Chapter 1

Introduction

In the present day scenario, the ability to exchange information seamlessly between business units, customers, and partners is vital for success; yet most organizations employ a variety of disparate applications that store and exchange data in dissimilar ways and therefore cannot talk to one another productively. Web services have evolved as a practical, cost-effective solution for uniting information distributed among critical applications over operating systems, platforms, and language barriers that were previously impassable. Web services use XML and related technologies to enable the seamless interoperability of Web-based applications. Web Services go beyond the functionality of simple Web pages; they provide dynamic application-to-application functionality that can be remotely invoked.

1.1 A Brief Review of Web services

Web services are software components that communicate using pervasive, standards-based Web technologies including HTTP and XML-based messaging. They are designed to be accessed by other applications. They vary in complexity from simple operations, such as checking a banking account balance online, to complex processes running CRM (customer relationship management) or enterprise resource planning (ERP) systems. Web services are hardware, programming language, and operating system independent. This means that applications written in different programming languages and running on different platforms can seamlessly exchange data over intranets or the Internet using Web services.

1.1.1 Definition

According to the World Wide Web Consortium, W3C, a Web service is a "*software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artefacts. A Web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols*".

1.1.2 Potential and problems

The service-oriented Web provides an attractive potential for tomorrow's interactions spanning a wide range of domains from e-economy to e-government. The viability of the claim has already been justified in several research prototypes of providing e-government services. (e.g., WebDG [1, 2], WebSenior [3], ARGOS [4]). Similarly, B2B integration through service composition allows services from different providers to be combined into a value-added composite service [5].

However the potentials of Web services are often challenged by the lack of perfect information about the quality of the services they provide. This is due to the fact that Web services are autonomous (i.e., provided by independent service providers), highly volatile (i.e., low reliability), and a priori unknown (i.e., new or no prior history) [5, 7, 8]. As a plethora of Web services are expected to compete in offering similar functionalities on the new service Web a key requirement is then to provide mechanisms for the quality access and retrieval of services. Web services may make promises about the provided service and its associated quality but may fail partially or fully to deliver on the promises bringing down the quality of the whole enterprise. So there is always a question to which extent the clients may believe the Web services to provide the required functionality, prior to their accesses. A Web service may fail partially or fully to deliver on these promises causing huge risks on the part of the clients. Thus, the challenge lies in providing a framework for enabling the

selection and composition of services based on some security mechanism. The usual security mechanisms, such as authentication and access control, cannot handle the problem of selection and composition. These mechanisms stop at the borders of verifying credentials and checking identities but cannot foretell how well services will behave and perform. Recently, focus is heading towards the social approaches that are based on trust and reputation to augment the usual security mechanisms [6]. Trust and reputation are not new concepts. They have been there in the field of Psychology, Social Science, and Economics etc. These concepts have been proliferated into computer science research for the design of security systems for interacting partners in the open system such as Web services.

1.1.3 Architecture and research issues

Web services can be deployed in different styles. The issues of trust and reputation for Web services across these deployment architectural styles are not the same. The figure 1.1 describes the general principle of deployment and discovery of a Web service.

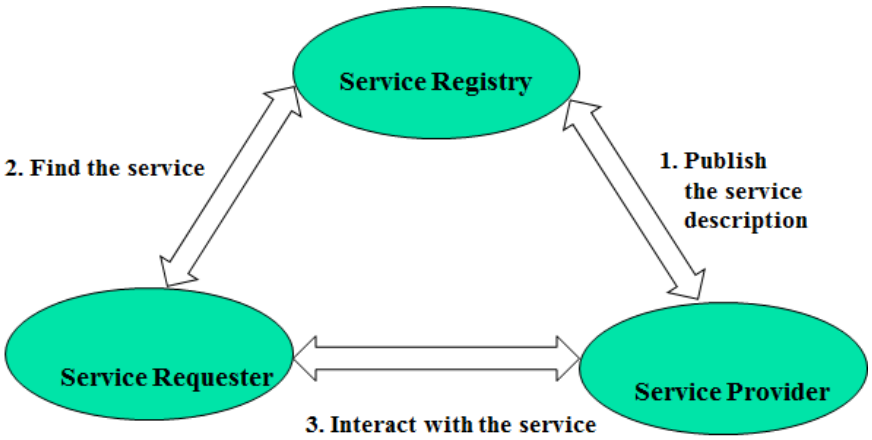

Figure 1.1: Web Service deployment and discovery architecture.

The idea behind the normal mechanism for deployment and discovery of Web services is for service providers to create Web services and to define an interface for invoking them. The service provider, after generating service descriptions

for those services, makes its services available to the world by publishing the corresponding service descriptions in a service registry. A service requester queries the service registry to find a service. While querying, the information included with the service description is used. The service registry answers to the requester with a service description that indicates where to locate the service and how to invoke it. The service requester can then bind to the service provider by invoking the service.

There are three common architectural styles for deployment and discovery of Web services [10, 11, 12]. They are single, composite, and communities.

Single Web service architecture

In single Web service deployment and discovery architectural style, Web services work in a standalone manner to accept and fulfil users' requests. The functionalities provided by single Web services are limited. For example, Airline Reservation service, Hotel Reservation, and Car Rental service.

Composite Web service architecture

A user’s desired complex functional and/or non-functional requirement cannot be met by a single Web service with limited capability in its entirety, but could be possibly met by appropriately integrating and composing a set of available services [11]. For example, whenever a customer makes a request for booking a ticket for a planned journey, a Flight booking Web service may accept the request containing the journey dates, origin and destination of, type of tickets (one way or round trip), and number of passengers to a Flight Booking Web service and after processing these information, it may return name of the airline, flight timings, and prices etc. to the customer. To synthesise these information, the Flight booking Web service must make a series of request to other Web services. Practically, it would request the name of the airline, ticket prices, and timing on the specified route from the Airline Reservation service; hotel accommodation availability status and room tariff form the Hotel Reservation service; options and charges of cars the Car Rental Web service.

There are two possible composition architectures – mediator based composition and choreography-based composition (Figure 1.2). In the former approach, a new composite Web service is generated. It acts as a mediator among the participating Web services. User directly communicates with the composite Web service which, in turn interacts with the participating services transparently. Therefore, each and every message exchange between the user and the participating services is channelled through the mediator. In the latter approach, the message exchange channels or links are established between the participating services themselves. Thus, the user communicates directly with the respective services.

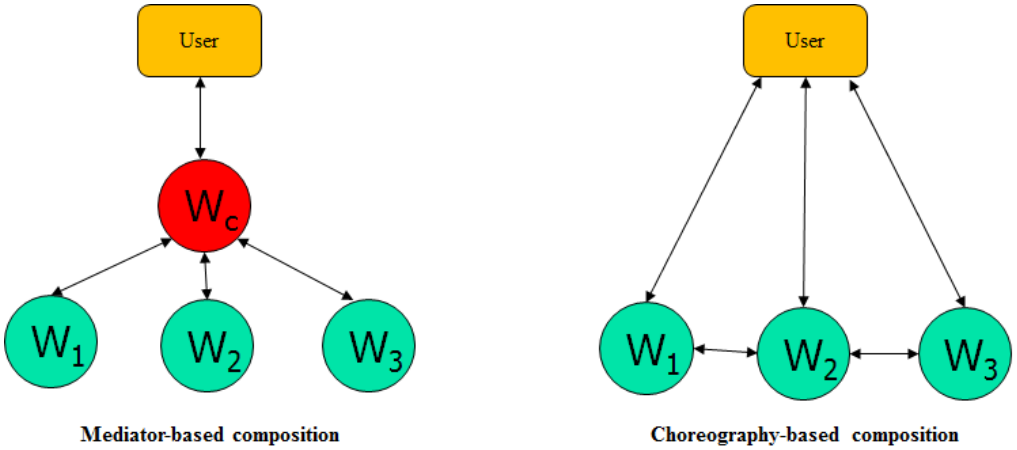

Figure 1.2: Two different Web service composition architectures

Community Web service architecture

Many Web services have been developed by independent providers. The functionalities these Web services offer are sufficiently homogeneous enough to allow for market competition, based on various factors like reliability, efficiency, financial charges etc., to happen. To ease and improve the process of Web services discovery, grouping the services offering the same functionality into communities is common [12]. Such an architecture is to facilitate the discovery of Web services by improving their visibility and to increase their

overall performance with the cooperation inside communities. In community architecture (figure 1.3) a master Web service always leads a community. The master Web service can be any one from the list of Web services in the community or a dedicated Web service that is independently developed (e.g., application designer) from other Web services that are advertised in the registry. The Web service that leads a community never participates in any composition. It is loaded only with mechanisms related to community management like Web services attraction and retention.

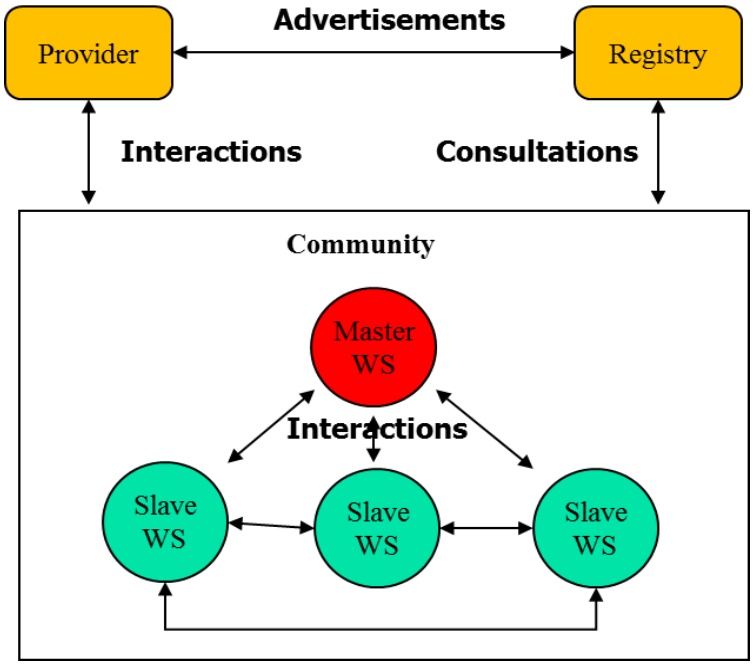

Figure 1.3: Community Web service architecture.

Research Issues

The issues regarding the design of trust and reputation systems for Web services have been identified in [9]. Some of these issues are common across the three architectures while some are specific to a particular architecture.

Issues for single Web service architecture

A Community Web service constitutes composite or single Web services. A composite Web service is made from other single Web services. Therefore,

fundamental to all architectures is the single Web service. The main objective of trust and reputation models for single Web service architecture is mainly to help the users in selecting the appropriate Web service that best achieves their requests. The issues in the design of trust and reputation models for single Web service are still prevalent in the composite and community Web services. The major issues for a comprehensive trust and reputation model in single Web service architecture are:

- Quality of service (QoS) parameters used for computation of trust and reputation measures and consideration of a service user's preference of one QoS parameter over another QoS parameter.
- Interpretation of the trust and reputation measures. Interpretation can be based on subjective judgement or objective criteria. The former semantic is prone to unfair rating while the later one offers high robustness.
- Consideration of dynamism in Web service behaviour as the performance of a Web service is subject to change over the time (improve or deteriorate).
- Assignment of initial trust and reputation value to a newcomer Web service.
- Credibility of the information sources used to build the trust and reputation model to avoid collusion and deception problems.
- Dependency between the recommendation given to a certain service and the credibility of the majority of ratings.

Keeping in these issues in mind, the major research issues in the design of trust and reputation models for single Web service are summarized in Table 1.1.

Issues for composite Web service architecture

Composite architecture encompasses the process integration and organization of a set of services to provide a certain set of functional and/ or non-functional requirements that cannot be accomplished by a single Web service. For the

composite architecture, the identified major challenges are determining the contribution of each component in the composition process and the problem of task allocation among components. Table 1.2 highlights the major research issues in the design of trust and reputation models in composite architecture.

Table 1.1: Major research issues for Single Web service architecture.

Sl. No.	Research Issues
1	Selection of QoS parameters based on which the evaluation of the trust and reputation of the Web services are to be performed.
2	The methods of evaluation of these QoS parameters.
3	Assignment of initial trust and reputation values to newcomer Web services
4	Adaption of the trust and reputation values to the dynamic behaviour of the Web services.
5	Safeguarding the trust and reputation values from collusion and deception problems

Table 1.2: Major research issues for Composite Web service architecture.

Sl. No.	Research Issues
1	All issues from Table 1.1.
2	Evaluation of the performance of participating Web services in the overall composite Web service.
3	Assessment of the trust of the participating Web services when performance cannot be fully observed.
4	Managing the collaboration and allocation of tasks among participating Web services.
5	Evaluation of the effect of malicious constituents on the reputation and performance of the composite Web service.

Issues for community Web service architecture

In the community-based architecture, several Web services offering the same functionality are grouped into clusters to ease their discovery process and to increase the overall performance.

In the community-based architecture, joining communities and the influence of that joining on the performance and reputation of the community have been the key challenges. Dealing with trust and reputation becomes more ramified and more issues to consider, in addition to those of the single and composite. In Table 1.3, we summarize the major research issues for community-based architecture.

Table 1.3: Major research issues for Community Web service architecture.

Sl. No.	Research Issues
1	All issues from Table 1.1. and Table 1.2
2	Evaluation of the reputation of a community in such a dynamic environment where member Web service continuously leaves or joins the community.
3	Cooperation among the community members and its effect on the reputations of the individual members and the reputation of the whole community.
4	Methods and basis for selection of Web services as parts of the community.
5	Influence of malicious Web services on the reputation and performance of the community.

.

1.2 Motivation

Predicting trust values is a key element of modelling and managing trust. It is of critical importance when the interaction is to be conducted at a future point in time [14]. Prediction used in the study of trust and reputation are of two classes. In the first class the prediction is focused on "existence of trust" among entities. In other words, it focuses on determining the existence of trust between two entities in a social setup. The other class of the trust prediction work is on the prediction of "trust values" either qualitatively and quantitatively in a future time spot. The basic goal of prediction or forecasting is to generate a model of the process under observation, and then to use the model to predict values that have not yet been measured. These models can be a global model or a local model. In a global model, the relationship between the input and the output values is described by a single analytical function over the whole input domain. On the other hand, local modelling only creates a specific model that describes the systems behaviour for a given input. The input for which the prediction has to be performed is only known at the prediction time. Again prediction can be one-step-ahead prediction or k-step-ahead prediction. The concept of predicting or forecasting values is not new. It has been used in different areas of applications such as electric power forecasting, stock market forecasting, weather forecasting protein structure prediction etc. The methodologies are also very well developed – Markov Model [13], Artificial Neural network [15], Holt-Winter forecasting method [16], Kalman filter method [17] and Bayesian networks [18] are some approaches worth mentioning

Dragoni proposed in [19] a rationale-based classification scheme for the trust-based Web services selection approaches resulting in three classes:

1. Direct experience-based approaches, in which consumers use the past experience of direct interactions with a service to form the trust for that service or the provider of the service. In these approaches, trust is computed as a rating of the level of performance of the interacting party. The party's

performance is assessed over multiple interactions checking how good and consistent it is at doing what it says it will.

2. Trusted Third-Party (TTP) approaches, in which a trusted third party provides an assessment of a service or the provider in place of the consumer. Based on this assessment, the consumer builds the trust for the service or the provider. The third party could be a central authority or a distributed community comprising of several "individual members". The rationale behind TTP approaches is that consumers must trust the third party they decide to consult and the final decision is based on the assessments provided by the TTP.
3. Hybrid approaches combine the techniques from the Direct experience-based approaches and Trusted Third-Party (TTP) approaches to build integrated frameworks. The rationale behind these approaches is that by combining the two aforementioned methodologies the resulting integrated framework improves some weaknesses of the constituent methodologies, and thus the overall assessment of online services.

The present research explored the effectiveness of machine learning tools, like clustering, Gaussian process and nonlinear time series principle in the prediction of direct trust or recommended trust for a future time point. The study also examined the use of Gaussian process in handling of reputation feedback value from the sources with different noise distribution. Common assumption is that feedbacks are contaminated by a noise drawn from the same distribution [111]. Such assumption may not be valid always in the real life as each source is independent from the other.

1.3 Contribution

The research work proposed a trust evaluation method based on a hybrid approach of combining the direct experience and third party experience. We proposed a behaviour modelling framework for evaluation of trustworthiness of a service provider based on the theory of Markovian process and the principle

of non-linear time series prediction. Markovian process models facilitate learning the dynamic and state-based behaviour of a Web service. When the question of finding the trust/reputation at a future time comes, the Markov process will select the next probable behaviour pattern of Web service provider and a local predictor is used to predict the future trust/reputation value. Our local predictor is a Gaussian process (GP). A GP regression technique has been applied for the prediction of trust value at a future time point. Applications of the proposed methodologies on synthetic as well real life datasets showed better efficiency over existing techniques.

Kernel of a GP represents the prior belief about the input data. Each kernel represents a structural pattern in the input data. So selection of kernel which will give the best predictive mean is a challenging task. We proposed a multiple GP models each with different kernels working in an ensemble. The model selection approach based on the likelihood is used to select the best ensemble member(s) for the prediction task at hand. For reducing the computation cost, an adaptive "Sleep and Recovery" mechanism is used to make the weaker model wait till the input data is found to be suitable for its prediction. The experimental verification showed the superiority of the models based on this ensemble framework.

In Web services environment, reputation-based trust systems helps to determine trustworthiness of a service provider in future interactions. In these systems, trust is quantified as some predicted probability values. However, the prediction variance is generally ignored. The prediction variance reflects the error (precision) that is introduced by a reputation source in its feedback. Therefore, the prediction variance can be utilized to assess the accuracy of a reputation prediction made by a Web service user. A reputation aggregation model based on heteroscedastic GP is implemented. The model is capable to handle changes of reputation over time. Accepting the prediction variance in the aggregation of feedbacks from different sources with different error distributions has also been proposed.

1.4 Thesis Structure

The rest of the thesis is organized as follows.

In Chapter 2, we give an overview of the concept of trust and reputation as they are used in different areas. The properties of the trust which need to be incorporated in any trust model are presented. Second part of the chapter gives a comprehensive report on the trust and reputation models available in the literature. First the general models are discussed followed by those model meant for web services.

Chapter 3 reports our proposed models of trust based on Markov chain, Hidden Markov model and time series prediction using Gaussian process regression (GPR). We present the background on Markov chain, Hidden Markov model, time series prediction and Gaussian process first. Then the formulation of our proposed models are discussed in details. Finally the model is evaluated empirically using synthetic and real life datasets.

Chapter 4 is an exposition on our next models. The kernel of a GP model represents the prior belief about the function to be leaned. Different kernels can represent different structures in the data. We proposed an ensemble of GP models for the prediction of trust based on the theory of addition and multiplication of kernels. The ensemble member is selected by using an adaptive approach to reduce the computational cost. Using this ensemble framework, the models of chapter 3 are extended to more efficient models. The extended models are experimentally verified and compared with some existing models.

Chapter 5 is on our third model. While collecting the feedbacks, the variance of estimation is an important information that can filter the malicious feedback from the final trust prediction. Many models have been reported in the literature

to do this filtering. Such systems do not give a good result if the noise present in the feedbacks from different sources does not follow a single probability distribution. We proposed a model based on heteroscedastic Gaussian process for handling such types of noise in the feedback. We experimentally showed that our system is capable of handling the noise properly.

Chapter 6 is the concluding part of the thesis. We make some concluding remarks and also highlight future direction of research.