*, *computation phase* and *analysis phase*. Each of these phases is described below:

5.3.1 Input Phase

The input phase for the development of the trustworthiness determination system consists of identifying the inputs such as *trust values* and *timeslot* from the information repository for computation. These input factors are discussed in detail in Section 5.5.

The input phase for the development of the reputation determination system consists of identifying the inputs such as *recommendation opinions*, *credibility* and *time delay*. These input factors are discussed in detail in Section 5.6.

5.3.2 Computation Phase

As stated in the previous section, the two systems that are being developed are the *trustworthiness* and *reputation determination systems*. The details of both these systems are described as follows:

- ***Trustworthiness determination system:*** As stated in Chapter 3, in order to determine the trustworthiness of a service provider, an appropriate methodology is needed that can determine the time delay. In Chapter 4, a *logistic function* is proposed for this purpose. It is important to consider time delay as the capability of the service provider may change over time. To capture this variability, the

consumer assigns a weightage to the timeslots in which he is assessing the service provider. In general, the timeslots nearest the interaction have more importance than distant timeslots.

Using logistic decay, the relative importance of each timeslot in the pre-interaction time phase is given by assigning an appropriate weight to each timeslot which results in the rate of decay or time delay. Trust values from the consumer's information repository are then used to determine the trustworthiness value according to the rate of decay.

- ***Reputation determination system:*** The reputation determination system is a little more complex than the trustworthiness determination system due to the additional input parameter, *credibility*. The computational approach proposed in this thesis for reputation determination is the fuzzy logic-based approach which was discussed briefly in the previous chapter. The system development starts with the *fuzzification of the input factors* followed by choosing an appropriate *fuzzy inference method*. Then the *fuzzy rule base* for the reputation system is defined which helps in the creation of an *initial Fuzzy Inference System (FIS).* To train this initial FIS using the training data, a *neuro-fuzzy-system-based soft computing approach* such as *Adaptive-Network-based Fuzzy Inference System (ANFIS)* is used. Once a trained FIS is obtained, it can then be used to determine the reputation value by inputting the recommendation opinion, credibility and time delay.

5.3.3 Analysis phase:

After the systems have been built, the next phase is to analyse them to validate that these systems can be used for the selection of a service provider. Validation includes different numerical techniques which are discussed in detail in later sections.

5.4 Flow of Activities

In this section, an overview of the steps involved to determine the *trustworthiness* or *reputation* of a service provider using the proposed systems is discussed followed by further discussion of the systems detail in the following sections.

5.4.1 Direct Interaction Scenario:

The flow of activities in determining the trustworthiness of a service provider is depicted in Figure 5.6.

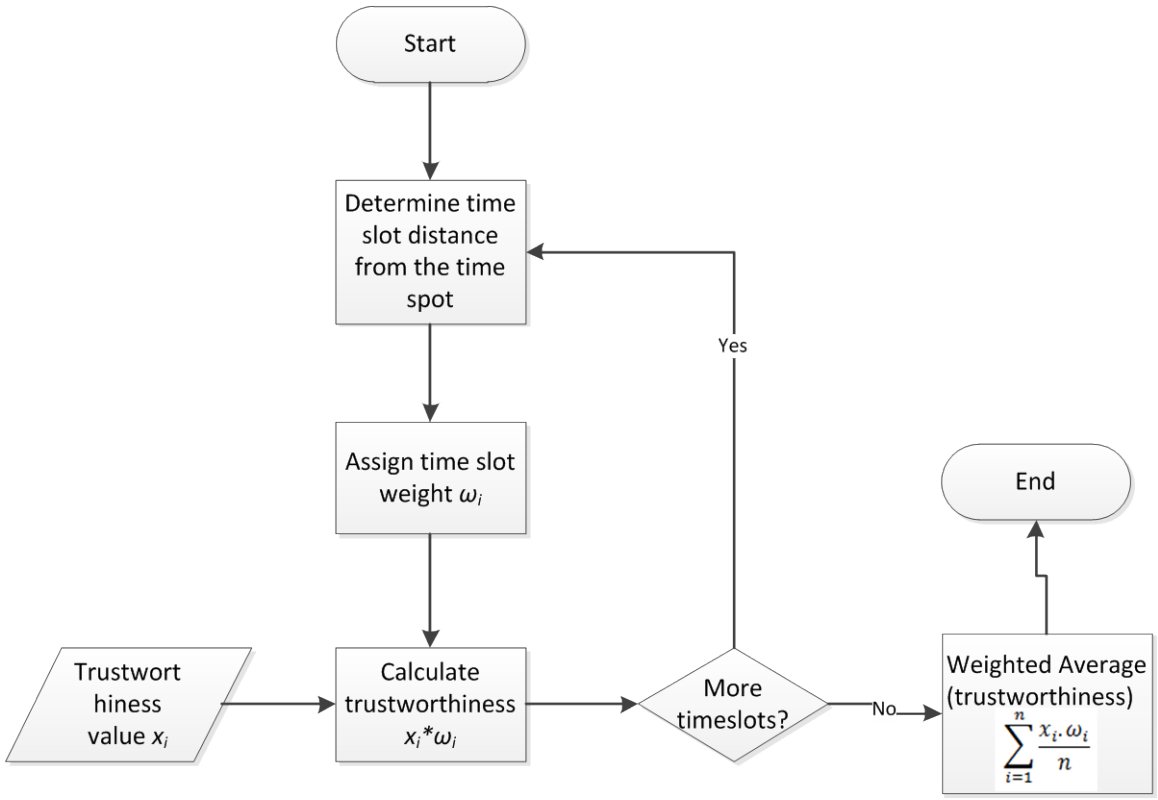

Figure 5.6: Trustworthiness evaluation in pre-interaction time phase

Determine timeslot distance: The role of this step is to determine how far the timeslot is from the time spot. Here time spot refers to the current time in focus.

Assign timeslot weight: The role of this step is to assign a weight to a timeslot based on its distance from the time spot. In other words, the weight is proportional to the distance of timeslots.

Trustworthiness input: The role of this step is to provide a trust value input from a database. This database contains the trust values based on the previous interaction and are used to calculate trustworthiness (e.g. of the service provider). In this trust determination system, trust is represented by an integer number to score the trust value.

Calculate trustworthiness: This is the core step in which the actual trustworthiness of a service provider is calculated. The calculated trustworthiness value is computed based on the pre-determined interactions and assigned timeslot weight.

Aggregate trustworthiness: For each timeslot, the trustworthiness calculation is done. After this, the weighted average is used to determine the final trustworthiness value. The detail of this process is given in Section 5.5.

5.4.2 Indirect Interaction Scenario:

The flow of activities in determining the reputation of a service provider is depicted in Figure 5.7.

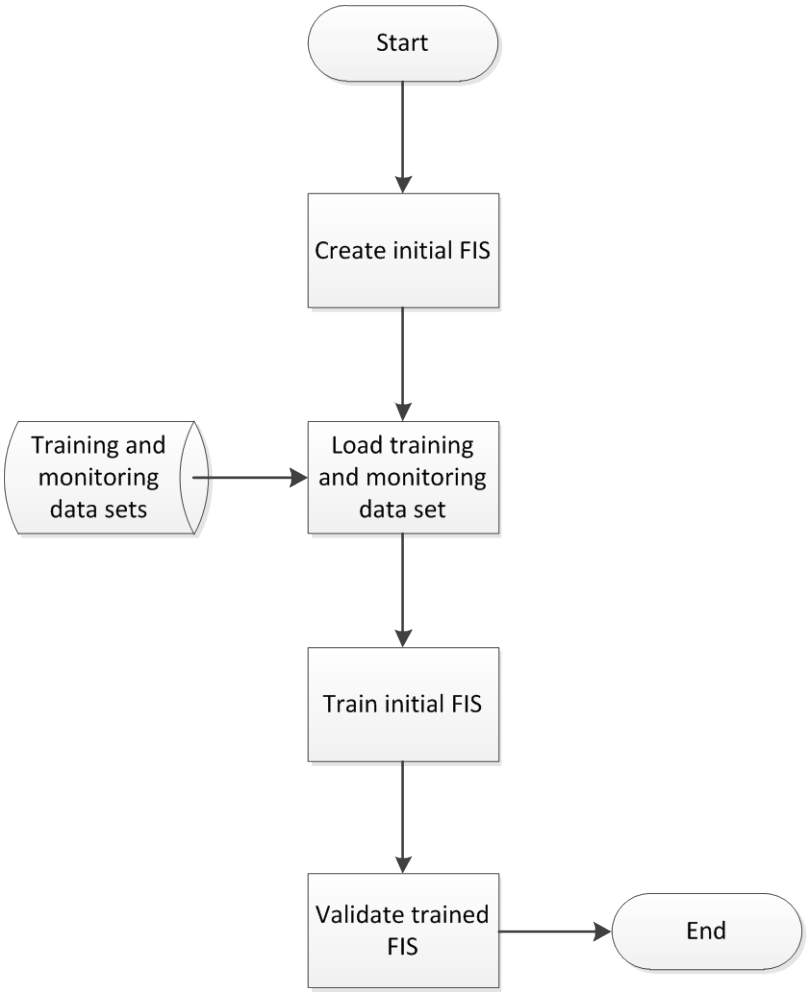

Figure 5.7: Reputation evaluation in the pre-interaction time phase

Create initial FIS: The purpose of this step is to create an initial Fuzzy Inference System (FIS). This initial structure is necessary for further processing. The initial structure consists of fuzzified input variables and a fuzzy rule-base. The structure can then be used later in the training process.

Load training and monitoring data set: This role of this step is to load the training data set from the training and monitoring data set. When the training data set is prepared, it is composed of training data and monitoring data. Both of these types of data are required in training the initial FIS.

Train initial FIS: This role of this step is to carry out the actual training process using a *neuro-fuzzy-based soft computing approach* called *Adaptive-Network-based Fuzzy Inference System (ANFIS).* In this step, the training data is fed into the initial FIS and the training process is executed. The training process stops when the best generalization capability of the reputation determination system is obtained. The output of this process is trained FIS.

Validate trained FIS: This step is the final step and it validates the system before it can be used to determine the reputation of the service provider. Validation techniques are discussed later in this chapter.

5.5 Direct Interaction Scenario-based Trust Determination

In this section, the detail of the trust determination system by the direct interaction scenario is discussed. The system development starts with the input factors followed by an explanation of the system development process and ends with the system analysis. As discussed previously, the system development for the indirect interaction scenario is based on the deterministic approach. Figure 5.8 depicts the inputs and outputs of the trustworthiness determination system.

As mentioned in Figure 5.6, the series of steps in the process of direct interaction-based trustworthiness assessment are to:

1- determine the distance from the timeslot and determine their weights
2- calculate the trust value according to the timeslot weight
3- aggregate them to find the final trustworthiness value.

Step 1 is explained in Section 5.5.1, step 2 is explained in Section 5.5.2 and step 3 is explained in Section 5.5.3.

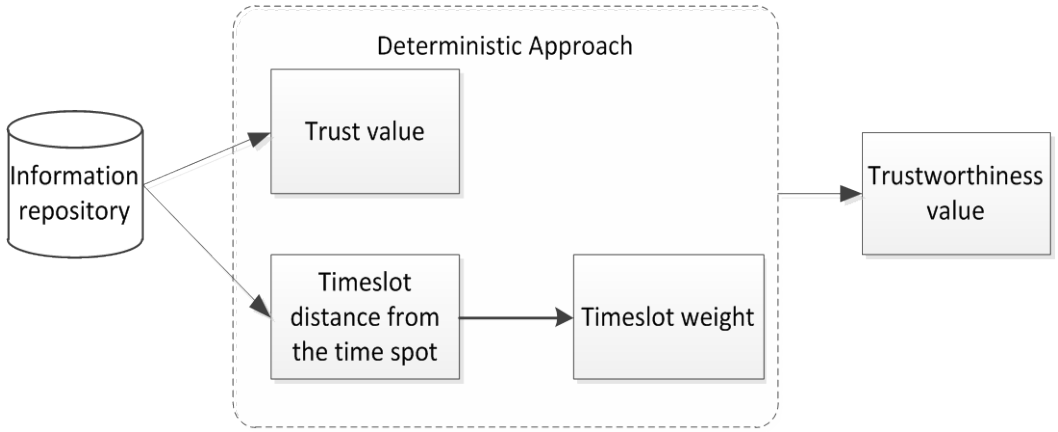

Figure 5.8: Deterministic-based trustworthiness determination model

5.5.1 Determining the Timeslot Distance and Assessing Timeslot Weight

The first step in this process is to determine the timeslot distance from the time spot at which the trust-based decision has to be made. One way of achieving this is to assign a weight to each timeslot according to its nearness to the time spot. A maximum value of 1.0 is assigned to the most recent timeslots with respect to the time spot, the weight of the timeslots then successively decreases until it reaches its minimum value. The trust values in the most recent timeslots have a greater influence on the final trustworthiness value than the trust values in older timeslots. As discussed previously, the logistic decay function is used in this thesis to represent the decay of time [2]. To use this function, consider n timeslots t_1, t_2, …, t_n, then the corresponding weights for each timeslot t_i is given by:

$$W_i(\Delta t_i) = A + \frac{K - A}{(1+e^{-B(\Delta t_i - M)})^{1/2}} \quad (1)$$

where $\Delta t = t_c - t_i$, is the time interval between the interaction time spot t_c and the time spot in consideration of t_i.

The properties of this logistic decay function are controlled by the constants A, K, B, and M, where A is the lower asymptote; K is the upper asymptote; B the growth rate and M is the time of maximum growth. If the first five timeslots have $W_i \approx 1$ and if

it declines to 0.5 between 5th to 20th timeslots, the decay curve obtained is shown in Figure 5.9

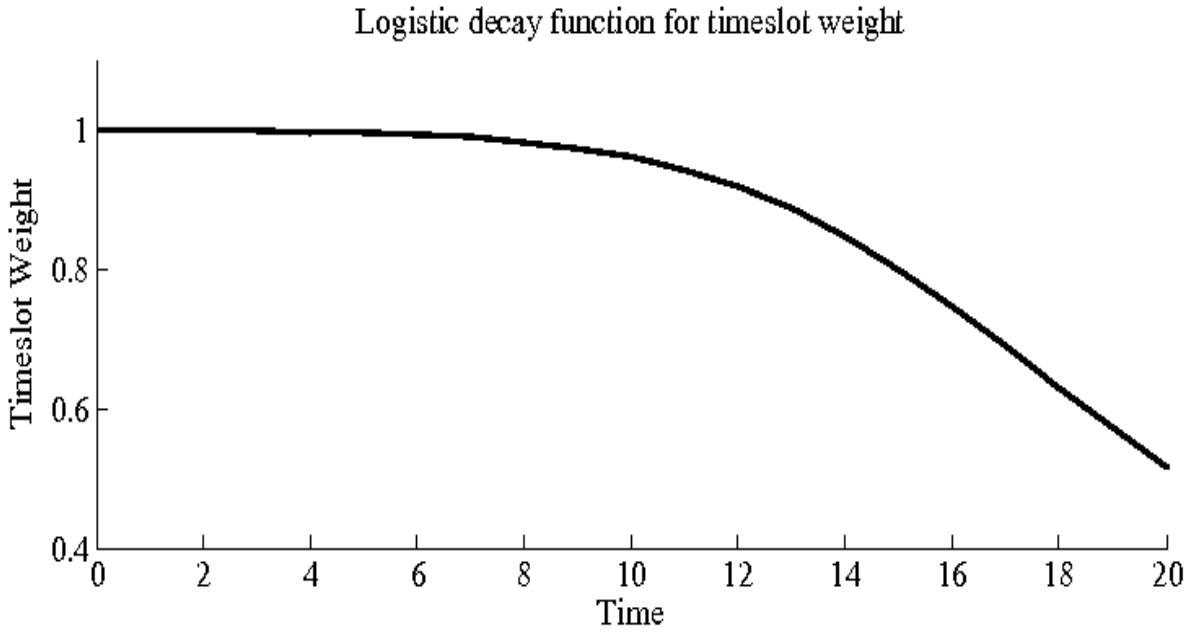

Figure 5.9: Logistic decay function for timeslot weights

For this curve, *A*=1; *K*=0.2; *B*=0.25 and *M*=15, which gives a weight to each timeslot in such a way that the most recent timeslots have more weight and the distant timeslots have a lower weight. The first few timeslots are closest to the current time and therefore have maximum weight ($w_t = 1$), then the weight decreases for subsequent timeslots and remains constant after reaching a minimum value (0.2, determined by the constant *K* in Equation 1). Once the weight specifying the importance value has been determined, the next step is to determine the trust value in that timeslot.

5.5.2 Determining Trust Values According to Timeslot Weight

In this step of the process, the timeslot weight is multiplied by the trust value based on consumer's past experience with the service provider. This trust value is retrieved from the consumer's information repository. The weighted trust value is calculated as:

$$\text{weightedTrust} = t.\omega$$

where t is the trust value in a given timeslot and ω is the weight assigned to this timeslot.

The weighted trust value represents the relative importance to the trust value according the timeslot weight that is, a decrease in timeslot weight decreases the trust value accordingly. Once all the weighted trust values for all the timeslots in consideration are determined, the next step is to calculate the final trustworthiness of the service provider, which is discussed in next section.

5.5.3 Trustworthiness Calculation

In this step of the process, the weighted average of all the weighted trust values is calculated. The weighted average as compared to the *mean* of the trust values gives the relative importance of the trust values according to the timeslot weights. By combining the steps from the last sections, the final trustworthiness of the service provider is calculated as:

$$\text{Trustworthiness} = \sum_{i=1}^{m} \frac{\text{trust}_i.\omega_i}{m}$$

where $trust_i$ is the trust value, ω_i is the weight of timeslot i and m is the total number of the timeslots under consideration. For example, if the consumer considers three recent timeslots all with a weight 1 and with trust values of 3.5, 3 and 2.7 respectively, the trustworthiness of the service provider is then calculated as:

$$\text{Trustworthiness} = \frac{3.5\times1+3\times1+2.7\times1}{3} = 3$$

Using this process, the service user can determine the trustworthiness of the service provider if it has a direct past-interaction history in each timeslot in consideration. In the case where there is no direct past interaction history, then it can use the indirect interaction scenario to calculate reputation, which is discussed in the next section.

5.6 Indirect Interaction Scenario-based Reputation Determination

In this section, the detail on the development of the reputation determination system is discussed. The system development starts with the input factors followed by an explanation of the system development process and the system analysis. As discussed previously, the system development is based on the soft computing-based

approach. An overview of the reputation determination-based approach is given in Figure 5.10.

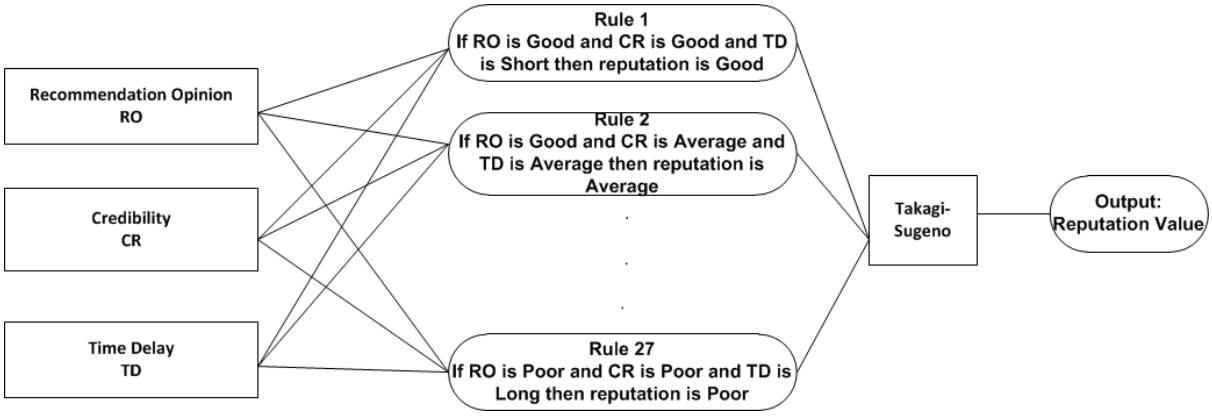

Figure 5.10: Soft computing-based reputation determination model

As shown in Figure 5.7, the series of steps in the process of indirect interaction scenario-based reputation determination are:

1- Create the initial FIS by considering the *input variables*, *fuzzy linguistic control rules*, *the fuzzy inference method* and *the output value* as depicted in Figure 5.10.
2- Load training and monitoring data sets after the initial FIS has been generated.
3- Train the initial FIS on the basis of loaded data. Steps 2 and 3 are parts of the *experiment* process which explains the training of the initial FIS.
4- Validate trained or obtained FIS. This step is carried out under the analysis of the produced system.

Each of these steps and the processes to carry out these steps are discussed in detail in the following sections.

5.6.1 Input Variables

In this third section of the identified input factors for reputation determination namely, Recommendation Opinion, credibility and time delay are discussed in detail. These three input factors should be combined to determine the reputation value of the

service provider. In other words, the recommendation opinion from the RUs should not be taken as the final reputation value of the service provider unless it is weighted against the credibility and time delay factors. This will give a more accurate contribution by the RU about the reputation of a service provider.

It should also be considered that the aforementioned input variables have their numeric as well as semantic representations due to their fuzzy nature [1]. Therefore, using a fuzzy logic-based approach where these variables can be represented semantically as well as numerically will increase the robustness of the service provider selection process in cloud computing. The first step towards building a neuro-fuzzy-based soft computing approach is to convert input variables into fuzzy variables.

5.6.1.1 Fuzzification of Input Factors

To this point, this thesis has discussed the factors that affect reputation which can be used as input to the system. As a first step, the input variables need to be fuzzified. Fuzzification is the process of converting crisp quantities to fuzzy values [3]. Fuzzy values are formed by identifying some of the uncertainties in the crisp values. Fuzzy membership functions represent the conversion of fuzzy values from crisp quantities. In our fuzzy system, the three input fuzzy variables are: recommendation opinion (RO), credibility (CR) and time delay (TD), and output is the reputation contribution RU_i of recommending user *i*.

The fuzzy membership function is defined on a fuzzy set. The input variables RO and CR can be categorized using linguistic terms such as ‘poor’, ‘average’ and ‘good’ [1]. Therefore, fuzzy sets that are used in this thesis for variables RO and CR are [Poor, Average, Good]. Similarly, fuzzy sets for TD variable are [Long, Average, Short]. The fuzzy membership function can be represented graphically by different shapes. In my approach, I consider each input variable is represented using G-Bell shape curves where the G-Bell is characterized by parameters a, b, c as illustrated in Figure 5.11.

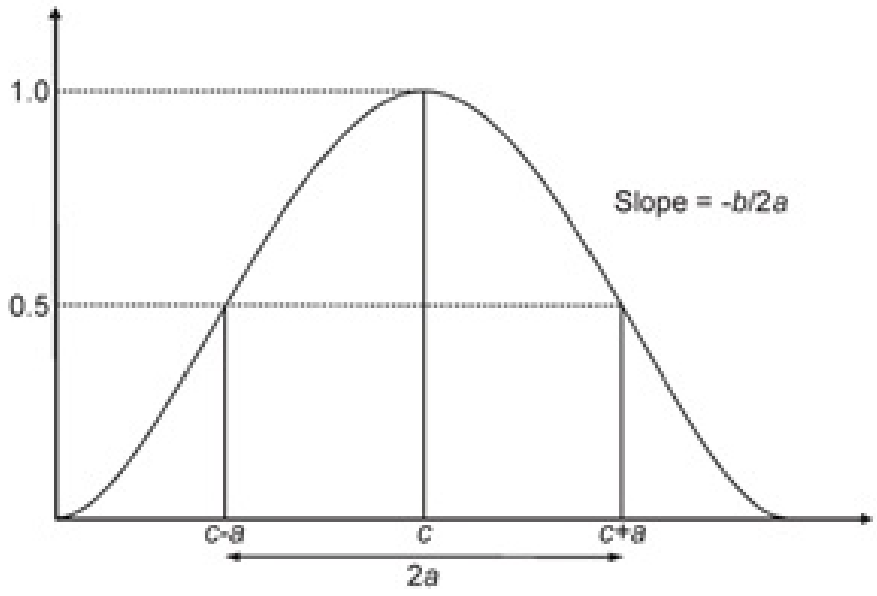

Figure 5.11: Generalized Bell Curve and Parameters [reproduced from [1]]

Since the proposed ANFIS-based system for reputation determination is trained using the training (numerical) data, it is beneficial to extract models from numerical data which represent the behavioral dynamics of the reputation system [4]. Such a system can be used to predict the behavior of the underlying system.

The G-Bell shape curves were chosen because of their smoothness and to the point notation to specify fuzzy sets. These curves have the advantage of being smooth and nonzero at all points. On the basis of the G-Bell shape, membership functions for input variable are shown in Figure 5.12.

The representation of the input variable credibility is similar to the representation of the recommendation opinion, as shown in Figure 5.12. The representation of time delay is given in Figure 5.13.

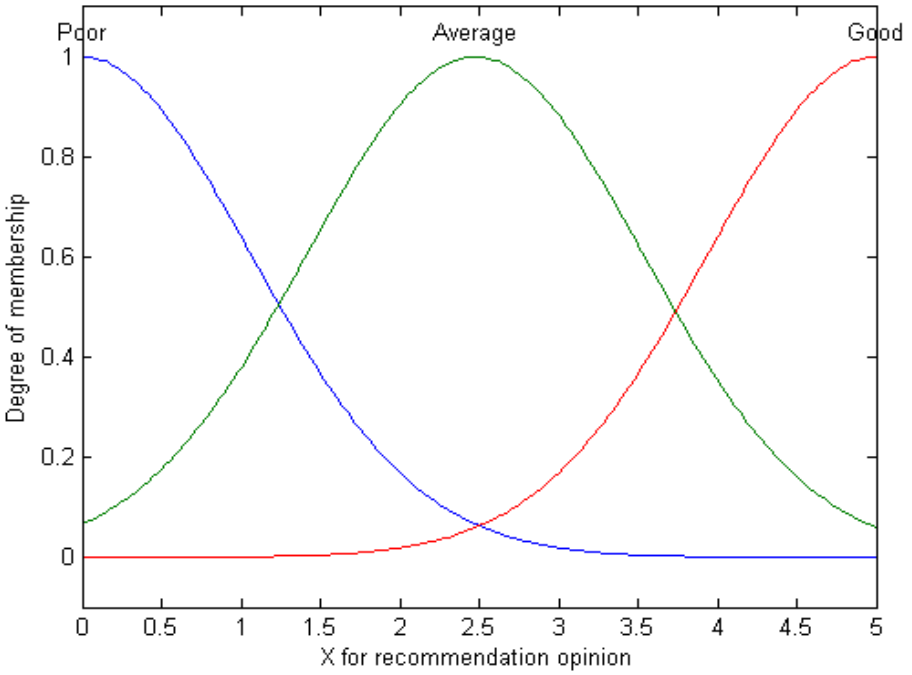

Figure 5.12: Membership functions for recommendation opinion

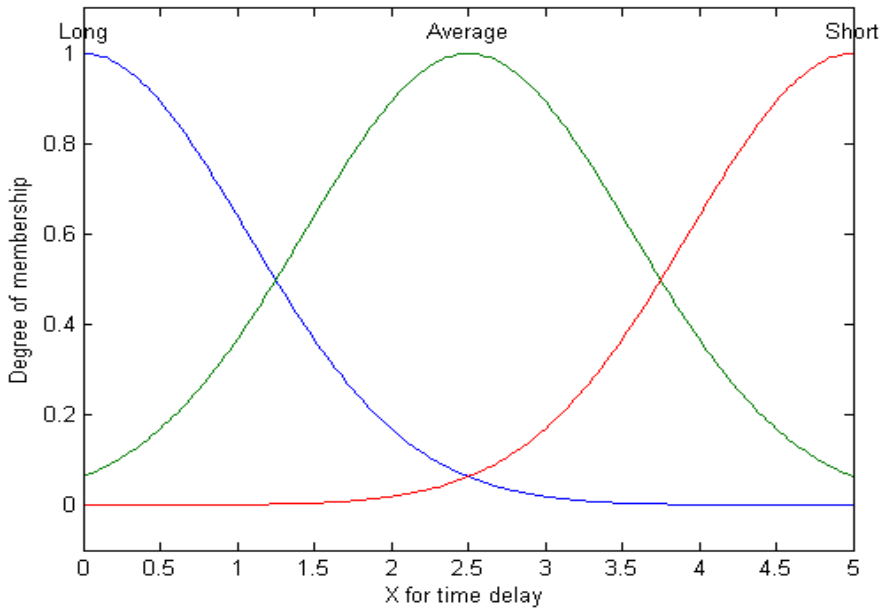

Figure 5.13: Membership functions for time delay

Based on these membership functions, the input variables can be fuzzified. After the fuzzification of the input variables, a fuzzy inference method is needed to map the input variables to the output value. In the next sub-section, the fuzzy inference method for the reputation determination system is discussed.

5.6.2 Fuzzy Inference Method

In the previous section, a fuzzy model to represent each of the reputation factors that were considered important in specifying reputation was discussed. However, a method is still needed to map the input space corresponding to these factors to the output space (reputation). In the previous sub-section, it was highlighted that the input space will consist of recommendation opinion (RO), credibility (CR) and time delay (TD) and that the output space will consist of a single dimension, which is the reputation contribution RU_i of recommending user *i*. This is done by a fuzzy inference method that maps from an input space to an output. For this mapping, fuzzy linguistic control rules are used. A typical fuzzy linguistic control rule is as follows:

IF *X* THEN *Y*.

This rule states that if the condition *X* is satisfied, then perform action *Y*. Conditional part *X* of the rule is also called antecedent and *Y* is called consequent. However, it should be noted that the antecedent part may consist of a composite condition joined together by the connectives AND and OR. There are two well-known fuzzy inference approaches, namely the Mamdani and Takagi-Sugeno (T-S) approaches. The Takagi-Sugeno approach will be used for our fuzzy system because it has the ability to utilize human experience in the form of datasets to generate a fuzzy inference system that is capable of mimicking human intelligence. In our problem, available datasets from the trust database of the consumer and RUs can be used to train fuzzy systems, using a suitable learning algorithm.

The antecedent part of both the Mamdani and T-S approaches is the same, however the major difference between the two approaches lies in the form of the output achieved. The output of Mamdani is in the form of fuzzy sets whereas the output of the T-S method is in the form of a linear crisp function. Different forms of output

have significant consequences for the inference approach. The problem is to determine which inference approach can be used to achieve the required goal. For reputation assessment, each of the input factors discussed above influences the value of the output that will be achieved. The T-S approach has the benefit of returning a single crisp value which is a more practical measure of reputation. The Mamdani approach, on the other hand, returns different outputs in the form of fuzzy sets from the same inputs because of the different approaches to aggregation and defuzzification. The T-S approach has the additional benefit of utilizing a tuning algorithm to tune input and output parameters, based on the data set available from the past experiences. Each data set consists of input values and the target output reputation value can be used for training the T-S model. This is a very effective method which is widely researched and produces reliable results and hence is used in my approach. In the next sub-section, the detail of the T-S method used for our system is discussed.

Takagi-Sugeno Inference Approach

The antecedent part of the Takagi-Sugeno rule system is the same as the Mamdani approach. The consequent part consists of a linear equation. So, a rule in the Sugeno system takes the form [5] as follows:

IF(x_1 is X_1 AND x_2 is X_2…, x_n is X_n)
THEN(y_q=a_{q0}+$a_{q1}x_1$+…$a_{qn}x_n$)

where x_1, x_2 are scalar inputs; X_1, X_2 are fuzzy sets; a_{q0}, a_{q1}, …a_{qn}, are real numbers; and y_q is the consequent of the rule.

A system with m fuzzy rules of the T-S form is proposed. The form of crisp output is

$$y(\underline{x}) = \frac{\sum_{q=1}^{m} \alpha_q \left(a_{q0} + \sum_{s=1}^{n} a_{qs} x_s \right)}{\sum_{q=1}^{m} \alpha_q} \tag{2}$$

In equation (2), α_q is the firing strength of rule q. The actual approach of fuzzy inference in this case has the following steps [1]:

a) fuzzify inputs
b) obtain the firing strength α_q associated with each rule q
c) obtain the output function of y_q associated with each rule q using the firing strength α_q
d) obtain the overall output $y(\underline{x})$ using expression (2) given above

To obtain the output from the inference approach, fuzzy rules are needed which are discussed in the next section.

5.6.3 Fuzzy Linguistic Control Rules

For our fuzzy system, the first-order T-S approach for fuzzy inference with a linear function on the right-hand side is used. A typical premise would take the form 'Credibility is good'. Therefore, the structure of the left-hand side rule in shorthand notation is

IF ((RO is X_1) AND (CR is X_2) AND (TD is X_3))

where

$X_j i=1,\ldots,3$ denotes in each case the fuzzy sets corresponding to the linguistic terms [POOR, AVERAGE, GOOD] for $j=1,2$ (RO, CR) and [LONG, AVERAGE, SHORT] for $j=3$ (TD).

A typical q^{th} rule has the form:

Reputation $y_q = a_{q0} + a_{q1}$*RO+a_{q2}*CR+a_{q3}*TD

Here $a_{q0}, a_{q1}, \ldots, a_{q3}$, are parameters. A tuning algorithm can be used to tune these parameters along with parameters of the input membership functions to obtain the best generalization capability of FIS. This is one of the major benefits of using the T-S system.

The complete q^{th} rule in short form notation, therefore, is

IF ((RO is X_1) AND (CR is X_2) AND (TD is X_3)) THEN
$a_{q0} + a_{q1}$*RO+a_{q2}*CR+a_{q3}*TD

If there are n inputs and K fuzzy sets, then the total number of possible fuzzy rules is equal to K^n. As previously discussed, there are 3 fuzzy sets for each input variable, therefore the total number of rules possible is 3^3=27 taking into account all possible combinations of the inputs.

5.6.4 Reputation Value (output)

As discussed earlier in this chapter, the output value of the reputation determination system is a reputation value which represents the reputation contribution of a RU. In this thesis, the scale of the output variable is from 0 to 5 where 0 represents the lowest (worst) reputation and 5 represent the highest (best) reputation. The detail of how to obtain output is given in Section 5.6.2.1.

5.6.5 Soft Computing-Based Approach for Reputation Determination

As discussed in Section 5.3.2, after creating the initial FIS which represents a neural network, a soft computing-based approach is used to train this FIS. The merger of fuzzy logic with neural networks falls under the soft computing-based approach which solves computing problems based on imprecision, uncertainty and partial truth by incorporating human expert knowledge in the computing process [6], [7]. Soft computing allows us to solve the reputation determination problem using the training dataset and creates a self-adaptive system based on the domain knowledge since the characteristics of soft computing-based systems are the ability to process information, adapt to changing environmental conditions and learn from the environment.

The reason for using the soft computing-based approach to solve the problem is that traditional computing lacks the flexibility which is present in soft computing. In this thesis, I am considering this flexibility on the basis of the representation of input variables as approximate values which also represent semantics, along with other benefits. For example, soft computing methodologies learn from experience; they have ability to universalise the domain where there is no direct experience and they can perform parallel processing using computer architecture. However, this flexibility comes at the cost of accuracy. Since soft computing is based on approximation, it has less accuracy compared to traditional computing. However, a

certain degree of imprecision can be tolerated in some systems. For example, for a service consumer who is utilizing a bandwidth service from an ISP service provider, the bandwidth requirement is 24 Mbps. Since the bandwidth level fluctuates due to several factors, bandwidth between 24 Mbps to 20 Mbps is still considered to be very good for the consumer. But bandwidth below 8 Mbps is not acceptable. This situation can be represented by a fuzzy membership function, as depicted in Figure 5.14:

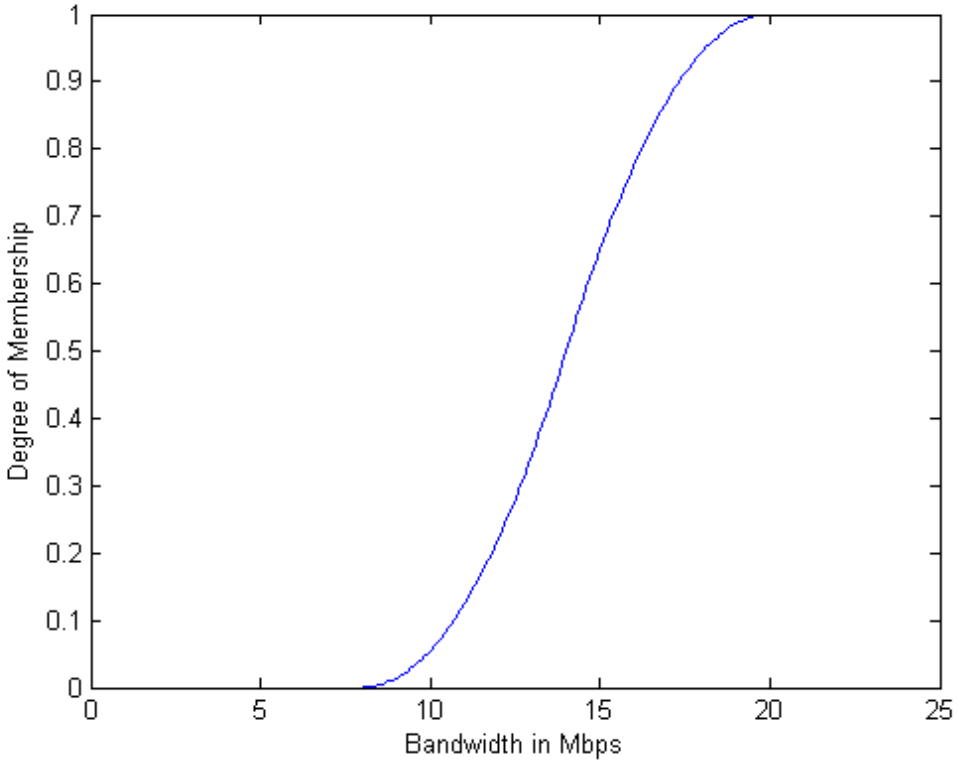

Figure 5.14: Representation of degradation of bandwidth

In the above figure, although the bandwidth is imprecise, it is useful in determining the level of quality of service (good, average or bad) based on the consumer's tolerance.

Different combinations of soft computing methodologies give rise to hybrid intelligent systems. One well known hybrid system is the neuro-fuzzy system which consists of Neural Networks (NN) and Fuzzy Logic (FL). Due to the benefits of this hybrid system, which are discussed in the next sub-section, it is used in this research.

Neuro-fuzzy systems

As discussed, the soft computing paradigm consists of fuzzy logic, neural networks, genetic computing and probabilistic reasoning methodologies. Each of these methods has its own advantages and drawbacks. The drawbacks of one methodology can be covered by the benefits of another methodology by merging them. Merging NN and FL results in the following different characteristics [6]:

1. Neuro-fuzzy systems can be developed by this merger in which fuzzy systems can be tuned using NN. Adaptive neuro-fuzzy systems fall into this category.
2. Fuzzy neural networks can be developed by this merger. This network retains the properties of NN with the fuzzification of some of the elements. Fuzzy logic, in this network, can be used to determine the learning steps of the NN structure.
3. Fuzzy-neural hybrid systems can be developed by this merger in which both fuzzy logic and neural networks perform separate tasks for separate sub-systems. Depending on the application, NN can be utilized for prediction whereas fuzzy logic can be used to control the system.

The above discussion proves that, due to the base properties of neural networks and fuzzy logic, a range of real world problems can be addressed by their merger and gives rise to different algorithms to address different characteristics of this merger. A well-known algorithm for neuro-fuzzy systems is the Adaptive-Network Based Fuzzy Inference System (ANFIS) [8], [9]. In the next section, the detail of this algorithm is discussed.

Adaptive-Network based Fuzzy Inference System (ANFIS)

In this sub-section, type III ANFIS architecture and algorithms are discussed. By combining neural networks and fuzzy logic, both fuzzy reasoning and network calculation are available simultaneously [10]. ANFIS consists of antecedent and consequent parts which can be defined by the fuzzy inference structure. The ANFIS structure is shown in Figure 5.15:

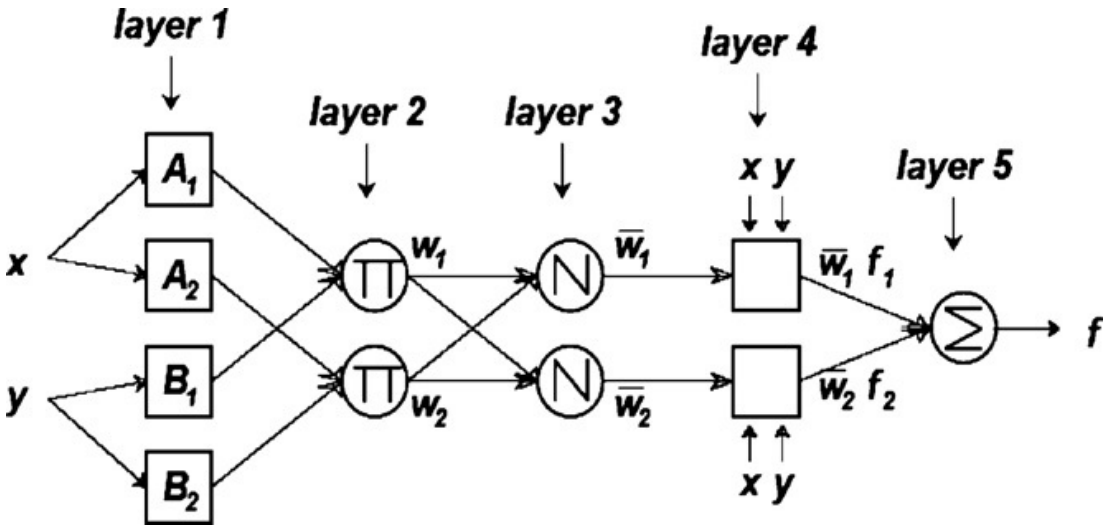

Figure 5.15: The type III ANFIS structure with two inputs and one output (adopted from [10])

The ANFIS structure consists of five layers and is equivalent to a multi-layered neural network. The process starts with fuzzification in the first layer which is followed by the implication process using AND rule in the second layer. On the basis of the output of implication, the third layer normalizes the Membership Functions (MFs) which can then be used in the conclusion of the fuzzy rules for the consequent part. By summing up the outputs from layer four, the fifth layer computes the output of the fuzzy system. If there are two inputs, each with two labels, the feed-forward equations of the ANFIS structure for each input, shown in Figure 5.15 according to type III rules, are as follows:

$$\omega_i = \mu_{A_i}(x_1) \times \mu_{B_i}(x_2), i = 1,2 \quad (3)$$

$$\bar{\omega}_i = \frac{\omega_i}{\omega_1+\omega_2}, i = 1,2 \quad (4)$$

$$f = \frac{\omega_1 f_1+\omega_2 f_2}{\omega_1+\omega_2} = \bar{\omega}_1 f_1 + \bar{\omega}_2 f_2 \quad (5)$$

As mentioned in Section 5.5.1, G-Bell shape curves are used to represent input variables. For the ANFIS algorithm, usually bell-shape membership functions are used for input variables with minimum grade equal to 0 and maximum grade equal to 1 such as:

$$\mu_{A_i}(x) = \frac{1}{1+\left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{b_i}} \quad (6)$$

$$\mu_{A_i}(x) = \exp\left\{-\left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{b_i}\right\} \quad (7)$$

where $\{a_i, b_i, c_i\}$ are the parameters of MFs which affect the shape of MFs and these parameters can be tuned using learning algorithms.

Learning algorithms

The important thing to note in ANFIS is the learning algorithms. There are several learning algorithms available which provide different optimization techniques. Choosing one of these algorithms is a trade-off between computation complexity and performance. In addition to this, there is a trade-off between precision and interpretability. Determining which learning algorithm should be used for the ANFIS is a debatable issue. This thesis agrees with Jang's statement [8] that the selection of learning algorithms depends on a case-by-case situation and the decision should be left to the users. In my experiment, *Gradient Descent (GD) only with least square estimation* is employed which is a computationally complex hybrid algorithm compared to *GD only* and *GD only with one pass of least square estimation* but it produces more accurate results.

5.6.6 Experiment

After describing the initial setup for our experiment in the previous section, our experiment process is now discussed. Using MATLAB's fuzzy logic toolbox, the following four major steps were performed for ANFIS training:

1. Training and monitoring datasets were loaded.
2. Initial FIS was created.
3. Initial FIS was trained.
4. Trained FIS was validated.

Before detailing the experiment process, the source of the training data and what is input-output data is explained and an explanation of our dataset is also given.

5.6.6.1 Obtaining Training Data

ANFIS uses a training data set to tune input and output fuzzy variables. Obtaining training data is a critical part of the whole process [11]. The first step is input selection. Input vectors should cover important aspects of the system without which optimum results cannot be achieved. The omission of any important input points may result in some output values that are inappropriate. The second step is selecting the values of the output vector based on the considered inputs from the training data set. The output vectors selected for training the data set must confirm the rule base. An artificial dataset containing values that span all possible scenarios for the considered inputs in a real-life situation is generated. These values were processed on linguistic rules to generate reputation values (output).

Each tuple of our training dataset consists of input vectors of the form [RECOMMENDATION OPINION, CREDIBILITY, TIME DELAY] and an output value that represents reputation. As an example, one tuple of our dataset is [3.6 3.7 3.2] with output 3.94. This tuple corresponds to the following linguistic rule:

IF RO is Good AND CR is Good AND TD is Average THEN REPUTATION is Good

588 such tuples encompassing every aspect of the reputation system were generated, 97 of which were used for monitoring purposes, leaving 491 data points for training purposes. The monitoring data set was also used for the cross-validation of the FIS structure, alongside the training data set during the training.

5.6.6.2 Training Process to Reputation Determination System

The training and monitoring data sets are used with tuning algorithms. The tuning algorithm used was the adaptive-network-based fuzzy inference system (ANFIS) algorithm [8], since it tunes the parameters of both the input membership function as well as the output coefficients [12]. For the studies carried out in this thesis, the version of ANFIS implemented in the Fuzzy Logic Toolbox of MATLAB was used. Initially, the data set was split into three parts: (1) a training set; (2) a monitoring set; and (3) a generalization test set. In the training stage, the training set error to derive the adjustments in the parameters was used. This training set error is the difference

between the outputs provided by the model and the actual value for each instance. The monitoring set, which is unseen by the training model was used, defining the output training to monitor the generalization capability of the model. Thus, the monitoring set error = (predicted value – actual value)^2 goes through to minimum before increasing, even though the training set error is still decreasing. This minimum corresponds to the best generalization capability of the model and training ceases when this is achieved. The final training error achieved was 0.2076 and the final monitoring error achieved was 0.2054.

5.6.7 Analysis

Using the experimental investigation of the fuzzy logic-based reputation measure, three fuzzy sets corresponding to the linguistic term set [Good, Average, Poor] were investigated to span the input space of RO and CR input factors and three fuzzy sets corresponding to the linguistic term set [Long, Average, Short] to span the input space of the TD input factor. The following curves for the input membership functions were obtained, as shown in Figure 5.16 – 5.18:

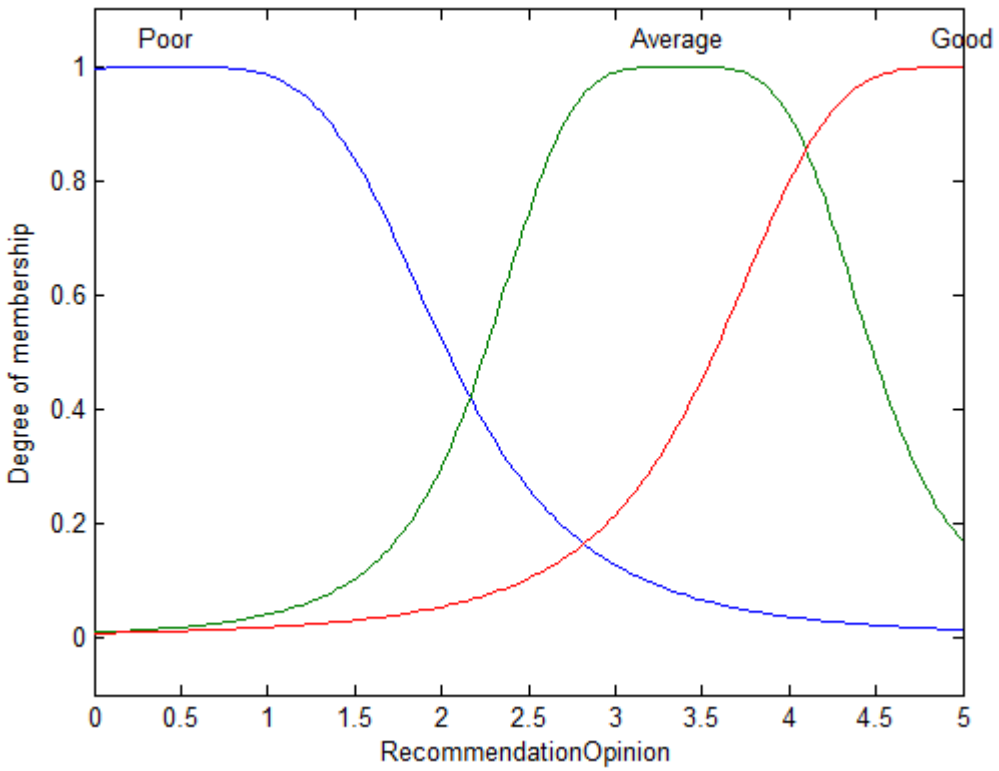

Figure 5.16: Membership function for Recommendation Opinion

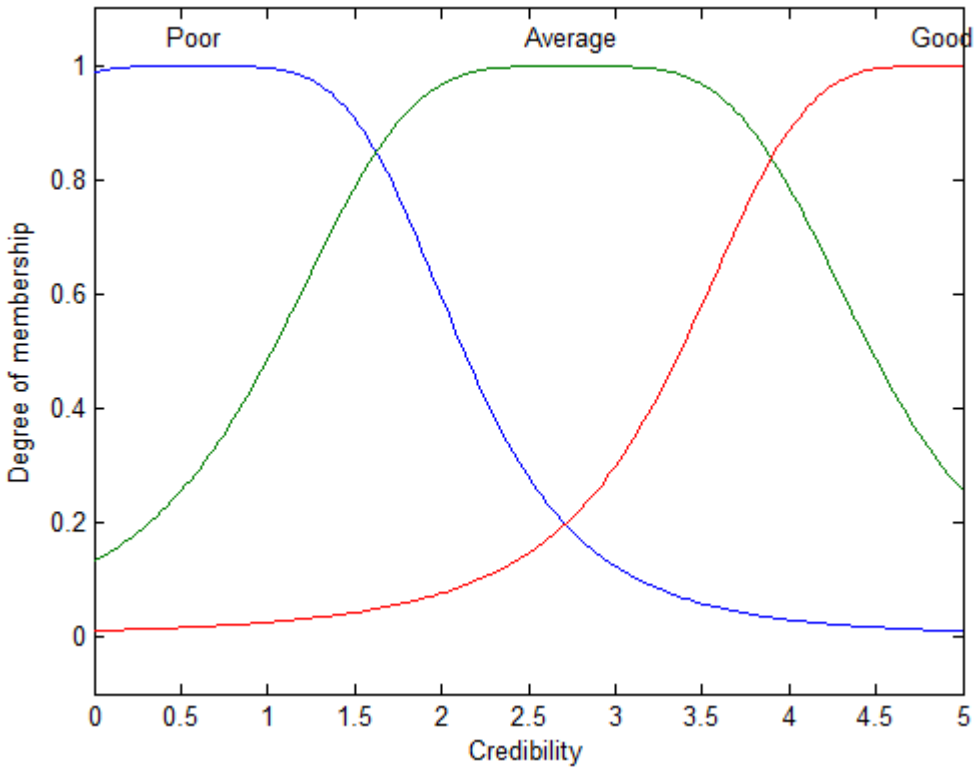

Figure 5.17: Membership function for Credibility

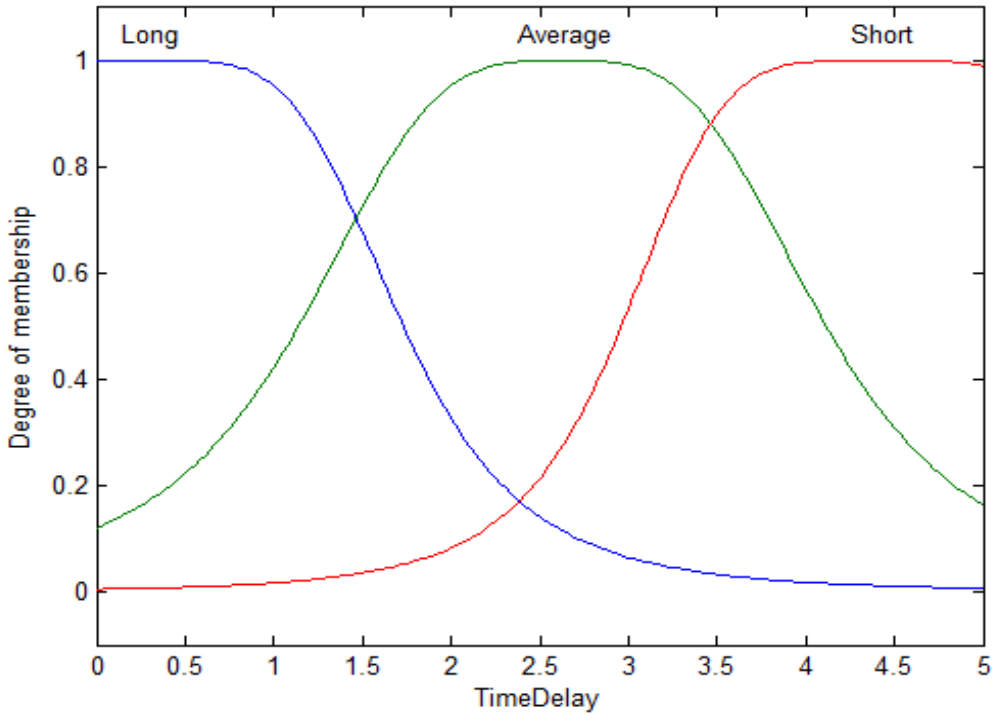

Figure 5.18: Membership function for Time Delay

Note that the 'Poor' membership function for the Recommendation Opinion and Credibility in Figures 5.8 and 5.9 spans to the right which implies that the 'Poor' recommendation and credibility has a great influence on the reputation of the system. It is also noted that the 'Average' membership function in the Recommendation Opinion in Figure 5.8 is skewed to the right which further emphasizes the influence of the 'Poor' data set in the Recommendation Opinion, whereas the 'Average' membership function in Credibility in Figure 5.9 spans wide which implies that it has a great influence on ascertaining reputation. The 'Short' membership function in Figure 5.10 spans to the left which implies that the time decay function is slow, indicating that the Time Delay has less influence on reputation when compared to other input variables namely, Recommendation Opinion and Credibility.

Model Validity

In order to study the validity of the model developed, the following three methods [12] are used:

1) input values corresponding to the training data were loaded and the output obtained from FIS was checked against the output value of the actual training data set;
2) input values that were not in the original training set were run with this fuzzy model;
3) sensitivity studies were conducted, changing each of the input factors one by one about a nominal point from the initial measured data.

The first two types of validations can be achieved using the ANFIS interface in MATLAB. In both cases, correct results were obtained within an acceptable tolerance. These validations indicate that our input data set includes all of the representative features of our reputation model. In the case of the second validation method, our model predicts output values as expected in response to input values that were not in the original training set. A discussion on the third validation technique, sensitivity analysis, is given in the next sub-section.

Sensitivity Study for Fuzzy Trust Model

Sensitivity analysis is a technique used to determine how different values of an independent variable will impact on a particular dependent variable [13]. Quantitative sensitivity analysis is used for quality assurance and the validation of model-based analysis. There are three broad categories of sensitivity analysis methods: statistical (or probabilistic) methods, mathematical methods and graphical methods. For the sensitivity analysis of our model, mathematical methods are implemented which are useful for deterministic and fuzzy-based models.

The sensitivity of a model output to the range of variation of an input can be assessed by mathematical models. Sensitivity is generally defined in terms of relative change in the output. These methods assess the impact of the range of variation in the input values on the output instead of simply the result variance of the output due to the variance in the inputs [14]. Most important inputs can be identified using mathematical models. In addition to this, these models are useful to identify inputs that require further data identification and research.

For the sensitivity analysis of our model, the Nominal Range Sensitivity Analysis (NRSA) method, which is also known as local sensitivity analysis or threshold analysis, is adopted. In this method, the effect of varying only one of the model inputs across its range of plausible values on model outputs is observed, while holding all other inputs at their nominal or base-case values [15]. The difference in the model output due to the change in the magnitude of the input variable is referred to as the sensitivity or swing weight of the model to that particular input variable [16]. The sensitivity can also be represented as positive or negative percentage change compared to the nominal point. If the sensitivity analysis is repeated for any number of individual inputs, the sensitivity index can be created which is calculated as follows:

$$\text{Sensitivity} = \frac{\text{Output}_{\text{max input}} - \text{Output}_{\text{min input}}}{\text{Output}_{\text{nominal point}}}$$

where $Output_{max\ input}$ and $Output_{min\ input}$ represent the maximum and minimum range for the output variable and $Output_{nominal\ point}$ represents the output value selected as

base point.

Using the NRSA method for the generated FIS, the results of our sensitivity studies are given in Tables 5.3 – 5.5; the input vector consists of the following factors: [Recommendation Opinion, Credibility, Time Delay]. These results are obtained by perturbing each good, poor and average recommendation opinion and credibility in addition to each long, average and short time delay. The fuzzy trust model is then used to find the overall measure of reputation for each of these perturbed values. Let us initially consider all the results and then the results with respect to some of the input factors in turn.

Almost all of the values generated for reputation indicate a movement in the correct direction, that is, an increase in input values leads to an increase in reputation and vice versa for a disturbtion about the nominal point in a single input factor. The exceptions are indicated by an asterisk (*). However, even for these, while the movement with respect to nominal points might be slightly in the wrong direction (test case number 4 in Table 5.4), if two adjacent points corresponding to a change in the same input factor are considered, it can be noticed that one of these points indicates a movement in the correct direction.

In order to test the sensitivity of my proposed model, I first defined the boundaries in which each fuzzy set falls for input variables. Let us consider that the value of reputation for poor Recommendation Opinion, poor Credibility and long Time Delay is approximately between 0 and 1.5, for average Recommendation Opinion, Credibility and Time Delay is between 1.6 and 3.5 and for good Recommendation Opinion, Credibility and short Time Delay is between 3.6 to 5. Note that in reality, each of these will be fuzzy sets. However, for the purpose of the qualitative discussion given below, the above ranges will suffice.

Almost all points obtained by the change in the input variable, except for test cases 4, 13, 14 in Table 5.4 and test cases 6, 11, 12 in Table 5.5, result in a change in reputation in the expected direction. Hence, the Fuzzy Trust Model seems to adequately model the relationship between input factors and the overall reputation.

The influences of some of these input factors are discussed in turn.

Recommendation Opinion

Let us consider the input factor Recommendation Opinion. The following observations can be made:

1) for poor Recommendation Opinion test cases 3, 9 and 10 (in Table 5.3) and test case 2 (in Table 5.4);
2) for average Recommendation Opinion test case 2 (in Table 5.3) and test cases 3, 9 and 10 (in Table 5.4);
3) for good Recommendation Opinion test cases 2, 3, 9 and 10 (in Table 5.5).

An improvement in the Recommendation Opinion leads to an improvement in reputation and a deterioration in the Recommendation Opinion leads to a deterioration in reputation, which is the expected result. It is observed that the first perturbation in the value of Recommendation Opinion in Table 5.3 takes the value of reputation from poor to average. In test case 2 in Table 5.4, the perturbation in the value of the Recommendation Opinion takes the value of reputation from average to poor. This indicates that the Recommendation Opinion has a significant influence on reputation.

Credibility

Let us next consider the input factor, Credibility. The following observations can be made:

1) for poor Credibility test cases 4, 5, 11 and 12 (in Table 5.3);
2) for average Credibility test cases 4, 5, 11 and 12 (in Table 5.4) and test case 4 (in Table 5.5);
3) for good Credibility test cases 5, 11 and 12 (in Table 5.5).

Almost all indicate that improvement in Credibility leads to improvement in reputation and vice versa with the exception of a few test cases. These test cases are numbers 11 and 12 in Table 5.5 and number 4 in Table 5.4. However, the other test cases in the same tables indicate movement in the right direction. In nearly all cases, Credibility has a strong influence on reputation for all subsets. In some cases, this influence is more pronounced (e.g. test case 4 in Table 5.5). Hence Credibility, with the Recommendation Opinion, is a significant indicator of reputation.

Time Delay

Let us finally consider the input factor, Time Delay. The following observations can be made:

1) for long Time Delay test cases 6, 7, 13 and 14 (in Table 5.3);
2) for average Time Delay test cases 6, 7, 13 and 14 (in Table 5.4);
3) for short Time Delay test cases 6, 7, 13 and 14 (in Table 5.5).

Almost all indicate movement in the right direction; that is, reducing time delay (higher value) increases reputation and increasing time delay (lower value) decreases reputation. The exceptions are test cases 13, 14 in Table 5.4 and test case 6 in Table 5.5. Although these test cases indicate movement in the wrong direction, all other points in those tables with respect to time delay lead reputation in the right direction.

Hence, it can be concluded that Credibility and Recommendation Opinion have a significant impact on reputation followed by Time Delay. In other words, the effect of Time Delay is slowly changing which is consistent with our model.

Table 5.3: Input test cases derived from varying different inputs about two nominal points for 'poor' reputation

Test Case Number	Input Vector	Reputation
1	[0.8 0.6 1]	0.95 nominal pt 1
2	[1.5 0.6 1]	1.91
3	[0.1 0.6 1]	0.67
4	[0.8 1.1 1]	1.21
5	[0.8 0.1 1]	0.72
6	[0.8 0.6 1.5]	0.93
7	[0.8 0.6 0.5]	0.94
8	[0.5 1 0.8]	1.04 nominal pt 2
9	[1 1 0.8]	1.22
10	[0.1 1 0.8]	0.88
11	[0.5 1.4 0.8]	1.19
12	[0.5 0.6 0.8]	0.82
13	[0.5 1 1.4]	1.07
14	[0.5 1 0.2]	0.94

Table 5.4: Input test cases derived from varying different inputs about two nominal points for 'average' reputation

Test Case Number	Input Vector	Reputation
1	[2.3 2 2.2]	2.13 nominal pt 1
2	[1.9 2 2.2]	1.42
3	[2.7 2 2.2]	2.84
4	[2.3 1.6 2.2]	2.15 *
5	[2.3 2.4 2.2]	2.14
6	[2.3 2.4 1.8]	1.98
7	[2.3 2.4 2.6]	2.25
8	[3 2.8 3]	2.93 nominal pt 2
9	[2.5 2.8 3]	2.59
10	[3.5 2.8 3]	3.34
11	[3 2.3 3]	2.77
12	[3 3.2 3]	3.34
13	[3 2.8 3.5]	2.61 *
14	[3 2.8 2.5]	3.37 *

Table 5.5: Input test cases derived from varying different inputs about two nominal points for 'good' reputation

Test Case Number	Input Vector	Reputation
1	[3.6 3.7 3.2]	3.94 nominal pt 1
2	[4.1 3.7 3.2]	4.12
3	[3.1 3.7 3.2]	3.56
4	[3.6 3 3.2]	3.42
5	[3.6 4.2 3.2]	4.05
6	[3.6 3.7 2.8]	4.22 *
7	[3.6 3.7 3.8]	3.93
8	[4.1 4.5 4.3]	4.18 nominal pt 2
9	[3.8 4.5 4.3]	4.11
10	[4.6 4.5 4.3]	4.62
11	[4.1 3.9 4.3]	4.23 *
12	[4.1 4.9 4.3]	3.92 *
13	[4.1 4.5 4.8]	4.33
14	[4.1 4.5 3.7]	4.11

From the trained system, the TP SLA Manager can obtain R_i, reputation value given by one recommending user *i*. Reputation R_k of the service provider *K* is the aggregation of *n* recommendations (reputation values) of all recommending users. It is given by:

$$R_K = \sum_{l=1}^{n} R_l / n \quad (8)$$

5.6.8 Analysis Result

By using a soft computing-based approach, important insight into the relationship between the input factors and the output reputation value was gained [12]. It was noted that an improvement in input factors leads to an improvement in reputation and a deterioration in input factors leads to a deterioration in reputation. However, the degree of the influence of a particular input factor is also dependent on the values associated with other input factors [12]. Therefore, in some cases, a change in a particular input factor results in a major change in the reputation value, even resulting change in the fuzzy category range of reputation. But it may also be possible that it has much smaller influence. It was only possible to identify the cases that result in a major change in the reputation value through the use of fuzzy system model. To explain with an example the working of the fuzzy logic-based reputation determination system, consider a consumer 'A' who wants to select a video conferencing service for his business needs. There are two service providers, namely service provider 'B' and service provider 'C' that closely match the business and operational requirements of consumer 'A'. In order to make a selection from the list of potential service provider candidates, consumer 'A' requests reputation assessments from the TP SLA Manager. On receiving the request, the TP SLA solicits recommendations from RUs for service providers 'B' and 'C'. Suppose RU1 and RU2 send their recommendations about service provider 'B' and RU1 and RU3 send their recommendations about service provider 'C'. Using the soft-computing-based approach introduced earlier, the TP SLA Manager computes the final reputation values of service providers 'B' and 'C'. Based on RO, CR and TD values, the TP SLA Manager computes the reputation value provided by one RU. For example, using our reputation assessment methodology, the reputation values provided by RU1 and RU2 about service provider 'B' are 3.93 and 4.11 respectively. The reputation values provided by RU1 and RU3 for service provider 'C' are 4.12 and 3.56, respectively. Using equation (8), the TP SLA Manager computes the final reputation value of service provider 'B' as 4.02 and the final reputation value of service provider 'C' as 3.84. On the basis of the high reputation value, service user 'A' may want to enter into an interaction with service provider 'B'.

5.7 Conclusion

For SLA evaluation in the pre-interaction time phase, the solution for direct and indirect interaction between consumers and service providers was discussed. A deterministic approach for trustworthiness evaluation was used for cases of direct interaction. For reputation determination on the basis of recommendations from RUs, a soft computing-based approach was used. Recommendations along with credibility and the time factor were used as input and the output was the reputation contribution of a recommending user *i*. Reputation contributions from all RUs were then aggregated using equation (8) to reach the final reputation of a service provider. Once a consumer selects a service provider on the basis of his capability to provide the promised service, the next step in SLA management is to monitor the performance of a service provider. Performance evaluation of the service provider is discussed in the next section.

5.8 References

[1] E. Chang, F. Hussain, and T. Dillon, *Trust and Reputation for Service-Oriented Environments: Technologies For Building Business Intelligence And Consumer Confidence*: John Wiley & Sons, 2005.

[2] Z. Rehman, O. K. Hussain, and F. K. Hussain, "Multi-Criteria IaaS Service Selection based on QoS History", in *Advanced Information Networking and Applications (AINA)*, March 25 - 28, 2013, Barcelona Spain, pp.1129-1135.

[3] S. Sivanandam, S. Sumathi, and S. Deepa, *Introduction to Fuzzy Logic using MATLAB* Springer, 2007.

[4] M. A. Denai, F. Palis, and A. Zeghbib, "Modeling and control of non-linear systems using soft computing techniques", *Appl. Soft Comput.,* vol. 7, pp. 728-738, 2007.

[5] T. Takagi and M Sugeno, "Fuzzy identification of systems and its applications to modeling and control", *IEEE transactions on Systems, Man, and Cybernetics,* vol. 15, pp. 116-132, 1985.

[6] A. Zilouchian and M. Jamshidi, *Intelligent Control Systems Using Soft Computing Methodologies*, CRC Press, Inc. 2000.

[7] L. Zadeh, "The roles of fuzzy logic and soft computing in the conception, design and deployment of intelligent systems", in *Software Agents and Soft Computing Towards Enhancing Machine Intelligence*. vol. 1198, H. Nwana and N. Azarmi, Eds., ed: Springer Berlin / Heidelberg, 1997, pp. 181-190.

[8] J. S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system", *IEEE transactions on systems, man, and cybernetics,* vol. 23, p. 665, 1993.

[9] J. S. R. Jang, "Neuro-Fuzzy and Soft Computing-A Computational Approach to Learning and Machine Intelligence [Book Review]", *IEEE transactions on automatic control,* vol. 42, p. 1482, 1997.

[10] M. A. Shoorehdeli, M. Teshnehlab, A. K. Sedigh, and M. A. Khanesar, "Identification using ANFIS with intelligent hybrid stable learning algorithm approaches and stability analysis of training methods", *Applied Soft Computing,* vol. 9, pp. 833-850, 2009.

[11] J. S. R. Jang, *Input selection for ANFIS learning, Proceedings of IEEE 5th International Fuzzy Systems*, , 1996, Vol. 2, pp. 1493 - 1499.

[12] E. Chang and T. S. Dillon, "A usability-evaluation metric based on a soft-computing approach", *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 36, pp. 356-372, 2006.

[13] H. Christopher Frey, Amirhossein Mokhtari, and Tanwir Danish, "Evaluation of Selected Sensitivity Analysis Methods Based Upon Applications to Two Food Safety Process Risk Models", Dept. of Civil, Construction, and Environmental Eng., North Carolina State Univ., Raleigh, NC, 2003.

[14] M. G. Morgan, M. Henrion, *Uncertainty: A Guide to Dealing With Uncertainty in Quantitative Risk and Policy Analysis*: Cambridge University Press: Cambridge, NY, 1990.

[15] A. C. Cullen and H. C. Frey, *Probabilistic techniques in exposure assessment: a handbook for dealing with variability and uncertainty in models and inputs*: Springer, 1999.

[16] M. G. Morgan, *Uncertainty: a guide to dealing with uncertainty in quantitative risk and policy analysis*: Cambridge University Press, 1992.

CHAPTER 6

PERFORMANCE RISK IN POST-INTERACTION TIME-PHASE

6.1 Introduction

In the previous chapter, the pre-interaction time phase assessment for *SLA management* was discussed in detail. After the assessment of a service provider's capability, selecting the service provider and forming the Service Level Agreement (SLA) with that service provider, the consumer needs to assess the performance of the service provider in the post-interaction time phase. As mentioned earlier, the assessment deals with determining the performance of the service provider against the defined SLAs. This is done in two parts. Firstly, the probability of the service provider not performing according to the SLA is determined and secondly, the financial loss is determined. In this chapter, the performance assessment of the service provider according to the agreed SLA is discussed. This assessment gives an idea of the QoS for services provided by the service provider to the consumer and guides the consumer to take a future course of action.

The solution for a service provider's performance assessment in the post-interaction time phase consists of two parts: namely, a performance assessment in the *current timeslot* and a performance assessment in a *future timeslot*. Figure 6.1 depicts both scenarios.

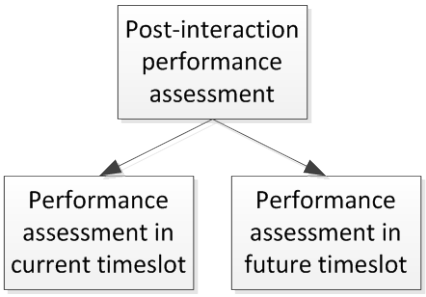

Figure 6.1: Post-interaction performance assessment scenarios

The main aim of presenting a solution in this chapter is to develop a *performance assessment system* in the cloud environment by considering the scenarios discussed in Figure 6.1. This performance assessment system is part of a *risk-based decision*

support system which enables a consumer to make an informed decision about the continuity of the service, the overview of which was discussed in Chapter 4. Based on the aforementioned scenarios, different *performance risk assessment* approaches will be used to assess the performance of the service provider. This chapter is organized as follows: Section 6.2 presents the overview of performance assessment in different timeslots in post-interaction time phase. To carry out performance assessment a conceptual framework is presented in Section 6.3. Section 6.4 discusses the performance assessment scenario when consumer's point of interaction is in current timeslot. Section 6.5 discusses the performance assessment scenario when consumer's point of interaction is in future timeslot. Working of performance assessment methodologies is demonstrated in Section 6.6 using a case study. Section 6.7 concludes the chapter.

6.2 Overview of performance assessment in different timeslots

As highlighted in Chapter 3, when a consumer assesses the performance of a service provider through the TP SLA Manager, different scenarios may arise, such as the following:

(a) the consumer is assessing the performance of the service provider in a timeslot which he is currently in, as shown in Figure 6.2; or
(b) the consumer wants to assess the performance of a service provider in a future timeslot of the post-interaction time phase, as shown in Figure 6.3.

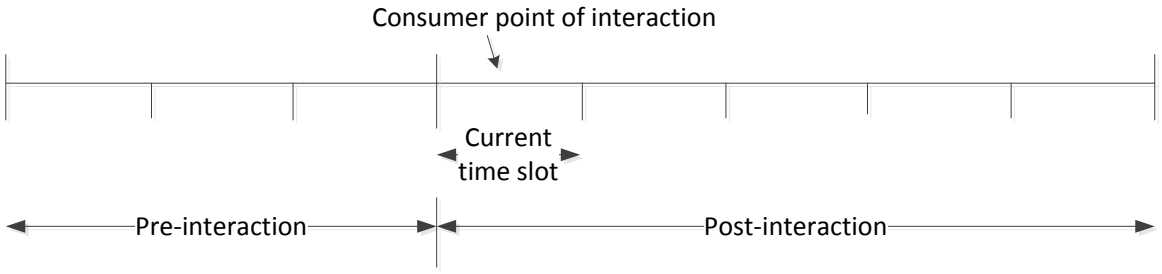

Figure 6.2: Current timeslot interaction scenario

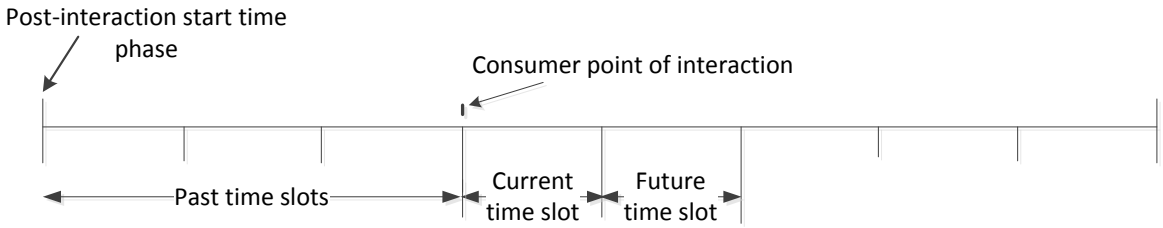

Figure 6.3: Future timeslot interaction scenario

These two scenarios are discussed briefly here.

Current timeslot interaction scenario: If the consumer's point of interaction is the current timeslot as depicted in Figure 6.2, a performance risk assessment method based on the *probabilitic approach* is used. This method takes a run-time performance parameter as an input along with the Service Level Objectives (SLOs) promised in the SLA and outputs service deviation levels for the specified timeslot. The detail of this method is given in Section 6.4.

Future timeslot interaction scenario: If the consumer's point of interaction is in a future timeslot, as depicted in Figure 6.3, a performance risk assessment method based on *exponential smoothing* is used. For performance assessment in a future timeslot, the consumer must use service deviations in past timeslots. In this thesis, once the consumer performs the performance assessment in a current timeslot and determines the service deviation levels, these service deviations levels are then stored in a central information repository, as discussed in Chapter 4, which can be later used in determining service deviation levels in a future timeslot. The detail of the prediction method is given in Section 6.5.

In the next section, the conceptual framework for the *performance assessment system* is presented.

6.3 Conceptual Framework

To develop the *performance assessment system* in the post-interaction time phase, I propose the conceptual framework depicted in Figure 6.4:

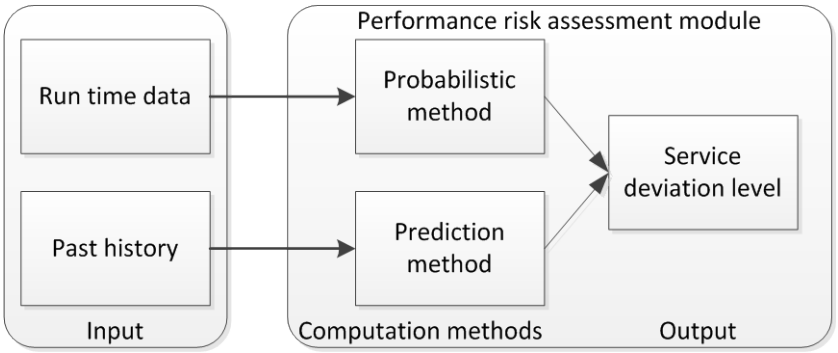

Figure 6.4: Conceptual framework for performance assessment system

The components of this framework are explained below:

Input: This part of the proposed system provides the input to start the performance assessment process. As explained in the previous section, if the consumer's point of interaction is in the current timeslot, the input in this case is the run-time data of the parameters on which the performance is being assessed. For example, if the performance parameter or SLO is the *response time* of an internet service and the performance metrics is in seconds, the run-time value of the *response time* in seconds is obtained. For cloud service users, the run-time parameter values can be obtained through the service provider's measurement interface, as discussed in the next section. If the consumer's point of interaction is in a future time spot and the consumer is assessing it in the current timeslot, the past interaction history or service deviation levels in past timeslots should be considered. A prediction method, such as *exponential smoothing,* needs past data to predict future data with the help of past service deviation levels stored in a central information repository.

Performance risk assessment module: This component of the proposed framework consists of performance risk computation methods and output. Computation methods have been mentioned earlier and will be discussed further in the following sections. The output of the performance risk assessment is the *service deviation levels* which indicate the deviation from the threshold of a SLO. This output is then used to calculate the financial risk, as discussed in the next chapter.

After discussion of different consumer interaction scenarios and the conceptual framework in the following sections, the detail of the computation methods for each scenario is given.

6.4 Ascertaining Performance Risk when the Consumer's Point of Interaction is in the Current Timeslot

If the consumer wants to assess the real time performance of a service provider in the timeslot which he is in at present, as shown in Figure 6.2a, then run-time SLA parameter values are used against SLOs given in the SLA to determine any service level deviation. The process of performance risk assessment in this case is depicted in Figure 6.5.

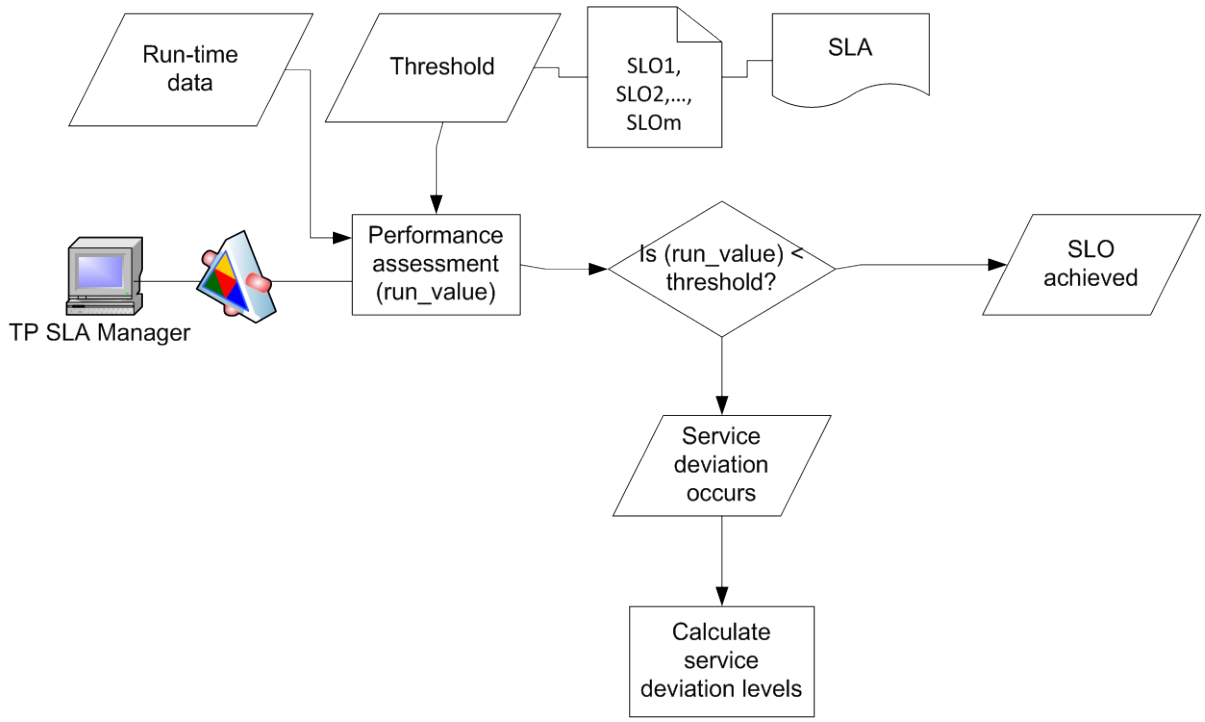

Figure 6.5: Performance risk assessment in the current interaction timeslot

TP SLA Manager: Whenever a consumer wants to assess the performance of a service provider, it requests the TP SLA Manager to carry out the performance risk assessment on his behalf, as discussed in Chapter 4. The TP SLA Manager uses the *performance risk assessment module* for this purpose. To observe the service deviation continuously, the TP SLA Manager monitors the service on *continuous intervals,* as discussed in Section 4.2.

Threshold: This is one of the inputs that is required by the *performance assessment* process. Threshold is the minimum level of service defined in SLOs. This input is required to compare it with its run-time value.

Run-time data: This input parameter is the run-time or actual value of an SLO defined in the SLA. As indicated in previous chapters, this data can be obtained from the TP SLA Manager through the service provider's measurement interface [1]. The service provider may choose a third-party service provider to display run-time parameters in the cloud environment to reduce the burden of monitoring service levels[2].

Performance assessment: This is an important step in the overall process. It consists of the following further steps:

- Determining the probability of occurrence of all events: the probability of occurrence (relative frequency) of all events related to a performance assessment criterion (SLO) over a specified monitoring interval by utilizing run-time parameter values is determined. If a discrete random variable *Q* represents the levels of service for a particular interval, the Probability Mass Function (pmf) for this random variable is given by [2]:

  $$f: q \rightarrow P(Q = q) \text{ or } f(q) = P(Q = q)$$

  where variable *q* indicates the values within the range of the random variable. The relative frequency for each class interval is then calculated as:

  $$P(Q = q) = \frac{Fr}{n} \quad (1)$$

  where *Fr* denotes the frequency of a service level on class intervals representing that service and *n* represents the promised service level. By definition of pmf:

  $$P(Q) = \sum_{i=1}^{n} \frac{Fr_i}{n} = 1 \quad (2)$$

  where *i* represents the class intervals.

[2] One such example is http://www.speednet.net.

Is run-value less than threshold? This step follows the performance assessment process and can be explained as follows:

- Each run-time performance value is compared with a given threshold of SLO and the service deviation level is determined as a result.

Calculate service deviation levels: In the previous step, it was determined that service deviation occurs. This step quantifies how much service deviation has occurred. This step is explained as follows:

If a discrete random variable *R* represents service deviation levels and a variable *r* indicates the values within the range of the random variable, the probability of service deviation is calculated as:

$$P(R = r) = \frac{P(q_i)}{\sum_{i=0}^{i<threshold} P(q_i)} \quad (3)$$

Consequently, a probability mass function for random variable *R* is given as:

$$P(R) = \sum_j P(r_j) = 1, \text{where} j \text{ is the scale of the service} \quad (4)$$

A graph is generated using the probability mass function that represents the service deviation levels. An example of such a graph is depicted in Figure 6.6, where the scale of the service deviation level is from 0 to 100 and x, y and z represent some values on this scale. This output can be used later in the process of analysing financial risk which will be discussed in detail in the next chapter.

6.5 Ascertaining Performance Risk when a Consumer's Point of Interaction is in the Future Timeslot

If the consumer's point of interaction is in a future point in time from the current timeslot in which he is at present, as shown in Figure 6.3, then he should utilize the service deviation levels of the service provider from the beginning of the post-interaction time phase to the current timeslot to predict and determine future service deviation levels. This assessment can only be done if the consumer has a past interaction history with the service provider in the same context. The steps of this process are depicted in Figure 6.7.

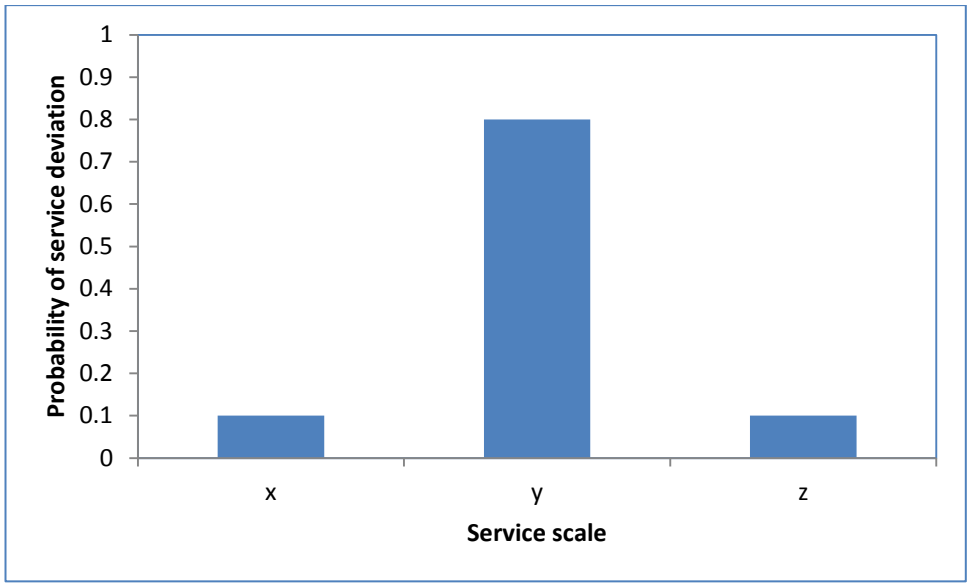

Figure 6.6: Service deviation levels

The explanation of the performance risk assessment process in the current timeslot is given in Section 6.6, using a case study.

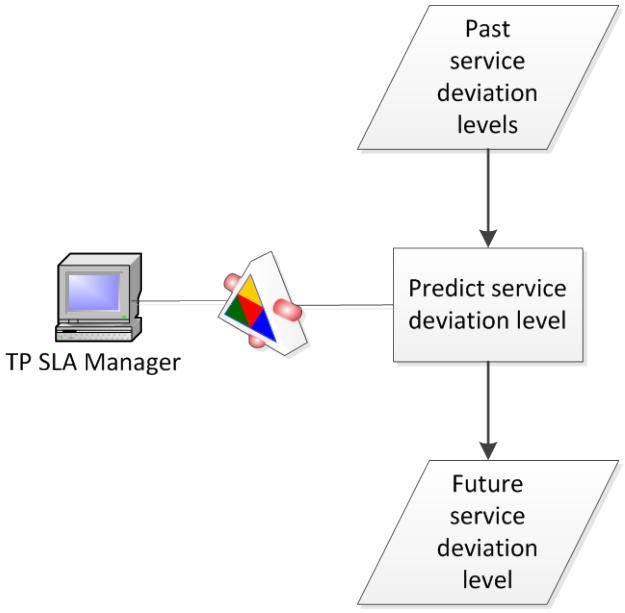

Figure 6.7: Performance risk assessment in a future interaction timeslot

TP SLA Manager: The consumer requests the TP SLA Manager to perform the performance risk assessment in a future timeslot. The TP SLA Manager may monitor

the performance of a service provider on *regular intervals* as discussed in Section 4.2.

Past service deviation levels: As discussed in Section 6.3, service deviation levels are stored in a central information repository. In order to determine future service deviation levels at timeslot say 'ft1' of the post-interaction time phase, it is proposed that the consumer should consider all previous service deviation levels from the beginning of the time space to the current timeslot 'ct'. Since the context of interaction is the same as the previous interactions, the scale of service deviation levels should be the same for all past interactions.

Predict service deviation level: In this step, a prediction method is used to predict the service deviation levels on the basis of inputs (past service deviation levels). There are several prediction techniques in the existing literature which can utilize the past interaction history and predict the future risk value. I use *exponential smoothing* for the prediction of service deviation levels in timeslot 'ft1' because this technique makes it possible to predict each service deviation level in turn. For example, if there are two past timeslots, namely timeslot 1 and timeslot 2, with service deviation levels represented by scales x, y and z on a percentage scale, the service deviation levels for a future timeslot are depicted in Figure 6.8.

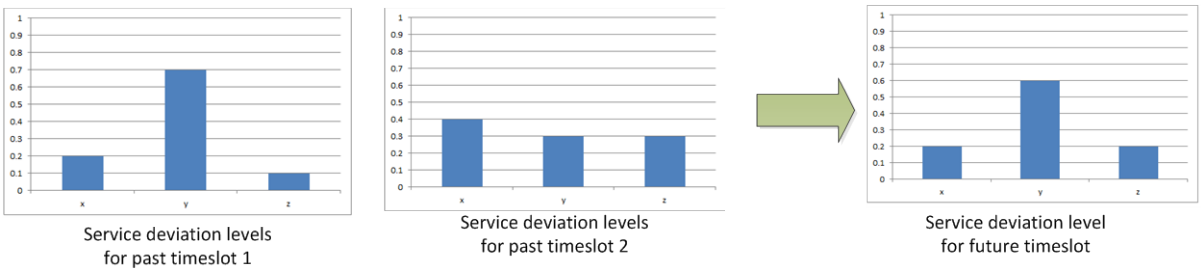

Figure 6.8: Prediction for service deviation levels in a future timeslot

For the prediction process shown in Figure 6.8, the service deviation levels for scale x from past timeslots are first taken and a future service deviation level for scale x is determined. A similar process is then repeated for scales y and z. The sum of the service deviation levels of the obtained graph, as shown in Figure 6.8, equals 1 which represents the probability mass function (pmf). In the following sub-section, the detail of exponential smoothing is discussed.

Exponential Smoothing

To forecast future values, exponential smoothing weighs past observations with exponentially decreasing weights [3]. In other words, it assigns exponentially decreasing weights to older observations. To assign the weights to a time series of observations, it uses one or more smoothing parameters. Using exponential smoothing, the deviation levels *DL* for timeslot *t* can be given as:

$$\mathrm{DL_t} = \alpha \mathrm{y_{t-1}} + (1-\alpha)\mathrm{DL_{t-1}}, 0 < \alpha \leq 1 \;\; \mathrm{t} \geq 3 \qquad (5)$$

where *DL* stands for the smoothed observation, *y* stands for the original observation and α is the smoothing parameter. The subscripts refer to timeslots. The smoothing scheme starts by setting DL_2 to y_1 which means that the smoothed series starts with the smoothed version of the second observation. This basic equation can be expanded as:

$$\mathrm{DL_t} = \alpha \mathrm{y_{t-1}} + (1-\alpha)[\alpha \mathrm{y_{t-2}} + (1-\alpha)\mathrm{DL_{t-2}}]$$

$$= \alpha \mathrm{y_{t-1}} + \alpha(1-\alpha)\mathrm{y_{t-2}} + (1-\alpha)^2 \mathrm{DL_{t-2}} \qquad (6)$$

By substituting DL_{t-2}, DL_{t-3} and so forth until we reach DL_2, the above equation can be rearranged as:

$$\mathrm{DL_t} = \alpha \sum_{\mathrm{i=1}}^{\mathrm{t-2}} (1-\alpha)^{\mathrm{i-1}} \, \mathrm{y_{t-i}} + (1-\alpha)^{\mathrm{t-2}} \mathrm{DL_2}, \;\; \mathrm{t} \geq 2 \qquad (7)$$

It should be noted that a larger value of α will result in a rapid change in the smoothed value of DL_t and a smaller value of α will result in a slower change.

Future Service Deviation Level: As a last step in the prediction method, the output (service deviation levels for future timeslot) is generated. This output was discussed earlier. The explanation of the performance risk assessment process in a future timeslot is given in the next section, using a case study.

6.6 Example of Determining Performance Risk in Cloud Computing

Consider a service user 'A' who wants to select a video conferencing service for his business needs. From user 'A's perspective, video conferencing is important as business meetings are arranged with his business customers through this service. Not

being able to conduct business meetings may result in financial loss and might damage the business reputation of service user ‘A’.

Let us consider that service user ‘A’, on a particular day, expects to secure a business contract of $15,000 through his video conferencing meetings. Since services such as video conferencing depend on bandwidth, let us consider that service provider ‘B’ has the maximum capacity to provide a bandwidth of 24 Mbps. Further, let us consider that service provider ‘B’ establishes a threshold of 12 Mbps in the SLA against which the bandwidth service is expected to be delivered. If service provider ‘B’ delivers bandwidth below 12 Mbps, these bandwidth levels are considered as service deviation levels since they deviate from the threshold. It is important to measure the service deviation levels in order to measure the performance of service provider ‘B’. To exemplify the scenario, let us assume that during the course of service delivery (video conferencing), service user ‘A’ receives 6 Mpbs bandwidth which may help in the continuity of the service but at the cost of a low quality service. Particularly, in the case of video conferencing, poor quality can result in a loss of commitment of service user ‘A’ to his business clients which could have a severe impact, such as financial loss. Therefore, service provision below the specified threshold should be considered as not acceptable Quality of Service (QoS).

From the above discussion, there are two scenarios in which the service user might be interested. Service user ‘A’ might be interested in assessing the performance of service provider ‘B’ in the current timeslot or it might be the case that service user ‘A’ is interested in assessing the performance of service provider ‘B’ in a future timeslot. Both of these cases are discussed in detail for the given case study.

6.6.1 Performance Risk Assessment in the Current Timeslot

Suppose a consumer wants to calculate the probability that the bandwidth falls below the threshold (12 Mbps) for the video conferencing service. By using the techniques discussed in Section 6.4, the TP SLA Manager determines the level of bandwidth delivery of the service provider in an advance time period of 24-hour duration. If Q in equation (1) represents the levels of bandwidth for a 24-hour interval, then using equation (2), the following graph is obtained on the basis of observed bandwidth levels:

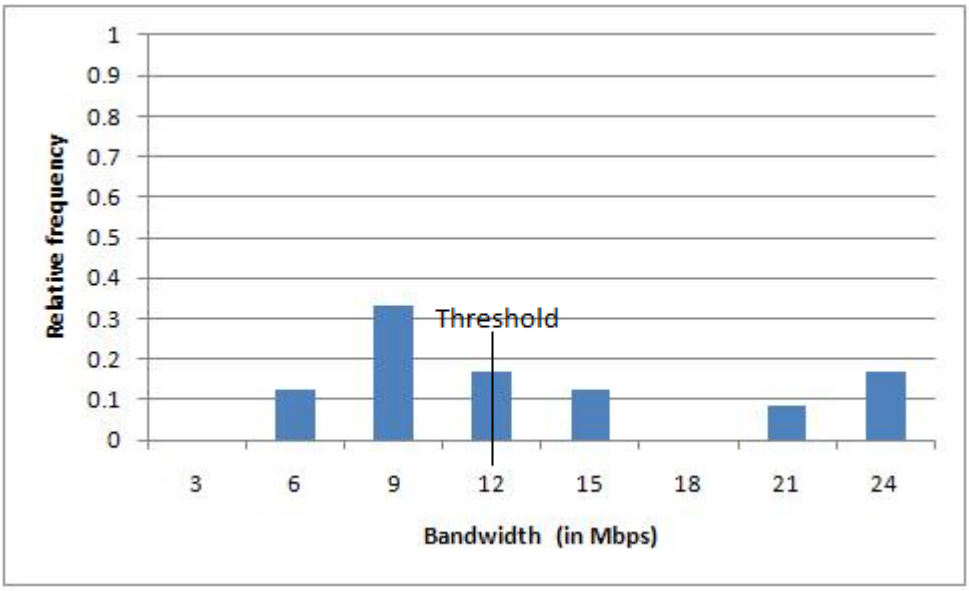

Figure 6.9: Probability mass function for bandwidth levels

In Figure 6.9, where $q < 12$, it indicates the chances of the service not meeting the required threshold. To determine the level of service degradability, this portion of the distribution is converted into another distribution that represents the service deviation levels of a bandwidth. Using equation (3), if *R* represents the service deviation levels and the threshold is 12 Mbps, then equation (3) can be rewritten as:

$$P(R = r) = \frac{P(q_i)}{\sum_{i=0}^{i<12} P(q_i)}$$

Consequently, the probability mass function for random variable *R* is given as:

$$P(R) = \sum_j P(r_j) = 1, \text{where} j = 0,3, \ldots,9$$

One last step that needs to be performed when the service level distribution is obtained is to normalize the service deviation level to a percentage scale. This step is needed so that the impact of the service level deviation can be determined on the financial expectation of a consumer, which is discussed in the next chapter. After obtaining the service deviation distribution and normalizing it to a percentage scale, the following graph is obtained:

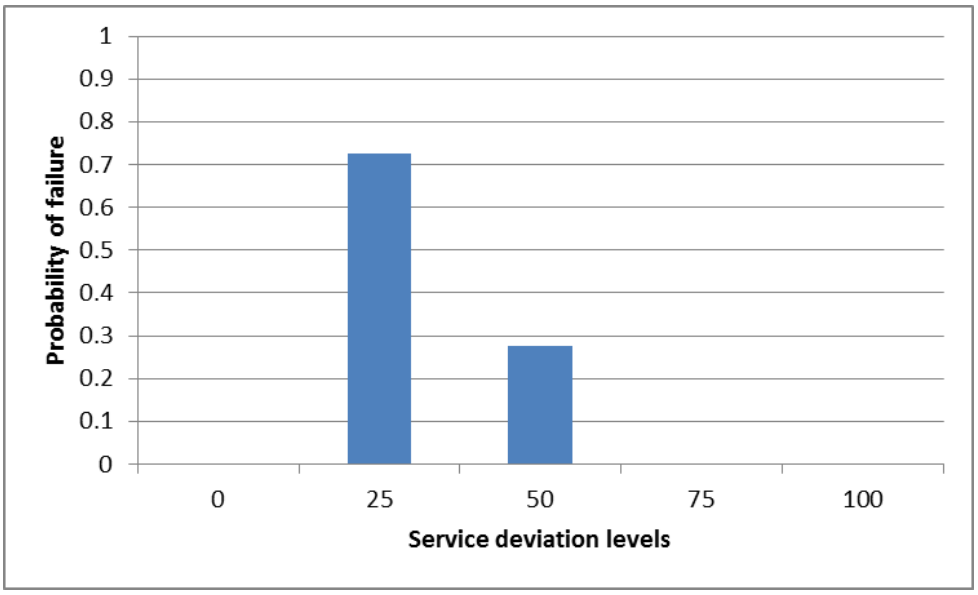

Figure 6.10: Level of service deviation from minimum bandwidth requirement

If the consumer point of interaction is a future timeslot, as discussed at the start of the case study, then using *exponential smoothing,* service deviation levels are obtained, which is explained in next section.

6.6.2 Performance Risk Assessment in a future timeslot

The case study which was used in the previous section is continued here. With the help of exponential smoothing, the service deviation level graph for the future timeslot is obtained. For this calculation, three timeslots are used, where timeslots 1 and 2 are past timeslots and timeslot 3 is the current timeslot. The service deviation levels for future timeslot 4 need to be determined. Figures 6.11 – 6.13 depict the service deviation levels for timeslots 1, 2 and 3, respectively.

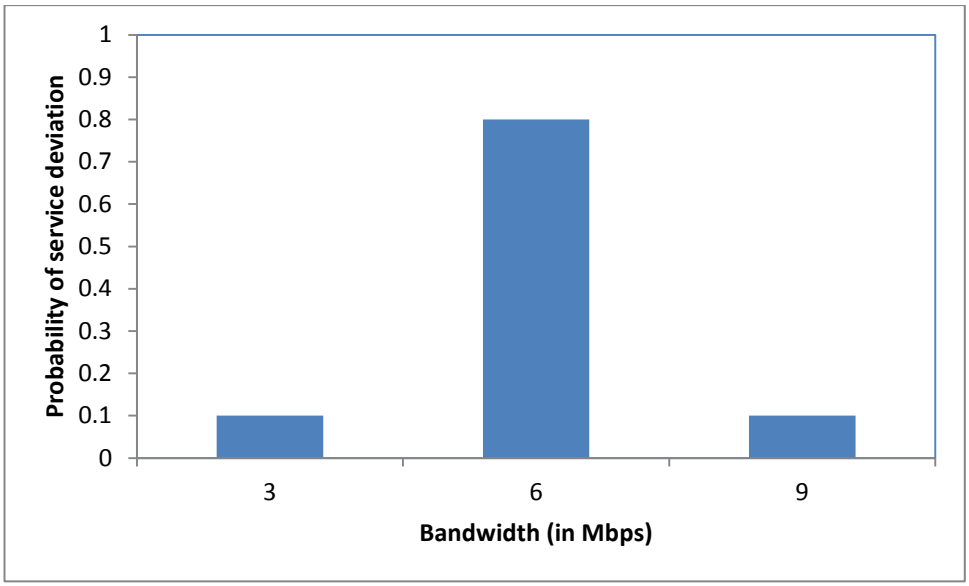

Figure 6.11: Service deviation levels in timeslot 1

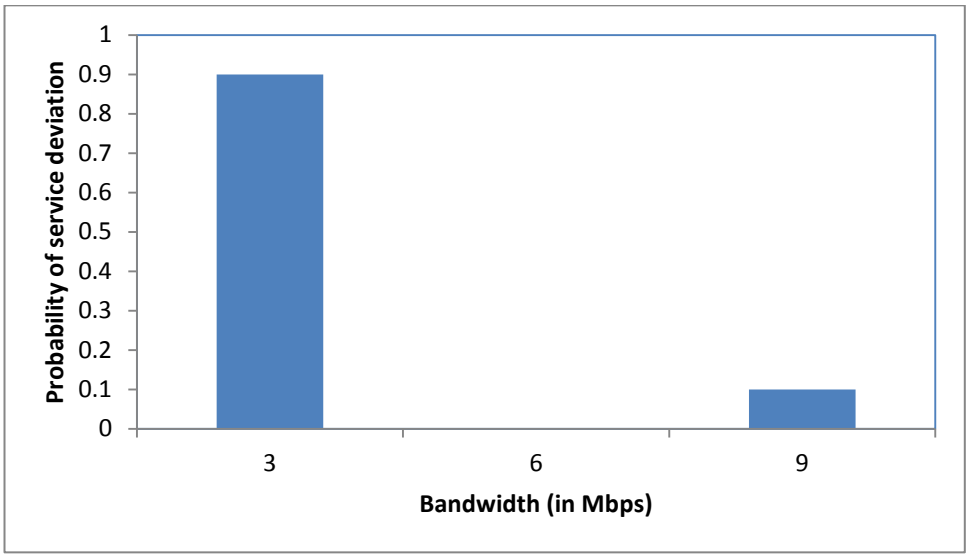

Figure 6.12: Service deviation levels in timeslot 2

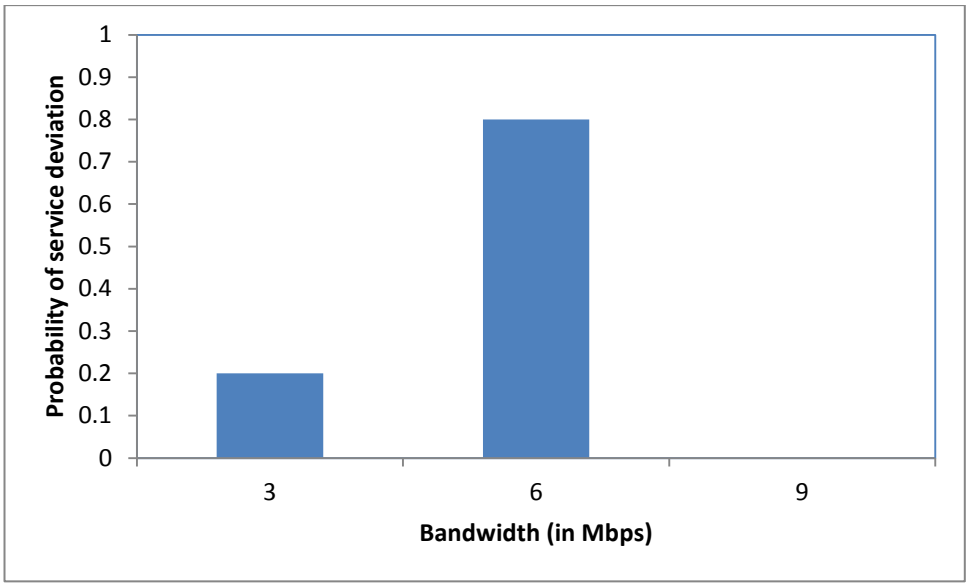

Figure 6.13: Service deviation levels in timeslot 3

For the following calculations, α=0.5. From the above figures, for bandwidth 3, $y_1 = 0.1, y_2 = 0.9, y_3 = 0.2$. Using equation 7, DL_4 is calculated as:

$$DL_4 = 0.5[(0.5)^0(0.2) + (0.5)^1(0.9)] + (0.5)^2(0.1) = 0.3$$

For bandwidth 6, $y_1 = 0.8, y_2 = 0, y_3 = 0.8$. and the following result is obtained:

$DL_4 = 0.6$

For bandwidth 9, $y_1 = 0.1, y_2 = 0.1, y_3 = 0.0$. and the following result is obtained:

$DL_4 = 0.1$

Based on above calculations and converting the final graph to a percentage scale, the graph depicted in Figure 6.14 is obtained:

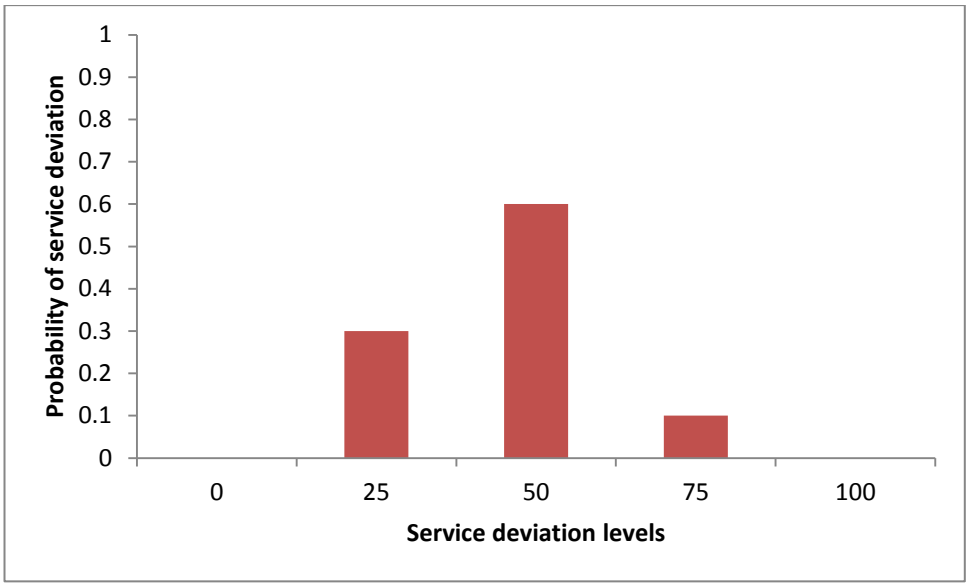

Figure 6.14: Service deviation levels in timeslot 4

The curves obtained in Figure 6.10 and 6.14 are called *Deviation Level* (DL) which will be used in the financial risk assessment in the next chapter.

6.7 Conclusion

This chapter has described the performance assessment of a service provider in the post-interaction time phase. Since the cloud is a dynamic environment, the cloud service level may fluctuate, hence it is important for the consumer to monitor service levels after entering into an interaction with the service provider. Two cases have been considered for the performance risk assessment of a service provider and methods for both of these scenarios have been discussed in detail. For performance assessment in a current timeslot, a probabilistic approach has been used which compares run-time service parameters (SLO) of service providers with a threshold of a service level defined in SLA. For performance assessment in a future timeslot, a prediction method has been discussed. In both cases, service deviation levels are obtained and normalized to a percentage scale. After service deviation levels have been attained, the impact of these deviations on the financial outcome of the consumer needs to be determined. The next chapter discusses the financial risk assessment for this purpose in the post-interaction time phase.

6.8 References

[1] J. Burton S. Kaliski and W. Pauley, "Toward risk assessment as a service in cloud environments", presented at the Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, Boston, MA, 2010.

[2] M. Modarres, *What every engineer should know about reliability and risk analysis,* Vol. 30: CRC Press, 1993.

[3] *Engineering Statistics Handbook*. Available: http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc431.htm

CHAPTER 7

DETERMINING FINANCIAL RISK IN THE POST-INTERACTION TIME PHASE

7.1 Introduction

In the previous chapter, an approach for *performance risk assessment* in the post-interaction time phase was discussed. The next step for *SLA management* in the post-interaction time phase is to determine the *financial loss* due to the impact of the determined service deviation level. This chapter focuses on the detail of determining the financial loss. As discussed in Section 3.3, when a consumer enters into an interaction with a service provider, he expects to obtain financial gain from the services by the service provider achieving the desired outcomes. Not achieving these outcomes results in financial consequences called *financial loss*. This thesis explores the view that, as a result of financial loss, a consumer may need to put in extra investment to achieve the business objectives as opposed to what was expected initially. This is classified as financial loss.

The main aim of this chapter is to propose an approach to *financial risk assessment* in the cloud environment. The financial risk assessment system not only determines the financial loss due to the non-conformance of a service provider to the SLA, but also determines the additional cost that contributes to the financial loss, such as the cost of migrating a service to another service provider. Both the aforementioned objectives can be achieved by assessing financial risk based on *dependable* and *non-dependable criteria* which was discussed in Chapter 4. The *financial risk assessment* in the post-interaction time phase includes the determination of the *levels* and the *magnitude* of *financial loss*. In other words, the level of financial loss shows how much extra resources a consumer has to invest to achieve the business objectives.

This chapter is organized as follows:

A framework for financial risk assessment is presented in Section 7.2. The determination of extra resource investment due to dependable and non-dependable criteria along with the loss curve is discussed in Section 7.3. Section 7.4 presents a

case study to demonstrate the working of the framework. The conclusion of the chapter is given in Section 7.5.

7.2 Conceptual Framework

The solution to the given problem consists of three steps: determining the extra resource investment based on dependable criteria, determining the extra resource investment based on non-dependable criteria and combining these assessments to determine the level and magnitude of financial loss. These steps are depicted in Figure 7.1:

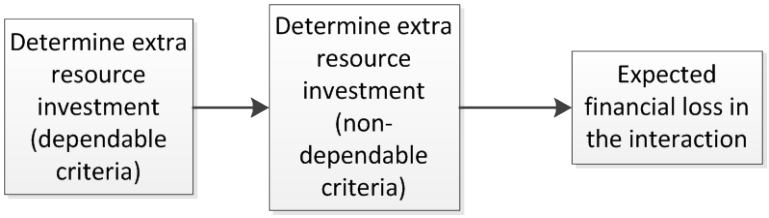

Figure 7.1: Steps of determining financial risk in an interaction

In this section, the conceptual frameworks to achieve the aforementioned steps are discussed. The frameworks to determine the extra resource investment (from dependable criteria) and the extra resource investment (from non-dependable criteria) are depicted in Figures 7.2 and 7.3, respectively.

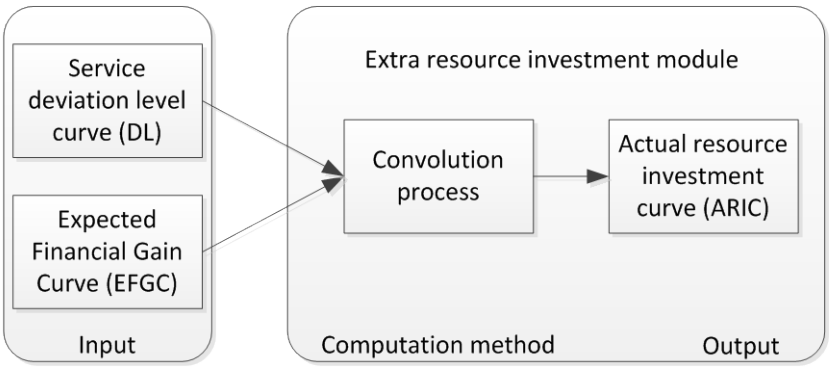

Figure 7.2: Conceptual framework for determining the extra resource investment (dependable criteria)

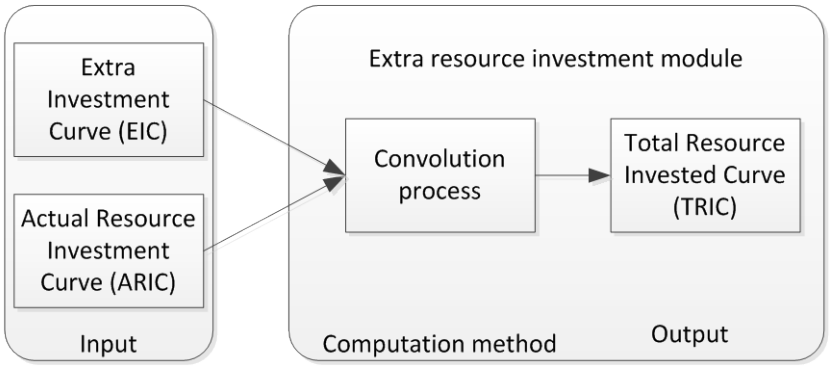

Figure 7.3: Conceptual framework for determining the extra resource investment (non-dependable criteria)

The components of these frameworks are discussed below:

Input: For dependable criteria such as response time and throughput, the input is in the form of the *service level deviation curve (DL)* that was determined on the basis of the performance of a service provider, as discussed in the previous chapter. The second input for Figure 7.2 is to determine the worth of financial resources that a consumer is going to invest in the interaction and this is determined by the *Expected Financial Gain Curve* (EFGC). The process of determining this curve will be discussed in detail in Section 7.3.

For Figure 7.3, the input *Extra Investment Curve* (EIC) represents the extra financial amount that a consumer needs to invest in case of financial loss due to non-dependable criteria. The other input for Figure 7.3 is the *Actual Resource Investment Curve* (ARIC) which represents the actual financial amount that a consumer needs to keep at stake due to the performance risk in the dependable criteria as opposed to the EFGC.

Extra resource investment module: This component of the proposed framework consists of the extra resource investment computation method and its output in both dependable and non-dependable criteria cases.

- ***Computation methods:*** In the literature, various techniques have been proposed for financial risk assessment. One such approach by which financial risk can be analyzed is convolution [1]. Convolution is the integral operator which expresses the amount of overlap and impact of one function as it shifts

over the other. Convolution has been used effectively in the risk assessment of applications such as power generation systems, to determine the expected demand not supplied by determining the effect of load demand on the generation capacity of the generation system. In this thesis, convolution is used to assess financial risk based on both dependable and non-dependable criteria. Further discussion of the convolution process is given in Section 7.3.1.1.

- ***Output:*** The output in the case of dependable criteria is the *Actual Resource Investment Curve* (ARIC) which is the actual financial amount that a consumer needs to keep at stake due to performance risk in dependable criteria as opposed to the EFGC. The output for non-dependable criteria is the *Total Resource Invested Curve* (TRIC) which represents the total extra financial resources that a consumer has to invest in an interaction due to loss based on dependable and non-dependable criteria to achieve the business objectives that were perceived at the start of the interaction.

The third and final step in determining financial risk, as depicted in Figure 7.1, is determining the expected loss in the interaction which is explained as follows:

Expected loss in the interaction: After determining the TRIC, a consumer has to analyse this curve to determine the levels of expected loss. Both magnitude and level of loss is important to enable a consumer to make an informed decision about the continuity of the service, using the *decision support system* that is proposed in the next chapter. To determine the magnitude of loss, the possibility theory is used to ascertain numerically the different levels of loss.

Although it is possible to ascertain the numerical level of loss in the interaction using probability distribution, there are several disadvantages to this approach, hence this thesis uses possibility distribution. In probability distribution, a non-zero value must be assigned to an element from its given set of UoD whose likelihood of occurrence is very high . Moreover, the sum of all probability distribution of the elements should equates to 1, no matter whatever value is assgiend to selected element [8]. Possibility distribution, on the other hand, does not prevent assigning the likelihood

of 1 to other elements if the likelihood of 1 is already assigned to an element [2]. The detail on *determining expected loss in the interaction* is given in Section 7.3.3.

To determine the levels of loss linguistically, a fuzzy inference system is used. A fuzzy inference system is a mathematical model that deals with uncertainties. Uncertainty may arise when a consumer has to make decision whether to continue with the current service provider or not on the basis of levels of loss that can be experienced and how this corresponds to his risk attitude. Humans have the capability of reaching a decision based on uncertainties using cognitive processes, which is better represented and used when the uncertainty is represented in linguistic terms rather than numbers [3]. So, in the proposed decision support system, the loss is not only represented numerically but also linguistically so that loss levels can be combined with the risk propensity of the consumer, which is also represented linguistically (as discussed in the next chapter) to compute the risk-based recommendation.

For determining linguistic levels of loss using the fuzzy inference system, input variables for the proposed *decision support system* such as *loss* need to be converted into fuzzy input variables by defining fuzzy membership functions. Fuzzy membership functions can be defined either based on the mathematical model developed on the basis of experiments on already existing systems or based on expert knowledge. In the context of the given problem, since no experimental model is available, expert knowledge is therefore utilized. To determine the shape of the membership functions for loss input variables, based on the opinions of several experts in the context of the given problem, trapezoidal curves are selected [4], as discussed in Section 7.3.3.

After presenting the conceptual frameworks, in the next section the overview of financial risk determination in the post-interaction time phase is discussed in detail.

7.3 Overview of extra resource investment determination in the post-interaction time phase

In this section, the overview of extra resource investment determination, based on dependable and non-dependable criteria, is explained.

7.3.1 Dependable assessment criteria for extra resource investment

The process of determining the *Actual Resource Investment Curve* (ARIC), based on dependable criteria, is depicted in Figure 7.4:

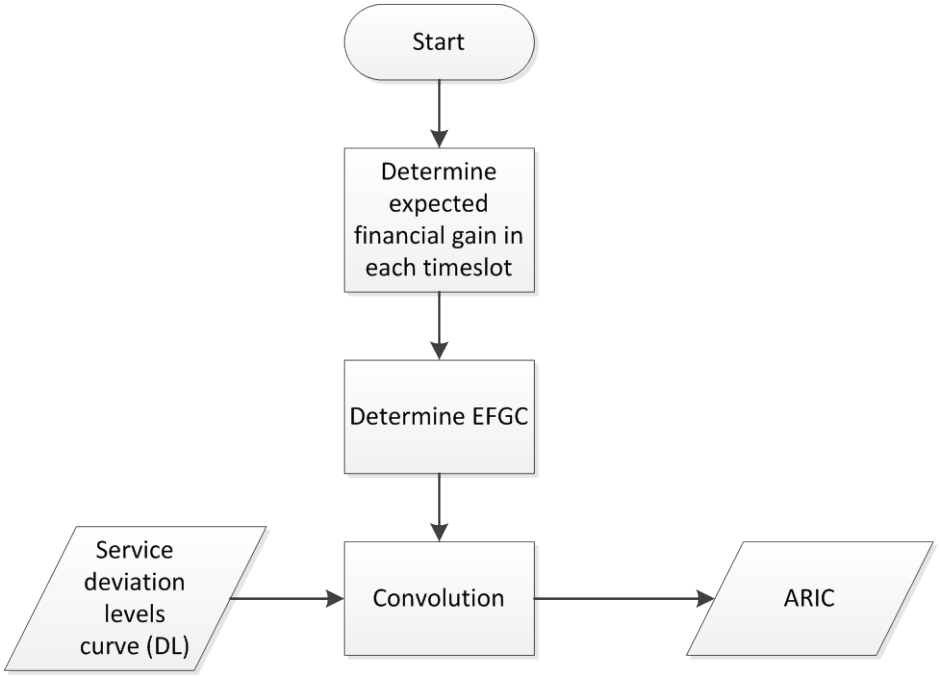

Figure 7.4: Process of determining ARIC

The steps in Figure 7.4 are explained below:

Determine expected financial gain in each timeslot: The consumer expects to achieve financial gain in each timeslot if the service is delivered as promised by the service provider.

The probability of financial gain or investment in each timeslot can be represented by financial gain or investment levels, as depicted in Figure 7.5, where x, y and z represent the timeslots in which the investment in dollars was made or, in other words, x, y and z are the timeslots representing the levels of investment:

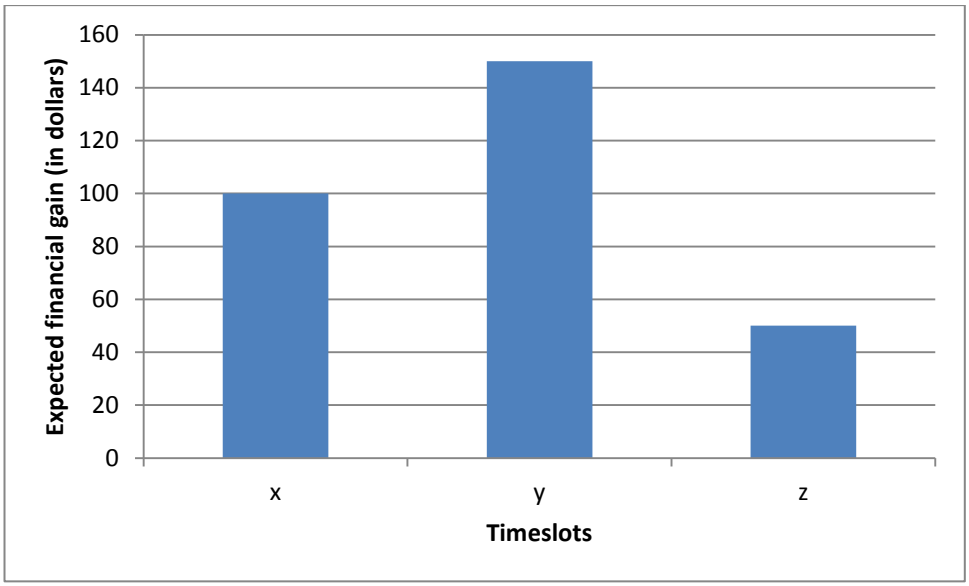

Figure 7.5: Expected financial gain in each timeslot

Determine EFGC: From the expected financial gain levels shown in Figure 7.5, the next step is to generate the *Expected Financial Gain Curve* (EFGC) which represents the cumulative distribution of the expected financial gain that was anticipated throughout the time period. This is achieved by plotting the *cumulative probability density function* of the amount of resources invested over the time period [5], [6].

Service deviation levels curve (DL): The service deviation levels show the variations in the performance of the service as determined in the previous chapter. These service deviation levels are represented by a random variable called *Deviation Levels (DL).*

Convolution: This step is the core process for determining the extra financial resources to be kept at stake on the basis of dependable criteria. As depicted in Figure 7.2, the convolution process is employed to determine the Actual Resource Investment Curve (ARIC). A discussion on how the convolution process is used in this thesis is given in the following sub-sections.

7.3.1.1 Determining Extra Financial Resource using Convolution

An introduction to convolution and its use in risk assessment was discussed earlier in this chapter. In this section, its application to the given problem is discussed.

Mathematically, the convolution of *X* and *Y*, both of which are independent random variables, is given by:

$$Z = X \oplus Y \quad (1)$$

For convolution, both the random variables need to be on a uniform scale. To use convolution for the given problem, the random variables *DL* and *EFGC* should be on same scale, therefore I normalize these variables to a percentage scale. Using equation (1), convolution for the given random variables can be written as:

$$ARIC = DL \oplus EFGC$$

where:
ARIC= Actual Resource Invested Curve,
DL= random variable indicating service deviation levels,
EFGC= Expected Financial Gain Curve.

In other words, convolution is the sliding of the *EFGC* over the projections of service deviation levels. To perform this sliding operation, the following formulae is used:

$$ARIC(x) = \sum_{i=1}^{n} p_i * EFGC(x - DL_i) \text{ if}(x - DL_i) \geq 0$$
otherwise
$$ARIC(x) = \sum_{i=1}^{n} p_i \text{ if}(x - DL_i) < 0$$
(2)
where:
n= the number of service deviation levels,
x= the point at which *ARIC* has to be determined,
DL_i= magnitude of service deviation level *i*,
p_i = magnitude of occurrence of service deviation level *i*,
EFGC(*x*-*DLi*)= Expected Financial Gain Curve at point (*x*-*DLi*).

The effect of the service deviation levels DL_i over each point of curve (*x*) is determined and after the convolution process, the *ARIC* curve is produced which represents the extra financial resources to be kept at stake due to the non-conformance of a service provider to the agreed SLA. The whole process of financial risk assessment based on dependable criteria is explained through an example in

Section 7.4. After determining the extra financial resources to be kept at stake based on dependable criteria, a consumer may also be interested to determine the additional cost of service migration in addition to financial loss in the form of the ARIC. This additional cost can be determined using non-dependable criteria since it is independent of the performance of the service provider. This is discussed in the next section.

7.3.2 Non-dependable assessment criteria for extra financial resources

The process of determining the *Total Resource Invested Curve* (TRIC) based on non-dependable criteria is depicted in Figure 7.6:

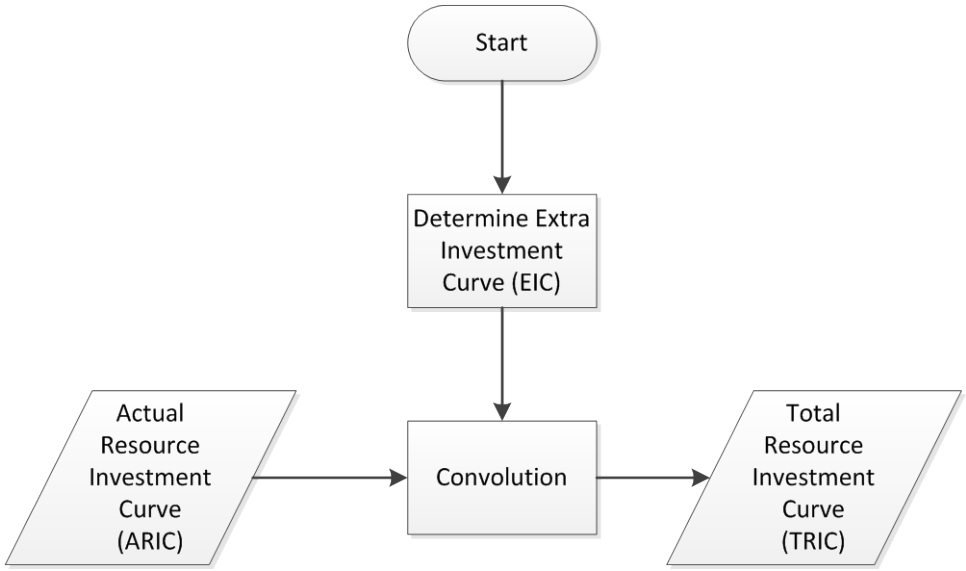

Figure 7.6: Process of determining TRIC

The steps in Figure 7.6 are explained below:

Determine extra investment curve: As discussed earlier, the *Extra Investment Curve* (EIC) represents the extra financial amount due to non-dependable events such as the service migration cost that a consumer needs to invest in case of service deviation of a service provider. This amount is in addition to the extra resource investment that has been determined due to dependable criteria. There may be two scenarios in relation to this extra payment. It may be a once-off payment or it may be spread over a period of time. In the second case, the probability mass function (pmf) is used to determine the distribution of payment over a period of time which is represented by

EIC and hence is one of the random variables for the determination of the total resource invested curve.

Convolution: After determining the EIC, the next step is to use it as input along with the ARIC to determine the TRIC. Since the ARIC and EIC are both random variables, their convolution is given by:

$$\mathrm{TRIC} = \mathrm{EIC} \oplus \mathrm{ARIC}$$

where:
TRIC= Total Resource Invested Curve
EIC= Expected Investment Curve
ARIC= Actual Resource Investment Curve.

To determine the TRIC, the following formulae is used:

$$\mathrm{TRIC}(x) = \sum_{i=1}^{n} p_i * \mathrm{ARIC}(x - l_i) \ \text{if}(x - l_i) \geq 0$$

otherwise

$$\mathrm{TRIC}(x) = \sum_{i=1}^{n} p_i \ \text{if}(x - l_i) < 0 \qquad (3)$$

where,
n= the number of extra financial resources needed for migration
x= the point of migration amount at which *TAIC* is to be determined
l_i= level of extra financial resource from *EIC*
p_i= probability of that extra level of financial resource
ARIC($x - l_i$)= Actual Resource Investment Curve value at point ($x - l_i$).

After the curve has been obtained, the TRIC is compared with the ARIC to indicate the extra level of resources that a consumer has to keep at stake for the migration of the service. Using the TRIC, a consumer has to determine the level and magnitude of loss. This process is explained in the next section

7.3.3 Determine the loss curve due to extra resource investment

After determining the extra resource investment due to dependable and non-dependable criteria, a consumer needs to determine how much loss has occurred, as depicted in Figure 7.1. As discussed in Section 7.2, a consumer may experience different levels of loss in an interaction and it is proposed that a consumer should

determine the magnitude and levels of loss numerically and linguistically. The process of determining levels and magnitude of loss in an interaction is discussed, as depicted in Figure 7.7:

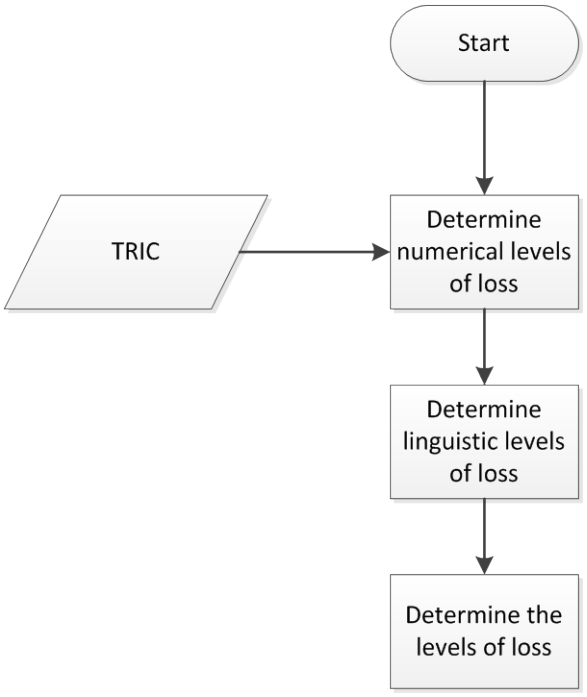

Figure 7.7: Process of determining levels of loss

The steps in Figure 7.7 are explained below:

Determine numerical levels of loss: This step uses the TRIC that is obtained in the previous section and converts it into a curve with numerical levels of loss.

To determine the numerical levels of loss from the TRIC, the possibility theory is used. The likelihood of occurrence of an element termed as 'degree of evidence' is represented as 'm(E)', where 'E' is the event or an element from the universe of discourse of the variable 'L' which represents the possibility distribution of loss in the interaction. The UoD of the loss variable is defined by the following set:

Loss ={0,1,2,3,…,100} where each element has a unit of %. To determine the numerical level of loss in the ineraction using the possibility distribution, the consumer should determine the focal elements with their degree of membership from UoD which ranges from {0 … 100}. For the variable 'L', from UoD, those elements for which degree of membership is greater than 0 are called focal elements.

In order to ascertain the numerical level of loss in the interaction by using possibility distribution, the consumer should first identify the focal elements along with their degree of evidence from the universe of discourse which ranges from {0…100}. From the universe of discourse, those elements with a degree of evidence greater than zero are called the 'focal elements' of the variable 'L'. These elements represent the impact of loss in the interaction. The degree of membership of an element from the UoD should be in the interval [0,1] and the cumulative sum of the degree of membership of all the focal elements from the UoD should satisfy the condition [2]:

$$\sum_{E \in L} m(E) = 1 \quad (4)$$

where:

'E' represents the focal elements belonging to the variable 'L',

m(E) represents the degree of membership of the focal element.

To determine the focal elements and possibility distribution of loss in the interaction, the consumer, from his *maximum investment capacity,* should determine the levels of extra financial resources that he has to keep at stake while interacting with the service provider.

The *maximum investment capacity* of a consumer is defined as the maximum extent to which he can invest his resources while interacting with the service provider. To achieve this:

- The consumer should determine the probability mass function (pmf) of the TRIC in interacting with the service provider. The pmf of the TRIC shows the probability of an amount from the required resources that the consumer has to keep at stake throughout the duration of interaction.
- The consumer should then determine the point on pmf of the TRIC which represents his maximum investment capacity in the interaction, which is termed as 'x'.
- The ordinates of the TRIC after point 'x' are taken as the focal elements from its UoD to represent the loss in the interaction.

- The portion of TRIC after point 'x' should be converted to a loss distribution by determining the pmf of that portion.
- The scale of loss distribution should be normalized in the range between 0 – 100 so that it can be used in the next step to determine the linguistic levels of loss.

Determine linguistic levels of loss: To determine the linguistic level of loss in an interaction, a fuzzy input variable *Loss* is used which is one of the input variables for the *risk-based decision support system* as discussed in Chapter 4. The UoD of the input membership function *Loss* is defined in the range of {0, 1, 2, 3, ... 100}. The UoD is divided into six fuzzy sets using the terms, *Extremely Low*, *Low*, *Low Medium*, *Medium High*, *High* and *Extremely High*. The membership function of the input Loss in an interaction is defined, using a trapezoidal curve, as shown in Figure 7.8:

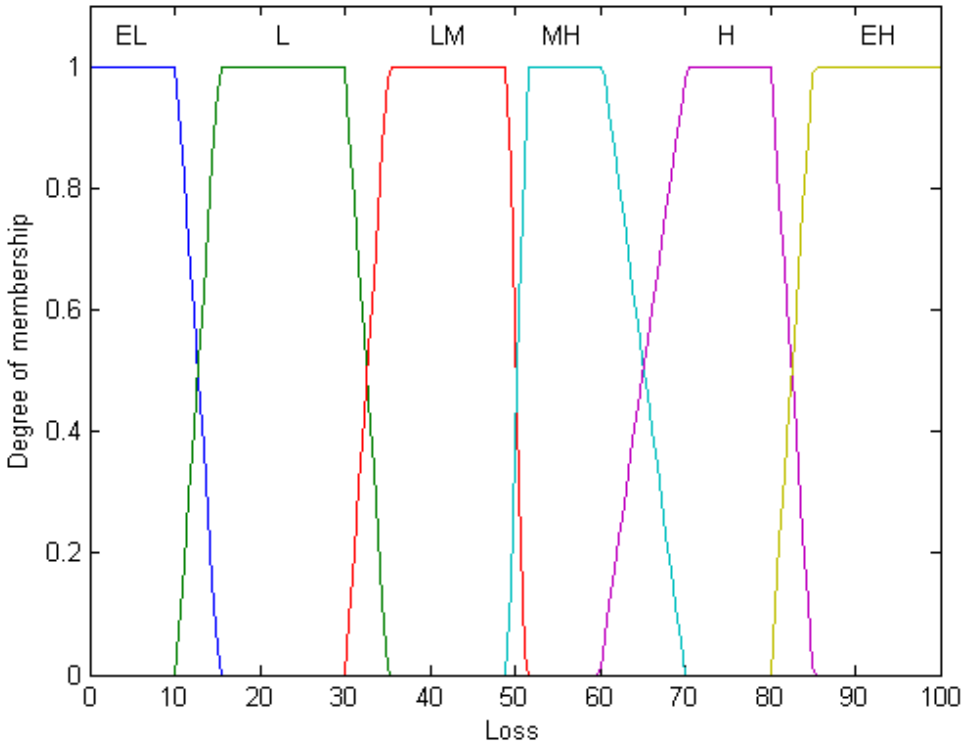

Figure 7.8: Membership function of the input Loss

The corresponding membership function for each predicate within the UoD for the input variable is defined as:

$$\mu_{\text{Extremely Low}}(\text{Loss}) = \begin{cases} 1 \text{ if } 0 < x < 10 \\ \frac{15-x}{5} \text{ if } 10 < x < 15 \\ 0 \text{ if } 15 < x < 100 \end{cases}$$

$$\mu_{\text{Low}}(\text{Loss}) = \begin{cases} 0 \text{ if } 0 < x < 10 \\ \frac{x-10}{5} \text{ if } 10 < x < 15 \\ 1 \text{ if } 15 < x < 30 \\ \frac{35-x}{5} \text{ if } 30 < x < 35 \\ 0 \text{ if } 35 < x < 100 \end{cases}$$

$$\mu_{\text{Low Medium}}(\text{Loss}) = \begin{cases} 0 \text{ if } 0 < x < 30 \\ \frac{x-30}{5} \text{ if } 30 < x < 35 \\ 1 \text{ if } 35 < x < 45 \\ \frac{50-x}{5} \text{ if } 45 < x < 50 \\ 0 \text{ if } 50 < x < 100 \end{cases}$$

$$\mu_{\text{Medium High}}(\text{Loss}) = \begin{cases} 0 \text{ if } 0 < x < 45 \\ \frac{x-45}{5} \text{ if } 45 < x < 50 \\ 1 \text{ if } 50 < x < 65 \\ \frac{70-x}{5} \text{ if } 65 < x < 70 \\ 0 \text{ if} 70 < x < 100 \end{cases}$$

$$\mu_{\text{High}}(\text{Loss}) = \begin{cases} 0 \text{ if } 0 < x < 65 \\ \frac{x-65}{5} \text{ if } 65 < x < 70 \\ 1 \text{ if } 70 < x < 80 \\ \frac{85-x}{5} \text{ if } 80 < x < 85 \\ 0 \text{ if } 85 < x < 100 \end{cases}$$

$$\mu_{\text{Extremely High}}(\text{Loss}) = \begin{cases} 0 \text{ if } 0 < x < 80 \\ \frac{x-80}{5} \text{ if } 80 < x < 85 \\ 1 \text{ if } 85 < x < 100 \end{cases}$$

The degree of membership (DOM) of a variable to a fuzzy set defines the magnitude of participation of that variable within that fuzzy set. The DOM of a variable is determined by inserting the selected input parameter into the horizontal axis of its UoD and taking its projection vertically at the point of intersection with the fuzzy sets. The DOM to the fuzzy set is represented by the point on the vertical axis to which the input variable intersects.

Determine the levels of loss: By using the membership function, the focal elements of variable 'L' should be converted into fuzzy sets, based on the level of truth to which the variable 'L' quantifies to the fuzzy sets. To convert a focal element *x* of variable 'L' to the defined fuzzy sets the, consumer needs to determine the possibility to which that element *x* corresponds with the defined fuzzy sets of fuzzy variable *Loss.* This is achieved by seeing the overlap between degree of membership of the input value *x*, with the degree of membership to which that input value *x* corresponds to a particular fuzzy set from the membership function. The possibility that the fuzzy set 'A' of the variable *Loss* will occur, based on the degree of membership of input *x*, is given by [7]:

$$\Pi(A) = \max\{\min[\pi(x), \mathrm{DOM}_A(x)]\} \quad (5)$$

Equation (5) is repeated for each focal element *x* from the UoD for variable 'L' to determine the possibility of occurrence of a fuzzy set 'A'. The possibility values obtained by this process represent the levels and magnitude of loss which can be utilized in the decision making process, which will be discussed in the next chapter. To explain the financial risk assessment approach proposed in this chapter, the case study that was introduced in the previous chapter is used.

7.4 Example of Determining Financial Risk in Cloud Computing

Continuing the case study from the previous chapter, consumer 'A', through his investment of resources in service provider 'B', wants to benefit financially over a specific period of time. However, due to service degradation, the expected outcome may not be achieved which may lead to the service user experiencing a financial loss. The service user expects to achieve financial gain in each time interval if the service is delivered as promised by the service provider. From such an expected financial gain in each timeslot over a time period, it is proposed that the TP SLA Manager plots the *Expected Financial Gain Curve* (EFGC). In the case study, consumer 'A', over a period of 24 hours, expects to gain a benefit of $15,000 as a result of a service provided by service provider 'B'. In this case study, the time space is 24 hours and is divided into 3 timeslots of 8 hours each. In 24 hours, as a result of a video conference service, let us suppose that the service user expects to achieve a financial gain, as shown in Figure 7.9.

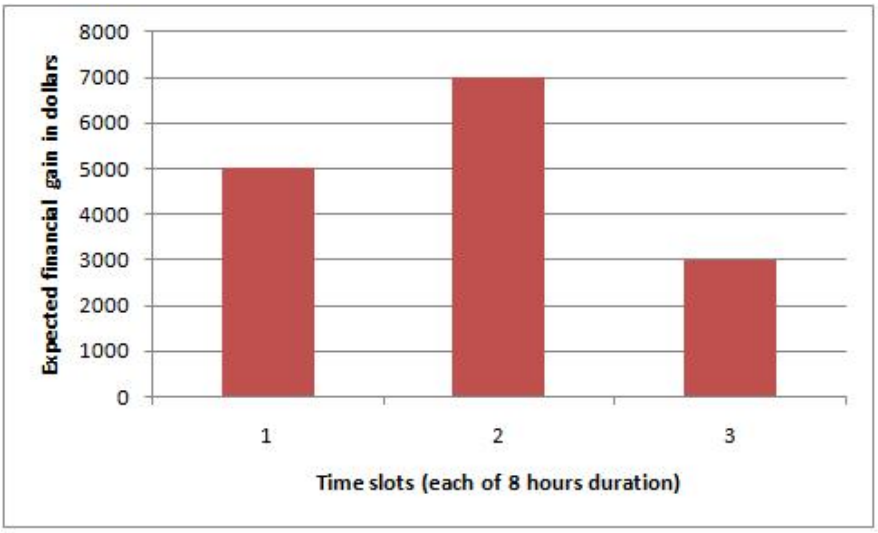

Figure 7.9: Expected financial gain in each timeslot

The *Expected Financial Gain Curve* (EFGC) can be generated from the expected financial gain levels in each slot, as given in Figure 7.9. The EFGC is depicted in Figure 7.10.

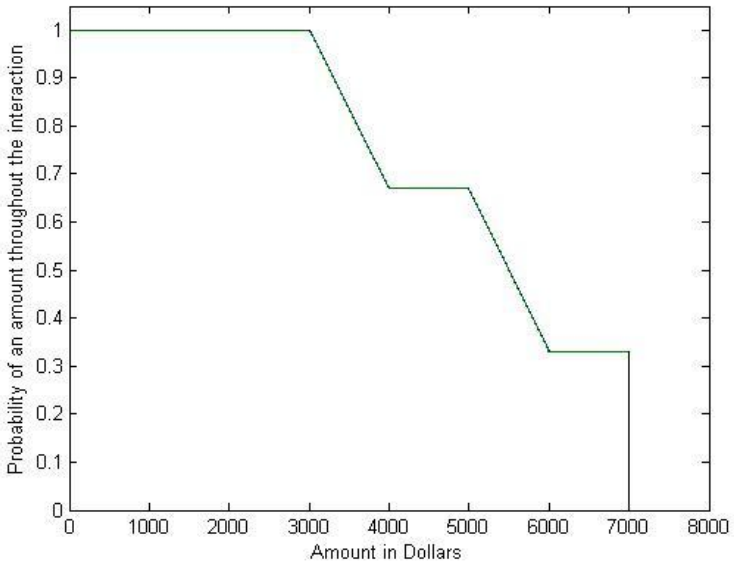

Figure 7.10: Expected Financial Gain Curve

To determine the *Actual Resource Investment* Curve (ARIC), *Deviation Levels* (DL) are used for the case study depicted in Figure 6.10. Using equation (2), the ARIC is determined which is depicted in Figure 7.11:

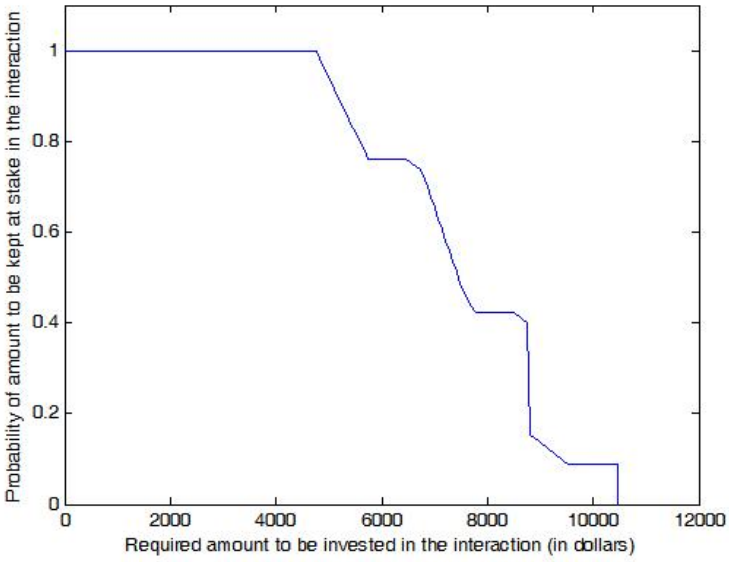

Figure 7.11: Curve representing the financial amount to be kept at stake due to service degradation

The inflated curve shown in Figure 7.11 can be used to determine the TRIC. Now, in the case of the *Actual Resource Invested Curve* (ARIC) if consumer 'A' wants to migrate a service to another service provider, then additional costs add to this curve. Let us consider that the migration cost for consumer 'A' to transfer a service to another provider is $3000 and that consumer 'A' opts to make a payment over a period of the next 24 hours. The Extra Investment Curve (EIC) is depicted in Figure 7.12:

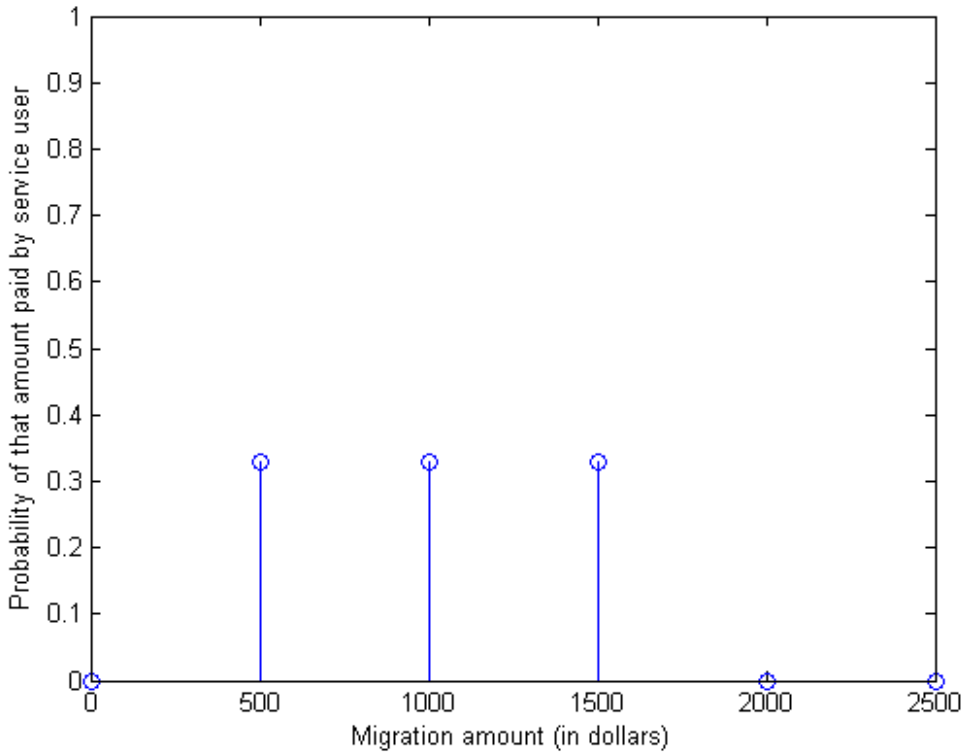

Figure 7.12: The probability of an extra level of resources for migration costs

To determine the TRIC using the random variables ARIC and EIC, equation (3) is used. The resultant curve, the TRIC is then compared with the ARIC to show the inflation. as depicted in Figure 7.13.

The TRIC can also be compared with EFGC to show the inflation as compared to original investment in the interaction. This comparison is depicted in Figure 7.14.

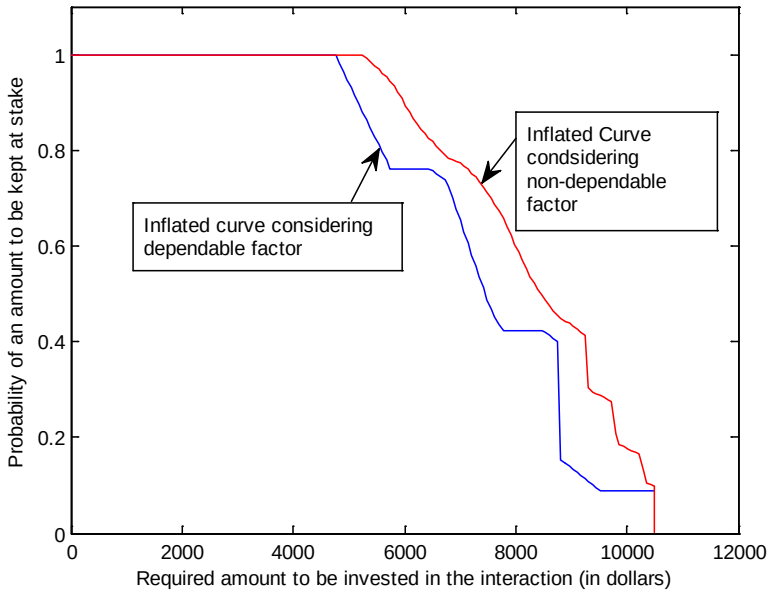

Figure 7.13: Comparison of Actual Resource Investment Curve and Total Resource Investment Curve

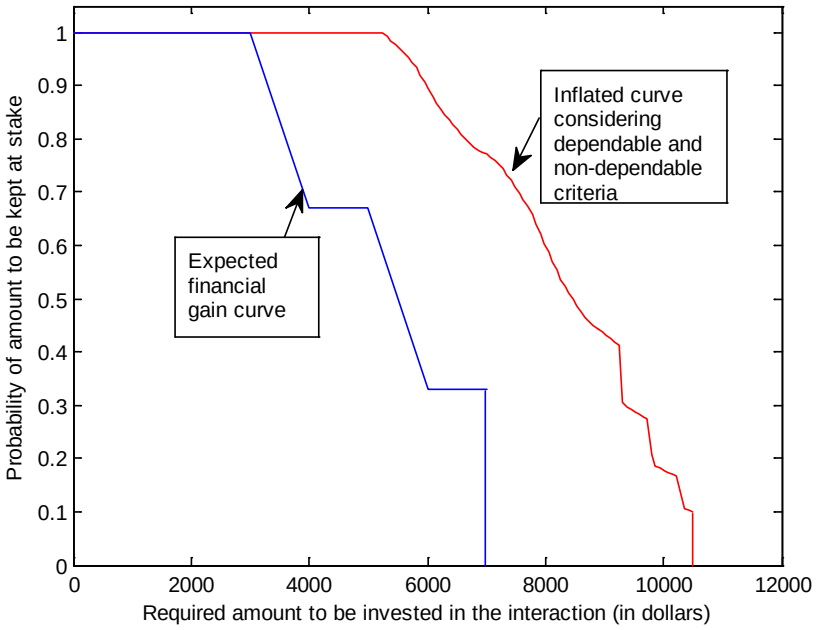

Figure 7.14: Comparison of Expected Financial Gain Curve and Total Resource Investment Curve

Next, to determine the numerical level of loss by using possibility theory, let us consider the TRIC which consumer 'A' ascertains in interacting with service provider 'B' and assume that the maximum investment capacity of consumer 'A' in the interaction is $7,000. Based on his maximum investment capacity, consumer 'A' can execute the steps mentioned in the previous section to identify the focal elements of the loss of the interaction, along with their degree of evidence to transform it into a possibility distribution. After the point of maximum investment capacity, the TP SLA Manager needs to determine the level to which the consumer needs extra financial resources in the interaction. The maximum investment capacity point on the abscissa is represented by 'X', as shown in Figure 7.15. On the TRIC, this point represents the probability of the consumer not achieving the full benefit of the expected returns of his resources in the interaction, as opposed to what was expected by the consumer at the start of the interaction.

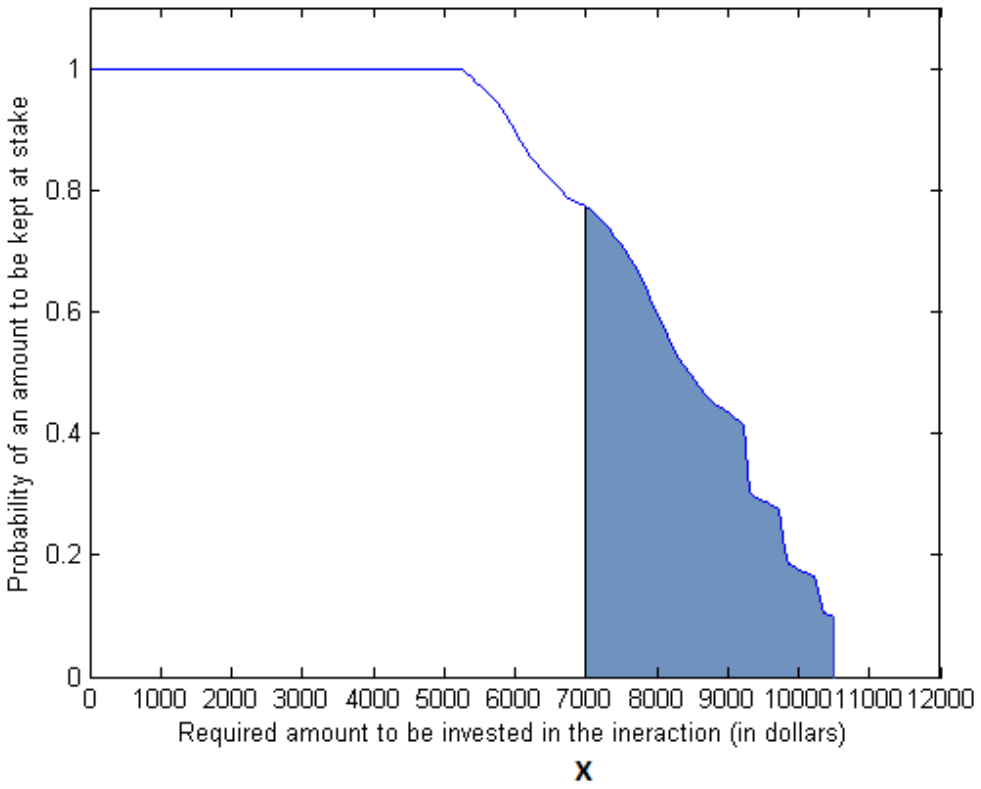

Figure 7.15: Total Resource Investment Curve with maximum loss bearing capacity point 'X'

The shaded area in Figure 7.15 indicates the probability of financial loss due to performance degradation of service provider 'B'. The focal elements from the UoD of variable 'L' are represented in Figure 7.16.

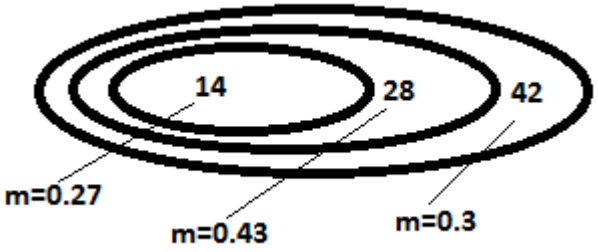

Figure 7.16: The focal elements and their degrees of evidence for variable 'L'

To determine the linguistic level of loss in the interaction, the TP SLA Manager transforms the focal elements to identify the fuzzy sets using equation (5). This process results in fuzzy sets for the input variable *Loss* as:

$$\Pi_{\text{Loss}}(\text{EL}) = \max\{\min[\pi(x), \text{DOM}_{\text{EL}}(x)]\} = 0.2$$

$$\Pi_{\text{Loss}}(\text{L}) = 0.43$$

$$\Pi_{\text{Loss}}(\text{LM}) = 0.3$$

$$\Pi_{\text{Loss}}(\text{MH}) = 0$$

$$\Pi_{\text{Loss}}(\text{H}) = 0$$

$$\Pi_{\text{Loss}}(\text{EH}) = 0$$

Once the fuzzy sets, along with the possibility of occurrence of each predicate from the membership function of the input variable *Loss* have been determined, the TP SLA Manager can determine the impact of these loss levels on the risk attitude of the consumer to obtain the recommendation to proceed or not proceed in the interaction, as discussed in the next chapter.

7.5 Conclusion

In this chapter, the financial risk using the dependable and non-dependable assessment criteria is determined and analysed. The expected financial resources that

the consumer intends to invest in the interaction are determined. The impact of service deviation levels and the migration cost on the consumer's financial expectation is determined using a technique called 'convolution'. The result of convolution is then used to determine the level and magnitude of loss that a consumer may experience as a result of service degradation and migration cost. The magnitude and level of loss determined in this chapter will be used as an input to the risk-based decision support system in the next chapter to enable the consumer to make an informed decision about the continuity of the service.

7.6 References

[1] W. Li, *Risk assessment of power systems models, methods, and applications*, John Wiley & Sons, 2005.

[2] D. Dubois, H. Prade, and E. Harding, *Possibility theory: an approach to computerized processing of uncertainty* vol. 2: Plenum press New York, 1988.

[3] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing", *Communications of the ACM,* vol. 37, pp. 77-84, 1994.

[4] B. Bouchon-Meunier, M. Dotoli, and B. Maione, "On the choice of membership functions in a mamdani-type fuzzy controller", 1996.

[5] O. Hussain, T. Dillon, F. K. Hussain, and E. Chang, "Probabilistic assessment of financial risk in e-business associations", *Simulation Modelling Practice and Theory,* vol. 19, pp. 704-717, 2011.

[6] R. E. Walpole, R. H. Myers, S. L. Myer, and K. Ye, *Probability and statistics for engineers and scientists* Vol.8: Prentice Hall Upper Saddle River NJ, 1993.

[7] R. C. Berkan and S. Trubatch, *Fuzzy System Design Principles*: Wiley-IEEE Press, 1997.

[8] O. Hussain, "Risk measurement, prediction and management in e-business and service oriented environment / Omar Khadeer Hussain", PhD, Curtin University, 2008.

CHAPTER 8

RISK-BASED DECISION SUPPORT SYSTEM

8.1 Introduction

In the previous chapter, the process of *determining financial loss* in the post-interaction time phase was discussed. The next step of *SLA management* in the post-interaction time phase is to enable a consumer to make an informed decision about the continuity of the service by combining the financial loss levels and the consumer's risk attitude. As mentioned in Chapter 4, this will be achieved by a *risk-based decision support system*. A risk-based decision support system consists of two inputs, namely *Loss* and *Risk Propensity* and an output which is a *recommendation* to a consumer as to whether to continue interaction with the service provider or not.

The different levels of loss in an interaction were determined in the last chapter. As highlighted in Chapter 3, in relation to decision making, the risk attitude of a consumer should be considered as it is one of the important factors that influence the decision. Risk Propensity indicates the current tendency of a consumer to the level of risk in an interaction and is the main factor in decision making [1]. It should be noted that no two consumers have the same risk attitude which affects their decision making. Also, the risk attitude of a consumer might not be same throughout. It is therefore important to consider the accurate Risk Propensity of a consumer at a given period of time and to utilize it with the level of loss to determine the recommended decision. The focus of this chapter is to consider the risk attitude of the consumer and determine its impact on loss levels.

Similar to loss levels, Risk Propensity levels are also uncertainties which can be represented by an input fuzzy variable [2]. With two fuzzy input variables, namely *Loss* and *Risk Propensity,* A fuzzy inference system is proposed to develop a *risk-based decision support system*. The main aim of this chapter is to discuss the detail of the proposed fuzzy inference system. This chapter is organized as follows: Section 8.2 discusses an overview of the risk-based support system. It presents the *fuzzy inference model* that I will use to develop the decision support system. The detail of *fuzzification* of the input and output variables, *fuzzy rules* and *defuzzification,* is discussed in Section 8.3. The demonstration of the proposed risk-based decision

support system is given in Section 8.4 using a case study. Section 8.5 is the conclusion of this chapter.

8.2 Overview of the risk-based decision support system

As discussed in Chapter 5, to capture human reasoning for the decision-making process, a *fuzzy inference system* is used. A fuzzy inference system for a risk-based decision support system with inputs and output is depicted in Figure 8.1:

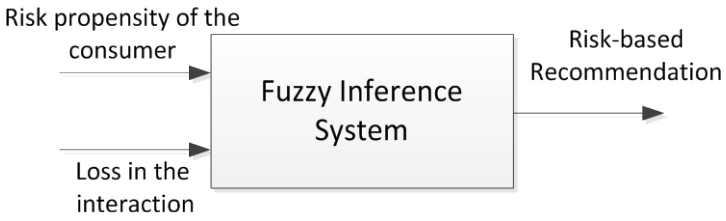

Figure 8.1: Overview of the fuzzy inference system for decision making

Based on Figure 8.1, the risk-based decision making process consists of the following steps as shown in Figure 8.2.

Determine Risk Propensity Level: The process starts with determining the risk attitude or *Risk Propensity* (RP) level of a consumer. The fuzzification of this input variable is discussed in detail in Section 8.3.1, in which categories of risk attitudes are defined.

Determine Maximum Risk Attitude: Based on the Risk Propensity level, the *Maximum Risk Attitude* (MRA) of a consumer is defined which represents the highest level of risk taking nature of a consumer in the interaction.

Determine Maximum Acceptable Loss Level: Based on the MRA, the consumer then needs to determine the *Maximum Acceptable Loss Level* (MALL) which represents the maximum loss level that a consumer accepts, based on his risk attitude.

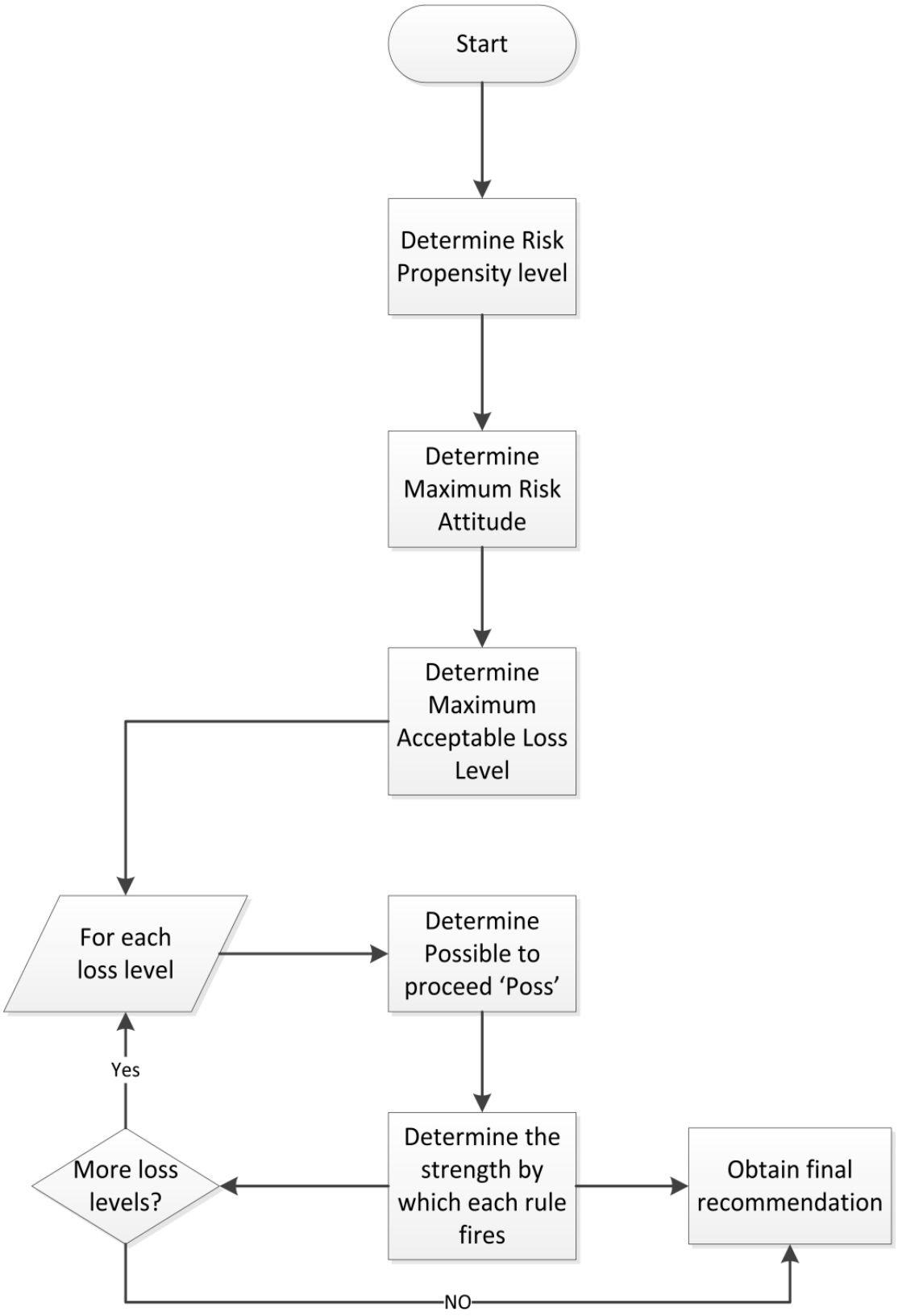

Figure 8.2: Steps in obtaining recommendations through the decision support system

Determine Possible to Proceed Poss: This step uses each *loss level* for the input variable *Loss* that was determined in the previous chapter, which is termed *Current Loss Level* (CLL). On the basis of CLL and MALL, an output represented by a variable *Poss* is determined. This variable determines whether it is possible to proceed with the service provider or not, given this level of loss. This process is repeated for each loss level (CLL).

Determine the strength by which each rule fires: On the basis of the *Poss* value, the recommendation is determined. If it is possible for the consumer to proceed in the interaction, then the recommendation from the fuzzy inference system is *Proceed,* otherwise the recommendation is *Don't Proceed.* This recommendation is represented by an output variable of the fuzzy inference system called *Risk-based Recommendation* (RR) and it is based on MRA, MALL, CLL and *Poss*. For more than one CLL, this process is repeated.

Obtain final recommendation based on aggregation: To quantify the output fuzzy variable, RR aggregation is done using the *Root Sum Square* method. The aggregation results in the output range on the output fuzzy sets. This range is then converted to a single crisp value using the defuzzification process. The crisp output represents the final recommendation from the fuzzy inference system to the consumer. This recommendation may be interpreted as a percentage on whether to proceed or not to proceed using output fuzzy sets.

In the next section, the working process of the fuzzy inference system is explained in detail.

8.3 Risk-based Decision Support System

In this section, the whole process of risk-based recommendation is discussed. To achieve this, the process of fuzzification, rule-base and defuzzification for fuzzy inference system [3] is explained in detail.

8.3.1 Fuzzification of the input variable: Risk Propensity (RP)

Figure 8.1 shows the fuzzy input variables *Loss* and *Risk Propensity* (RP). *Loss* was discussed in the previous chapter. The fuzzification of the second variable, RP, is discussed in this section.

To capture the different types of risk attitudes of a service consumer, three types of risk attitudes of a consumer are proposed for the fuzzy variable for Risk Propensity. They are:

- Risk Averse (RA): The consumer with an RA type of attitude can only take minimal risk in an interaction.
- Risk Neutral (RN): The consumer with an RN type of risk attitude proceeds in an interaction only if the advantages that he will achieve will balance the costs involved.
- Risk Taking (RT): The consumer with this type of risk attitude is ready to take risk no matter what level of loss may occur in an interaction.

The risk attitudes of the consumer in terms of accepting the magnitude of loss in the interaction can be arranged in the order of RT>RN>RA. This order is based on the assumption that consumers with a Risk Propensity level of risk averse accept a magnitude of loss up to EL = 1, consumers with a risk neutral type of risk attitude accept a magnitude of loss up to LM=1, and consumers with a risk taking type of risk attitude accept a magnitude of loss up to EH=1 in interacting with a service provider. Since the consumer has to select the risk attitude from the spectrum, it might not be a crisp value as it might overlap across different risk attitudes.

To define the fuzzy input variable called *Risk Propensity,* the universe of discourse (UoD) chosen in the range of 1-5; {1, 2, …, 5} is defined where each element represents a numeric value and is unit-less. Three fuzzy sets for input variable Risk Propensity are defined where each fuzzy set represents a type of risk attitude of the consumer, as discussed above. Therefore, the UoD is divided into three predicates, *Risk Averse*, *Risk Neutral* and *Risk Taking*. In this thesis, the input variable *Risk Propensity* is represented by a combination of a triangle and straight lines. The membership function is used to determine the strength to which a value from the UoD of the input variable quantifies to the defined predicates of the risk attitude of the consumer, as depicted in Figure 8.3.

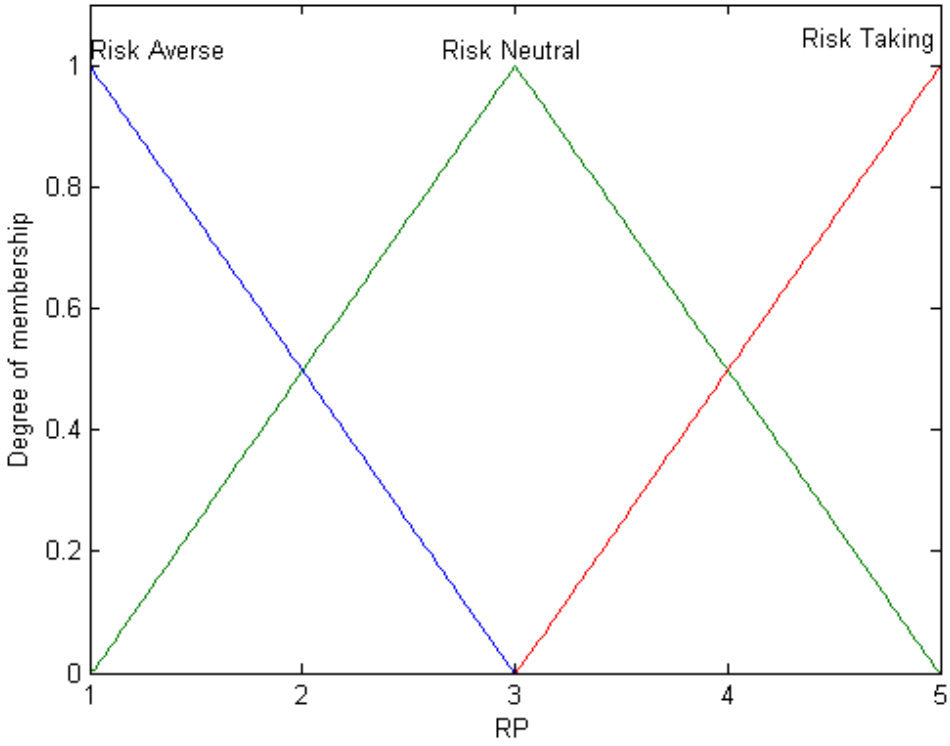

Figure 8.3: Membership function for Risk Propensity

For this input variable, the corresponding membership function is defined as:

$$\mu_{\text{Risk Averse}}(\text{RP}) = \begin{cases} \frac{3-x}{2} \text{ if } 1 < x < 3 \\ 0 \text{ if } 3 < x < 5 \end{cases}$$

$$\mu_{\text{Risk Neutral}}(\text{RP}) = \begin{cases} \frac{x-1}{2} \text{ if } 1 < x < 3 \\ \frac{5-x}{2} \text{ if } 3 < x < 5 \end{cases}$$

$$\mu_{\text{Risk Taking}}(\text{RP}) = \begin{cases} 0 \text{ if } 0 < x < 3 \\ \frac{x-3}{2} \text{ if } 3 < x < 5 \end{cases}$$

8.3.2 Fuzzification of output variable: Risk-based Recommendation (RR)

To define the fuzzy output variable called RR, the UoD in the range of 0-10; {0, 1, 2, …, 10} is defined. Since the aim of the proposed fuzzy inference system is to assist the consumer in informed decision making, the output variable (RR) consists of two fuzzy sets with predicates 'Proceed (P)' and 'Don't Proceed (DP)'. These predicates represent the possibilities that a consumer considers while making a decision. The

fuzzy sets are represented by straight lines spread over the UoD, as shown in Figure 8.4.

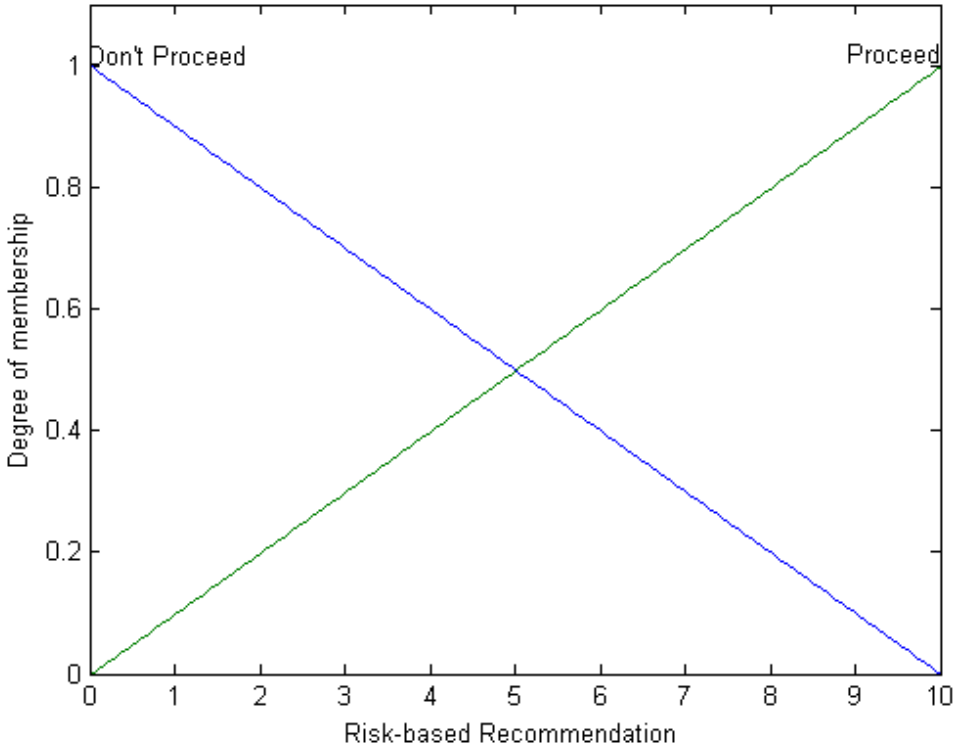

Figure 8.4: Membership function for Risk-based Recommendation

The corresponding membership function for each predicate within the UoD of the output variable is defined as:

$$\mu_{\text{Don't Proceed}}(\text{RRD}) = \frac{10-x}{10} \text{ if } 0 < x < 10$$

$$\mu_{\text{Proceed}}(\text{RRD}) = \frac{x-0}{10} \text{ if } 0 < x < 10$$

After the fuzzification of input and output variables, the rules for the proposed fuzzy inference system are defined, as discussed in the next section.

8.3.3 Rules for Fuzzy Inference System for Risk-based Recommendation

As previously discussed, to process the Risk Propensity of the consumer, the fuzzy nature of the risk attitude of the consumer needs to be taken into consideration which means that, with a change in Risk Propensity levels, the way the consumer assesses

the loss in the interaction changes. Therefore, it is important to accurately ascertain the Risk Propensity level. For example, the Risk Propensity of a consumer with a risk averse nature with (RA)=1 is utilized differently to determine the loss, compared to a consumer whose Risk Propensity nature is a combination of Risk Averse (RA) and Risk Neutral (RN) levels which are intermediate levels of Risk Propensity. This implies that a consumer should consider accurate Risk Propensity levels according to its Degree of Membership (DOM), before ascertaining its impact on the loss in an interaction with the service provider.

To determine the effect of the intermediate levels of the consumer's Risk Propensity on the level of loss in the interaction, the acceptable levels of loss by the consumer are defined in interacting with the service provider with the help of fuzzy rules, according to the various levels of his Risk Propensity. Additionally, for Risk Propensity levels that are combinations of different risk attitudes, the *Maximum Risk Attitude* (MRA) of the consumer is defined and based on this, the *Maximum Acceptable Loss Level* (MALL) is determined in the interaction while defining the fuzzy rules. As discussed in Section 8.3.1, the risk attitude of the consumer in relation to bearing loss can be arranged in order of RT>RN>RA. Hence, if a consumer's Risk Propensity is a combination of levels RA and RN with DOM of 0.3 and 0.7, respectively, then the maximum risk attitude of the consumer is RN=0.7. Based on this Risk Propensity level, by using fuzzy rules, the consumer should determine the maximum level of loss which can be termed as acceptable in interacting with the service provider. In Table 8.1, the fuzzy rules are defined which give the MALL by the consumer according to his RP nature in the interaction.

MRA			MALL
IF	RA=1	THEN	EL=1
IF	RN=0.1	THEN	L=0.2
IF	RN=0.2	THEN	L=0.4
IF	RN=0.3	THEN	L=0.6
IF	RN=0.4	THEN	L=0.8
IF	RN=0.5	THEN	L=1
IF	RN=0.6	THEN	LM=0.2
IF	RN=0.7	THEN	LM=0.4
IF	RN=0.8	THEN	LM=0.6
IF	RN=0.9	THEN	LM=0.8
IF	RN=1	THEN	LM=1
IF	RT=0.1	THEN	MH=0.34
IF	RT=0.2	THEN	MH=0.67
IF	RT=0.3	THEN	MH=1
IF	RT=0.4	THEN	H=0.34
IF	RT=0.5	THEN	H=0.67
IF	RT=0.6	THEN	H=1
IF	RT=0.7	THEN	EH=0.34
IF	RT=0.8	THEN	EH=0.67
IF	RT=0.9	THEN	EH=0.99
IF	RT=1	THEN	EH=1

Table 8.1: Fuzzy rules to determine Maximum Acceptable Loss Level according to the Risk Propensity of the consumer

In the previous chapter, the process by which the consumer determines the linguistic level of loss was determined by the predicates and magnitude of occurrence of those predicates. It is quite possible that there is more than one predicate of loss in an interaction with the service provider at a given time. Each predicate of loss needs to be taken in turn to assess whether it is acceptable or not according to the risk attitude of the consumer to manage the loss in an interaction. Three possibilities arise here according to the Risk Propensity nature of the consumer:

- the levels of loss in the interaction are acceptable to the consumer in full;
- the levels of loss in the interaction are not acceptable to the consumer at all;
- the levels of loss in the interaction are somewhat acceptable to the consumer.

For example, if the Risk Propensity of the consumer is RT=1, then his MALL is EH=1 according to Table 8.1 which indicates that he can accept all levels of loss in an interaction with the service provider. Contrary to this, if the Risk Propensity of a consumer is RA=1, then all levels of loss are beyond his maximum acceptable level. When a consumer's Risk Propensity is RN=0.6 and RT=0.4, then he partially accepts the level of loss in the interaction with the service provider. In each case, the recommended output from the fuzzy inference system to the consumer depends upon the level of his acceptance of the loss in the interaction, based on his risk attitude nature.

To process such cases in the proposed fuzzy inference system, a variable called *Possible to Proceed in the Interaction at this stage* (*Poss*) is introduced, as discussed in previous section, at the time of defining the fuzzy rules for the recommended risk-based decision output. This variable is determined for each fuzzy set of loss and it represents whether or not this fuzzy set, along with its magnitude of occurrence is acceptable to the consumer according to his risk attitude. Then consumer decides whether to proceed or not to proceed in the interaction based on consumer's acceptance level. Risk attitude is represented by a variable called *Current Risk Attitude* (CRA), as shown in Table 8.2. For each predicate of loss, there will be output *Proceed* or *Don't Proceed*. If in an interaction there is more than one predicate, then the process of determining the value for the variable *Poss* should consider each of these predicates, as depicted in Figure 8.2. The predicate of loss which the consumer currently examines to determine the value of the variable *Poss* is termed *Current Loss Level* (CLL).

To define the fuzzy rules for the fuzzy inference system to assist in decision making, the predicates for the input variable *Risk Propensity* is defined, as shown in Figure 8.3 and for the input variable *Loss*, as shown in Figure 7.8. Hence, the total rules for our fuzzy system are: 3X6=18. To model the fuzzy rules, the IF-THEN-ELSE structure is used. In the antecedent, the two inputs are connected through the AND operator. The consequent part consists of output *Proceed* (P) if the value of the variable *Poss* is 1 and if the value of *Poss* is 0, then the output is *Don't Proceed* (DP). The rules for the system to make a risk-based decision in an interaction are given in Table 8.2.

	CLL		CRA		RR		Poss		RR
IF	EL	AND	RA	THEN	P	IF	1	ELSE	DP
IF	L	AND	RA	THEN	P	IF	1	ELSE	DP
IF	LM	AND	RA	THEN	P	IF	1	ELSE	DP
IF	MH	AND	RA	THEN	P	IF	1	ELSE	DP
IF	H	AND	RA	THEN	P	IF	1	ELSE	DP
IF	EH	AND	RA	THEN	P	IF	1	ELSE	DP
IF	EL	AND	RN	THEN	P	IF	1	ELSE	DP
IF	L	AND	RN	THEN	P	IF	1	ELSE	DP
IF	LM	AND	RN	THEN	P	IF	1	ELSE	DP
IF	MH	AND	RN	THEN	P	IF	1	ELSE	DP
IF	H	AND	RN	THEN	P	IF	1	ELSE	DP
IF	EH	AND	RN	THEN	P	IF	1	ELSE	DP
IF	EL	AND	RT	THEN	P	IF	1	ELSE	DP
IF	L	AND	RT	THEN	P	IF	1	ELSE	DP
IF	LM	AND	RT	THEN	P	IF	1	ELSE	DP
IF	MH	AND	RT	THEN	P	IF	1	ELSE	DP
IF	H	AND	RT	THEN	P	IF	1	ELSE	DP
IF	EH	AND	RT	THEN	P	IF	1	ELSE	DP

Table 8.2: Fuzzy rules for a risk-based decision support system

Current Risk Attitude (CRA) in Table 8.2 represents the fuzzy sets to which the Risk Propensity of the consumer quantifies to its defined membership function. Table 8.3 gives the value of the *Poss* variable in Table 8.2 for each CLL by taking into consideration the MRA of the consumer and then determining if the CLL is less than or equal to the MALL. If this is true, then the value of *Poss* will be 1 for that CLL, as shown in Table 8.2, otherwise it will be 0.

	MRA		CLL ≤		Poss		Poss
IF	RA=1	AND	EL=1	THEN	1	ELSE	0
IF	RN=0.1	AND	L=0.2	THEN	1	ELSE	0
IF	RN=0.2	AND	L=0.4	THEN	1	ELSE	0
IF	RN=0.3	AND	L=0.6	THEN	1	ELSE	0
IF	RN=0.4	AND	L=0.8	THEN	1	ELSE	0
IF	RN=0.5	AND	L=1	THEN	1	ELSE	0
IF	RN=0.6	AND	LM=0.2	THEN	1	ELSE	0
IF	RN=0.7	AND	LM=0.4	THEN	1	ELSE	0
IF	RN=0.8	AND	LM=0.6	THEN	1	ELSE	0
IF	RN=0.9	AND	LM=0.8	THEN	1	ELSE	0
IF	RN=1	AND	LM=1	THEN	1	ELSE	0
IF	RT=0.1	AND	MH=0.34	THEN	1	ELSE	0
IF	RT=0.2	AND	MH=0.67	THEN	1	ELSE	0
IF	RT=0.3	AND	MH=1	THEN	1	ELSE	0
IF	RT=0.4	AND	H=0.34	THEN	1	ELSE	0
IF	RT=0.5	AND	H=0.67	THEN	1	ELSE	0
IF	RT=0.6	AND	H=1	THEN	1	ELSE	0
IF	RT=0.7	AND	EH=0.34	THEN	1	ELSE	0
IF	RT=0.8	AND	EH=0.67	THEN	1	ELSE	0
IF	RT=0.9	AND	EH=0.99	THEN	1	ELSE	0
IF	RT=1	AND	EH=1	THEN	1	ELSE	0

Table 8.3: Determining the value of the variable *Poss*

After the evaluation of the fuzzy rules, they must be aggregated and defuzzified in order to obtain a crisp value on the output membership function, which is discussed in the next section.

8.3.4 Defuzzification to Obtain Risk-based Recommendation (RR)

For the purpose of aggregation, the *Root Sum Square* (RSS) method is used. This method determines the square of each rule output which corresponds to a predicate in the output membership function and then sums them up to find its centroid. For the given problem, there are two predicates for the output membership function, therefore the aggregation output of all the rules for each predicate is determined by:

$$\mu_{\text{Proceed}} = \sqrt{\sum(\text{P})^2}$$

$$\mu_{\text{Don't Proceed}} = \sqrt{\sum(\text{DP})^2} \quad (1)$$

After the aggregation for the defuzzification process, the obtained values for each predicate are plotted on the output membership function to ascertain the range of the output. By using the centroid method, the scalar output of the fuzzy inference system is obtained which results in a crisp value. The obtained crisp value, when plotted on the output fuzzy set, represents the recommended risk-based decision from the fuzzy inference model. The centroid is given by equation:

$$\bar{x} = \frac{\sum x.\mu(x)}{\sum \mu(x)} \quad (2)$$

where x represents the points on the domain of RR on which membership $\mu(x)$ of the predicates 'P' and 'DP' are obtained after aggregation. In the next section, the working of the proposed fuzzy inference system is demonstrated using the case study.

8.4 Example of Risk-based Decision Making in Cloud Computing

Continuing the case study from the previous chapter with determined levels of loss, the risk-based recommendation is determined based on the Risk Propensity of the consumer. For consumer 'A', the predicates of loss from the previous chapter in interacting with service provider 'B' are EL=0.2, L=0.4 and LM=0.3. Let us consider that consumer 'A' chooses a value of 3.6 as his *Risk Propensity* in the interaction. The DOM of the predicates against which the *Risk Propensity* of consumer 'A' is quantified are:

$$\mu_{RP}(RN) = 0.7$$

$$\mu_{RP}(RT) = 0.3$$

Since RN<RT, the Maximum Risk Attitude (MRA) in this case is RT=0.3. Based on the MRA value, the Maximum Acceptable Loss Level (MALL) using Table 8.1 is MH=1. Using the proposed fuzzy inference system, the consumer determines the current loss level (CLL) in the interaction with service provider 'B' and compares it with the MALL based on the MRA. The value of the variable *Poss* is determined for each predicate and by using the rules defined in Table 8.2, the strength to which each rule fires is determined. Consequently, the DOM of the output fuzzy sets is

quantified and then aggregated using equation (1). Consumer 'A' obtains the following result after aggregation:

$\mu_{\mathrm{RRD}}(\mathrm{P}) = 1$

$\mu_{\mathrm{RRD}}(\mathrm{DP}) = 0.81$

If the DOM of each output predicate is plotted, the range in which the scalar output of the system exists is obtained, which is shown by the shaded portion of Figure 8.5. To obtain the scalar output, consumer 'A' needs to defuzzify the shaded region using equation (2) which results in the value 6.37. The DOM of the output fuzzy sets *Proceed* and *Don't Proceed* can be obtained by plotting the scalar output on the output membership function. The DOM on the fuzzy set *Proceed* is 0.64 and on the fuzzy set *Don't Proceed* is 0.36. In other words, it implies that, based on the Risk Propensity of the consumer and the level and magnitude of loss in interacting with service provider 'B', the strength by which the rules fire and quantify to the output fuzzy set *Proceed* and *Don't Proceed* are 64% vs 36%, respectively. On the basis of the recommendation given by the system, consumer 'A' may decide to continue the interaction with service provider 'B'.

Based on loss levels EL, L and LM and based on risk attitude levels RN and RT of the consumer for the given case study, it is logical for the fuzzy inference system to recommend to proceed in the interaction which is verified from the result of 64% recommendation to proceed. This decision support system helps the consumer to make an informed decision about continuing the service.

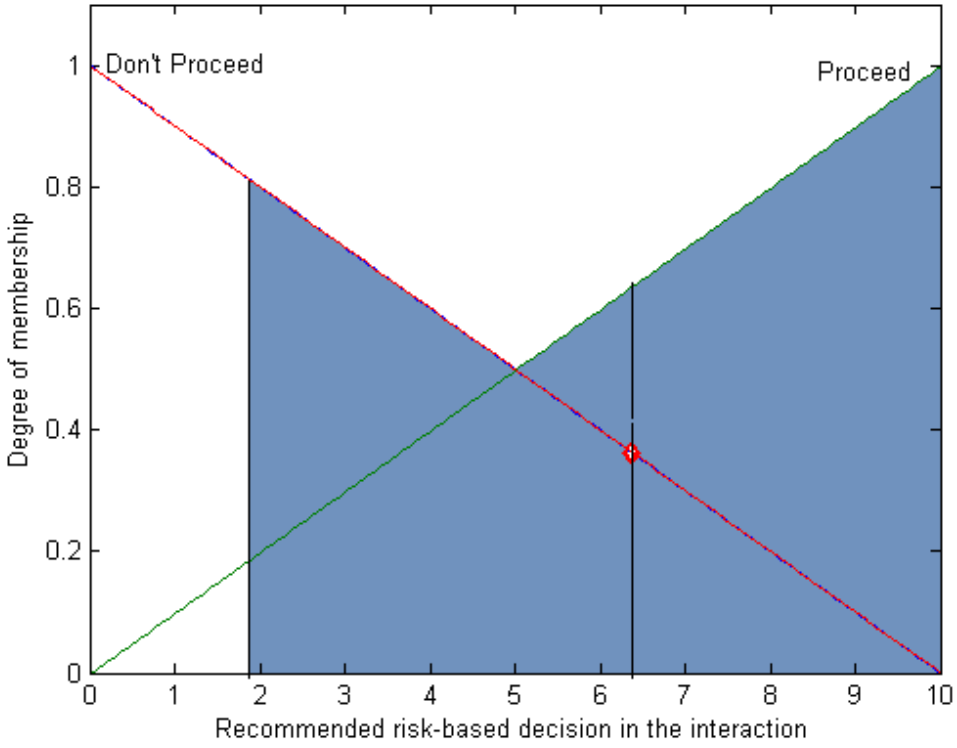

Figure 8.5: The range of output fuzzy sets in forming an interaction with service provider 'B'

8.5 Conclusion

In this chapter, a fuzzy inference system has been proposed which takes into account the risk attitude of the consumer and determines its impact on loss levels. It helps the consumer to manage loss in an interaction by assisting the consumer to make an informed decision about an interaction with the service provider. The output in the proposed system not only represents the predicates, such as proceed or don't proceed in the interaction, it also provides a recommendation as a percentage (quantitative representation) which is extremely helpful in assisting a consumer to make up his mind about the continuity of the service with the service provider.

8.6 References

[1] S. B. Sitkin and L. R. Weingart, "Determinants of risky decision-making behavior: A test of the mediating role of risk perceptions and propensity", *Academy of management Journal,* vol. 38, pp. 1573-1592, 1995.

[2] D. Liginlal and T. T. Ow, "Modeling attitude to risk in human decision processes: An application of fuzzy measures", *Fuzzy Sets and Systems,* vol. 157, pp. 3040-3054, 12/1/ 2006.

[3] O. K. Hussain, "Risk measurement, prediction and management in e-business and service oriented environment", Ph.D., Curtin Business School, Curtin University, Perth, 2008.

CHAPTER 9

IM PLEMENTATION AND EVALUATION

9.1 Introduction

In the previous chapters, each part of the proposed solution for SLA management in cloud computing has been discussed in detail. To validate and demonstrate the effectiveness of the proposed SLA management framework, in this chapter, a prototype system is developed and simulated with experiments to validate the proposed methodology. This system will accept consumer requirements for a service, select the best suited service provider, monitor the performance of that service provider, calculate the financial loss and recommend to proceed or not to proceed with that service provider. This system, called the *Trust and Risk based SLA management* (TR-SLAM) system, results in a web-based application for the selection and evaluation of service provider performance.

The prototype system is implemented in Java using Java EE 7, with third-party libraries for graphs, MATLAB and cloud simulation. For the development, the Eclipse IDE for Java EE developers is used. The purpose of this implementation is to simulate the SLA management in the pre-interaction and post-interaction time phases. The prototype is meant to reflect a software service in the cloud environment. Prototype development and implementation is explained through software development lifecycle phases. This chapter is arranged as follows: System requirements are discussed in Section 9.2. Section 9.3 presents an overview of the TR-SLAM system. Section 9.4 discusses the development of the proposed prototype. Details on the prototype implementation are discussed with the help of sequence diagrams in Section 9.5. The simulation of the prototype, along with its validation, is given in Section 9.6. The work in this chapter is summarized in Section 9.7.

9.2 System Requirements

Service provider selection and run-time SLA evaluation in cloud-like environments aim at providing QoS assessment in a dynamic way, according to the client's requirements. To build the proposed system for SLA management in cloud computing, the following requirements have been identified.

- ***Multi-layered-based approach*:** Since the TR-SLAM system results in a web-based application, a multi-layered-based approach is used [1] to implement the proposed solution. These layers include *user layer*, *application layer* and *data layer*. Each of these layers for the implementation is discussed in Section 9.3.
- ***Web Service-based solution*:** To develop a web-based application, a web service-based solution [2] is used which is more flexible in terms of protocols and development languages, is loosely coupled and works for heterogeneous platforms. It is best suited to implement the proposed solution for the cloud environment.
- ***Support cloud infrastructure*:** The TR-SLAM system should support the cloud infrastructure since the implementation is for SLA management in cloud computing.
- ***Capable of integrating with APIs*:** The TR-SLAM system should be capable of integrating with APIs that are available or developed to implement each part of proposed solution, as discussed in previous chapters.
- ***Integration/communication with external data sources*:** In the proposed framework in Chapter 4, there are two external data sources, namely, the knowledge base (recommender system) and the reputation database for the recommending users (RUs). The TR-SLAM system should be able to integrate or communicate with these external data sources.

Based on the identified requirements, in the next section, an overview of the TR-SLAM system is presented.

9.3 Overview of the Trust and Risk-based SLA Management (TR-SLAM) System

This section discusses the methodology for the TR-SLAM system development, the architecture and the implementation technologies/tools. The TR-SLAM system architecture is depicted in Figure 9.1.

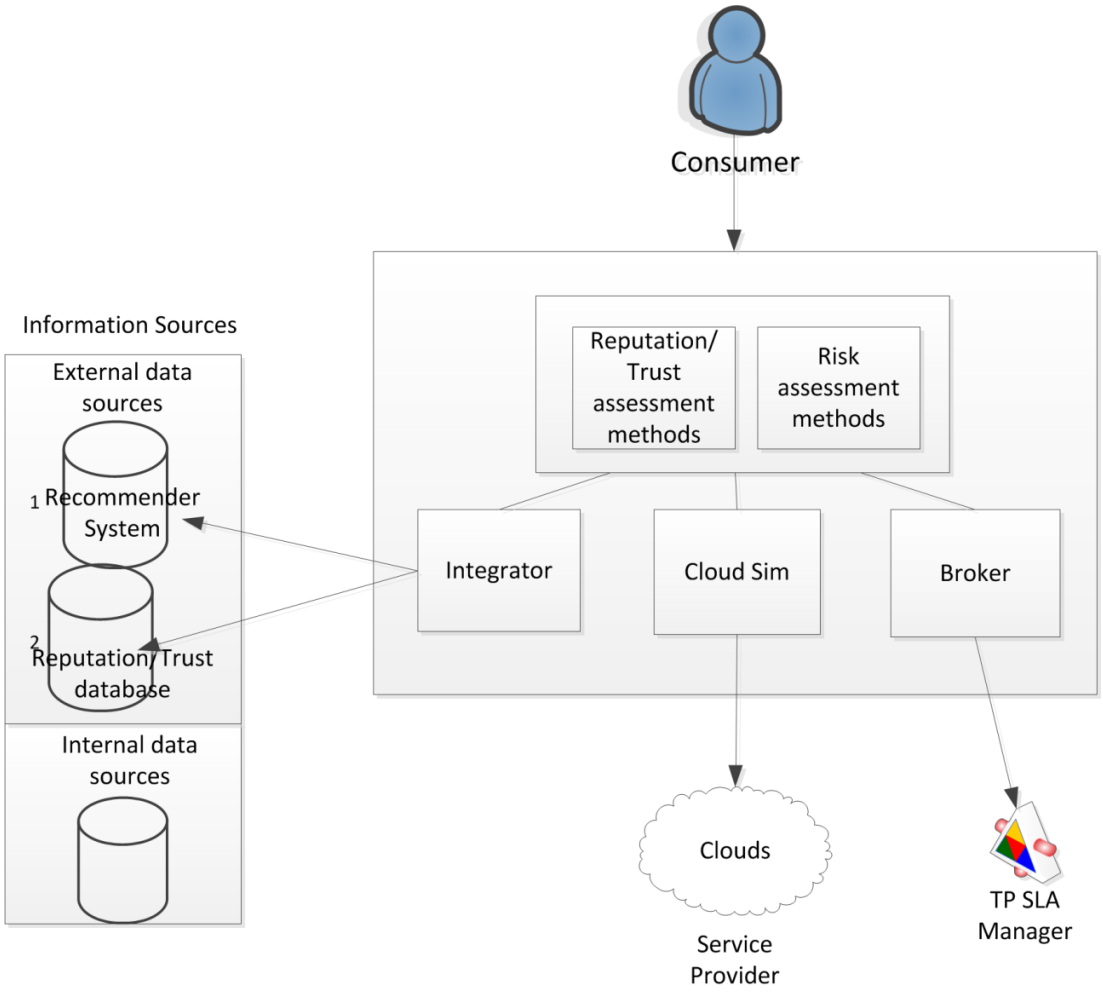

Figure 9.1: TR-SLAM system architecture

Information Sources: To process a reputation assessment request or risk assessment, the TR-SLAM system needs both internal and external data sources. External data sources consist of a Recommender System database and the reputation databases of Recommending Users. The details of these databases were discussed in Chapter 4. An internal data source consists of a database to store and retrieve the state of different objects in the TR-SLAM system. An internal data source makes it possible to create, read, update and delete (CRUD) objects.

Integrator: The role of the integrator in the TR-SLAM system is to relay a broker's query request to the external data sources [3]. When a broker receives a QoS evaluation request from the consumer, it delegates the task of querying external data sources to the integrator. The integrator performs the following functionalities: session management, translation of the incoming request to the respective schema,

forwarding the request to the respective data source, collecting the results, converting the results into respective formats and sending them back to the broker. Since the application being built is a web-based application, the integrator also has the responsibility of state preservation under session management. Retaining state information in web-based applications is very important for the successful completion of the consumer's request.

CloudSim: Cloud environments are very difficult to evaluate [4], the main reason being that it is very difficult to reproduce the desired result from cloud computing in real infrastructures. Simulation is a good way of evaluating the desired result before the actual implementation with real infrastructures as it allows the cloud user to repeat the experiment in a controlled environment without incurring costs. It also provides the cloud service providers with an opportunity to evaluate different resource leasing scenarios. Without simulation, a service user or service provider has to depend on theoretical or imprecise evaluations that may result in service degradation. Therefore, a simulation-based approach for the implementation of the TR-SLAM system has been adopted.

CloudSim is one of the tools that can be used to simulate the cloud environment. CloudSim is used in the implementation of the proposed prototype for the following reasons:

1- Time effectiveness: Less time and effort is needed to implement applications in the cloud environment [5].
2- Flexibility: CloudSim provides the flexibility to implement applications in the cloud environment with little programming effort.
3- Self-contained platform: The CloudSim platform can be used for clouds, service brokers, and the provisioning or allocation of policies.

Overall, CloudSim is a holistic software framework for modelling cloud environments and performance testing. Before the implementation is discussed, the CloudSim architecture is briefly overviewed.

The CloudSim architecture consists of the user code layer and the CloudSim layer. The user code layer allows configuration-related functionalities for hosts. The

CloudSim layer enables the simulation of cloud-based environments. This layer provides hosts for Virtual Machines (VMs), manages application execution and monitors the dynamic state.

Programmatically, CloudSim is built on SimJava and GridSim which are developed and well tested toolkits. CloudSim offers the basic components of cloud infrastructure, such as hosts (IaaS), virtual machines (PaaS) and applications (SaaS).

Broker: A broker in the TR-SLAM architecture is responsible for mediating between a consumer and a third-party service provider (TP SLA Manager). A broker manages the consumer's request for QoS evaluation. The consumer's request for QoS evaluation is initiated through the broker. The broker architecture consists of the user interface layer, the core services layer and the execution layer. Brokers then provide the functionalities of application and service interpretations, scheduling and job dispatching [6]. The application interpreter translates the consumer's requirement into what is to be executed by considering inputs through external data sources. The service interpreter determines the service requirements which include service location and service type. The scheduler schedules the job to be executed or the job to be submitted to the third-party service provider. The job dispatcher dispatches the job using the queuing model, depending on the scheduling. The broker needs to store the information about the management of the consumer's request which is kept in the internal data source of information sources.

Since a multi-layered approach has been used to build the TR-SLAM system, important layers for the proposed architecture are discussed in detail in the next section.

9.3.1 Application Layer

Depending on the system requirements discussed in the previous section, the system, which is scalable by design and modular to allow future extensions to extend the implementation as new research emerges, is developed. Keeping in view these salient features of the system, a "separation of concern" [7] approach is followed, which encourages the development of system components by observing the separation based on their focus. This enables each system's modules to be independent from

each other in their logic encapsulation and provide interface-based interaction with other systems. Following the above principles, a multi-layered approach is adopted to separate the system's actors from the ***business logic*** and ***data integration***. The system comprises the following layers, which are discussed in next subsections.

Business Logic

The business layer incorporates the application's business logic which may include the business calculations, data manipulation, information processing, data transformation and other similar operations requiring change in data elements to implement business logic. The business layer is implemented though the Model-View-Controller (MVC) pattern [8], which comprises the Model, View and Controller components, as depicted in Figure 9.2.

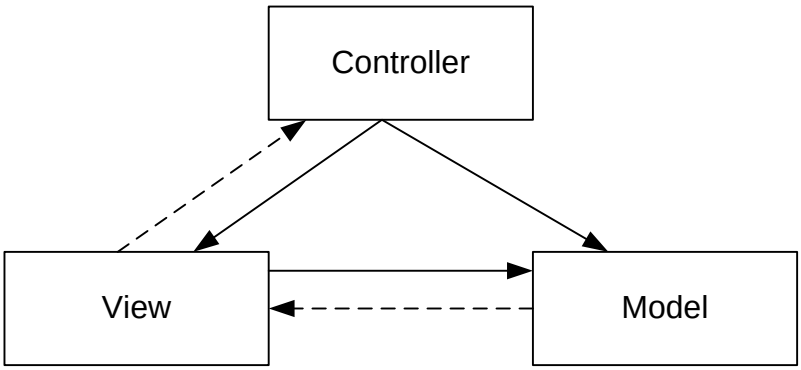

Figure 9.2: Generic MVC model

In the MVC pattern, the role of each component is described as:

Model: Model represents the data model to capture the entities' representation in the implementation of business logic. There are different approaches to implementing model components, including persistence in database or persistence in file system. In other words, Model represents and manages domain data and logic which needs to be stored for future retrieval.

View: View is the interface with which the user interacts. The user receives the instructions and interacts with the model's data to either display information or enter user input. The advantage of having View separate from the model (logic) is its de-coupling which enables the use of the same logic implementation for different

interface requirements such as for Web, for mobile devices, or for kiosks (GUI for any device, generally).

Controller: The role of the controller is to control the flow logic of the application. The controller acts as a coordinator, managing the flow logic of the application.

9.3.2 Integration Layer

The integration layer acts as a mediator between internal classes and external classes. Using façade architecture[3], the role of the integration layer in our implementation is depicted in Figure 9.3.

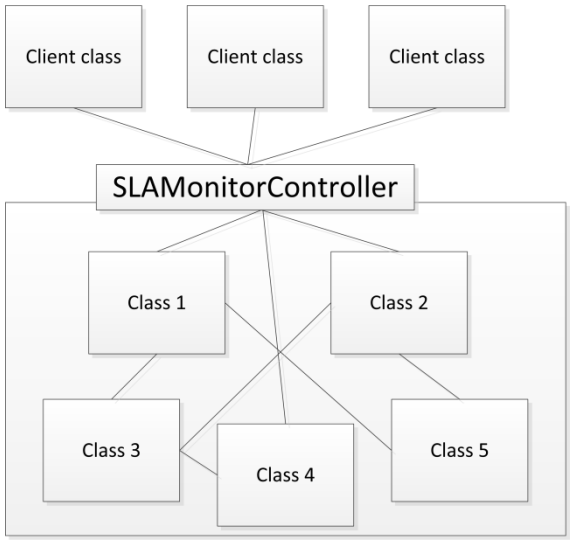

Figure 9.3: Integration pattern in the system implementation

In Figure 9.3, client classes interact with complex internal classes through Façade. The integration layer uses an adapter to connect with the *integrator* which integrates internal and external data. An XML-based strut configuration for the integration interface makes it possible to integrate the classes, as shown in the code snippet in Figure 9.5. The façade pattern hides the complexity of interactions by hiding the fine grained interactions. An *SLAMonitorController* in our architecture takes the central control by interacting with views, data sources and other controllers. The benefit of

[3] http://www.kirkk.com/modularity/2009/12/module-facade/ (Last accessed: 2/10/12)

this approach is that there is little change needed to the application when the behaviour of these functions needs to be changed. The *SLAMonitorController* plays the role of façade in our architecture.

9.3.3 Persistence Layer

This layer stores or retrieves the information from the internal data sources to read or write the states of objects in our system. In other words, this layer encapsulates the behaviour that is required to make objects persistent by read, write or delete the object from permanent storage [9]. In the TR-SLAM system implementation, a relational database is used as type of persistence mechanism. This layer takes care of save, delete or retrieves messages to an object. Since a layered approach is used for the implementation, the user interface layer, business logic layer and persistence layer are separated. The persistence layer consists of persistence data stores and their related persistence classes. The mapping from the relational database to classes called object-relational (OR) mapping provides a robust mechanism in such a manner that simple changes in the relational database do not affect the object-oriented code. Since Java tools and technologies are used for the implementation of the TR-SLAM system, Hibernate[4] is used for the object-relational mapping. Hibernate replaces the direct handling of the persistence database in a system implementation with high level object handling functions. The role of this Java library in object-relational mapping is depicted in Figure 9.4.

Figure 9.4 shows that entities in the entity-relationship diagram are converted to classes using Java beans. From these Java beans, Hibernate creates the XML files which can then talk to the databases. The code snippet generated by Hibernate for class *SelectedServiceProvider* is shown in Figure 9.5.

[4] http://www.javaworld.com/javaworld/jw-01-2005/jw-0124-strutshibernate.html?page=2 (Last accessed: 2/10/12)

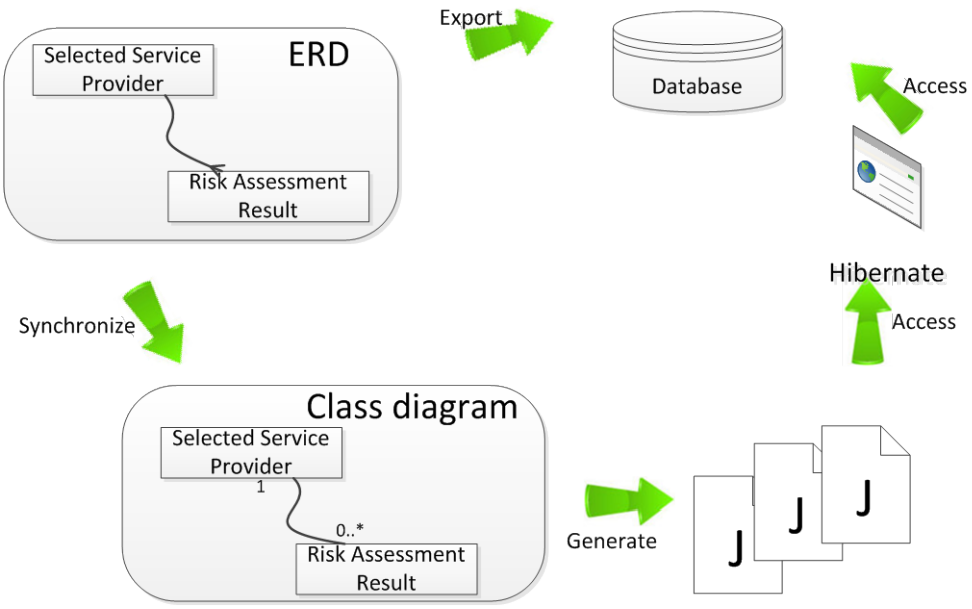

Figure 9.4: Object-relational mapping through Hibernate

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD//EN"
   "http://hibernate.sf.net/hibernate-mapping-2.0.dtd">

<hibernate-mapping>
    <class name="au.edu.curtin.debii.beans.SelectedServiceProvider"
        table="selectedserviceprovider">

        <id name="id" column="id">
            <generator class="native" />
        </id>

        <property name="SelectedSPID" column="SelectedSPID" />
        <property name="ReputationValue" column="ReputationValue" />

    </class>
</hibernate-mapping>

```

Figure 9.5: Code snippet for *SelectedServiceProvider* class

The code snippet in Figure 9.5 is the part of MVC pattern that is used to implement proposed system.

9.4 Development

After presenting a discussion of the design and architecture of the TR-SLAM system and its components in the previous section, in this section the detail of the development of this architecture is given.

9.4.1 Tools and Technologies

To develop the proposed system, a tool that fulfils our computational and performance requirements is needed. As discussed in Section 9.1, the proposed system is implemented in Java. In scientific projects, Java has been successfully used to achieve the aforementioned requirements [10]. Java power for programming in parallel and distributed environments, such as cloud computing, can be achieved by exploiting its OO, multi-paradigm communications and multi-threaded features. Since the proposed system is computationally intense, useful Java tools, such as JavaBeans and Servlets, are utilized.

The Model-View-Controller (MVC) design pattern was discussed in the previous section which provides "separation of concerns" by dividing the application into manageable parts. The tools and technologies that are used in this project are depicted in Figure 9.6. To implement the MVC pattern, the well-tested Struts framework is used, which is purely based on the MVC design pattern. The Struts[5] control layer consists of technologies such as Java Servlets, JavaBeans, ResourceBundles and XML. For Model, the Struts framework interacts with the persistence layer using Hibernate. For View, this framework works with Java Server Pages (JSP). The framework's controller uses the Action class to consult with Façade. It uses the ActionForm class to transfer data between Model and View. JavaBeans are used to represent Models. BeanUtils is most commonly used JavaBean utility to transfer data between ActionForm and Façade.

[5] http://struts.apache.org/1.x/index.html (Last accessed: 2/10/12)

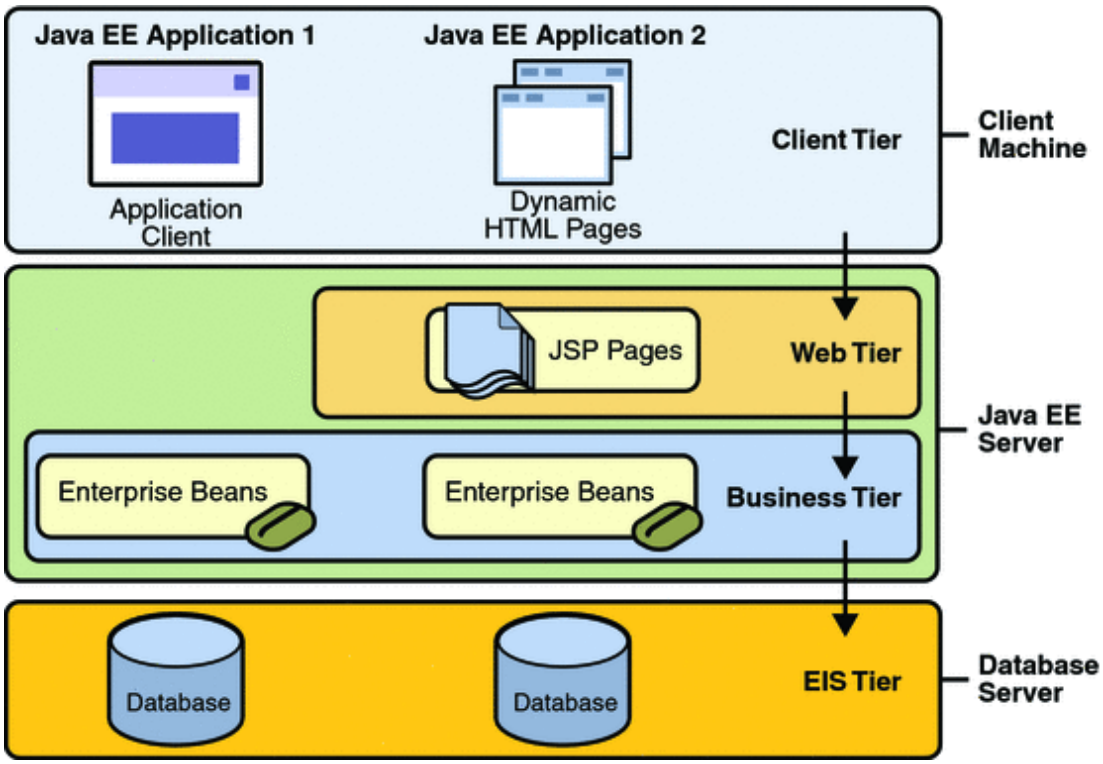

Figure 9.6: Java EE Architecture (adopted from[6])

9.5 Prototype Implementation

After discussing the multi-layered approach in the previous section, in this section the prototype implementation through sequence diagrams is discussed. Since the OO approach is used to develop the proposed system, a sequence diagram is helpful to understand the sequential interactions between objects. As shown in the figures in this section, two components are of prime importance. The first component is *SLAMonitorController* which was discussed in the previous section. This component plays a central role in our architecture and it plays the role of façade. The second important component which has been used in figures is *ComputeComponent*. The role of this component is to carry out calculations for financial risk assessment in the post-interaction time phase.

The implementation of the TR-SLAM system is divided into six phases. Each phase represents one complete activity or one use case and it may have a relation or

[6] http://abdennour-insat.blogspot.com.au/2012/09/application-architecture-tuto-1-j2ee.html (Last accessed: 5/11/12)

dependence on other use cases. Therefore, each phase is discussed in sequence of execution.

9.5.1 Pre-screening Phase

In Figure 9.7 the interaction of the components and utilization of the steps in the pre-screening layer is revealed. At the start of the interaction, the consumer submits his/her request through the search view by providing the context and search criteria. The *SearchHandler* component handles the request by passing it to the *recommender system* and obtaining a response from it. The response consists of a list of potential service providers who provide the service in the same context as the consumer requested. The response is displayed in *searchResultView*. Trust or reputation is then calculated using *TrustSequence* or *ReputationSequence* for each service provider in the list. The discussion of the search process or pre-screening process is limited to the details given in Section 4.3.

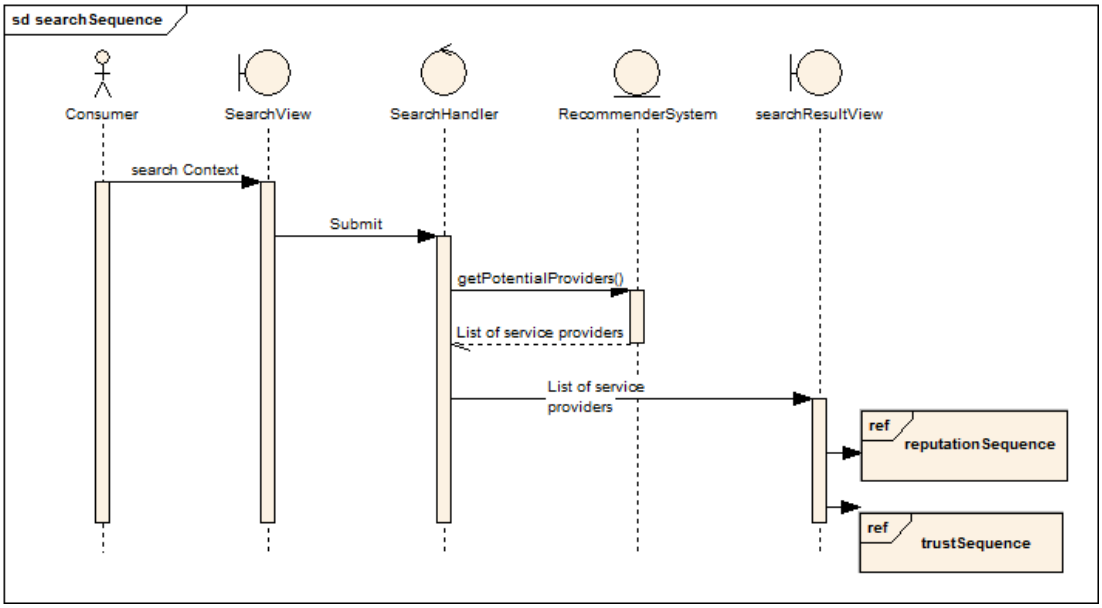

Figure 9.7: Prototype implementation: pre-screening phase

9.5.2 Trust and Reputation Assessment Phase

The trust and reputation assessment process in the pre-interaction time phase starts after *SearchHandler* passes the list of potential service providers to *SLAMonitorController*. *SLAMonitorController* uses the Java API for MATLAB

called *matlabcontrol* which uses the deterministic model and fuzzy inference model, as discussed in Chapter 5, to calculate the trust and reputation of service providers, respectively. Trust or reputation is based on the scenarios discussed in Chapter 5. In the case of the direct interaction scenario, the execution consists of the following steps that are depicted in Figure 9.8:

- *searchResultView* displays the list of potential service providers and *consumer* starts the trust assessment for service provider selection.
- For each service provider in the list, for each service assessment criterion *SLAMonitorController* calls *matlabcontrol* using the *TrustRequest(SPID, criterion)* method. Trust values for a particular service provider and a particular criterion are obtained from the trust database of the consumer.
- Once the trust assessment is completed for all service providers in the list, the result of this assessment is stored in a *TrustResult* data store.

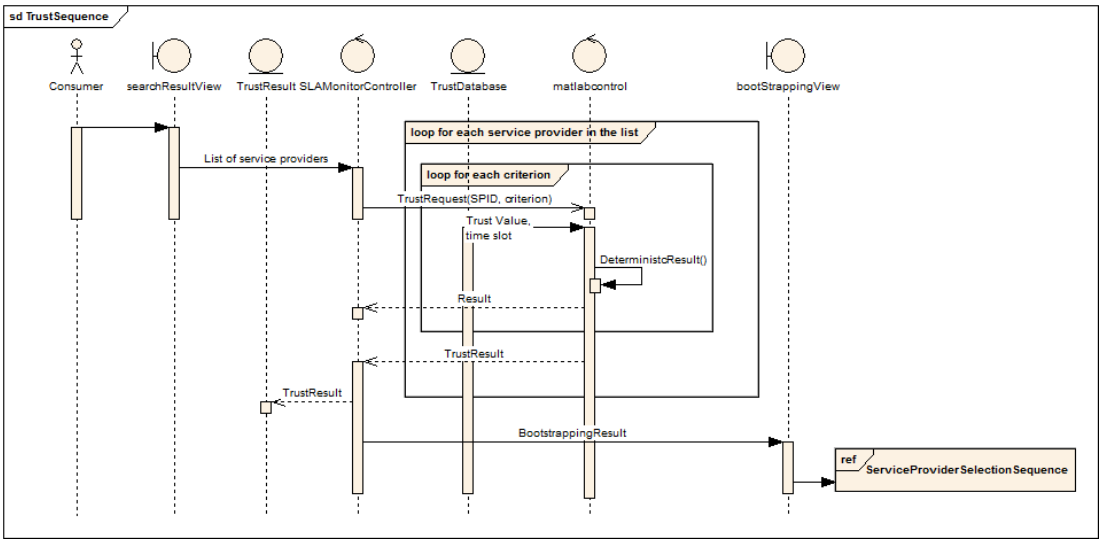

Figure 9.8: Prototype implementation: Trust determination

Similarly, in the case of the indirect interaction scenario, the execution consists of the following steps, as depicted in Figure 9.9:

- *searchResultView* displays the list of potential service providers and *consumer* starts the reputation assessment for service provider selection.

- For each service provider in the list, for each service assessment criterion *SLAMonitorController* calls *matlabcontrol* using the *ReputationRequest(SPID, criterion)* method.
- Once the reputation assessment is complete for all service providers in the list, the result of this assessment is stored in a *ReputationResult* data store.

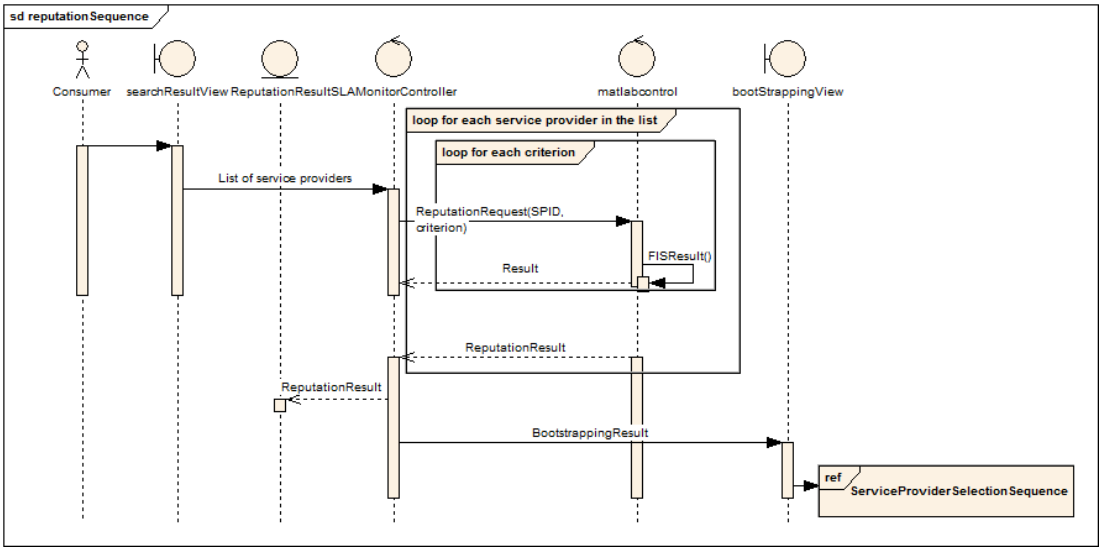

Figure 9.9: Prototype implementation: Reputation determination

The result of trust and reputation assessment for each service provider in the list is displayed in *bootStrappingView* of Figures 9.8 and 9.9. After this process, the next phase is service provider selection, which is discussed in the next section.

9.5.3 Service Provider Selection Phase

In this phase, after computing the trust or reputation, the result of *overall reputation* of all service providers in the list is displayed along with the selected service provider. *SLAMonitorController* computes the service provider with the highest reputation value. The result is displayed in *selectionView*. The sequence of service provider selection is shown in Figure 9.10.

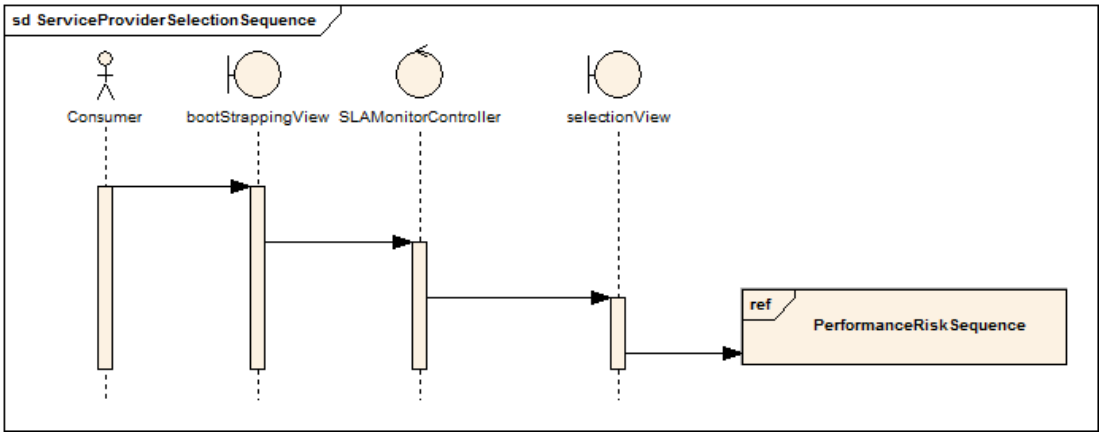

Figure 9.10: Prototype implementation: Service provider selection

9.5.4 Performance Risk Assessment Phase

From *selectionView* when the consumer proceeds further, the performance risk assessment starts. The sequence of performance risk assessment is depicted in Figure 9.11. In this figure, two scenarios for performance risk assessments are shown. In the case of performance risk assessment in the current timeslot, the sequence is as follows:

- *SLAMonitorController* sends the request to *SLAReader* which reads the SLA from the *SLA* data store.
- After receiving SLOs as a response, *SLAMonitorController* evaluates the performance for each criterion using the *ComputeComponent* which contains the logic to evaluate the performance.
- The result of the performance evaluation is stored in the *PerformanceResult* data store and displayed in *PerformanceView*.
- From *PerformanceView,* the user can proceed to Service Deviation view (*SDView*) which displays the service deviation levels. From this view, the user can proceed to *financialRiskSequence*.

In the case of performance risk assessment in a future timeslot, the sequence is depicted in the *else* part of the figure. In this part of the sequence, past performance levels are read from the *PerformanceResult* data store. The *ComputeComponent* calculates the performance risk using the method that has been discussed in Section 6.5. The remainder of the process is the same as for current timeslot scenario. The

result is displayed in *SDView* and from here, the user can proceed to *financialRiskSequence,* which is discussed next.

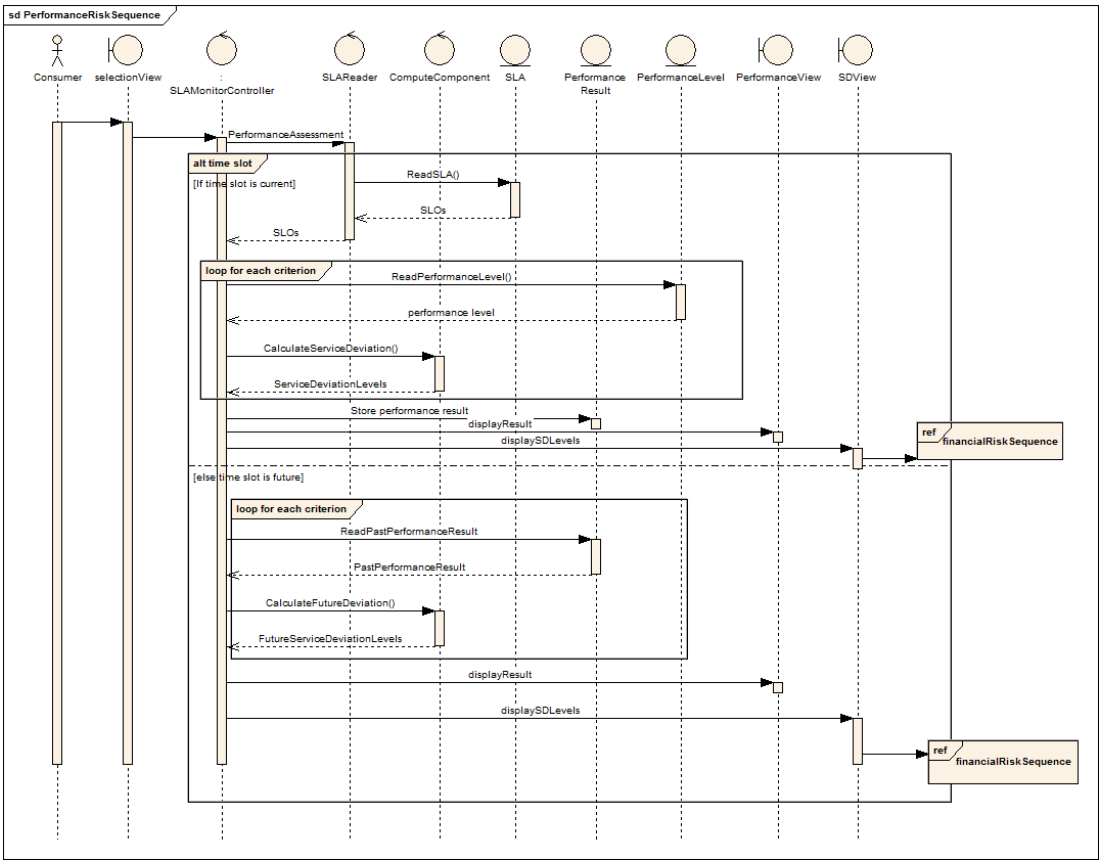

Figure 9.11: Prototype implementation: Performance assessment

9.5.5 Financial Risk Assessment Phase

Using the *SDView,* when the consumer proceeds to evaluate financial risk, the sequence depicted in Figure 9.12 is followed. The explanation of this sequence is as follows:

- The consumer enters the amount that he wants to invest in the interaction into *FinancialView*.
- When *SLAMonitorController* receives the input from *FinancialView,* it invokes the *CalculateEFI()* method on *ComputeComponent* to calculate the Expected Financial Investment.
- The result is then displayed in *EFIView*.

- When the consumer proceeds further from *EFIView*, the Expected Actual Investment curve is calculated using the *CalculateEAI()* method. This curve is then displayed in *EAIView*. From this view, the consumer can continue to evaluate financial risk using the *ExpectedLossSequence*.

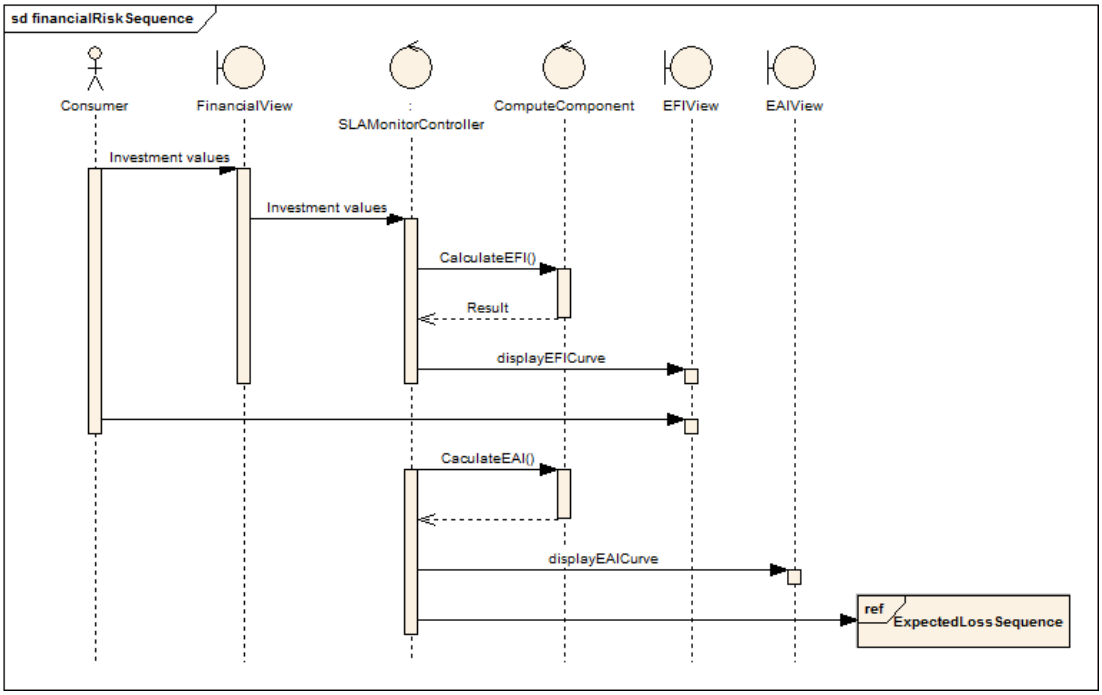

Figure 9.12: Prototype Implementation: Financial risk assessment

From *EAIView*, the consumer can go to Non-Dependable Criteria view (*NDCView*) which takes the consumer's input for the migration cost in dollars, as depicted in Figure 9.13. The request from *NDCView* is then sent to *ComputeController* which calculates the Total Resource Invested Curve (TRIC). The result is then displayed in *TRICView*. From this view, when the consumer proceeds further, loss levels are calculated using the *CalculateLossLevels* method which results in loss levels that can be utilized in risk-based recommendation, as discussed in the next section.

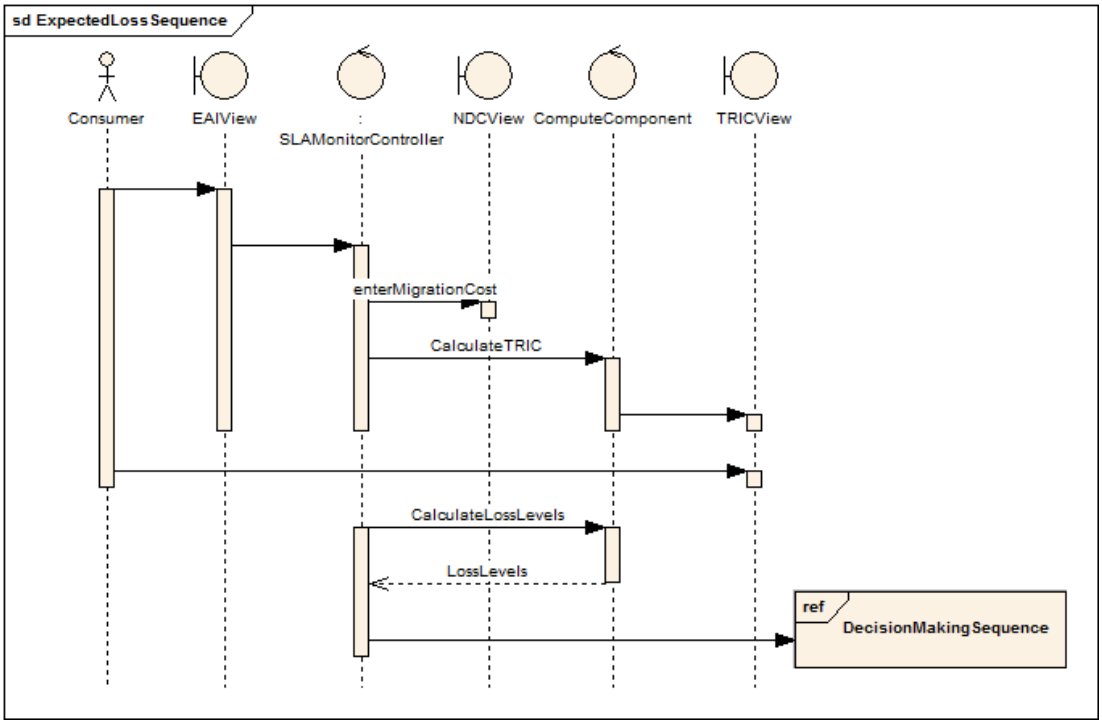

Figure 9.13: Prototype Implementation: Expected loss assessment

9.5.6 Risk-based Recommendation Phase

The sequence of risk-based recommendation is depicted in Figure 9.14. From *TRICView,* after calculating loss levels, *SLAMonitorController* displays the Risk Propensity view (*RPView*) where the consumer selects the level of risk propensity from the given levels that were defined in the previous chapter. Based on this selection, *SLAMonitorController* calls the calculate Risk-based Recommendation (*CalculateRR*) method which returns a risk-based recommendation to the Risk-based Recommendation View (*RRView*). The process of SLA management stops here.

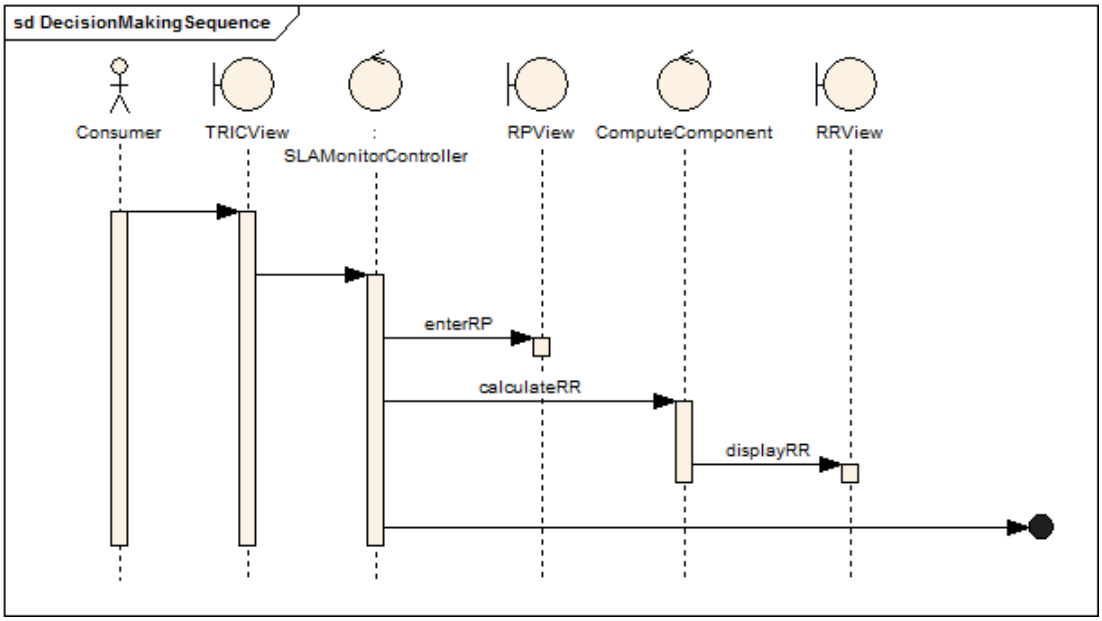

Figure 9.14: Prototype Implementation: Determining Recommended Risk

9.5.7 Database Schema

In this section, we discuss the database schema for our system. The schema is depicted in Figure 9.15. The entities are represented as *italicised*. From an implementation point of view, an entity *Service Provider* is created which stores the information about the service providers that are shortlisted in the pre-screening phase. Once a TP SLA Manager performs the trust or reputation evaluation on this list, the assessment results are stored in the entities *Trust Result* or *Reputation Result*. On the basis of trust and reputation results, the TP SLA Manager selects the service provider with the highest reputation value for service provisioning. The details of the selected service provider are stored in the *Selected Service Provider* which can then be used for performance assessment in the post-interaction time phase. The performance levels of the selected service provider can be recorded on the basis of the Service Level Objective (SLO) that is retrieved from *SLA Parameter* with SLO ID, SLO Name and the threshold. This information is then used to record the performance levels of the selected service providers in *Performance Levels.* The results of performance risk assessment are stored in *Performance Result* which stores the service deviation level for each criterion (SLOID) for the selected service provider's ID (SPID) with the timestamp.

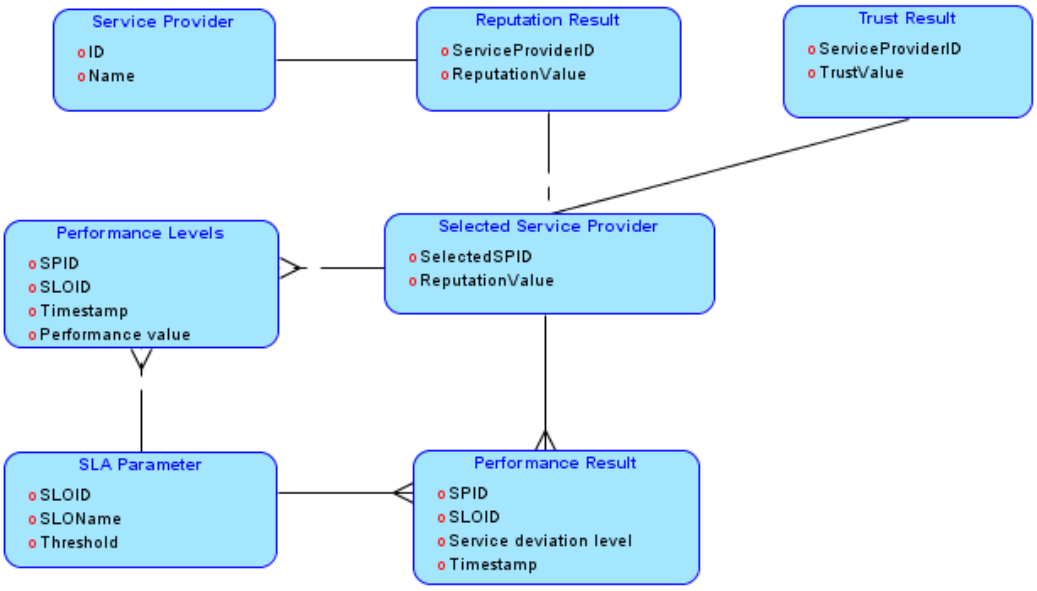

Figure 9.15: Database Schema for proposed prototype

After the discussion of the phases of implementation and the database schema of the proposed prototype, in the next section, the evaluation of the proposed prototype is given with the help of a case study.

9.6 Evaluation

In this section, a hypothetical case study is presented which is based on a real world scenario for the evaluation of the proposed prototype.

CoolBusiness is an SME that has a business set up in Australia. Most of their suppliers are from China and Singapore. To view their products and place orders, CoolBusiness needs a web conferencing service with their suppliers. For web conferencing, they need the following features:

- instant access from a browser with no software installation requirement
- high quality video
- Voice over IP (VoIP)
- recording of a conference session (video and audio) to keep track of important decisions
- real-time collaboration tools, such as a white board, for note taking

- annotation tools and one click-document sharing.

  Through this web conferencing service, CoolBusiness can connect to their suppliers at any time. Their suppliers don't need to install any special software for the conference. This is a cost-effective and time-saving solution to connect to suppliers in different countries and in different time zones.

9.6.1 System Walkthrough

Based on the requirements, CoolBusiness wants to select a trustworthy service provider based on the context of the interaction. They also want to monitor the performance of the service provider in the post-interaction timeslot based on the aforementioned criteria. The TR-SLAM system allows them to achieve their objectives. For the pre-screening phase and the reputation assessment phase, the main page of the application is shown in Figure 9.16 with CoolBusiness requirements input. These requirements are assessment criteria for trust and reputation assessment. Since the performance evaluation of the selected service provider has to be done based on the criteria in the post-interaction time phase, the main form also shows the CoolBusiness selection for the post-interaction timeslots, timeslot numbers and current or future timeslot selection.

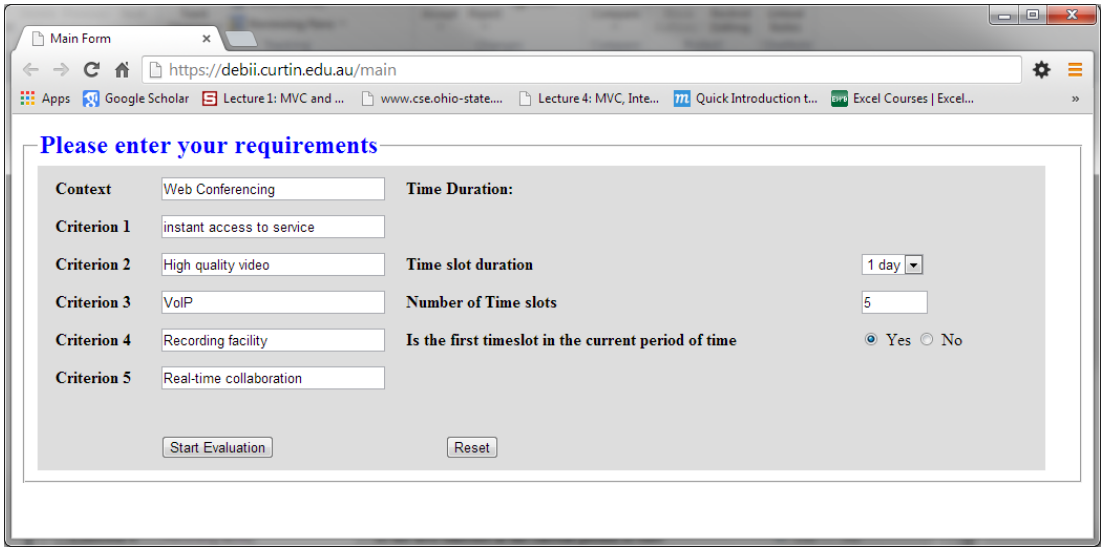

Figure 9.16: The main form of the application

When the 'Start Evaluation' button is clicked, the pre-screening process starts which returns a list of potential service providers who are able to provide a Web

conferencing service. The result of the search process is shown in Figure 9.17. This screen displays the results of the search or pre-screening process. Search process implementation was discussed in Section 9.5.1.

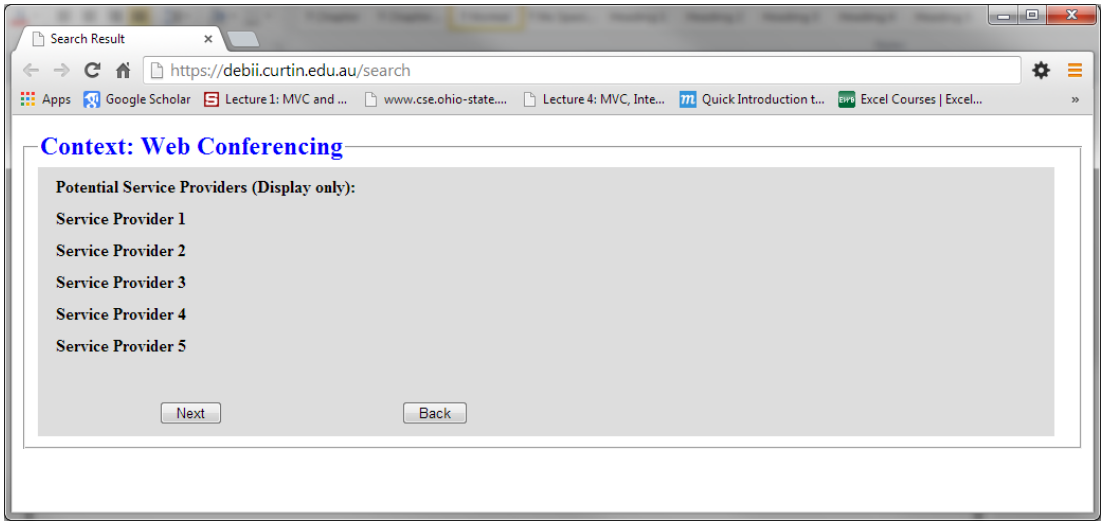

Figure 9.17: Service search result

Figure 9.17 contains two buttons. When the 'Next' button is pressed, the bootstrapping process for reputation starts which retrieves the reputation values from the external data source (trust or reputation database) for each criterion for each service provider. As discussed in Chapter 5, trust or reputation values can be calculated either on an internal opinion based on the past experience of a service consumer with the service provider or on the recommendation of Recommending Users (RUs). Figure 9.18 shows the trust/reputation calculation for each criterion for Service Provider 1.

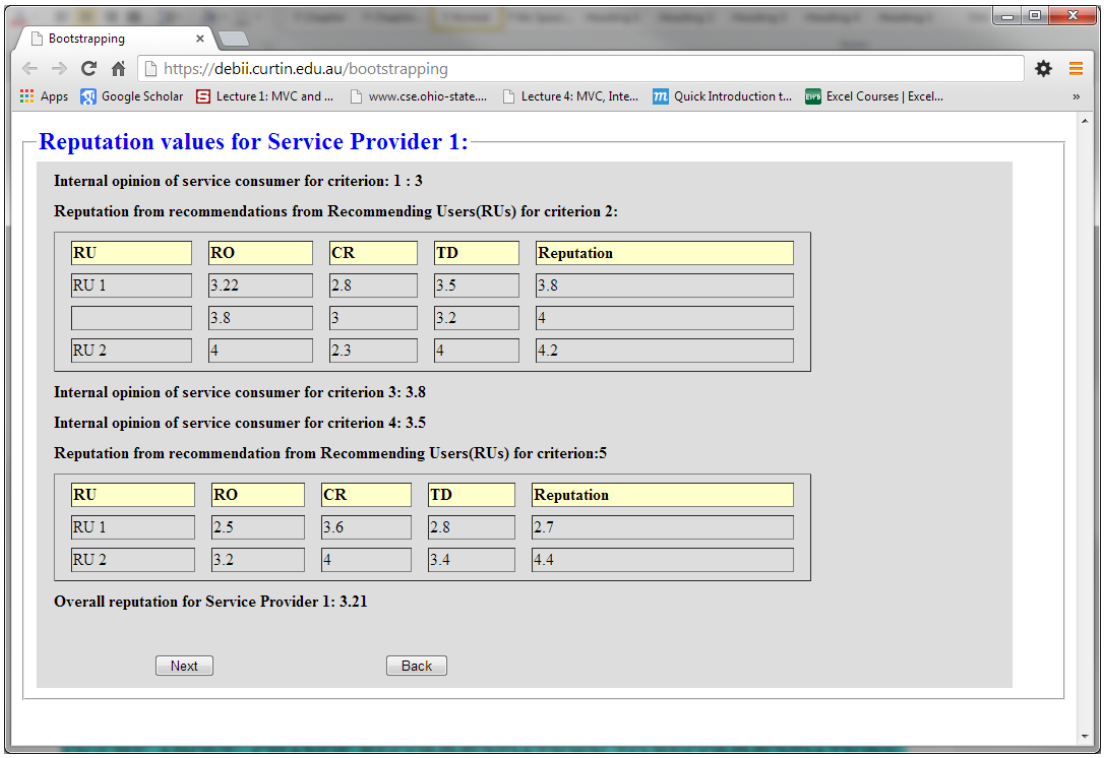

Figure 9.18: Bootstrapping process for reputation

Similarly, reputation values for other service providers listed in the Figure 9.17 are calculated by the TR-SLAM system. The detail of the implementation of the trust and reputation assessment process was discussed in Section 9.5.2. The result screen for the reputation assessment of all service providers is depicted in Figure 9.19. Based on the highest reputation value (4.55), service provider 3 is selected for the cloud service provisioning of the Web conferencing service.

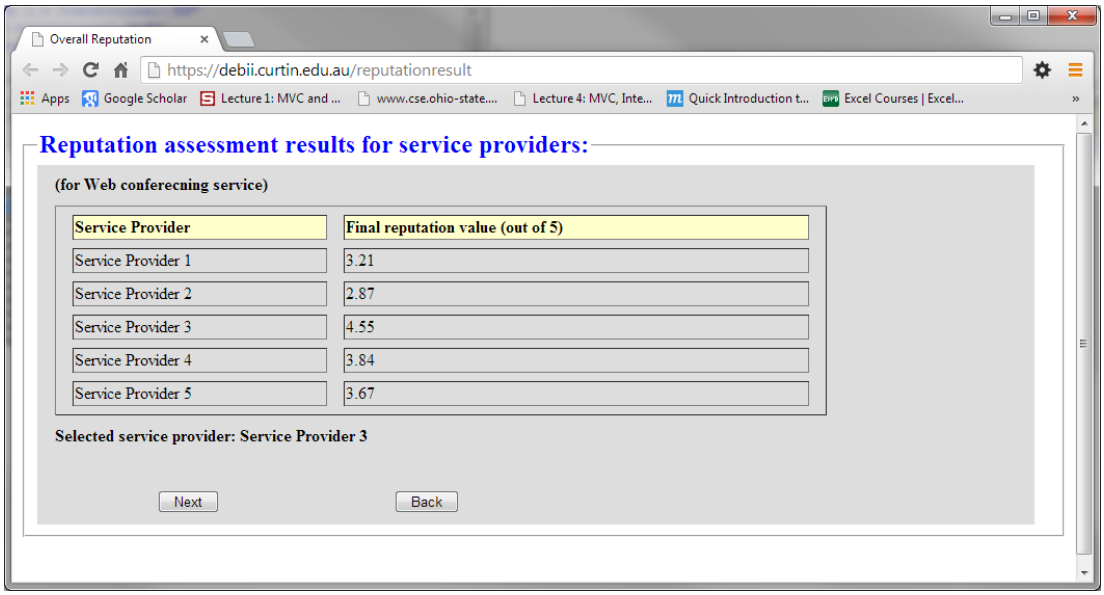

Figure 9.19: Reputation assessment result

When a user clicks 'Next' on the form shown in Figure 9.19, the performance assessment process, which was described in Section 9.5.4, starts. The next *views* or interfaces of the proposed simulation are related to SLA management in the post-interaction time phase which consists of simulations of *performance risk assessment*, *financial risk assessment*, *loss determination* and *risk-based recommendation*.

The first task in the post-interaction time phase is the creation of a simulation of the Web conferencing service using CloudSim. The performance data is obtained from the Web conferencing service simulation. On the basis of the recorded performance (observation) for Instant Access in current timeslot of 1-day duration, the result obtained is depicted in Figure 9.20 Note that performance risk assessment is done on a near-to-real time basis. The performance for the Instant Access criterion of the Web conferencing service is measured in response time, which is given in seconds.

It should be noted that although bar and line graphs are depicted in next few figures to demonstrate the result of evaluation for the case study, consumer does not require the knowledge to read these graphs. For consumer input screens (Figures 9.23, 9.26 and 9.28) and output screen (Figure 9.28) are easy to follow and understand.

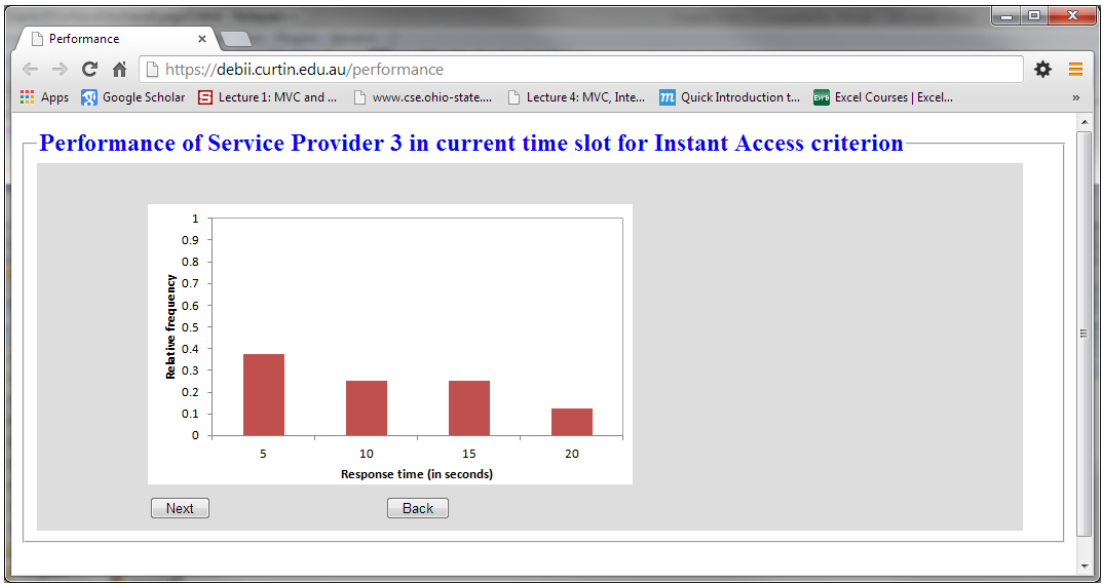

Figure 9.20: Performance observation

Suppose for Instant Access, CoolBusiness has a threshold of 15s, then a value above this threshold results in service deviation. On the basis of the performance data shown in Figure 9.20, the service deviation result for the Instant Access criterion of the Web conferencing service is depicted in Figure 9.21. This service deviation has been calculated by the TR-SLAM system for the duration of 1 day in the current timeslot. The levels of the service required have been described in the Service Level Agreement (SLA) between CoolBusiness and Service provider 3. Therefore, Figure 9.21 shows the levels of service deviation from the threshold. Note that service deviation levels are normalized to percentage scale.

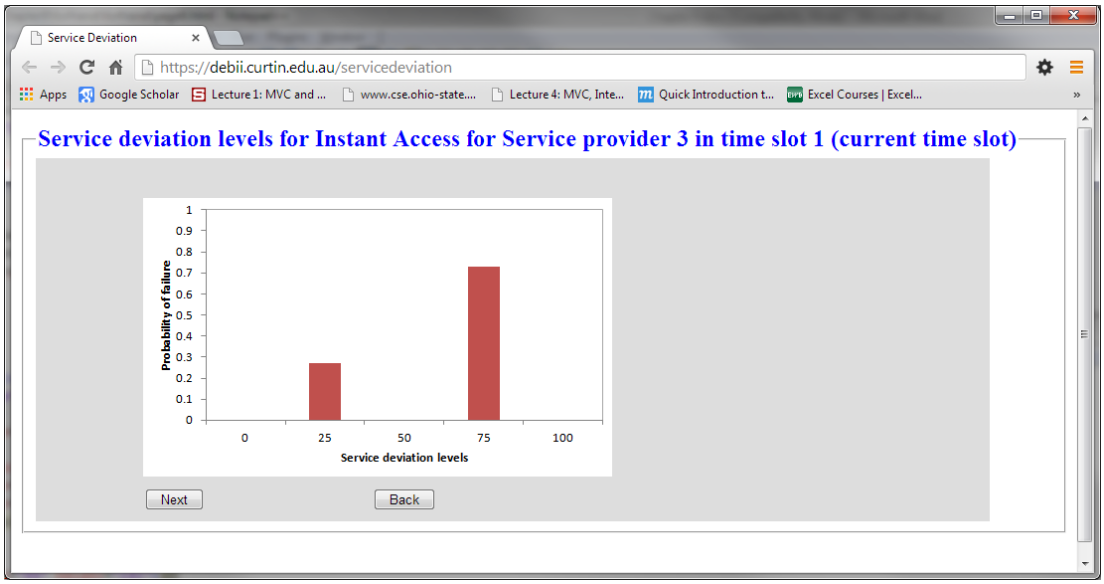

Figure 9.21: Service deviation levels in current timeslot

Figure 9.21 depicts the service deviation levels in the current timeslot. Similarly, the service deviation levels for timeslot 2, which is a future timeslot, are depicted in Figure 9.22.

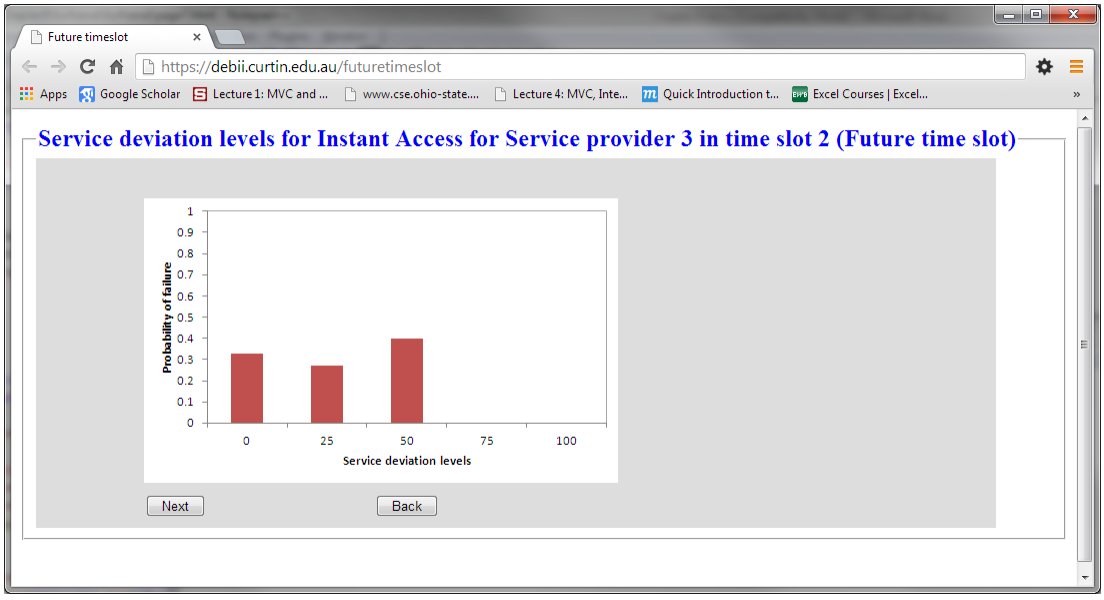

Figure 9.22: Service deviation levels in a future timeslot

When the user clicks ‘Next’, as shown in Figure 9.21 or 9.22, the next screen which is for the input of the financial expectations of the user is displayed. This screen is depicted in Figure 9.23. Assuming that CoolBusiness wants to do investments in

each timeslot, the amount of the investments shown in Figure 9.23 are entered by the user. For the first timeslot which is the current timeslot, the investment amount is $200.00. There are two possibilities for CoolBusiness to make this investment in the current timeslot: a *once-off payment* or a *payment over an interval of time*. CoolBusiness has decided to make this payment in three instalments by dividing the current timeslot duration of 1 day into three equal timeslots of 8 hours each. Similarly, the investment amounts in the other time intervals are also depicted in Figure 9.23.

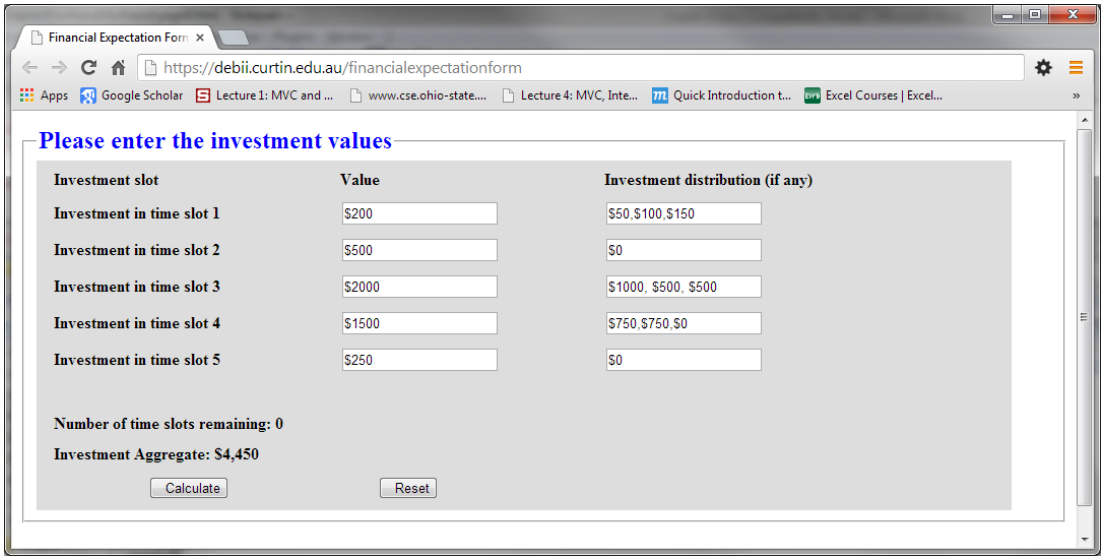

Figure 9.23: Financial expectation 'input' form

When the 'Calculate' button shown in Figure 9.23 is clicked, the result of the expected financial investment will be displayed as a cumulative curve, as shown in Figure 9.24. Notice that this curve is for investment 1.

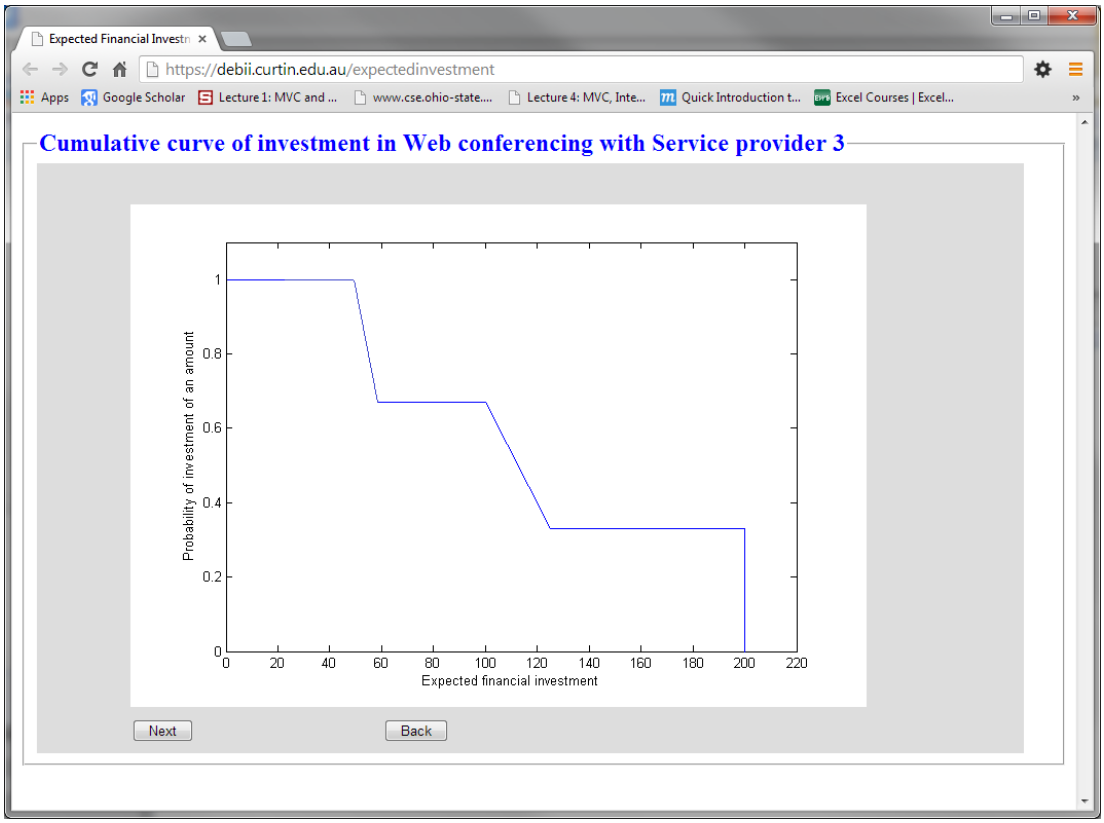

Figure 9.24: Expected financial investment curve

When the 'Next' button shown in Figure 9.24 is clicked, the extra resource investment is calculated based on a dependable criterion which is *instant access* in this case. The curve in Figure 9.25 indicates the extra resources that CoolBusiness will have to keep at stake to achieve the expected result in the Web conferencing service.

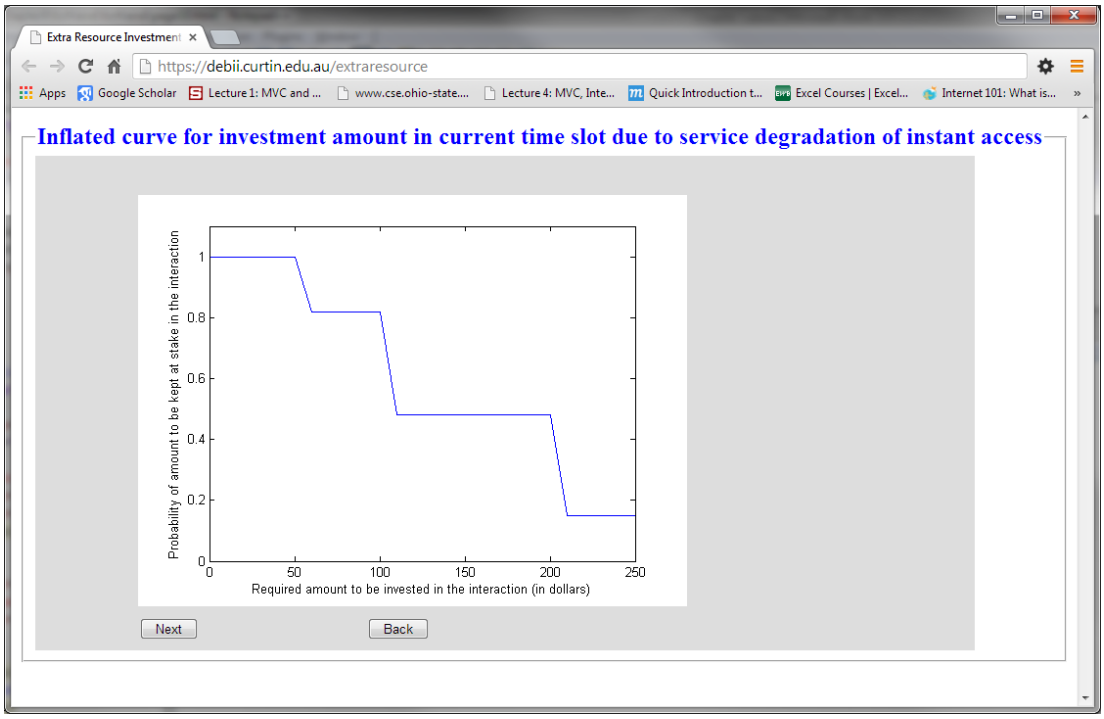

Figure 9.25: Extra resource investment curve (dependable criteria)

When the user clicks the 'Next' button shown in Figure 9.25, an input form is displayed which allows the user to enter the migration cost. Suppose CoolBusiness enters $1000 as a migration amount. This amount indicates the extra resource that has to be added to the extra resource investment curve depicted in Figure 9.25. The input form is depicted in Figure 9.26.

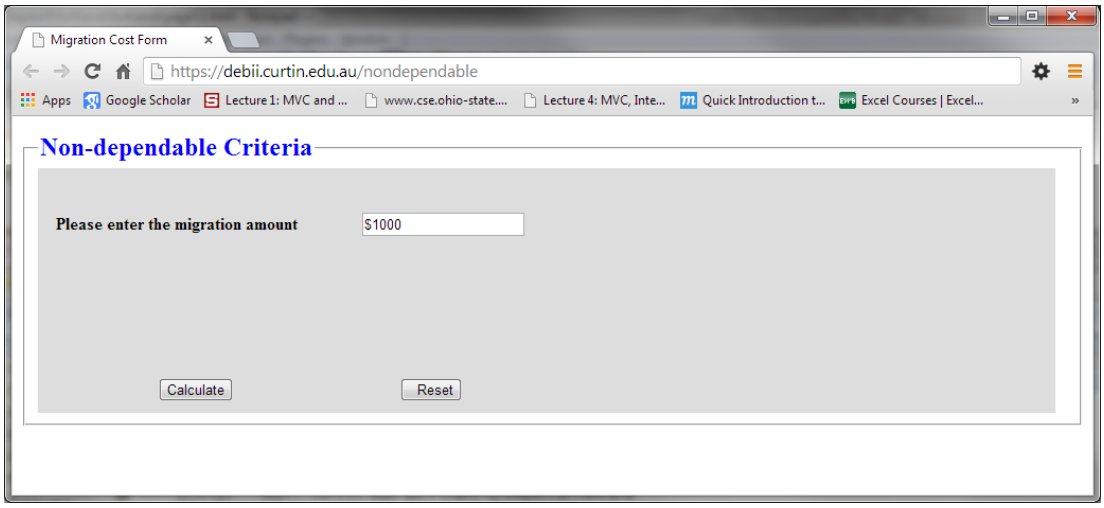

Figure 9.26: Extra resource investment (non-dependable criteria)

When theuser clicks ‘Calculate’, as shown in the above figure, the extra amount investment curve will be obtained, as discussed in Section 7.3.2. In the implementation, it is assumed that the payment of $1000 is spread over a period of 24 hours. This extra amount should be added to the already obtained extra resource investment curve to determine the total resource invested curve. As a result of this calculation, a comparison of the expected financial investment curve shown in Figure 9.24 and the total resource invested curve is shown in Figure 9.27 can be made.

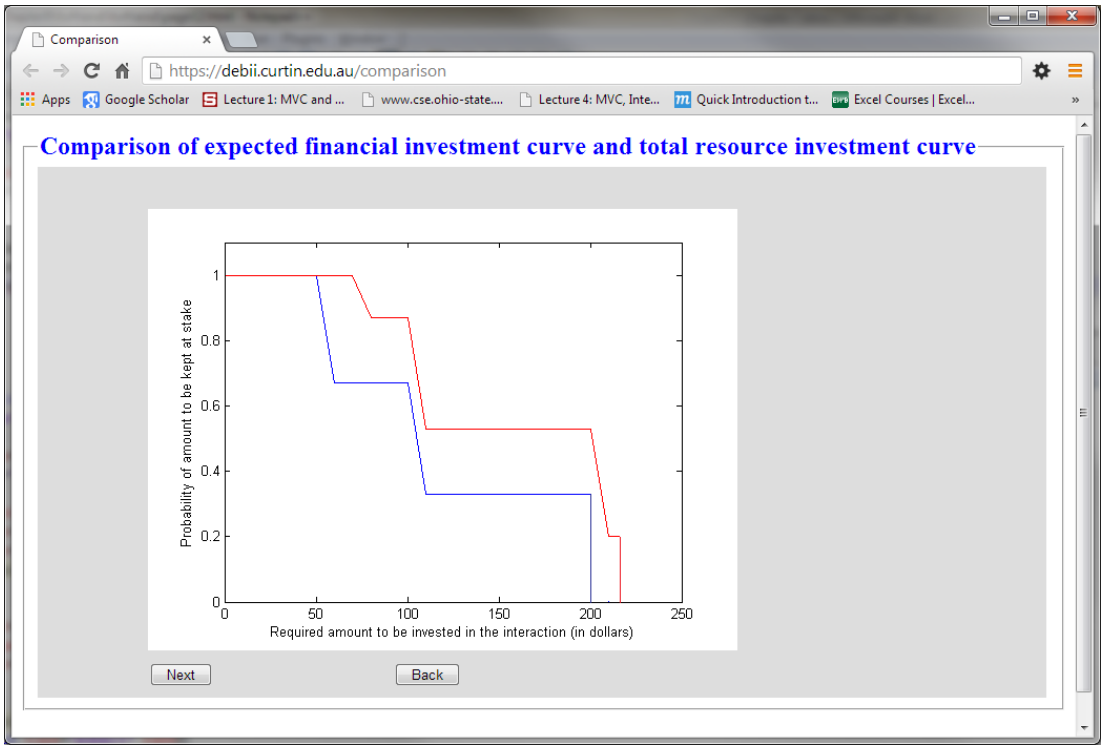

Figure 9.27: Comparison: extra resource investment curve and total resource invested curve

When the user clicks ‘Next’, the loss levels are calculated, as discussed in Section 7.3.3. To process the risk-based recommendation, the next step is to take the user’s input for his risk attitude. Assume that the consumer in this case selects a value of 1 on a scale of 1 to 5 for his risk propensity attribute. This value of risk propensity equates to a *Risk Averse* attitude. The input screen for risk propensity is depicted in Figure 9.2.8.

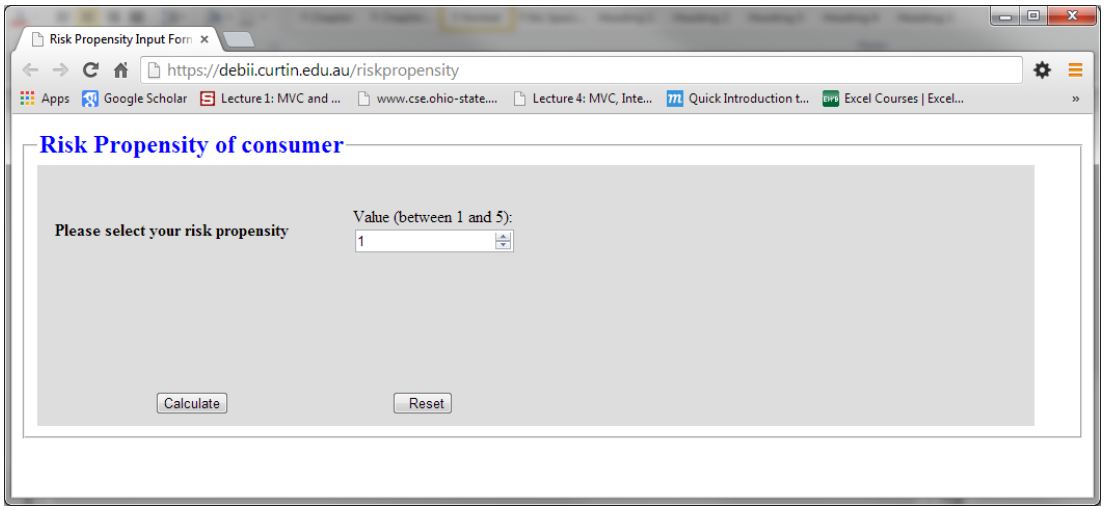

Figure 9.28: Risk propensity 'input' form

When the 'Calculate' button shown in Figure 9.28 is clicked, the risk-based recommendation is calculated using the method discussed in Chapter 8. The result of this recommendation is depicted in Figure 9.29 which is the final recommendation according to the determined loss levels and selected risk propensity of the consumer.

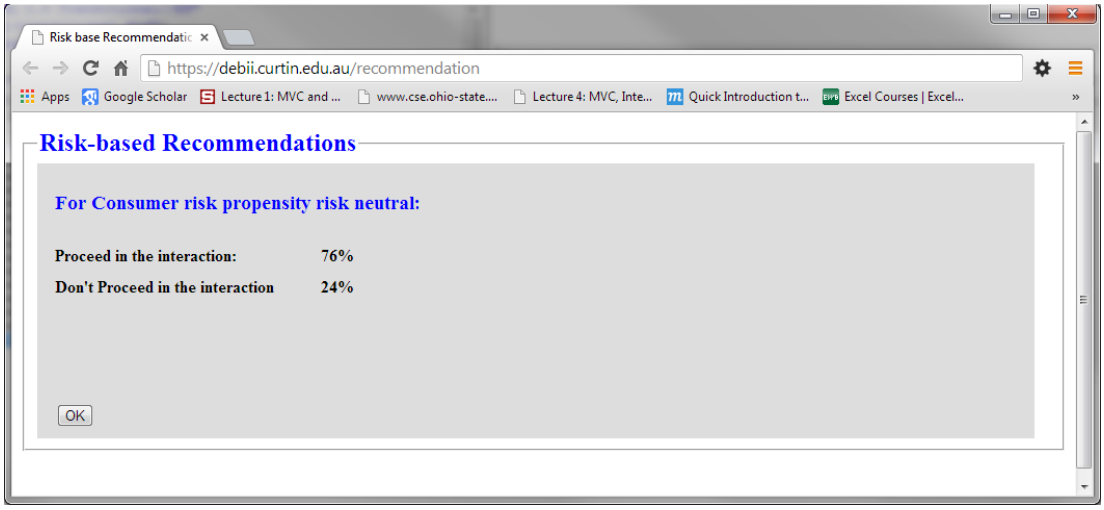

Figure 9.29: Risk-based recommendation

Based on the recommendation in Figure 9.29, CoolBusiness decides whether to proceed or not to proceed in the interaction. As demonstrated in the figure, the TR-SLAM system assists the consumer in decision making by facilitating the decision-making process with the help of percentages in relation to proceed or not to proceed.

Notice that this recommendation is based on one criterion (instant access to service) for investment in one timeslot (investment in timeslot 1). However, if the consumer chooses, he can assess the performance of the service provider by combining the different criteria in different timeslots.

9.7 Conclusion

In this chapter, the prototype system is developed to validate and demonstrate the usefulness of the proposed SLA management framework. The necessary components for the proposed prototype system have been defined which results in a web-based application. The prototype system has been discussed in detail with the help of sequence diagrams. A case study has been used to demonstrate the working of the proposed prototype system and a complete walkthrough has been provided for this case study. The application starts by accepting inputs for the context of interaction, assessment criteria and timeslots and ends with the recommendation to the service provider whether to proceed or not to proceed in the interaction.

9.8 References

[1] J. Alves-Foss, C. Taylor, and P. Oman, "A multi-layered approach to security in high assurance systems", 37th annual Hawaii International conference on System Sciences, 2004.

[2] M. Stal, "Web services: beyond component-based computing", *Communications of the ACM,* vol. 45, pp. 71-76, 2002.

[3] C. Petrou, S. Hadjiefthymiades, and D. Martakos, "An XML-based, 3-tier scheme for integrating heterogeneous information sources to the WWW", 10th International Workshop on Database and Expert Systems Applications, 1999, Florence, pp. 706-710.

[4] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities", 2009, pp. 1-11.

[5] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", *Software: Practice and Experience,* vol. 41, pp. 23-50, 2011.

[6] R. Buyya, R. Ranjan, and R. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services", *Algorithms and architectures for parallel processing,* pp. 13-31, 2010.

[7] P. Tarr, H. Ossher, W. Harrison, and S.M. Sutton Jr, "N degrees of separation: multi-dimensional separation of concerns", 21st International Conference on Software Engineering, 1999, pp. 107-119.

[8] A. Leff and J. T. Rayfield, "Web-application development using the model/view/controller design pattern", 2001, pp. 118-127.

[9] Scott W. Ambler, "The Design of a Robust Persistence Layer for Relational Databases", *AmbySoft Inc.White Paper, November, 2000,* Vol.28.

[10] V. Getov, G. Von Laszewski, M. Philippsen, and I. Foster, "Multiparadigm communications in Java for grid computing", *Communications of the ACM,* vol. 44, pp. 118-125, 2001.

CHAPTER 10

RECAPITULATION AND FUTURE WORK

10.1 Introduction

For effective *SLA management* in cloud computing, the selection of a trustworthy service provider before the start of an interaction and monitoring the performance of the selected service provider after the start of an interaction are important tasks. In the literature, there are several techniques for QoS assessment of a service provider before and after the interaction. However, these techniques cannot be applied to the dynamic nature of the cloud environment which needs continuous monitoring of the service to determine the probability of a service violation occurring and allowing the service user to overcome it. To overcome the gap in the literature related to *SLA management* in cloud computing, this thesis proposed an *SLA management* framework that helps the selection of a trustworthy service provider before the start of an interaction and then determines the performance risk and financial risk after the start of the interaction using run-time data of performance parameters. This framework then assists the service user to make an informed service-based decision. In this thesis, to solve the given problem, five different issues have been identified and addressed. In the next section, a recapitulation of the thesis is given.

10.2 Recapitulation

Some of the issues associated with the cloud computing environment which restrict businesses from moving their services to a cloud are related with *SLA management*. In the literature, attempts have been made to address the SLA management problem in the domain of e-Commerce in Service Oriented Architecture, grid computing and cloud computing. However, these techniques focused on assessing SLA from a service provider's point of view. None of the approaches in the literature considers the impact of SLA violation on business outcomes to the consumer that represent a better impact of service deviation. Further, in the cloud environment, SLA evaluation needs to take into account the run-time performance of a service provider in defined monitoring intervals. Such an SLA evaluation needs near-to-real time performance evaluation of the service provider. So, in the course of the research documented in this thesis, a broad issue to be addressed is formed which enables the consumer to

select an appropriate service provider based on his trustworthiness and evaluate the performance of the service provider against the SLA once the interaction starts, to ensure the achievement of the desired outcomes. Several sub-problems were addressed in order to solve the broad issue as follows:

1. To propose SLA management as a two-step process: the first which focuses on forming a SLA with a service provider who is capable of providing the required service; and the second which monitors or predicts the performance of the service on delivery to ensure the quality of service is according to the defined parameters as stated in the SLA.
2. To propose a trust or reputation assessment of a service provider in the pre-interaction time phase based on the direct or indirect interaction of the consumer with the service provider. Trust assessment in the direct interaction scenario is based on the past experience of a consumer with a service provider. Reputation assessment in the indirect interaction scenario is based on recommendations provided by recommending users about the trustworthiness of a service provider.
3. To propose performance risk assessment of the service provider in the post-interaction time phase for current or future timeslots, based on the run-time parameters of the service or the pervious performance history of the service provider.
4. To propose the financial risk assessment of the service provider to determine the total worth of the resources he intends to invest in the interaction in the post-interaction time phase. Financial loss is then determined numerically and linguistically.
5. To propose a methodology by which financial loss can be used with the risk attitude of the consumer to determine a recommendation to proceed in the interaction or not.
6. To validate the proposed SLA management framework by simulating experiments.

10.3 Contributions of the Thesis

The major contribution of this thesis to the existing literature is that it proposes an SLA management framework which enables the consumer to utilize the trust and reputation of the service provider to select the service at the start of the interaction

and then analyse the performance of the service provider and the financial risk in the interaction to make an informed decision as to whether to continue using the service or not. The complete solution for an SLA management framework is a combination of various proposed definitions and concepts, and six different methodologies which form the core contribution of this thesis to the existing literature. The definitions, concepts and the six different methodologies proposed in this thesis are:

1. a conceptual framework for SLA management where a third-party service provider performs the assessment of the service provider before and after the start of interaction.
2. a methodology for the assessment of the service provider before the start of an interaction which consists of a methodology of trustworthiness evaluation for direct interaction and a methodology of reputation evaluation for indirect interaction of a consumer with the service provider.
3. a methodology for the consumer to determine the performance risk after the start of the interaction with a service provider based on the performance criteria defined in the SLA.
4. a methodology for the consumer to determine the financial risk in interacting with the service provider and to determine the financial loss quantitatively and semantically in the interaction with the service provider.
5. a methodology to assist the consumer to make an informed risk-based decision on the basis of loss levels and the current risk attitude of the consumer in interacting with the service provider.

In the following, we briefly explain the contributions of this thesis to the existing literature.

Contribution 1: SLA Management Framework Before and After the Start of an Interaction

To achieve the financial outcomes of an interaction, the consumer must be assured of the capability of a service provider in providing the requested service and also the quality of the service in the cloud computing environment. This has been achieved by proposing an SLA management framework and dividing the interaction time period into the pre-interaction time phase and the post-interaction time phase. The

proposed framework consists of three pragmatic layers, each containing a dedicated set of services in addition to interacting with other layers. This framework consists of several components. It consists of a third-party service provider called the TP SLA Manager that carries out assessments on behalf of the consumer. The other important components of this framework include *Recommending Users* (RUs), *trust and reputation databases*, and *trust and risk assessment modules*.

To the best of my knowledge, this is the first attempt in the literature to propose an SLA management framework to ensure the business objectives of the consumer are met.

Contribution 2: Methodology for Assessment of the Service Provider before the Start of an Interaction

The second major contribution of this thesis to the existing literature is that it proposes a methodology for assessment in the pre-interaction time phase which enables a consumer to assess the trust or reputation of the service provider. This methodology consists of two scenarios, direct interaction and indirect interaction of a consumer with the service provider. For each scenario, the methodology is discussed in detail to assess the trust or reputation of a service provider. In the direct interaction scenario, trust assessment is based on the time decay function which indicates the elapse of time of the consumer's interaction with the service provider. In the indirect interaction scenario, a fuzzy logic-based approach has been used which is based on neuro-fuzzy algorithms such as ANFIS. Although in the literature, fuzzy systems have been proposed for reputation determination, to the best of my knowledge, none of the approaches used a neuro-fuzzy algorithm for this purpose. Using a neuro-fuzzy algorithm for reputation determination has advantages over a fuzzy system, based on the fuzzy inference method alone, as discussed in Chapter 5.

Contribution 3: Methodology to Determine Performance Risk in the Pre-Interaction Time Phase

The third major contribution of this thesis to the existing literature is that it proposes a methodology for the consumer to ascertain performance risk in an interaction with the service provider according to the given time scenarios. The methodology was

presented in Chapter 6. The proposed methodology enables the consumer to assess the performance of the service provider in the current timeslot using the run-time parameters of the service against the performance criteria given in the SLA. The consumer can also use his past interaction history with the service provider to determine the performance risk in a future timeslot. The result of the performance assessment is service deviation levels. To the best of my knowledge, there is no approach in the literature that determines the performance risk in the cloud computing environment by considering different time scenarios that result in service deviation levels.

Contribution 4: Methodology to Determine Financial Risk in the Post-Interaction Time Phase

The fourth major contribution of this thesis to the existing literature is that it proposes a methodology for the consumer to ascertain the financial risk in an interaction with the service provider. The methodology was presented in Chapter 7 and Chapter 8. The proposed methodology enables the consumer to determine the accurate worth of the resources that he has at stake throughout the time period of the post-interaction time phase. The worth of the consumer's resources is determined according to the variation in the dependable and non-dependable criteria. Then the financial loss is represented numerically and quantitatively so that both the level and magnitude of the loss can be determined and can be used further to make an informed decision. Using the proposed methodology, first the numeric values of the levels of loss are determined and then, using the fuzzy logic approach, these levels of loss are represented linguistically. The possibility theory is used to determine the level of loss along with its magnitude that can be used later in the fuzzy inference system for informed decision making.

To the best of my knowledge, none of the financial risk assessment methodologies in the literature determines the extra resources that a consumer has to keep at stake due to performance risk in cloud computing and then determines the magnitude and levels of financial loss.

Contribution 5: Methodology to Assist the Consumer to Make Informed Decision

The final major contribution of this thesis to the existing literature is that it proposes a methodology for risk-based decision making. Based on the level of loss, which can be represented as a fuzzy variable as discussed in the previous section, and the risk attitude or risk propensity of the consumer, the proposed methodology recommends the consumer to proceed or not to proceed in the interaction. This methodology is based on the fuzzy inference system which takes into account each level of loss and the current risk attitude level of the consumer and then makes a recommendation to the consumer accordingly. To the best of my knowledge, none of the approaches in the literature on informed decision making able to combine the risk attitude of the consumer which is represented numerically and linguistically with the level of loss in cloud computing.

10.4 Future Work

In this section, the future work which will be undertaken to strengthen the proposed methodology for SLA management in an interaction [1] will be discussed. It should be noted that the future work is not just limited to the discussed areas.

10.4.1 Embedding the SLA Management Methodology in the Central Control of Cloud Computing

Efforts have been made to create an automatic QoS control model [2] at the heart of cloud computing to ensure that QoS is maintained throughout the time period of the interaction. The proposed work in this thesis is a step towards this QoS model. In this thesis, QoS evaluation through trust and reputation has been mentioned. Other QoS parameters such as security of data, privacy and reliability should also be taken into account to make the consumer's experience fruitful. Therefore, the proposed SLA management framework can be embedded in an effort to develop an automatic QoS model for cloud computing.

10.4.2 Prediction of Service Provider's Performance

One of the areas where the research proposed in this thesis can be extended is the prediction of the performance of a service provider. Although one prediction method

has been used in this thesis to predict the performance of a service provider, in future, the author wants to incorporate prediction techniques, such as neural networks or chaos theory, to predict the performance of a service provider in a real-time scenario.

10.4.3 Reliability of a Cloud Service

In this thesis, performance risk and financial risk techniques are used to determine the risk associated with a service provided by a service provider. The author wants to extend this work to determine the reliability of a cloud service using credibility theory [3]. Credibility theory is a risk assessment method that handles the uncertainty associated with the randomness and fuzziness of a service operation. Reliability assessment based on credibility theory can be a part of the central control of cloud computing.

10.5 Conclusion

In this chapter, the work that has been undertaken has been recapitulated and documented. Several issues in the literature were highlighted and addressed, which prompted the work in this thesis. The various contributions to the literature as a result of the outcomes of the work done in this thesis were highlighted. A brief description of the further work which will be undertaken to extend the approaches developed in this thesis was given.

The work that has been undertaken in this thesis has been published as a part of the proceedings in a peer reviewed international journal and conferences. A complete list of all the publications arising as a result of the work documented in this thesis can be found at the beginning of this thesis.

10.6 References

[1] J. M. Myerson. (2013, November 19, 2013). Best practices to develop SLAs for cloud computing. Available: http://www.ibm.com/developerworks/cloud/library/cl-slastandards/cl-slastandards-pdf.pdf

[2] A. V. Dastjerdi, "QoS-aware and Semantic-based Service Coordination for Multi-Cloud Environments", PhD, Department of Computing and Information Systems, The University of Melbourne, 2013.

[3] Y. Feng, W. Wu, B. Zhang, and W. Li, "Power system operation risk assessment using credibility theory", *Power Systems, IEEE Transactions on*, vol. 23, pp. 1309-1318, 2008

APPENDIX A - PUBLICATIONS ARISING FROM THIS THESIS

Cluster Comput
DOI 10.1007/s10586-012-0232-9

A framework for SLA management in cloud computing for informed decision making

Adil Hammadi · Omar Khadeer Hussain · Tharam Dillon · Farookh Khadeer Hussain

Received: 2 July 2012 / Accepted: 3 October 2012

Abstract In cloud computing, service providers offer cost-effective and on-demand IT services to service users on the basis of Service Level Agreements (SLAs). However the effective management of SLAs in cloud computing is essential for the service users to ensure that they achieve the desired outcomes from the formed service. In this paper, we introduce a SLA management framework that will enable service users to select the best available service provider on the basis of its reputation and then monitor the run time performance of the service provider to determine whether or not it will fulfill its promise defined in the SLA. Such analysis will assist the service user to make an informed decision about the continuation of service with the service provider.

Keywords Service level agreement · Reputation · Transactional risk · Performance risk · Financial risk

A. Hammadi · O.K. Hussain (✉)
School of Information Systems, Curtin Business School, Perth, Australia
e-mail: O.Hussain@cbs.curtin.edu.au

A. Hammadi
e-mail: adil.hammadi@postgrad.curtin.edu.au

T. Dillon
Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, Australia

F.K. Hussain
Decision Support and e-Service Intelligence Lab, Quantum Computation and Intelligent Systems, School of Software, Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia
e-mail: Farookh.Hussain@uts.edu.au

1 Introduction

Big data, due to its huge volume and variety, is considered as the next frontier for innovation, competition and productivity [1]. Such data which is generated from numerous activities has resulted in a huge growth of information in the digital universe, and is a valuable asset for businesses, enterprises and users. However, before such data can be utilized by the various users for their benefit, appropriate data analytic tools are needed that assist them to understand the vast variety and volume of big data in real-time and synthesize meaningful knowledge from it. Big data analytics are the new generation of technologies designed to economically extract value from large volumes and a wide variety of data, by enabling high-velocity capture, discovery, and/or analysis [2]. *Business Intelligence* which involves performing analytics on the underlying data to synthesize actionable knowledge from it is one type of big data analytic techniques. Business Intelligence has various applications in different domains. In this paper, we focus on one such application of Business Intelligence, namely making informed service-based decisions in the domain of cloud computing.

Cloud computing is defined as *a parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements* [3]. By using cloud computing architecture, users can access massive computing resources for short time periods without having to build their own infrastructure. This is achieved in the form of services. Also, by using the cloud computing platform, service providers will be able to provide cost effective and on-demand IT services to the multiple-service users on a multi-tenancy basis for their various needs. Cloud computing architecture works on the vision of the services (required by the service users

Published online: 13 December 2012

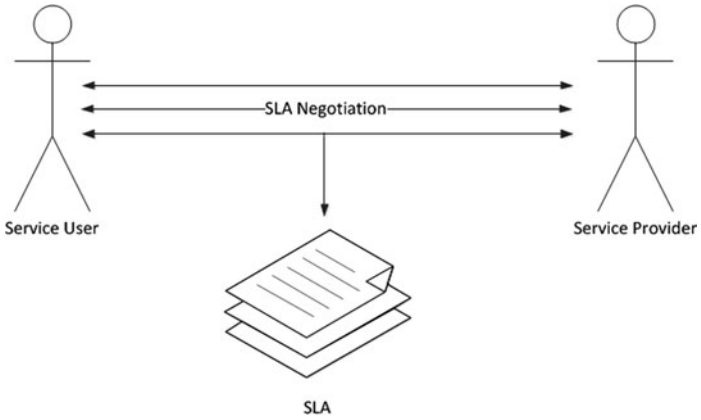

Fig. 1 SLA negotiation and formation

and provided by the service providers) being delivered as promised at the required capacity, time and need. However, the two main challenges that need to be addressed in order to ensure that this vision is achieved are:

(a) the service user must select a capable service provider (from the possible ones) from whom s/he can achieve the desired outcomes;
(b) the service provider has to ensure that s/he has the capability and ample resources to deliver on the agreed service to the service user. This will ensure in the achievement of financial gains to the service providers as an outcome of the service.

To guarantee Quality of Service (QoS), Service Level Agreements (SLAs) are established between a service provider and a service user. SLA is a formal contract and contains terms and conditions to which both the service provider and service user agree. The establishment of SLAs benefits both providers and users. For a service provider, the SLA provides performance metrics to be committed to, and for a service user, it provides assurance of QoS. SLAs are formed between a service provider and a service user through an iterative process of negotiations [4] as shown in Fig. 1. SLAs can be broken down into various smaller criteria called Service Level Objectives (SLOs). The SLOs describe in detail the QoS properties for the agreed-upon service. In other words, the SLOs also establish a threshold against which a service is expected to be delivered. The performance of the service provider is measured against this threshold.

1.1 Problem definition

Even though a SLA defines the agreed QoS properties to which the service provider will commit, there may be scenarios where it may not meet those properties (and hence the SLA) due to various factors such as incapability to meet the service, service degradation, service outages etc. Service degradation is defined as the reduction in the quality of the service, due to any event. This may have the potential to lead to an outage in the service. Service degradation and/or service outage may have an impact on both the service users and service providers in relation to the business outcomes to be achieved and may result in a loss of investment. For example, outages which occurred at Amazon Simple Storage Service (S3), AppEngine and Gmail in 2008 caused significant losses to service users [5]. In order to avoid such undesirable outcomes, *SLA management* is important to ensure that the run-time service properties meets the criteria that are established in the agreement. However, a framework for SLA management should be a two-step process; the first being the establishment of the SLA with a service provider who is capable of providing the required service, and the second being the monitoring or predicting of the performance of the service on delivery to ensure that the quality of service is according to the defined parameters as stated in the SLA. In the literature, SLA monitoring techniques are mentioned, however, none of these techniques considers SLA monitoring as a two-step conjoint process as discussed here. In this paper, we propose an approach for SLA management by which a service user is able to make an informed decision about the selection and continuity of the service with a service provider.

2 Literature review

In this section, we explore the work done in the area of SLA formation, negotiation and QoS evaluation, based on SLA for Service-Oriented Computing and Grid Computing. He et al. [6] discussed the SLA formation and negotiation process. The entire SLA process can be accomplished using agreement protocols such as a WS-Agreement where a service requester submits his/her request through a web portal. The client submits its requirement to a Business Rule Consultant which activates the agreement. The service provider's service templates are already available through a service

registry. The Business Rule Consultant maps the user requirements to available templates of service providers. After completing this phase, negotiation starts between negotiator agents. If negotiation is successful, a final agreement between agents is reached through an agreement factory. After this, to enforce the agreed SLA, monitoring begins. At this stage, for reasons of credibility, the authors propose the commissioning of a third-party agent to monitor the SLA. Quillinan et al. [7] mention that the Quality of Service (QoS) provided by a service provider can be assessed on the basis of the SLA. When it comes to monitoring the SLA, penalties can be imposed as a result of violation by the service provider. Online monitoring is one way of monitoring service violations. Online monitoring means that the continuous monitoring of service is done based on monitoring intervals. These monitoring intervals depend upon agreement terms and can be in seconds, minutes, hours or days. A second monitoring approach discussed is reactive monitoring, whereas the third monitoring approach is offline monitoring. In terms of penalties for service violations, two types are defined: reputation-based penalty and monetary-based penalty. Khader et al. [8] discuss the reactive monitoring technique for SLA. After discussing online and offline monitoring, they mention that the reactive monitoring approach balances the trade-off between online and offline monitoring. Oliveira et al. [9] provide a benchmark to monitor SLA by using a grid system. They argue that since most of the monitoring techniques are embedded in a particular SLA specification, they rely on specific protocols which impede the wide-spread adoption of grid infrastructures. They introduced a benchmark called Jawari to monitor the SLA. Chang et al. [10] propose a measurement methodology for Quality of Service assessment. According to Chang et al., one way to obtain QoS assessment is by determining the trustworthiness of the service provider. By using a combination of their proposed metrics, the trustworthiness value of a service provider can be determined. In an extension of this work, Schmidt et al. [11] used these metrics to determine the reputation of a service provider. After each business interaction, the credibility of the service provider is adjusted which results in either an increase or decrease of its reputation. If the quality of service provided falls short of the promised service quality, the reputation of the agent is significantly decreased. This work, even though is significant, applies only to QoS assessment after the initiation of the service. However, sometimes the service user needs QoS assurance *before* the start of a business interaction. Hussain et al. [12] address pre-interaction QoS assessment using risk assessment techniques, whereby a service user determines the risk before any interaction with the service provider begins. The level of risk is determined by considering past trustworthiness values obtained through past experience with the service provider or through the recommendations obtained by third-party agents.

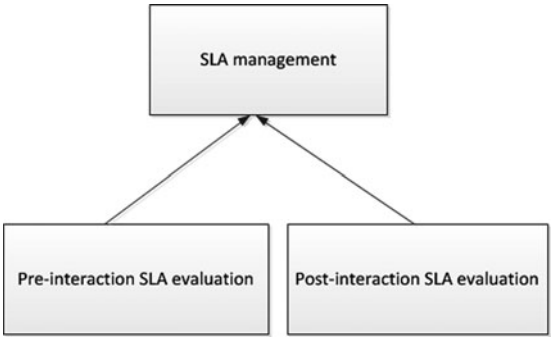

Fig. 2 SLA management framework

Even though a great amount of work has been done in the literature to ensure QoS based on the SLA, most of the work places an emphasis on either the pre- or post- service initiation phase. However, as discussed in the earlier section, in order to have a framework for SLA management, a conjoint assessment and consideration of the analysis of both these phases is required to make an informed decision. Such an approach is needed for a service user to make an informed decision of service selection and the continuation of the service with a service provider. In the next section, we propose such an approach for SLA management in cloud computing.

3 Proposed framework for SLA management in Cloud Computing

To address the problem raised in Sect. 1.1, in this section, we propose an SLA management framework comprising two parts: (a) pre-interaction SLA evaluation; and (b) post-interaction SLA evaluation, as shown in Fig. 2. Pre-interaction SLA evaluation assists the service user to select the service provider who has the best capability to commit to the established SLAs, whereas post-interaction SLA evaluation assists the service user to monitor the performance of the chosen service provider to ensure that the delivered service meets the service levels defined in the SLOs in different periods of time.

We term the total time over which SLA management has to be carried out as the 'time space'. In other words, time space is defined as that total period of time which the service user takes into consideration to ascertain the QoS while interacting with the service provider. However, as the capability of the service provider to deliver a service with the required level of quality varies over a period of time, in order for SLA management to be meaningful for both the time parts defined in Fig. 2, we divide the interaction time period into two different non-overlapping parts namely, 'pre-interaction' and 'post-interaction' time phases [12] as shown in Fig. 3.

The pre-interaction time phase is the portion of time before the start of service between the service user and service

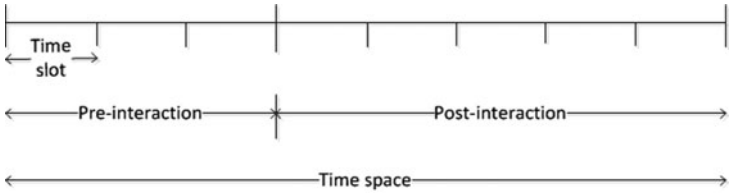

Fig. 3 Time space and related concepts

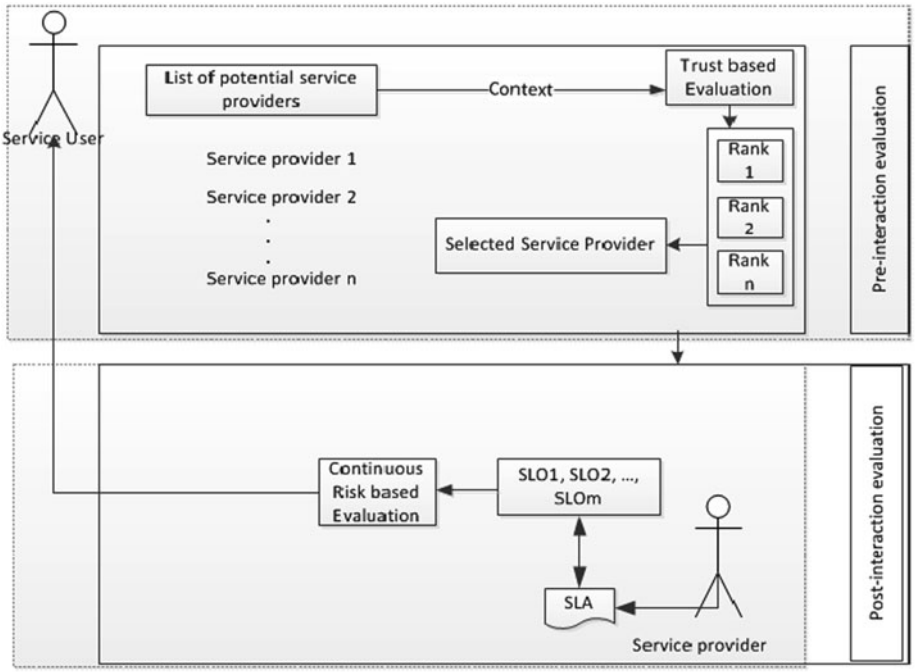

Fig. 4 Conceptual framework for QoS assessment in cloud computing

provider and the post-interaction time phase is the portion of time after the start of service between the service user and service provider. 'Time slot' is defined as a non-overlapping interval of time within the time space of the interaction. The time slot is obtained by dividing the time space into different equal, non-overlapping parts of time. QoS evaluation is done in each time slot. In the pre-interaction phase it is done by determining the reputation of the service provider in being capable of and committed to complying with the terms of the established SLAs; whereas, in the post-interaction phase, it is done by determining the transactional risk as a result of the service being provided by the service provider not committing to the defined metrics of the SLAs. In order to implement the solution in a pragmatic setting, we lay out the architectural foundation to make the solution scalable and modular. Modularity has been achieved by introducing a multi-layered architecture, as shown in Fig. 4.

In pre-interaction evaluation step of Fig. 4 for pre-interaction evaluation, there is a *trust-based evaluation* service which evaluates service providers who have the capability of fulfilling service user's request. The output of this evaluation is the selection of a service provider who has highest trust value. This output is then passed on to post-interaction evaluation step which then evaluates or predicts the performance of selected service provider against the formed SLAs by using the *continuous risk-based evaluation* service. To monitor SLA evaluation in pre- and post-interaction time phases in automatic fashion, we introduce a third-party service provider called Third-Party Service Level Agreement Monitor (TP SLA Monitor) who is responsible for executing a *trust-based evaluation* service and *continuous risk-based evaluation* service. A trust-based evaluation service consists of a *reputation assessment module* and a continuous risk-based evaluation service consists of a *risk assessment module*, as depicted in Figs. 5a and 5b, respectively.

To start the monitoring process, the TP SLA monitor should first establish the total time space for which QoS has to be assessed. It then divides this time space into pre-interaction and post-interaction time phases. QoS evaluation in the pre-interaction time phase is achieved by using a reputation assessment methodology by soliciting the recommendations from *Recommending Users* (RU). RUs in our framework are a group of users who receive a reputation query from TP SLA monitor and they reply on the basis of their past experiences with the service provider. Each RU stores past experiences in the form of trust values assigned to a service provider in an information repository [10] as depicted

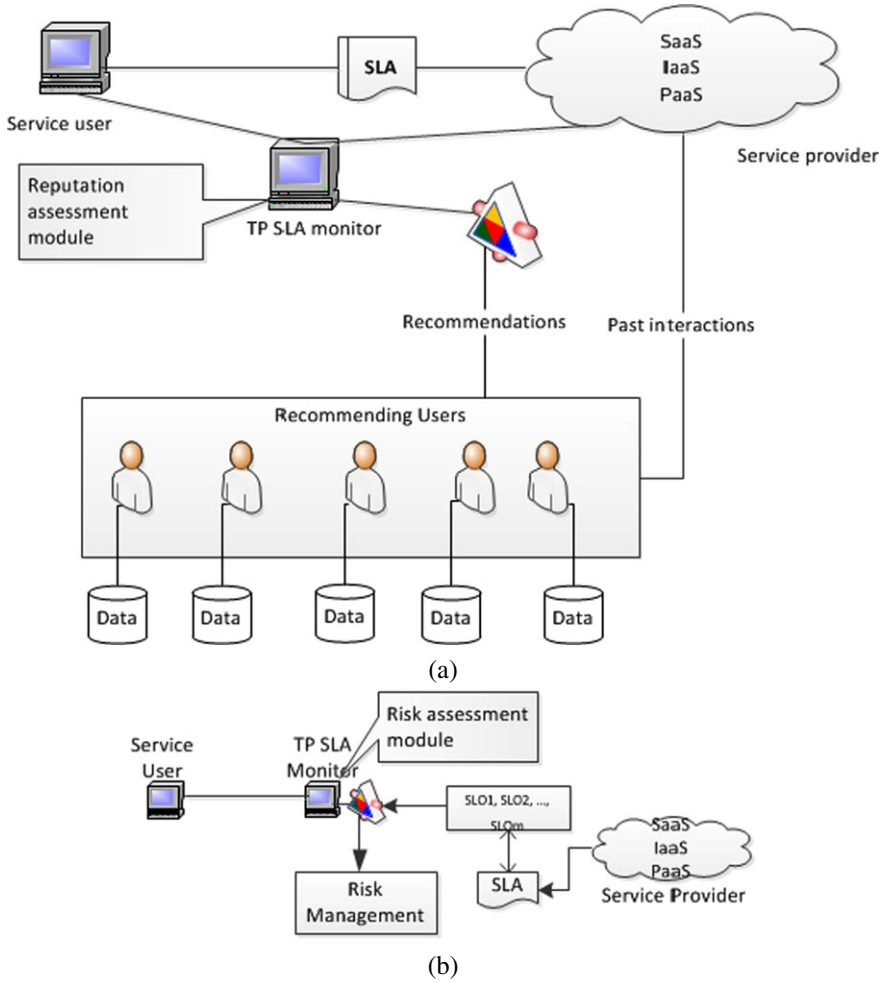

Fig. 5 (a) Framework for pre-interaction SLA evaluation. (b) Framework for post-interaction SLA evaluation

in Fig. 5a. In addition, TP SLA Monitor stores the credibility of each RU in providing opinions about a service provider in its information repository.

The TP SLA Monitor determines the reputation of the service provider before the start of a transaction considering the context and criteria of the SLA parameters to measure performance. For example, if the context of a transaction is bandwidth, initially, it is useful to determine the reputation of the service provider to deliver a promised bandwidth. Reputation is a general perception of a service provider's ability to provide a committed service which is useful for a cloud service user before it starts a transaction with a service provider. After the service user decides to enter into a service with the service provider, the TP SLA Monitor can assess or predict the performance of the service provider at run-time by evaluating its commitment to the defined SLOs, as shown in Fig. 5b.

For assessment of a service's performance at run-time, as proposed by [13], in our framework we consider that the TP SLA Monitor obtains the run-time SLA parameter of the service through the service provider's measurement interface when it is in the current time slot. For example, to evaluate the performance of the service provider offering a bandwidth service, the TP SLA Monitor needs to obtain the run-time value of the bandwidth and compare this with the bandwidth threshold given in the SLA. As a result of a comparison between the SLA threshold of a parameter and its run-time value, the TP SLA monitor determines the transactional risk. Transactional risk ascertains the subcategories of *performance risk* and *financial risk*. Performance risk represents the level of non-commitment of the service provider in the SLOs to the threshold values defined in the SLAs, whereas financial risk represents the financial impact to the service user as a result of this. In this paper, we utilize these two subcategories to ascertain the transactional risk. Such analysis can then be utilized by the service user to make a decision on the continuation of the service. In the next sections, we propose our approach for pre- and post-interaction SLA evaluation.

4 Pre-interaction SLA evaluation

The steps in this phase are as follows:

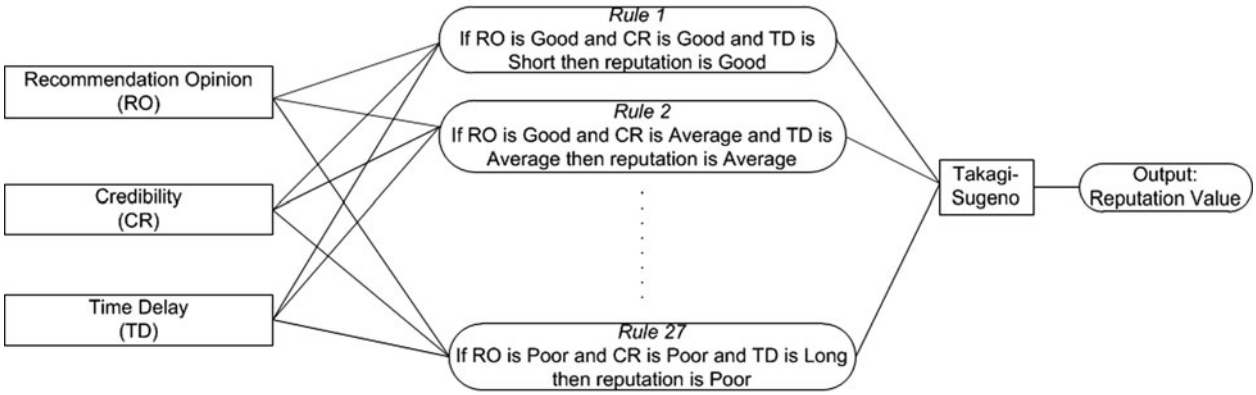

Fig. 6 T-S based Fuzzy Inference System

1. The TP SLA Monitor sends a reputation query soliciting about a service provider's reputation to RUs.
2. For those users with whom the TP SLA monitor has solicited recommendations in the past, it stores their credibility values in its information repository. Credibility here is defined as the trustworthiness of the opinion of an RU. In our proposed solution, credibility is represented on the scale of [0, 5] where 0 represents the lowest (poor) credibility of the RU and 5 represents the highest (good) credibility of the RU.
3. Only those RUs which have interacted with the service provider in the same context and criteria in which the service user wants to interact with the service provider reply to the reputation query.
4. The reply from each RU contains the trust value assigned to the service provider on the basis of its past interaction with that service provider and the time in which this interaction had occurred [11]. When a RU shares this trust value with the TP SLA Monitor, this value is called recommendation opinion. It is represented on a scale of [0, 5] where 0 represents lowest (poor) recommendation opinion given by RU and 5 represents the highest (good) recommendation opinion of RU.
5. The time factor is important since a RU can have multiple interactions with the same service provider in the same context but in different time slots. The time factor can be represented by a time delay which indicates the time elapsed since the last interaction of a RU with the service provider and we represent time delay on a scale of [0, 5] where 0 represents the long time delay and 5 represents the short time delay.
6. For each reputation query response received from a RU, the TP SLA Monitor consolidates the credibility value, time delay value and the recommendation opinion. The outcome of this consolidation is the *reputation* of the service provider provided by a RU.
7. After consolidating these factors for each RU, the TP SLA Monitor then aggregates all reputations by all RUs involved in the reputation query and gives the final reputation value for a service provider.

In our approach we use a fuzzy logic-based approach to consolidate the recommendation opinion of the RU, the credibility of their opinion and the time delay after the last interaction to ascertain the reputation value of a service provider in a particular context. We discuss about that further in the next section.

4.1 Fuzzy logic-based approach

To build a fuzzy logic-based system to determine reputation, we use the factors affecting reputation, namely, recommendation opinion (RO), credibility (CR) and time delay (TD) as input, and the output is the reputation contribution R_i of RUi. We need a fuzzy inference method to map input variables to the output value. Fuzzy linguistic control rules can be used for this mapping. For the fuzzy reasoning, we use the Takagi-Sugeno approach (T-S method) [14] which can be trained using the Adaptive-Neuro Fuzzy Inference System (ANFIS) algorithm. An example of input variables, linguistic control rules, fuzzy reasoning method and output value is given in Fig. 6.

One of the main advantages of the Takagi-Sugeno approach is the use of a tuning algorithm to achieve the best generalization capability of the fuzzy inference system, therefore we use the fuzzy inference system with the Adaptive-Neuro Fuzzy Inference System (ANFIS) algorithm. This approach is called a 'soft computing'-based approach, which is discussed in Sect. 4.2.

The fuzzy sets that we use for input variables RO and CR are [Poor, Average, Good] and TD variables are [Long, Average, Short]. All fuzzy variables are expressed on a scale of 0 to 5, as shown in Figs. 7 and 8 for input variables RO and TD, respectively.

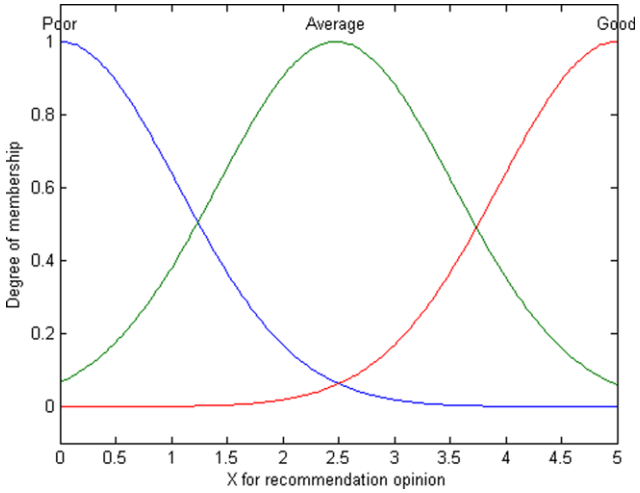

Fig. 7 Membership functions for Recommendation Opinion

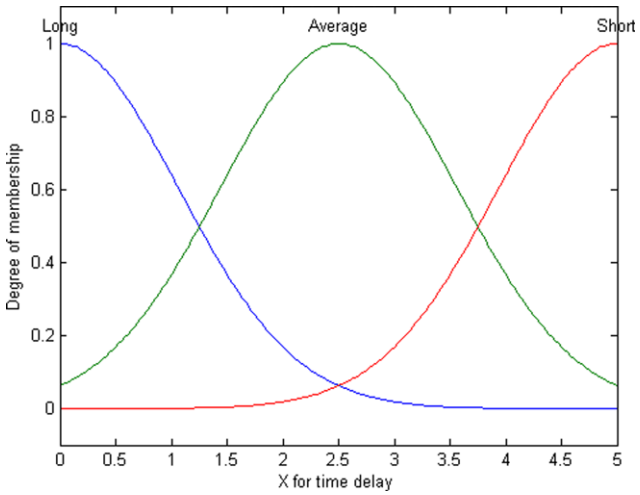

Fig. 8 Membership functions for Time Delay

4.2 Takagi-Sugeno inference approach

The antecedent part of the Takagi-Sugeno rule system is the same as the Mamdani approach. The consequent part of it consists of a linear equation. So, a rule in the Sugeno system takes the form [14]:

IF(x_1 is X_1 AND x_2 is $X_2, \ldots, x_n$ is X_n)

THEN($y_q = a_{q0} + a_{q1}x_1 + \cdots + a_{qn}x_n$)

where: x_1, x_2 are scalar inputs; X_1, X_2 are fuzzy sets; $a_{q0}, a_{q1}, \ldots, a_{qn}$, are real numbers; and y_q is the consequent of the rule.

We consider a system with m fuzzy rules of the T-S form. The form of crisp output is

$$y\left(\underline{x}\right) = \frac{\sum_{q=1}^{m} \alpha_q (a_{q0} + \sum_{s=1}^{n} a_{qs} x_s)}{\sum_{q=1}^{m} \alpha_q} \quad (1)$$

In (1), α_q is the firing strength of rule q. The actual approach to fuzzy reasoning in this case has the following steps:

(a) fuzzify inputs
(b) obtain the firing strength α_q associated with each rule q
(c) obtain the output function of y_q associated with each rule q using the firing strength α_q
(d) obtain the overall output $y(\underline{x})$ using expression (1) given above

We use the first-order Takagi-Sugeno (T-S) approach for fuzzy inference with a linear function for the right-hand side. The inputs on the left-hand side of the fuzzy rule will consist of the factors that affect reputation.

4.3 Working of the FIS for reputation

Using MATLAB's fuzzy logic toolbox, we performed the following four major steps for ANFIS training:

1. loaded training and monitoring datasets
2. created initial FIS
3. trained initial FIS
4. validated trained FIS

Before we detail the experiment process, we explain the source of the training data, what is input-output data, and provide an explanation of our dataset.

4.4 Obtaining training data

ANFIS uses a training data set to tune input and output fuzzy variables. Obtaining training data is a critical part of the whole process [15]. The first step is input selection. Input vectors should cover important aspects of the system without which optimum results cannot be achieved. The omission of any important input points may result in some output values that are either very large or negative. The second part is the values of output vectors in the training data set. The output vectors selected for the training data set must confirm to the rule base. We generated an artificial dataset containing values that span all possible scenarios for the considered inputs in a real-life situation. These values were processed on linguistic rules to generate reputation values (output).

Each tuple of our training dataset consists of an input vector of the form [RECOMMENDATION OPINION, CREDIBILITY, TIME DELAY] and an output value that represents Reputation. As an example, one tuple of our dataset is [3.6 3.7 3.2] with output 3.94. This tuple corresponds to the linguistic rule:

IF RO is Good AND CR is Good AND TD is Average

THEN REPUTATION is Good

We generated 588 such tuples encompassing every aspect of the reputation system, 97 of which we use for monitoring

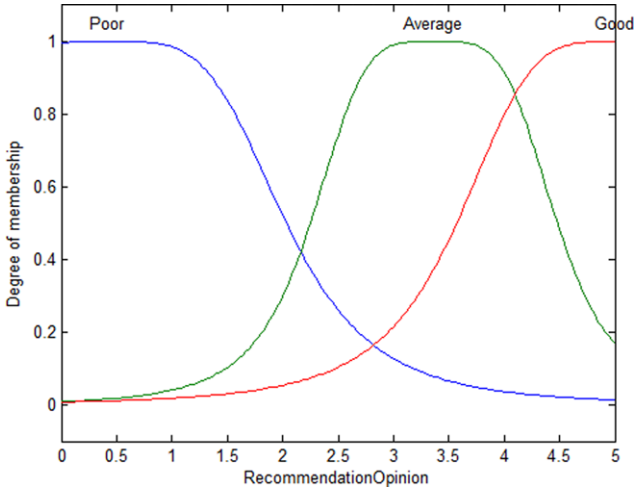

Fig. 9 Membership Function for Recommendation Opinion

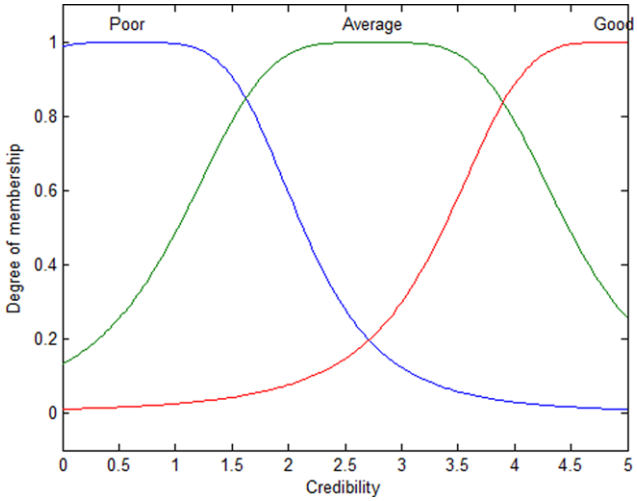

Fig. 10 Membership Function for Credibility

purposes, leaving 491 data points for training purposes. The monitoring data set was also used for cross-validation of the FIS structure alongside the training data set during the training.

4.5 Training process

The training and monitoring data sets are used with tuning algorithms. The actual tuning algorithm used was the adaptive-network-based fuzzy inference system (ANFIS) algorithm [16], since it tunes the parameters of both the input membership function as well as the output coefficients. For the studies carried out in this paper, the version of ANFIS implemented in the Fuzzy Logic Toolbox of MATLAB was used. Initially, we split the data set into three parts: (1) training set (2) monitoring set and (3) a generalization test set. In the training stage, we used the training set error to derive the adjustments in the parameters. This training set error is the difference between outputs provided by the model and the actual value for each instance. We used the monitoring set which is unseen by the training model, defining the output training to monitor the generalization capability of the model. Thus, the monitoring set error = (predicted value − actual value)^2 goes through to minimum before increasing, even though the training set error is still decreasing. This minimum corresponds to the best generalization capability of the model and training ceases when we achieve this. The final training error achieved was 0.2076 and the final monitoring error achieved was 0.2054.

4.6 Results obtained and discussion

Using experimental investigation of the fuzzy logic-based reputation measure, we investigate three fuzzy sets corresponding to the linguistic term set [Good, Average, Poor] to span the input space of RO and CR input factors and three

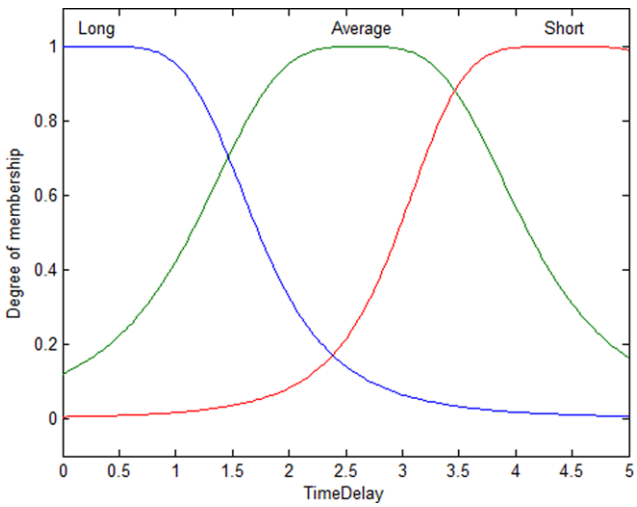

Fig. 11 Membership Function for Time Delay

fuzzy sets corresponding to the linguistic term set [Long, Average, Short] to span the input space of TD input factor. We obtain the following curves for the input membership functions shown in Figs. 9, 10, 11:

Note that the 'Poor' membership function for the Recommendation Opinion and Credibility in Figs. 9 and 10 spans to the right which implies that the 'Poor' recommendation and credibility has a great influence on the reputation of the system. We also note that the 'Average' membership function in the Recommendation Opinion in Fig. 9 is skewed to the right which further emphasizes the influence of the 'Poor' data set in the Recommendation Opinion, whereas the 'Average' membership function in Credibility in Fig. 10 spans wide which implies that it has great influence on ascertaining the reputation. The 'Short' membership function in Fig. 11 spans left which implies that the time delay function is slow, indicating that Time Delay has less influence on

Table 1 Input test cases derived from varying different inputs about two nominal points for 'poor' Reputation

Test case number	Input vector	Reputation
1	[0.8 0.6 1]	0.95 nominal pt 1
2	[1.5 0.6 1]	1.91
3	[0.1 0.6 1]	0.67
4	[0.8 1.1 1]	1.21
5	[0.8 0.1 1]	0.72
6	[0.8 0.6 1.5]	0.93
7	[0.8 0.6 0.5]	0.94
8	[0.5 1 0.8]	1.04 nominal pt 2
9	[1 1 0.8]	1.22
10	[0.1 1 0.8]	0.88
11	[0.5 1.4 0.8]	1.19
12	[0.5 0.6 0.8]	0.82
13	[0.5 1 1.4]	1.07
14	[0.5 1 0.2]	0.94

reputation when compared to other input variables namely, Recommendation Opinion and Credibility.

4.7 Model validity and sensitivity studies

In order to study the validity of the model developed, we used three different methods [17] as follows:

(1) input values corresponding to the training data were loaded and the output obtained from FIS was checked against the output value of the actual training data set;
(2) input values that were not in the original training set were run with this fuzzy model;
(3) sensitivity studies were conducted, varying each of the input factors in turn about a nominal point from the initial measured data.

The first two types of validations can be achieved using the ANFIS interface in MATLAB. In both cases, we obtained correct results within an acceptable tolerance. These validations indicate that our input data set includes all of the representative features of our reputation model. In the case of the second validation method, our model predicts output values as expected in response to input values that were not in the original training set. Discussion on the third validation technique, sensitivity analysis, is given in the next subsection.

4.7.1 Sensitivity study for fuzzy trust model

Sensitivity analysis is a technique used to determine how different values of an independent variable will impact on a particular dependent variable [18]. The results of our sensitivity studies are given in Tables 1–3; the input vector consists of the following factors: [Recommendation Opinion, Credibility, Time Delay].

These results are obtained by perturbing each good, poor and average recommendation opinion and credibility in addition to each long, average and short time delay. The fuzzy trust model is then used to find the overall measure of reputation for each of these perturbed values. Let us initially consider all the results and then the results with respect to some of the input factors in turn.

Nearly all of the values generated for Reputation indicate movements in the correct direction, that is, an increase in input values leads to an increase in reputation and vice versa for a perturbation in a single input factor about the nominal point. The exceptions are indicated by an asterisk (*). However, even for these, while the movement with respect to nominal points might be slightly in the wrong direction (test case number 4 in Table 2), if two adjacent points corresponding to a change in the same input factor are considered, we notice one of these points indicates a movement in the correct direction. Let us consider the value of Reputation for poor Recommendation Opinion, poor Credibility and long Time Delay is approximately between 0 and 1.5, for average Recommendation Opinion, Credibility and Time Delay is between 1.6 and 3.5 and for good Recommendation Opinion, Credibility and short Time Delay is between 3.6 to 5. Almost all points obtained by the change in the input variable, except for test cases 4, 13, 14 in Table 2 and test cases 6, 11, 12 in Table 3, result in a change in Reputation in the expected direction. Hence, the Fuzzy Trust Model seems to adequately model the relationship between input factors and the overall Reputation. The influences of some of these input factors are discussed in the next section.

Recommendation opinion Let us consider the input factor, Recommendation Opinion. We notice the following:

(1) for poor Recommendation Opinion test cases 3, 9 and 10 (in Table 1) and 2 (in Table 2);

Table 2 Input test cases derived from varying different inputs about two nominal points for 'average' Reputation

Test case number	Input vector	Reputation
1	[2.3 2 2.2]	2.13 nominal pt 1
2	[1.9 2 2.2]	1.42
3	[2.7 2 2.2]	2.84
4	[2.3 1.6 2.2]	2.15 *
5	[2.3 2.4 2.2]	2.14
6	[2.3 2.4 1.8]	1.98
7	[2.3 2.4 2.6]	2.25
8	[3 2.8 3]	2.93 nominal pt 2
9	[2.5 2.8 3]	2.59
10	[3.5 2.8 3]	3.34
11	[3 2.3 3]	2.77
12	[3 3.2 3]	3.34
13	[3 2.8 3.5]	2.61 *
14	[3 2.8 2.5]	3.37 *

Table 3 Input test cases derived from varying different inputs about two nominal points for 'good' Reputation

Test case number	Input vector	Reputation
1	[3.6 3.7 3.2]	3.94 nominal pt 1
2	[4.1 3.7 3.2]	4.12
3	[3.1 3.7 3.2]	3.56
4	[3.6 3 3.2]	3.42
5	[3.6 4.2 3.2]	4.05
6	[3.6 3.7 2.8]	4.22 *
7	[3.6 3.7 3.8]	3.93
8	[4.1 4.5 4.3]	4.18 nominal pt 2
9	[3.8 4.5 4.3]	4.11
10	[4.6 4.5 4.3]	4.62
11	[4.1 3.9 4.3]	4.23 *
12	[4.1 4.9 4.3]	3.92 *
13	[4.1 4.5 4.8]	4.33
14	[4.1 4.5 3.7]	4.11

(2) for average Recommendation Opinion test cases 2 (in Table 1) and 3, 9 and 10 (in Table 2);
(3) for good Recommendation Opinion test cases 2, 3, 9 and 10 (in Table 3).

All indicate that an improvement in the Recommendation Opinion leads to an improvement in Reputation and a deterioration in the Recommendation Opinion leads to a deterioration in Reputation, which is the expected result. We notice that the first perturbation in the value of Recommendation Opinion in Table 1 takes the value of Reputation from poor to average. In test case 2 in Table 2, the perturbation in the value of Recommendation Opinion takes the value of Reputation from average to poor. This indicates that Recommendation Opinion has a significant influence on Reputation.

Credibility Let us next consider the input factor Credibility. We notice the following:

(1) for poor Credibility test cases 4, 5, 11 and 12 (in Table 1);
(2) for average Credibility test cases 4, 5, 11 and 12 (in Table 2) and 4 (in Table 3);
(3) for good Credibility test cases 5, 11 and 12 (in Table 3).

Almost all indicate that improvement in Credibility leads to improvement in Reputation and vice versa with the exception of a few test cases. These test cases are numbers 11 and 12 in Table 3 and number 4 in Table 2. However, the other test cases in the same tables indicate movement in the right direction. In nearly all cases, we notice that Credibility has a strong influence on Reputation for all subsets. In some cases, the influence is more pronounced (e.g. test case 4 in

Table 3). Hence Credibility, with Recommendation Opinion, is a significant indicator of Reputation.

Time delay Now we consider the input factor, Time Delay. We notice the following:

(1) for long Time Delay test cases 6, 7, 13 and 14 (in Table 1);
(2) for average Time Delay test cases 6, 7, 13 and 14 (in Table 2);
(3) for short Time Delay test cases 6, 7, 13 and 14 (in Table 3).

Almost all indicate movement in the right direction; that is, reducing time delay (higher value) increases Reputation and increasing time delay (lower value) decreases Reputation. The exceptions are test cases 13, 14 in Table 2 and 6 in Table 3. Although these test cases indicate movement in the wrong direction, all other points lead Reputation in the right direction.

Hence, we conclude that Credibility and Recommendation Opinion have a significant impact on Reputation followed by Time Delay. In other words, the effect of Time Delay is slowly changing which is consistent with our model. From the trained system, the TP SLA monitor can obtain R_i, reputation value given by one recommending user i. Reputation R_k of the service provider K is the aggregation of n recommendations (reputation values) of all recommending users. It is given by:

$$R_K = \sum_{l=1}^{n} \frac{R_l}{n} \tag{2}$$

In the next section, we will demonstrate the working of the proposed FIS to determine the reputation of the service provider.

4.8 Example

To illustrate the working details of our proposed framework, let us consider the following hypothetical scenario. Consider a service user 'A' who wants to select a video conferencing service and Voice over IP (VoIP) service for his business needs. There are two service providers, namely service provider 'B' and service provider 'C' that closely match the business and operational requirements of service user 'A'. In order to do the pre-screening of potential service provider candidates, service user 'A' requests reputation assessments from the TP SLA Monitor. On receiving the request, the TP SLA solicits recommendations from RUs for service providers 'B' and 'C'. Suppose RU1 and RU2 send their recommendation about service provider 'B' and RU1 and RU3 send their recommendations about service provider 'C'. Using the soft-computing-based approach introduced earlier,

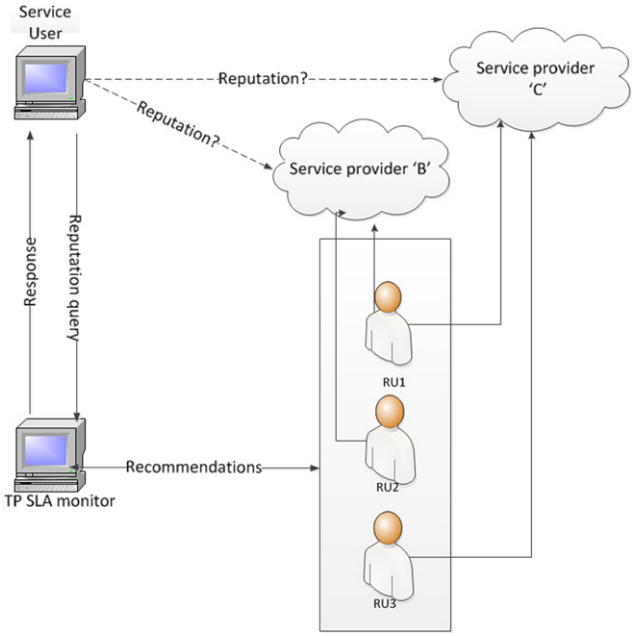

Fig. 12 Service selection process

the TP SLA monitor computes the final reputation values of service providers 'B' and 'C'. The process of selection is depicted in Fig. 12.

Based on RO, CR and TD values, the TP SLA monitor computes the reputation value provided by one RU. For example, using our reputation assessment methodology, the reputation values provided by RU1 and RU2 about service provider 'B' are 3.93 and 4.11, respectively. The reputation values provided by RU1 and RU3 for service provider 'C' are 4.12 and 3.56, respectively. Using Eq. (2), the TP SLA monitor computes the final reputation value of service provider 'B' as 4.02 and the final reputation value of service provider 'C' as 3.84. On the basis of the high reputation value, service user 'A' may want to enter into an interaction with the service provider 'B'.

After determining the reputation of a service provider and selecting it for service provision, a service user now needs to ascertain that the service provider will deliver the promised service. This can be assessed by risk assessment in the post-interaction SLA evaluation methodology which is discussed in the next section.

5 Post-interaction SLA evaluation

As depicted in Fig. 4, once a service selection process is complete, then starts using the *continuous risk-based evaluation* service which is implemented by the TP SLA monitor as shown in Fig. 5b. As discussed in Sect. 2, there are several methods to evaluate a service provider's performance after the initiation of the service in order to decide on the

continuation of the service, one of these being transactional risk analysis. In this section, we discuss how we use transactional risk analysis for SLA evaluation and management in the post-interaction time-phase.

Risk assessment is that phase of risk analysis where a hazard or threat is identified, quantified. This then leads to the evaluation and management in the risk evaluation and risk management phases. In its simplest form, risk assessment starts with the identification of a hazardous event E_i, followed by determining the likelihood or probability P_i of this undesirable event occurring [19]. The second phase of risk assessment is to determine the consequence C_i as a result of this event occurring. This consequence C_i, is a measure of the impact of E_i. The overall risk R then can be given as:

$$R = \sum_i P_i \cdot C_i \tag{3}$$

It should be noted that determination of P_i is objective and depends upon past experience or historical data, whereas the determination of C_i is subjective. In the literature, various techniques have been proposed for risk assessment. One such approach by which risk can be analyzed is convolution [20]. Convolution is the integral operator which expresses the amount of overlap and impact of one function as it shifts over the other. Convolution has been used in different applications such as power generation systems, to determine the expected demand not supplied by determining the effect of load demand on the generation capacity of the generation system. Mathematically, convolution of X and Y, both of which are independent random variables, is given by:

$$Z = X \oplus Y \tag{4}$$

In our approach, we utilize the principles of convolution and use it to ascertain the transactional risk of the service provider not meeting the defined threshold levels in the SLAs.

5.1 Example

Continuing with the case study introduced in Sect. 4.8, once service user 'A' enters into a business interaction with service provider 'B', it is important for him to monitor whether service provider 'B' provides the desired level of service to achieve his desired outcomes. From user 'A's perspective, video conferencing and VoIP services are important as business meetings are arranged with his business customers through these services. Not being able to conduct business meetings may result in financial loss and damage the business reputation of service user 'A'.

Let us consider that service user 'A' on a particular day expects to secure a business contract of $15,000 through his video conferencing meetings. Since services such as VoIP and video conferencing depend on bandwidth, let us consider that service provider 'B' has the maximum capacity to provide a bandwidth of 24 Mbps. Further, let us consider that service provider 'B' establishes a threshold of 12 Mbps in SLA against which the bandwidth service is expected to be delivered. If a service provider 'B' delivers a bandwidth below 12 Mbps, these bandwidth levels are considered as service level degradation since they deviate from the threshold defined in the SLAs. It is important to measure the service deviation levels in order to measure the performance of service provider 'B'. Moreover, let us consider that the video conferencing service does not work if the bandwidth falls below 6 Mbps.

5.2 Determining the service deviation levels

Suppose we want to calculate the probability that the bandwidth falls below the threshold (12 Mbps) for the video conferencing service. Depending on the time phase in which the current time slot is, the TP SLA monitor can either use the past performance of the service provider or predict the performance of the service provider in the future time slots, based on its past performance history by using the techniques mentioned in Hussain et al. [21]. By using the techniques we determine the level of bandwidth delivery of the service provider in an advance time period of 24-hour duration. If a discrete random variable Q represents the levels of bandwidth for a 24-hour interval, the Probability Mass Function (pmf) for this random variable is given by:

$$f : q \rightarrow P\,(Q = q) \quad \text{or} \quad f\,(q) = P\,(Q = q)$$

where variable q indicates the values within the range of the random variable.

Relative frequency for each class interval is then calculated as:

$$P\,(Q = q) = \frac{Fr}{n} \tag{5}$$

where Fr denotes the frequency of bandwidth on class intervals representing bandwidth and n represents the total bandwidth. By definition of pmf:

$$P\,(Q) = \sum_i \frac{Fr_i}{n} = 1 \tag{6}$$

where i represents the class intervals.

Using (5) for our case study, we obtain the graph shown in Fig. 13.

In Fig. 13, where $q < 12$ it indicates the chances of the service not meeting the required threshold. To determine the level of service degradability, we convert this portion of the distribution into another distribution that represents the service deviation levels of a bandwidth. If a discrete random

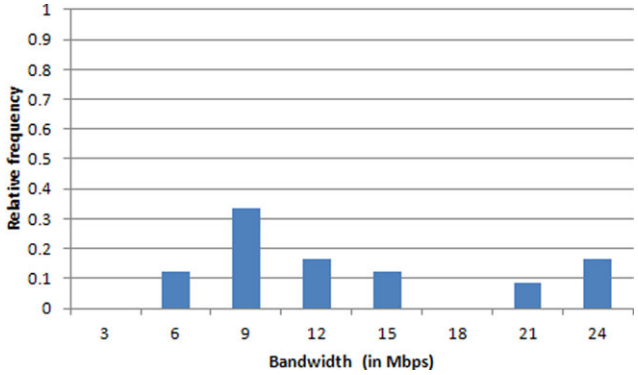

Fig. 13 Probability Mass Function for bandwidth levels

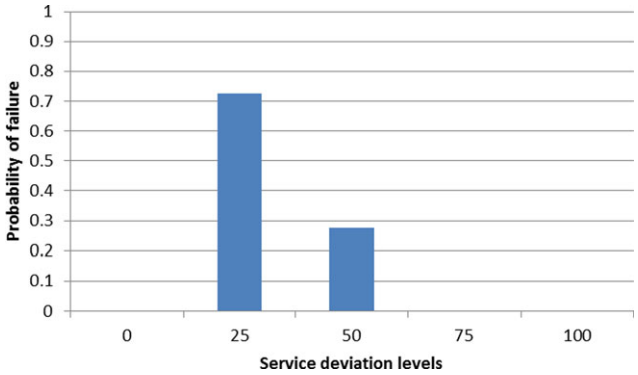

Fig. 14 Level of service deviation from minimum threshold requirements

variable R represents service deviation levels and a variable r indicates the values within the range of the random variable, the probability of the level of service deviation is calculated as:

$$P\,(R=r) = \frac{P(q_i)}{\sum_{i=0}^{i<12} P(q_i)} \tag{7}$$

Consequently, we obtain a probability mass function for random variable R given as:

$$P\,(R) = \sum_{j} P(r_j) = 1, \quad \text{where } j = 0, 3, \ldots, 9 \tag{8}$$

By normalizing service deviation distribution to a percentage scale, as shown in Fig. 14, we obtain the different levels of deviation from the threshold point defined in the SLAs. The next step in determining the transactional risk is to ascertain the impact as a result of the deviation in the service.

5.3 Determining the financial loss as a result of service deviation levels

Service user 'A', through his investment of resources in service provider 'B', wants to benefit financially over a specific period of time. However, due to service degradation,

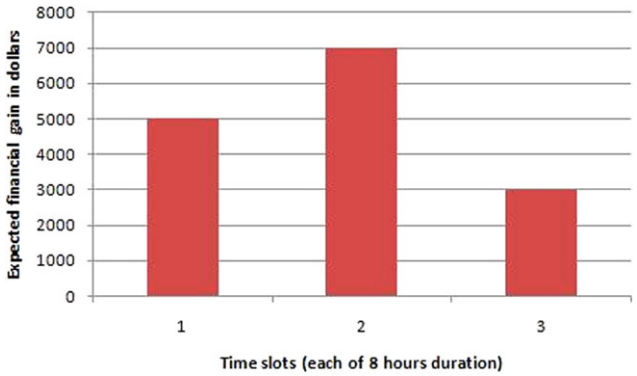

Fig. 15 Expected financial gain in each time slot

the expected outcome may not be achieved which may lead to the service user experiencing a financial loss. In order to determine this, the first step in this process is to determine the expected financial gain which user 'A' wants to achieve over a period of time. The service user expects to achieve financial gain in each time interval if the service is delivered as promised by the service provider. From such expected financial gain in each time slot over a time period, we propose that the TP SLA monitor plots the Expected Financial Gain Curve (EFGC).

5.3.1 Expected financial gain curve

We use our case study to demonstrate the process of plotting the Expected Financial Gain Curve (EFGC). As discussed in our case study, service user 'A', over a period of 24 hours, expects to gain a benefit of $15,000 as a result of a service provided by service provider 'B'. In order to capture the dynamic nature of risk over this period, we consider the methodology proposed by Chang et al. [10], where the time space of the business interaction is established and then divided into different time slots. In this case study, the time space is 24 hours and is divided into 3 time slots of 8 hours each. In 24 hours, as a result of a video conference service, let us suppose that the service user expects to achieve a financial gain, as shown in Fig. 15.

In order to determine the impact of service deviation levels on the expected financial gain, we first need to determine the collective expected financial gain that was anticipated throughout the time period. This is achieved by plotting the cumulative probability density function of the amount of resources invested over the time period [22, 23]. The curve, called an Expected Financial Gain Curve (EFGC), is shown in Fig. 16.

To determine the financial loss, the impact of service deviation levels on the expected financial gain has to be determined. We achieve this by convolution. However, for convolution both the random variables need to be on a uniform scale. So the expected financial gain first has to be converted

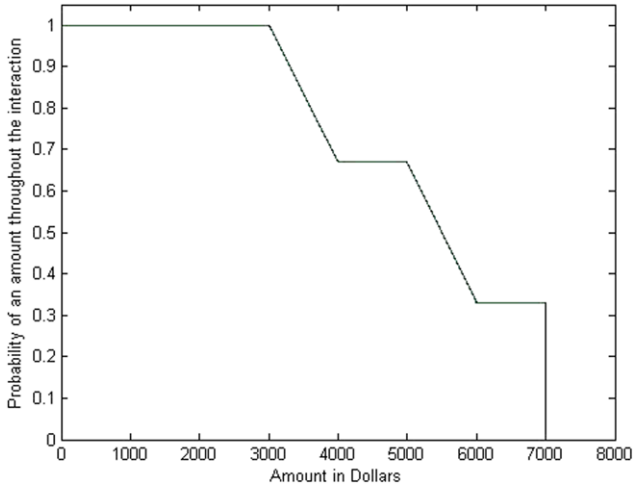

Fig. 16 Expected Financial Gain Curve

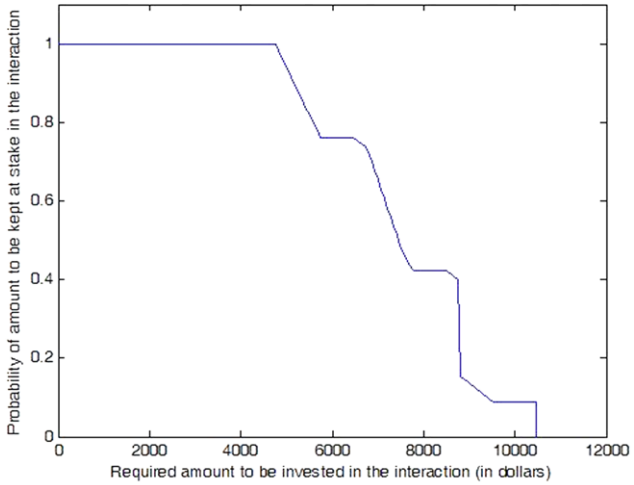

Fig. 17 Curve representing the financial amount to be kept at stake due to service degradation

to the percentage scale. Convolution results in a curve which we call the Actual Resource Investment Curve (ARIC). As opposed to the EFGC, the ARIC indicates the probability of an amount required to be kept at stake in order for service user 'A' to achieve the same outcome with service provider 'B'.

5.3.2 Convolution process

Using (4), convolution for our random variables can be written as:

$$ARIC = FL \oplus EFGC$$

where: $ARIC$ = Actual Resource Invested Curve, FL = random variable indicating service deviation levels, $EFGC$ = Expected Financial Gain Curve.

In other words, convolution is the sliding of the $EFGC$ over the projections of service deviation levels. We use the following formulae to perform this operation:

$$ARIC(x) = \sum_{i=1}^{n} p_i * EFGC(x - FL_i) \quad \text{for } (x - FL_i) \geq 0$$

or

$$ARIC\,(x) = \sum_{i=1}^{n} p_i \quad \text{for } (x - FL_i) < 0 \tag{9}$$

where: n = the number of service deviation levels, x = the point at which $ARIC$ has to be determined, FL_i = magnitude of service deviation level i, p_i = magnitude of occurrence of service deviation level i, $EFGC(x - FLi)$ = Expected Financial Gain Curve at point $(x - FLi)$.

The effect of the service deviation levels FL_i over each point of curve (x) is determined and after the convolution process, the $ARIC$ curve is produced, as shown in Fig. 17.

Comparing Figs. 16 and 17, it is evident that the curve in Fig. 17 is inflated compared to the curve in Fig. 16 because of the additional resources that need to be kept at stake to obtain the financial gain due to the different levels of service deviation or service degradation. These additional levels of resources can be viewed as the financial loss that arises as a result of service degradation. However an important point to note is the curve obtained in Fig. 17 is based on dependable criteria where a service provider fails to provide a required level of service. Apart from this, there may be other associated costs outside the dependence of the service provider. These costs are called as non-dependable criteria and in cloud computing they may be costs such as the migration cost that occurs due to the migration of a service from one cloud service provider to another. To achieve business goals, service user 'A' also needs to determine the effect of non-dependable criteria in addition to dependable criteria.

5.4 Determining the impact of additional factors when ascertaining financial risk as a result of service degradability

Continuing our case study, if service user 'A' chooses to migrate VoIP or video conference service from service provider 'B' to another service provider due to degradation of bandwidth service, the loss from other non-dependable factors such as migration cost adds to the financial loss that could be experienced as a result of service degradation. Let us consider that the migration cost for service user 'A' to transfer its service to another provider is $3000. This may be either a one-off payment or it may be spread over a period of time. Let us consider that service user 'A' opts to make a payment over a period of time with the probability mass function (pmf) as shown in Fig. 18.

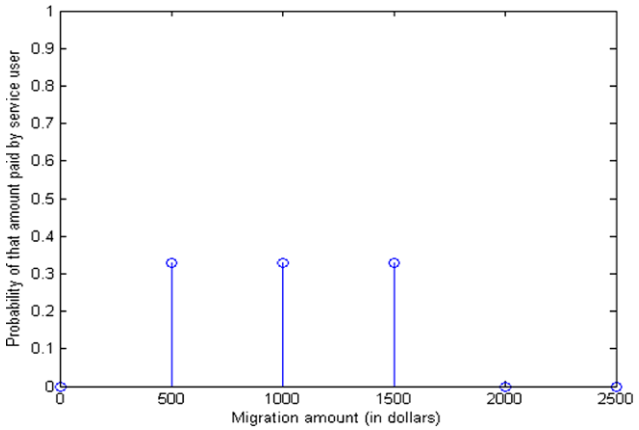

Fig. 18 The probability of extra level of resources for migration cost

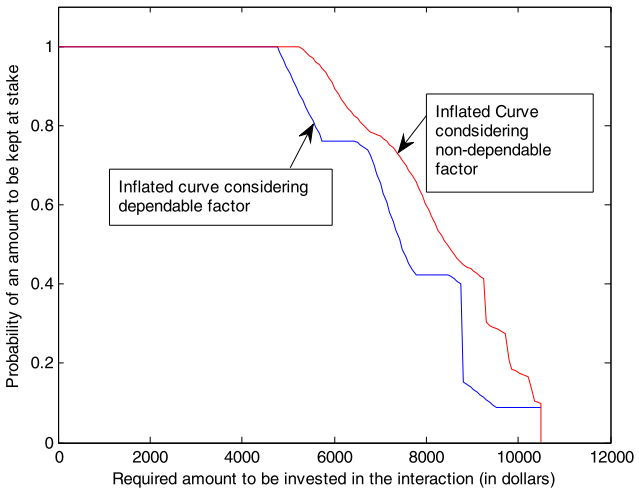

Fig. 19 Comparison of Actual Resource Investment Curve and Total Resource Investment Curve

We call the curve depicted in Fig. 18 the Extra Investment Curve (EIC). To determine the total financial resources that service user 'A' has to keep at stake, we need to combine the EIC with the ARIC. We achieve this through convolution and we obtain the Total Resource Invested Curve (TRIC) as a result. If we compare the two curves, namely TRIC and ARIC, we obtain the graph shown in Fig. 19.

It can be seen from Fig. 19 that the TRIC is inflated compared to the ARIC. This indicates the extra level of resources that service user 'A' has to keep at stake for the migration of bandwidth service. The higher the levels and probabilities of the resources in the EIC, the greater will be the inflation in the TRIC. Further, the comparison between the total resource investment curve (TRIC) and the expected financial gain curve given in Fig. 20 indicates the extra amount that service user 'A' has to keep at stake to achieve business objectives.

The difference between the expected financial gain curve and the inflated curve due to dependable and non-dependable criteria is shown in Fig. 21. This curve is termed a 'loss of investment' curve.

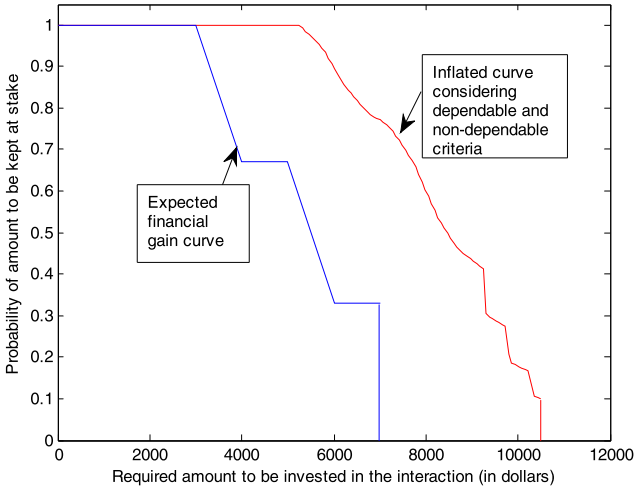

Fig. 20 Comparison of Expected Financial Gain Curve and Total Resource Investment Curve

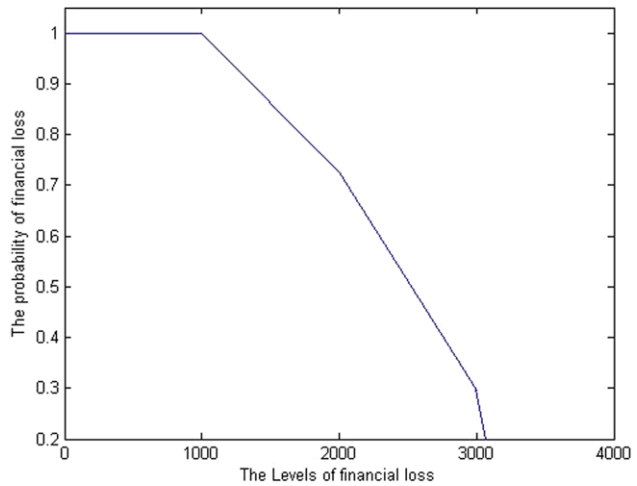

Fig. 21 The loss curve in the interaction

Initially, the service user was expecting to achieve the financial gain as shown in Fig. 16 but due to possible level/s of service degradation as shown in Fig. 14, the service user has to keep extra amounts of resources at stake to achieve the desired outcomes. The TP SLA monitor informs about the determined analysis to the service user who may then decide whether to take the risk management step and continue the service with the service provider or migrate to a new service provider. Having such continuous evaluation of service provider performance on regular intervals allows the service user to take timely action to ensure that s/he achieves the desired outcomes.

6 Conclusion

In this paper, we proposed a framework for SLA management which consists of pre-interaction and post-interaction SLA evaluation processes. In the pre-interaction SLA evaluation process, we used a fuzzy-logic-based approach to determine the reputation of a service provider on the basis of recommendations provided by a group of users. This reputation value is used to select the best possible service provider for the requested service. In the post-interaction time phase, we used a transactional-risk-based approach to determine the level of service degradability in the performance of a service provider and the impact to the service user as a result of that. By using analysis, the service user can make an informed decision regarding the continuity of a service. In our future work we aim to propose a decision making technique that considers the output of these two approaches conjointly and recommend an informed decision on the continuity of the service to the service user.

References

1. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H.: Big data: the next frontier for innovation, competition, and productivity. McKinsey Global Institute, May (2011)
2. Gantz, D.R.J.: (2011). Extracting value from chaos. Available at http://idcdocserv.com/1142. Accessed on June 5, 2011
3. Buyya, R., Pandey, S., Vecchiola, C.: Market-oriented cloud computing and the cloudbus toolkit. In: The First International Conference on Cloud Computing (CloudCom), Beijing, pp. 24–44 (2009)
4. Sun, Y.L., Perrott, R., Harmer, T.J., Cunningham, C., Wright, P., Kennedy, J., Edmonds, A., Bayon, V., Maza, J., Berginc, G., Hadalin, P.: SLA-aware resource management. In: Grids and Service-Oriented Architectures for Service Level Agreements, pp. 35–44. Springer, New York (2010)
5. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Commun. ACM **53**, 50–58 (2010)
6. He, Q., Yan, J., Kowalczyk, R., Jin, H., Yang, Y.: Lifetime service level agreement management with autonomous agents for services provision. Inf. Sci. **179**, 2591–2605 (2009)
7. Quillinan, T.B., Clark, K.P., Warnier, M., Brazier, F.M.T., Rana, O.: Negotiation and monitoring of service level agreements. In: Grids and Service-Oriented Architectures for Service Level Agreements, pp. 167–176. Springer, New York (2010)
8. Khader, D., Padget, J., Warnier, M.: Reactive monitoring of service level agreements. In: Grids and Service-Oriented Architectures for Service Level Agreements, pp. 13–22. Springer, New York (2010)
9. de Oliveira, E., Pfreundt, F.-J.: Monitoring service level agreements in grids with support of a grid benchmarking service. In: Grids and Service-Oriented Architectures for Service Level Agreements, pp. 1–11. Springer, New York (2010)
10. Chang, E., Hussain, F., Dillon, T.: Trust and Reputation for Service-Oriented Environments: Technologies for Building Business Intelligence and Consumer Confidence. Wiley, New York (2005)
11. Schmidt, S., Steele, R., Dillon, T., Chang, E.: Building a fuzzy trust network in unsupervised multi-agent environments. In: On the Move to Meaningful Internet Systems Workshops, Agia Napa, Cyprus, pp. 816–825 (2005)
12. Hussain, O.K., Chang, E., Hussain, F.K., Dillon, T.S.: A methodology to quantify failure for risk-based decision support system in digital business ecosystems. Data Knowl. Eng. **63**, 597–621 (2007)
13. Kaliski, B.S. Jr., Pauley, W.: Toward risk assessment as a service in cloud environments. In: 2nd USENIX Conference on Hot Topics in Cloud Computing, Boston, MA, pp. 1–7 (2010)
14. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. Syst. Man Cybern. **15**, 116–132 (1985)
15. Jang, J.-S.R.: Input selection for ANFIS learning. In: 5th International Conference on Fuzzy Systems, New Orleans, LA, pp. 1493–1499 (1996)
16. Jang, J.-S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. **23**, 665–685 (1993)
17. Chang, E., Dillon, T.: A usability-evaluation metric based on a soft-computing approach. IEEE Trans. Syst. Man Cybern., Part A, Syst. Hum. **36**, 356–372 (2006)
18. http://www.investopedia.com/terms/s/sensitivityanalysis.asp#axzz1z0Oi7az5
19. Modarres, M.: What Every Engineer Should Know About Reliability and Risk Analysis. Dekker, New York (1993)
20. Li, W.: Risk Assessment of Power Systems Models, Methods, and Applications. Wiley, New York (2005)
21. Hussain, O., Chang, E., Hussain, F., Dillon, T.: Determining the Failure Level for Risk Analysis in an e-Commerce Interaction. Advances in Web Semantics, vol. I, pp. 290–323. Springer, Berlin (2009)
22. Hussain, O., Dillon, T., Hussain, F.K., Chang, E.: Probabilistic assessment of financial risk in e-business associations. Simul. Model. Pract. Theory **19**, 704–717 (2011)
23. Walpole, R.E., Myers, R.H.: Probability and Statistics for Engineers and Scientists, 5th edn. Macmillan Co., London (1993)

Adil Hammadi is a Ph.D. student at School of Information Systems, Curtin University, Perth, WA. In 2006, he received his Master of Science (by research) degree from Deakin University, Australia. In 1998, he received Master of Information Technology degree from Charles Sturt University, Australia. His areas of interests are Software Engineering, Data Mining, Business Intelligence, Trust and Risk based Informed Decision Support Systems. He is a member of IEEE, ACS and AUSOUG.

Omar Khadeer Hussain received his Ph.D. degree in Computer Science in 2008 from Curtin University. Since 2008, he is currently a Research Fellow with the School of Information Systems at Curtin University. His research interests are in the area of Risk Assessment and Management, Trusted Computing, Informed Decision Support Systems. He is a member of the IEEE and the IEEE Computer Society.

Tharam Dillon received the Ph.D. degree in electrical and computer systems engineering from Monash University, Australia, in 1974. He is the Editor-in-Chief of the International Journal of Computer Systems Science and Engineering and Engineering Intelligent Systems. He has published more than 750 papers published in international conferences and journals. He is the author of five books and the editor of five books. His current research interests include Web semantics, ontologies, Internet computing, e-commerce, hybrid neurosymbolic systems, neural nets, software engineering, database systems, and data mining. Prof. Dillon is the head of the International Federation for Information Processing (IFIP) International Task Force WG2.12/24 on Semantic Web and Web Semantics, the Chairman of the IFIP WG12.9 on computational intelligence, the IEEE Industrial Electronics Society Technical Committee on Industrial Informatics, and the IFIP Technical Committee 12 on Artificial Intelligence.

Farookh Khadeer Hussain received the Bachelor of Technology degree in computer science and computer engineering; the M.S. degree in Information Yechnology from the La Trobe University, Melbourne, Australia; and the Ph.D. degree in Information Systems from Curtin University of Technology, Perth, Australia, in 2006. He is currently a Faculty member at School of Software, Faculty of Engineering and Information Technology, University of Technology, Sydney, Ultimo, NSW. His areas of active research are trust, reputation, trust ontologies, data modeling of public and private trust data, semantic web technologies and industrial informatics. He works actively in the domain of making informed business decisions (business intelligence) through the use of trust and reputation technology. He is interested in the application of trust and reputation as a technology, as a business analysis and intelligence tool, and the applications of trust and reputation to various domains.

2012 26th International Conference on Advanced Information Networking and Applications Workshops

A Framework for SLA Assurance in Cloud Computing

Adil M. Hammadi, Omar Hussain

Digital Ecosystems and Business Intelligence Institute
Curtin University
Perth 6102, Australia
e-mail: {adil.hammadi@postgrad, o.hussain@cbs}.curtin.edu.au

Abstract— **One of the challenges of cloud computing is SLA assurance. On-demand and self-serve nature of cloud requires ascertaining real-time QoS assurance to meet SLA specifications of the consumer. A third-party service provider could be used to provide on demand QoS assessment service. We propose a SLA monitoring framework to address the real-time QoS assessment issue. The framework consists of two modules namely, reputation assessment module and transactional risk assessment module. Third-party service provider equipped with such assessment modules can provide real-time assessment for consumer's informed decision making to continue using a service or to migrate to another service provider in the case of service degradation.**

I. INTRODUCTION

There are various definations of Cloud computing proposed in the literature. One definition proposed by Buyya et al. [1] is "*a parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements*". The major applications of cloud computing include social networking, business, scientific and engineering applications. The IDC (International Data Corporation) has indicated that, due to its great potential, spending on IT cloud services will increase from about $16 billion in 2008 to about $42 billion in 2012. Cloud computing is still in its infancy but it is expected to reach the productivity phase in just 2-5 years [1]. Currently, Cloud services are offered by large vendors like Amazon, Google, Microsoft, SalesForce, Sun MicroSystems and IBM.

Cloud computing is based on a grid computing and service-oriented computing paradigm. It shares common features such as service discovery, service description, service composition and service management, with service-oriented computing. It is based on the architectural principles and software specifications of service-oriented computing to connect computers and devices using standardized protocols across the internet [2]. However, cloud is different from service-oriented computing in providing flexible on-demand services with flexible cost models for the services [3]. Virtualization is one of the key concepts of cloud computing which enables dynamic service provisioning by dynamically allocating or de-allocating resources based on demand. The economic benefit of cloud makes it an attractive choice for many real-world businesses. In cloud, since services are requested on demand, the service charges are applied only to the service that has been consumed. There is no upfront cost for the service infrastructure and there are no charges for service subscription.

However, apart from all the benefits of cloud computing, several issues still need to be resolved, one of which is service level management. SLA management is intended to ensure that the defined service meets certain criteria that were established in the agreement. Failure to meet the terms of the agreement will result in service level degradation. Since cloud computing promises "self-serve" which has to be done in an automatic manner, a good optimization algorithm is needed that takes dynamic resource allocation into account based on the user demand. Such an algorithm should be capable of notifying the resource manager, thereby enabling it to make real-time evaluation and adjustments when there is an unexpected occurrence in the cloud environment. In the absence of such an algorithm, it is difficult to automatically manage resources without the violation of service level objectives [4].

The occurrence of service level degradation in the cloud has an impact on business outcomes. This impact in the business domain could be measured in terms of not achieving the desired outcome and loss of investment. The issue of service level degradation is related to run-time Quality of Service (QoS) evaluation. QoS is a metric that defines the quality benchmark which the given service should meet. In other words, we need efficient QoS assurance software that takes into account the self-serve and on-demand nature of the cloud in order to avoid financial loss that may be incurred due to service level degradation. Such QoS assurance software helps in the making of informed decisions regarding service re-composition as a result of service level degradation.

The problem is how to develop such QoS software in the cloud since QoS assessment methods developed for traditional IT environments do not fit the dynamic nature of the cloud. One way to address this problem is to develop an on-demand QoS assessment service based on the concept that, just as cloud is on-demand, the QoS assessments applied to the cloud can be on-demand as well [5]. This QoS assessment service can be implemented by a trusted third-party service provider that provides unbiased assessment to both the consumer and service provider. The same concept has been discussed by Dan et al. [6] who suggested that service monitoring can be delegated to third-party service providers. Keynote Systems, Inc. and Xaffire, Inc. are examples of such third-party service providers.

978-0-7695-4652-0/12 $26.00 © 2012 IEEE
DOI 10.1109/WAINA.2012.280

In business applications, service level degradation affects not only the expected financial outcome for both consumer and service provider, but also the reputation of the service provider. A QoS parameter such as *trust* measures the consumer's confidence in the service provider and it is a key factor for the success of business transactions. Another QoS parameter by which the success of a business transaction can be measured is *risk*. Risk expresses the expected financial loss and the level of financial gain achieved for both consumer and service provider as a result of not achieving the desired outcomes. The concepts of trust and risk are interrelated and complement each other. Low reputation may indicate high risk and high reputation may indicate low risk. Trust and risk assessments help the consumer to make an informed decision about whether to continue using service/services or to re-compose services in the case of service level degradation.

From the above discussion, we conclude that trust and risk assessments have to be done in an automatic fashion because of the self-serve and on-demand feature of the cloud. In this paper, we propose efficient risk and reputation assessment methodologies for a third-party assessment provider.

II. SLA Monitoring Architecture in the Cloud Computing

In this section, we discuss cloud architecture and the fact that it leads to the issue of dynamically allocating and de-allocating resources while preserving SLA compliance. In other words, the benefits of self-serve, on-demand features of cloud come at the cost of service level violations. The cloud paradigm will help us to understand the problem.

A. Cloud Computing paradigm

To fully explain what constitutes a cloud computing system, we discuss a cloud computing reference model. Cloud computing can be classified into three main categories: Infrastructure/Hardware-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS).

IaaS provides physical resources such as storage, computers, network etc., at system level. This physical infrastructure is managed by core middleware which provides a run-time environment for applications to utilize physical resources. Core middleware consists of virtualization technologies which are further classified as hardware virtualization and software virtualization. Hardware virtualization provides partitioning of physical resources such as CPU and memory using virtual machines.

PaaS is a layer above IaaS and provides the environment for software development. For this purpose, PaaS consists of tools, programming environments and configuration management based on a virtual machine layer. Examples of PaaS are Google AppEngine and Microsoft Azure.

SaaS, with the support of IaaS and PaaS, offers services to consumers where different social, business, engineering etc. applications are offered and utilized.

The cloud computing categories discussed above can be represented by the cloud computing reference model given in Fig. 1.

Fig. 1. Cloud computing reference model (adopted from Buyya et al.[1])

Since leveraging of resources is done at the middleware layer through virtualization, an efficient resource allocation mechanism is needed at the physical layer, taking into account dynamic nature of cloud. This automatic scaling needs automatic QoS assessment in the cloud.

B. Importance of automatic QoS assessment in cloud

Automated control, at the heart of the cloud computing paradigm as discussed above, takes away the human control element [5]. In traditional IT environments, trained humans monitor and control QoS issues. In a cloud environment automated QoS control mechanisms are needed to replace their human counterparts. In this paper, we will introduce such QoS assessment methodologies that can be used for automatic service monitoring by a third-party service provider to achieve near real-time SLA assessment.

As a part of automatic self-assessment, a service provider needs to perform continuous evaluation of its run-time environment for each tenant according to the SLA. To remove the burden of monitoring, a trusted third-party is the best choice to provide unbiased opinion.

We discussed the importance of trust and risk in Section I. To achieve real-time QoS assessment, we propose an SLA monitoring framework that consists of two modules: the trust assessment module and the risk assessment module. This framework will provide the benefits of a real-time QoS assessment mechanism that also incorporates the user's feedback about the service quality provided by a service provider. In the next section, we discuss our proposed framework which is depicted in Fig. 2 below.

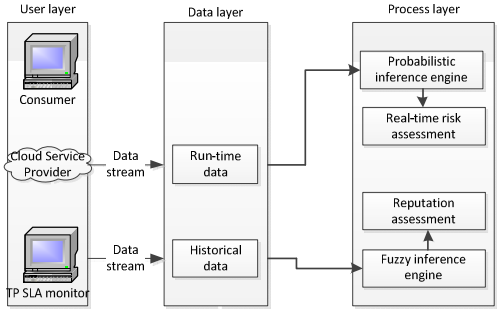

Fig. 2. SLA monitoring framework

III. Factors Required For a SLA Monitoring Framework

As shown in Fig. 2, our proposed framework consists of a Third-Party Service Level Agreement monitor (TP SLA Monitor), a service provider that monitors near real-time performance of service provider based on the SLA. The TP SLA Monitor consists of two modules for QoS assessment: reputation assessment module and risk assessment module.

Although both modules are independent, they complement each other. To start the monitoring process, the TP SLA Monitor should first define the total time space for which QoS has to be assessed. It then divides this time space into pre-interaction and post-interaction time phases [7]. QoS evaluation in pre-interaction time phase is achieved by using reputation assessment methodology by soliciting the recommendations given by *Recommending Users* (RU). RU in our framework is a group of users who received a reputation query from a consumer and they reply on the basis of their past experiences with the service provider. The TP SLA Monitor aggregates the opinions of RU to reach the final reputation value of the service provider as depicted in Fig. 3. This reputation value then can be used to determine near real-time assessment of SLA parameters in the post-interaction time phase using risk assessment methodology.

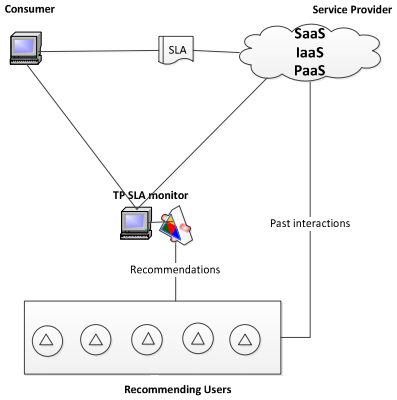

Fig. 3. Pre-Interaction time phase assessment model

A. Importance of trust in cloud computing

When a consumer establishes a business association with a service provider, the TP SLA Monitor can provide information on the reputation of the service provider in the context of the transaction. For example, if the context of the transaction is bandwidth, the reputation of the service provider to provide promised bandwidth is determined by the TP SLA Monitor. The task of this Monitor is to seek opinions (trust values) about the service provider's capability to provide the requested service, such as bandwidth, from the RU who used this service. The TP SLA Monitor determines the reputation of the service provider before the start of a transaction considering context of transaction only, without going into specific details regarding SLA parameters to measure the performance. In our example, initially it is useful to determine the reputation of the service provider to deliver a promised bandwidth. This evaluation needs the context of the transaction only, not the threshold of bandwidth promised by the service provider in the SLA. Reputation is a general perception of a service provider in providing committed service which is useful for a cloud service consumer before it starts a transaction with a service provider. We discuss the reputation calculation methodology of the TP SLA Monitor in the Section IV.

B. Importance of risk in cloud computing

After the consumer decides to enter into a transaction with the service provider, the TP SLA Monitor can assess the performance of service provider at run-time by evaluating the risk. For the calculation, only necessary information needed by the TP SLA Monitor is derived from the SLA between consumer and the service provider. This SLA information contains the conditions and thresholds of the SLA parameters. In our example, the TP SLA Monitor evaluates reputation of the service provider in providing bandwidth and found it trustworthy. Now the TP SLA Monitor evaluates the performance of the service provider based on the bandwidth threshold defined in the SLA and by comparing actual bandwidth value at run-time with the threshold. For transactional risk evaluation, the TP SLA Monitor requires necessary data such as SLA parameter values at run-time. The TP SLA Monitor could obtain the required run-time SLA parameters though privileged access to service provider's measurement interface [5] or by probing a consumer transaction [6]. As a result of comparison between SLA threshold of a parameter and its run-time value, failure levels are determined which indicate the non-compliance of a service provider with SLA. With the help of failure levels and SLOs, the TP SLA Monitor can determine the risk in the transaction. We discuss this risk assessment approach in Section V.

IV. Trust-based Approach to Determine Pre-Interaction QoS Assessment for Decision Making

We proposed in Section III that the TP SLA Monitor should be equipped with reputation assessment and risk assessment capabilities. In this section, we suggest an efficient reputation assessment methodology that can be used by the TP SLA Monitor to assess the reputation of a service provider based on the opinions provided by the RU. The proposed reputation assessment methodology is based on our initial work on trust and reputation assessment in service-oriented computing [8]. We apply this work to reputation assessment in the cloud.

A. Reputation assessment methodology

For QoS assessment in the pre-interaction time phase using reputation assessment, there are two fundamental issues: trust representation and source of trust values. Formally, these two issues can be expressed as:

1) Trust can be represented as semantics, pictorially or numerically. Chang et al. [9] introduced a mechanism to convert these representations on a scale of [0,5]. We use this scale in our paper to represent trust.
2) Trust values should be stored in an e-Commerce, e-Business or cloud environment. Chang et al. [9] proposed

an information repository (trust and reputation database) for this purpose. They proposed that each agent in an e-Commerce or e-Business environment should have its own information repository where it stores the trust values derived from its past experiences with other agents. RU in our model possess such an information repository to share their experiences with the TP SLA Monitor.

A reputation assessment request from a consumer is processed in our framework as depicted in Fig. 3. We describe the process in following steps:

1. The TP SLA Monitor sends a reputation query about a service provider's reputation to a network of RU.
2. The TP SLA monitor has previously solicited these RU in the past and stores their credibility values in its information repository. Credibility here is defined as the trustworthiness of the opinion of a RU. It is represented on a scale of [0, 5].
3. Only those RU reply to the reputation query who have interacted with the service provider in the same context in which consumer wants to interact with the service provider.
4. The reputation query reply from each RU contains the trust value assigned to the service provider on the basis of its past interaction with that service provider and the time in which this interaction has occurred [10]. When a RU shares this trust value with TP SLA Monitor, this value is called Recommendation Opinions.
5. Time factor is important since a RU can have multiple interactions with the same service provider in the same context but in different time slots. The time factor can be represented by a time delay which indicates the time elapsed since the last interaction of a RU with the service provider and we represent time delay on a scale of [0,5].
6. For each reputation query response received from a RU, the TP SLA Monitor needs to consolidate credibility value, time delay value and the Recommendation Opinion. The outcome of this consolidation is the *reputation* of the service provider provided by a RU.
7. After consolidating these factors for each RU, the TP SLA Monitor then needs to aggregate all reputations by all RU involved in the reputation query. It will give the final reputation value for a service provider.

From the above discussion, we identified the factors that affect the reputation of a service provider, namely recommendation opinion of RU, credibility of their opinion and passage of time after the last interaction of a RU with a service provider (time delay), we need a methodology to: (1) quantify each of these factors individually and, (2) combine these factors to produce a final reputation value for a service provider. We use a fuzzy logic approach to achieve these objectives as given in the next sub-section.

B. Fuzzy logic-based approach

From the above discussion, it is clear that recommendation opinion (RO), credibility (CR) and time delay (TD) should be combined in order to arrive at the correct reputation value of the service provider. Since the values of recommendation opinion, credibility and time delay, as expressed by human beings, are approximations and cannot be expressed by exact or certain values, we need a mathematical model that can represent these approximations and can compute reputation on the basis of approximate values.

By equipping the TP SLA monitor with approximation logic, we achieve a human-like intelligent system. This can be achieved using fuzzy logic. Therefore, in our fuzzy system, the three input fuzzy variables are: recommendation opinion (RO), credibility (CR) and time delay (TD), and output is the reputation contribution R_i of recommending user *i.* We need a fuzzy inference method to map input variables to the output value. Fuzzy linguistic control rules can be used for this mapping. For the fuzzy reasoning, we use the Takagi-Sugeno approach (T-S method) as suggested by Chang et al. [9] for the abovementioned problem. An example of input variables, linguistic control rules, fuzzy reasoning method and output value is given in Fig. 4.

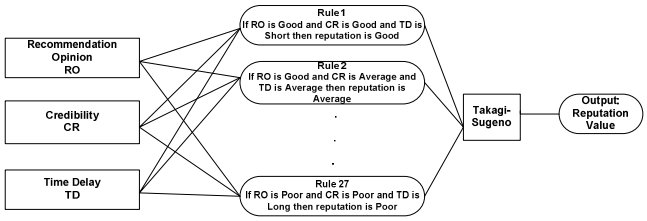

Fig. 4. T-S based Fuzzy Inference System

Using the T-S-based fuzzy inference system has advantages over the popular Mamdani approach in our problem. One of the main advantages is the use of a tuning algorithm with the Takagi-Sugeno approach to achieve best generalization capability of the fuzzy inference system. Therefore, we developed a fuzzy inference system using the Adaptive-Neuro Fuzzy Inference System (ANFIS) algorithm based on the fuzzy model proposed by Chang et al. This approach is called a 'soft computing' based approach.

Fuzzy sets that we use for variables RO and CR are [Poor, Average, Good] and fuzzy sets for TD variable are [Short, Average, Long]. All fuzzy variables are expressed on a scale of 0 to 5. As an example, we show an input variable recommendation opinion with its fuzzy sets in Fig. 5.

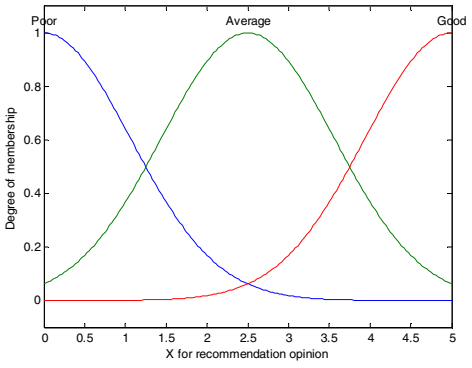

Fig. 5. Membership functions for Recommendation Opinion (adopted from [9]).

From the trained system, the TP SLA monitor can obtain R_i, reputation value given by one recommending user i. Reputation R_k of the service provider K is the aggregation of n recommendations (reputation values) of all recommending users. It is given by:

$$R_K = \sum_{l=1}^{n} R_l / n \quad (1)$$

To illustrate the working details of our proposed framework, let us hypothesize the following scenario. Consider a consumer 'A' who wants to select a video conferencing service and Voice over IP (VoIP) service for his business needs. There are two service providers, namely service provider 'B' and service provider 'C' that closely match the business and operational requirements of consumer 'A'. In order to do the pre-screening of potential service provider candidates, consumer 'A' requests reputation assessments from the TP SLA monitor. On receiving the request, the TP SLA solicits recommendations from RU for service requesters 'B' and 'C'. Suppose RU1 and RU2 send their recommendation about service requester 'B' and RU1 and RU3 send their recommendations about service requester 'C'. Using the soft computing based approach introduced earlier, the TP SLA monitor computes the final reputation values of service requesters 'B' and 'C. The process of selection is depicted in Fig. 6 below:

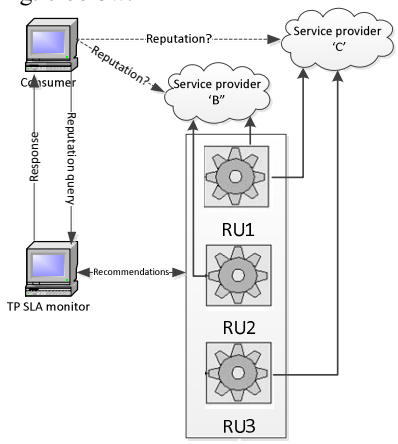

Fig. 6. Service selection process

Based on RO, CR and TD values, the TP SLA monitor computes the reputation value provided by one RU. For example, using our reputation assessment methodology, the reputation values provided by RU1 and RU2 about service provider 'B' are 3.93 and 4.11 respectively. The reputation values provided by RU1 and RU3 for service provider 'C' are 4.12 and 3.56 respectively. Using equation (1), the TP SLA monitor computes the final reputation value of service provider 'B' as 4.02 and the final reputation value of service provider 'C' as 3.84. The reputation value obtained is the indication of level of risk in the interaction. On the basis of high reputation value, consumer 'A' may want to enter into an interaction with the service provider 'B'. This is called the post-interaction time phase and QoS assessment on this phase is given in the next section.

V. Transactional Risk-based Approach to Determine Post-Interaction QoS Assessment for Decision Making

Although a high reputation value in the pre-interaction time phase indicates low risk in the interaction, it does not eliminate the risk completely. We use a risk assessment methodology to assess the run-time performance of the service provider in the post-interaction time phase. We are interested in *transactional risk* which depends upon *performance risk* and *financial risk*. Performance risk represents the level of failure in an interaction due to the non-occurrence of desired outcomes; whereas, financial risk represents the financial consequences that could be experienced as a result of the failure of the interaction.

If a service provider fails to deliver a service according to SLOs, these events are captured by Probability of Failure (PoF). Performance risk is then determined by PoF of not achieving the SLOs against the criteria defined in the SLAs.

The consequence of PoF is the impact on the consumer's investment in a business transaction. Consequence of Failure (CoF) determines the effect of the PoF on the consumer's expectations.

A. Transactional risk assessment approach

In order to determine transactional risk on the basis of performance risk and financial risk, we use mathematical operator, 'convolution' [11]. Convolution is the integral operator which expresses the amount of overlap and impact of one function as it shifts over the other. Convolution has been used in different applications such as power generation systems, to determine the expected demand not supplied by determining the effect of load demand on the generation capacity of the generation system. Mathematically, convolution of X and Y, both of which are independent random variables, is given by:

$$Z = X \oplus Y \quad (2)$$

The principals of convolution can be utilized in transactional risk assessment to accurately calculate the effect of the PoF on the consumer's expectations in terms of financial gain over a given period of time.

B. Determining cost due to service level degradation

In our framework, represented in Fig. 2, the TP SLA monitor can access the SLA parameters that set the threshold for the service provider's performance. In order to calculate the PoF, the TP SLA monitor obtains the run-time values of the SLA parameters and compares these values with the threshold. The process of calculating the PoF is based on the probability mass function (pmf) of the SLA parameter, such as bandwidth, in a given time phase. The pmf obtained in this way represents the performance of the service provider in the given time phase. Based on the threshold given in the SLA for that parameter, another pmf is created from the performance pmf. This pmf represents the level of deviation from minimum threshold requirements. Since the TP SLA Monitor has to use the PoF in convolution, it needs to convert PoF and CoF on a uniform scale. Therefore, the TP SLA Monitor

converts pmf representing failure levels into percentage scale by normalizing failure distributions. Finally, it obtains the PoF which it can use in the convolution operation.

In the presence of PoF, the expected outcome may not be achieved which may lead to the consumer experiencing a financial loss. In order to determine that, the first step in this process is to determine the expected financial gain over a period of time. The TP SLA Monitor divides the post-interaction time phase into time slots of equal intervals [9]. The consumer expects to achieve financial gain in each time slot if the service delivered is as promised by the service provider. From such expected financial gain in each time slot over a time period, the TP SLA Monitor plots the Expected Financial Gain Curve (EFGC). From this curve, it determines the collective expected financial gain that was expected to be achieved throughout the time period. Finally, the TP SLA Monitor needs to convert the cumulative EFGC into a percentage scale for the sake of uniformity to use with the PoF in the convolution.

Convolution is the sliding of the EFGC over the projections of failure levels. Convolution results in a curve which is called Actual Resource Investment Curve (ARIC). As opposed to the EFGC, the ARIC indicates the probability of an amount required to be kept at stake in order for the consumer to achieve the same outcome from the service provider. This cost is determined using dependable criteria. However, the impact of the financial loss given by curve ARIC needs to be combined with costs that can be incurred due to non-dependable criteria. In the next sub-section, we use the convolution method to determine the financial loss due to non-dependable criteria.

C. Determining cost due to non-dependable criteria

As a result of service degradability, the TP SLA Monitor needs to determine the impact of additional factors in order to ascertain financial risk. These additional factors are outside the control of a service provider, they are called non-dependable criteria. In the business domain, the consumer may need to invest extra resources to avoid the financial loss due to service degradability. To determine the impact of this extra investment on the Actual Resource Investment Curve (ARIC) obtained previously, the TP SLA Monitor uses convolution process using the equation (2). But for convolution process, as it has been discussed, both variables should be on same scale. This can be obtained by randomizing both random variables a percentage scale. After the convolution operation, the TP SLA Monitor obtains Total Resource Invested Curve (TRIC) which indicates the total resources that must be invested by the consumer in order to avoid financial loss. Finally, the total financial loss curve can be obtained by the TP SLA Monitor. Interested readers are referred to our previous work on this transactional risk assessment in cloud computing [11].

VI. DISCUSSION

The result of our experiment using the proposed real-time QoS framework indicates that the TP SLA Manager can make effective use of both trust assessment and risk assessment methodologies to conduct a run-time evaluation of service provider's performance. It can be seen that both of these methodologies complement each other by first selecting services on the basis of trust values and then evaluating risk in the interaction and, in case of high transactional risk, referring again to a trust evaluation of an alternate service for migration purposes. The continuous run-time evaluation of service provider performance enables the SLA assurance manager to make an informed decision to continue a service or to migrate to another trustworthy service provider.

VII. SUMMARY

In this paper, we addressed service level management issue by introducing a Third-Party Service Level Agreement monitor (TP SLA Monitor) which consists of a trust assessment module and a risk assessment module. We addressed the issue of real-time QoS evaluation of a service provider's with the help of these modules. Introducing a third-party service provider to do the real-time QoS assessment by combining trust and risk assessment techniques is a novel approach that has not been used before for cloud computing. From the outset, our framework addresses the specific requirements of evaluating the trustworthiness of service providers. However, it generalizes the requirement of automatic process execution on the cloud to offer enterprises agility. The actual implementation of our framework remains a future work.

REFERENCES

[1] Buyya R., S.P., Christian Vecchiola, *Market-Oriented Cloud Computing and the Cloudbus Toolkit.* CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE, 2010.

[2] Huhns, M.N., *Service-oriented computing: Key concepts and principles.* IEEE internet computing, 2005. **9**(1): p. 75.

[3] Dillon, T., Cloud Computing: Issues and Challenges 2010 24th IEEE International Conference on Advanced Information Networking and Applications. 2010. 27.

[4] Armbrust, M., et al., *A view of cloud computing.* Commun. ACM, 2010. **53**(4): p. 50-58.

[5] Burton S. Kaliski, J. and W. Pauley, Toward risk assessment as a service in cloud environments, in Proceedings of the 2nd USENIX conference on Hot topics in cloud computing. 2010, USENIX Association: Boston, MA. p. 13-13.

[6] Dan, A., et al., *Web services on demand: WSLA-driven automated management.* IBM Systems Journal, 2004. **43**(1): p. 136-158.

[7] Hussain, O.K., A methodology to quantify failure for risk-based decision support system in digital business ecosystems. Data & knowledge engineering, 2007. **63**(3): p. 597.

[8] Hammadi A.M., Dillon T.S., Chang E., *Soft Computing Approach to Trust and Reputation Systems.* Expert Systems with Applications, 2011 (under review).

[9] Chang E., Hussain F., and Dillon T., Trust and Reputation for Service-Oriented Environments: Technologies For Building Business Intelligence And Consumer Confidence. 2005: John Wiley \& Sons.

[10] Schmidt, S., et al., Building a Fuzzy Trust Network in Unsupervised Multi-agent Environments, in On the Move to Meaningful Internet Systems 2005: OTM Workshops. 2005. p. 816-825.

[11] Hammadi A.M., Hussain O. Transactional Risk Assessment-based Approach for Service Degradability Management. in IEEE International Conferenece on e-Business Engineering. 2011. Beijing, China.