uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Bo Zhang

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (Systems Science)

GRADE / DEGREE

Systems Science

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Architectures for Online Reputation Systems

TITRE DE LA THÈSE / TITLE OF THESIS

Dr. Morad Benyoucef

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Dr. Bijan Raahemi

Dr. David Wright

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Architectures for Online Reputation Systems

Bo Zhang

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the MSc degree in System Science

School of Information Technology and Engineering
Faculty of Engineering
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-41688-4
Our file Notre référence
ISBN: 978-0-494-41688-4

Abstract

E-commerce provides a platform where complete strangers from all over the world have the chance to transact with each other. Reputation is a key factor in the success of such transactions, therefore reputation systems are in widespread use to help participants build their online reputation. This thesis analyzes current reputation systems and studies how to design architectures for online reputation systems. Based on the shortcomings found in current reputation systems, this thesis proposes and implements a reputation architecture called ORAS (online reputation aggregation system) Plus which provides a portable reputation mechanism for Web users. With ORAS Plus, users can search others' global reputation and obtain their own reputation cards which can be displayed at any website. ORAS Plus can help users know someone's online behaviors all over the Internet. More importantly, it allows users to take their existing reputation from one website to other websites where they are unknown to others.

Acknowledgements

I would like to thank my thesis supervisor, Professor Morad Benyoucef at the Telfer School of Management, for the help in my study and life.

I also wish to thank my team members, Zhuosong Duan and Hui Li, who gave me a lot of help in my research.

At last, I want to say thank you to my parents and my girlfriend, Yuanyuan Lu, who have been supportive to me all the time.

Publications

Benyoucef , M., Duan, Z., and Zhang, B. (2007), "Portable Online Reputation Systems", in *Proceedings of the Networking and Electronic Commerce Research Conference (NAEC 2007)*, Riva Del Garda, Italy, October 18-21, 2007 (to appear)

Table of the contents

List of Tables

List of Figures

Chapter 1 Introduction

Introduction

The explosive growth of E-commerce known as the New Economy has come to life in the last decade. E-commerce provides a platform which has risk because transactions commonly take place between participants who never transacted with each other before and participants often have insufficient information about the products and the service providers. These force the participants to accept the risk of prior performance, such as paying the money before receiving goods. Although the participants do not have the opportunity to see or try the goods, they may have the confidence as long as they trust the sellers. A trustworthy seller therefore has a significant advantage in the platform of E-commerce which consists of many unknown and uncertain factors [11].

Although plenty of new opportunities are created by E-commerce, many uncertain factors involved in E-commerce constrain its development. The uncertainty of the sellers' reputation is the main reason that many potential buyers do not want to participant in E-business. Trust is considered as the essential factor that determines E-commerce's success or failure and the ability to build users' online reputation is an important factor of successful E-commerce. One of the major challenges for E-commerce is how to establish the trust relationship between participants. Establishing trust in E-commerce is hard, because participants cannot judge others by physical documents used in traditional society and most of the participants involved in E-commerce have not interacted with each other before.

In order to solve these problems, reputation systems are introduced as a main technology to reduce the uncertain factors appeared in E-commerce and play a very important role in online transactions. "A reputation system is specifically developed to collect, analyze and

produce users' reputation which later can be used to determine their level of trustworthiness before E-commerce business transactions can commence [2]." In the Internet world, people can only judge others by the online reputation values they have. A reputation system is an effective way not only to facilitate the trust in E-commerce but also to promote the growth of online shopping.

Reputation is a collectable and measurable term in E-commerce, and it can be calculated based on the ratings given by different participants and communities. People judge an individual's reputation with two major means: the direct experience with this individual and the ratings about her/him from other people. The direct experience is usually put in the first place as the first hand evidence to know someone's reputation. In common situation, the direct experience with someone is hard to get, so the ratings or comments from the people who have direct experience with this individual are desirable and important. More and more reputation systems appear nowadays, especially in E-commerce, and they are trying to build their own reputation mechanisms that help the strangers trust each other and reduce the risk that participants will afford in online transactions.

1.1 Motivation

Reputation systems seek to address the development of trust by recording the reputation information of different participants. However, current reputation systems have some shortcomings. A major one is that since most existing reputation systems are restricted to specific or individual websites, relevant information about a participant may come from different websites and from the transactions that were not mediated by any authority [24]. Participants cannot use their existing good reputation built on one website throughout the Internet. Instead, they have to rebuild their online reputation when they register to a new website or community. By far, we do not see current reputation systems cooperating with each other to allow participants to take their online reputation from one website to another.

For instance, someone with a good reputation on one website may become an untrustworthy stranger on other websites, although these websites provide similar services. Participants in E-commerce need a mechanism to help them make their existing reputation portable.

Another shortcoming is that participants sometimes can only obtain someone's reputation on one website without knowing that the person may have bad reputation records on other websites. By far, there is not any website that could search someone's reputation all over the Internet. Therefore, a searching engine that can get all the reputation information about a target participant will be very helpful. And also, the different rating mechanisms appeared in different websites may make participants confused and difficult to judge others' reputation. An aggregation system that can make different types of reputation into one standard should be introduced to help participants to judge someone's reputation easily and quickly. Based on the discussion before, we focus on designing a reputation architecture which can solve these major shortcomings of current reputation systems.

1.2 Research Objectives

The first research objective is to study current reputation systems and find out how to design reputation architectures. Meanwhile, making a classification for reviewed reputation architectures and finding out the deterministic factors of building a portable reputation system.

The second research objective is to propose a reputation architecture which provides portable reputation mechanism. In order to solve the problems we discussed before, the proposed architecture should have the ability to collect participants' online reputation from different websites, compute participants' global reputation values and make participants' reputation portable.

The third objective is to implement a proof of concept prototype of the proposed architecture.

The remainder of the thesis is organized as follow. Section 2 introduces the background acknowledge related to E-commerce and reputation systems. Section 3 reviews related work on reputation systems, makes a classification for reviewed reputation systems and analyses the important factors in building a successful portable reputation system. Section 4 introduces the methodologies which are used in the design and the implementation. The architecture of the online portable reputation system (ORAS Plus) is proposed in section 5. Section 6 specifies the details of the implementation. Finally, the contribution and future work are concluded in section 7.

Chapter 2

Background

2.1 Trust

Trust is hard to define because it manifests itself in many different forms. The term of trust is being used with a variety of meanings in the literature.

Gambetta (1988) [14] proposed the following definition of trust containing the concept of dependence and interpreted trust as Reliability Trust.

Definition 1 Trust is the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends.

The following definition is inspired by McKnight & Chervany (1996) [13].

Definition 2 Trust is the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible.

This definition implicitly includes some aspects of trust such as dependence, reliability and utility which means positive outcome comes from positive utility. It also mentions risk as a part of trust and trusting party will accept the situational risk.

Similar to Definition 1 and 2, most of the definitions concentrate on the action or behavioral aspects of trust while some cover the context-dependent nature of trust [34, 35]. What is the

meaning of trust in e-commerce? Chang et al. [29] gave a definition of trust that can be used in e-commerce.

Definition 3: Trust is defined as the belief 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 time slot.

Because it is difficult to capture the notion of trust, trust is thought as a non-computational concept by some economists. According to Williamson (1993) [12], the notion of trust should be avoided when modeling economic interactions due to the lack of measurement and the computation model for trust should be done within the domains of sociology and psychology.

According to [11], there are two fundamental differences between traditional and online environment regarding how trust and reputation are. Firstly, the traditional evidences of trust and reputation used to observe and depend on in the physical world are missing in the online environment. Secondly, in the physical world trust information is hard to be shared and usually constrained to local or specific communities. The Internet can efficiently spread trust information on a global scale.

Why do we need trust in e-commerce? Transactions have begun to move away from traditional face-to-face business to being more on the Internet. More and more individuals and companies move their business to websites and benefit from e-commerce. The infrastructure for e-commerce could be client-server network, peer-to-peer (P2P) network, or mobile network. No matter what kind of network environment it uses, the participants are anonymous in most cases. This may cause different kinds of dishonest behaviors. The sellers may only provide part of the service promised or deliver the poor quality products to buyers.

Some buyers may receive the products without paying the money. Trust and trust technology have come to solve these problems that could affect the development of e-commerce. It helps participants build online trust and boosts their confidence about e-commerce. On the other hand, consumers will choose the one with good online reputation, and this will force companies to improve their online reputation by providing better on-demand services. From this aspect, "trust is a crucial ingredient in any mutual relationship and where transactions are carried out in an anonymous, pseudo-anonymous or non-anonymous distributed environment to provide the agreed to Quality of Service [29]." The study has shown that the use of basic trust and reputation technologies generate substantial interest among online users as well as facilitate success in business volumes [32].

2.2 Trustworthiness and Reputation

Trustworthiness is defined as a measure of the level of trust that the trusting agent has in the trust agent [29], so trustworthiness can be seen as the trust with scale or level.

The concept of reputation is close to trustworthiness. According to the Concise Oxford dictionary, the basic meaning of reputation is: Reputation is what is generally said or believed about a person's or thing's character or standing. In the literature, reputation is defined from different aspects. Mui et al. [36] defined reputation as the 'perception that an agent creates through past actions about its intentions and norms'. Sabater J. and Sierra C. [37] defined reputation as an 'Opinion or view of one about something'. Abdul-Rahman A. and Hailes S. [38] defined reputation as 'an expectation about an Agent's behavior based on information about the past behavior'.

Chang et al. [29] classified reputation as the reputation of the quality of product and the reputation of the quality of service. They gave the advanced definition of reputation in e-commerce.

Definition: In service-oriented environments, reputation is an aggregation of the recommendations from all of the third-party recommendation agents and their first-, second- and third-hand opinions as well as the trustworthiness of the recommendation agent in giving correct recommendations to the trusting agent about the quality of the product or service [29].

In E-commerce, reputation is introduced as a computable term. It can be centralized, computed by a trusted third party, like a Business Bureau or e-auction website; or it can be decentralized, computed independently by each peer after asking other peers for recommendations [28].

Reputation can relate to a group or to an individual. A group's or a community's reputation can be modeled as the average of all its members' individual reputations and the more trustable the group is, the more reputable all its members are. Tadelis' study shows that an individual belonging to a given group will inherit a priori reputation based on that group's reputation [15].

2.3 Reputation systems

It is very important for the participants, both buyers and sellers, to estimate each other's reputation before transactions. "Not only do buyers need to trust sellers, but also sellers need to trust buyers [24]." Buyers need to trust the sellers sufficiently with the expected delivery time, the quality of the products and other services promised by sellers. At the same time, sellers must trust the prospective buyers and believe that the buyer is seriously considering the purchase and will not cheat in the transaction. The mutual trust is based on the sufficient reputation information which is enough to judge each other's trustworthiness. In other words, in order to estimate someone's trustworthiness, participants must try to collect enough

evidence related to their potential partners' reputation. Reputation systems are the platforms that provide reputation information and support such estimations.

The lack of experiences, knowledge or information about other persons may lead us to refrain from the interactions. Reputation systems can provide us with relevant experiences that others have had with someone [18] and help people establish mutual trust. Reputation systems have the ability to collect and aggregate the feedbacks from participants' past online behaviors, and then publish them as reputations to help participants trust each other. Feedbacks or ratings are based on a participant's past transactions and help other participants learn about the transaction partners they are dealing with. The more reputation value a participant has, the more trustworthy others will feel about this participant. The participants will decide who to trust after viewing others' reputation which informs their ability and reliability. Furthermore, reputation systems also encourage participants to be more trustworthy and discourage deceiving behaviors [6].

An effective reputation system should have the ability to help users locate trustworthy participants and exchange resources securely with confidence. According to Resnick and Zeckhauser (2000) [16], a reputation system must meet three challenges. It must provide information that allows buyers to distinguish between trustworthy and non-trustworthy sellers, encourages sellers to be trustworthy and discourages the participation from those who are not.

Trust and reputation systems are used in successful commercial online applications. The aggregated ratings about a given participant generate a reputation score which can help other participants to decide whether or not to transact with that participant in the future. According to [11], reputation systems can be called collaborative sanctioning systems to reflect their collaborative nature. Therefore, reputation systems have a positive effect on

market quality. Some researches have indicated that reputation systems can encourage market actors to participate in transactions [19].

Although it is still hard to be achieved, a reputation system should be impossible or difficult for an individual to change identity or pseudonym to erase the connection to his/her past behavior. According to Resnick et al. [16], reputation systems must have the following three properties to operate at all:

1. Entities must be long lived, so that with every interaction there is always an expectation of future interactions.
2. Ratings about current interactions are captured and distributed.
3. Ratings about past interactions must guide decisions about current interactions.

The foundation of a reputation system is the transactions between participants. As it is impossible for a participant to transact with all the other participants to know their reputation information, it is essential that reputation systems are introduced to develop a global reputation for each participant. Feedback is a critical factor for the operation of a reputation system. There are three types of reputation systems classified by different kinds of feedback: positive reputation systems, negative reputation systems and hybrid reputation systems. In order to encourage honest participants, the positive reputation systems reward good behaviors. In contrast the negative reputation systems will punish the malicious behaviors such as cheating. The participants in both types of the reputation systems start with a neutral reputation, such as zero. Points are either added or deducted depending on the behaviors of participants. In the negative reputation systems, points are taken away as punishment, while in the positive reputation systems points are added only as a reward for good behaviors. In the hybrid reputation systems, both kinds of behavior (positive and negative) are considered as point values. The point distribution of a hybrid reputation system produces broader gradients between honest and dishonest behaviors, making it easier to distinguish between

participants with good, bad, and neutral reputation [42]. Nowadays the hybrid reputation systems are used popularly on e-auction websites, but on some websites users can still choose to switch to positive or negative reputation systems from hybrid reputation systems.

It is a fact that trust is hard to build and easy to destroy. Sometimes people care more about someone's negative ratings, although most of the reputation systems running now are hybrid systems (e.g. eBay). Also, all reputation systems have a shortcoming that new participants cannot make deals with others because they have little reputation information to make them trustworthy. Because of this shortcoming, reputation systems would be ineffective for promoting cooperative participants in some cases.

There are two types of trust management methods: the top-down type (e.g. one with a trusted third party) and the bottom-up type (e.g. one where participants share reputation information). They are also the two major ways to built reputation architectures. The top-down management method controls all the trust information from the top of the system as a third party. In the bottom-up management method, participants circulate and share reputation information among themselves to promote cooperative behavior.

The reputation systems are popularly be classified by two categories: the centralized reputation systems and the distributed reputation systems. They are two mayor technical principals for building reputation systems.

Most current reputation systems are based on centralized architecture. "In centralized reputation systems, information about the performance of a given participant is collected as ratings from other members in the community who have had direct experience with that participant [11]." A centralized reputation system needs a third party to collect feedback and publish them as reputation information, so it adopts the top-down management manner. The

centralized reputation system itself does not promote someone's reputation to other participants, it just provide a platform for users to publish the transaction feedbacks that can be seen by anyone in the system. A user need to decide whether to trust a particular participant based on the ratings collected by the third party. The reputable participants are likely to result in more outcomes than the disreputable ones in transactions, so the centralized reputation systems encourage participants to be reputable and honest.

According to [11], the centralized reputation systems have two fundamental aspects. One is the centralized communication protocol that allows participants to provide ratings about their transaction partners, as well as to obtain potential transaction partners' reputation from the centrally controlled system. The other one is a reputation computation engine which is used to calculate each participant's reputation score based on collected ratings and other information related to reputation. The centralized communication protocol is the core part of every reputation system and the reputation computation engine could help users easily consume the reputation information provided by the central authority.

Most distributed reputation systems are designed for the distributed network, such as P2P networks. Some of the researchers classify the centralized reputation systems and the distributed reputation systems by the network environment. The reputation systems used in a distributed network are usually thought as the distributed reputation systems. G. Swamynathan [26] proposed five main characteristics that a distributed network should have.

1. There is no central coordination and no central database.
2. A transaction must run to completion or it is not counted.
3. Transactions occur between participants and participants have semi-permanent identities.
4. Participants are autonomous and self-interested.
5. Each participant can evaluate the outcome of a transaction independently and subjectively.

A distributed reputation system can be either top-down or bottom-up type. However, in the distributed reputation system, it does not need a third party to appear in the architecture to keep the system working, so the bottom-up management method is more feasible to be used to design a distributed reputation system. There is no central location for participants to submit ratings or obtain other participants' reputation scores in a distributed system. Each participant records the transaction experience with other participants, and shares the information directly with the one who needs this information. Anyone who considers transacting with a target participant must try to obtain the distributed reputation scores or information from as many community members as possible who have had direct experience with that target participant. A distributed communication protocol is usually be used in distributed reputation systems to help participant to submit or obtain ratings from other members in the community. Not every distributed reputation system has a reputation computation engine like the centralized reputation system, participants need to evaluate the reputation information collected from their neighbors before they make the decision about whether to trust a given participant or not.

According to [26], participants in reputation systems play the following roles: reputation holder, reputation seeker, reputation evaluator and reputation storer. The third parties in centralized reputation systems act as the reputation holder, the reputation evaluator and the reputation stoter, and participants play all of the four roles in distributed reputation systems.

There are many kinds of reputation computation engines used in current reputation systems. Most of the reputation computation engines are based on the following theories: Simple Summation or Average of Ratings, Bayesian Systems, Discrete Trust Models, Belief Models, Fuzzy Models (Mamdani method [39] and T–S method [41]) and Flow Models. The Simple

Summation or Average of Ratings method is widely used by current e-business websites such as eBay, Amazon and Yahoo.

2.4 P2P networks

Peer-to-peer (P2P) networks are networks in which peers cooperate to perform a critical function in a decentralized manner [20]. Peer-to-Peer networks represent an environment well suited for distributed reputation systems. Every member considered as a node in the P2P network plays both as a client and a server, and is therefore sometimes called a servant. Users can share their own resources directly without constraining to any specific website. Acorrding to Jøsang et al. (2006) [11], there are two phases in the use of P2P networks. The first is the search phase, which is about to locate the servant where the requested resource resides. Some search phase can rely on centralized functions such as Napster [11] which has a resource directory server. There are also the search phase using distributed functions in some P2P network, for example Gnutella and Freenet [11]. The other one is the download phase. After the requested resource has been located, the download phase transfers the resource from the exporting servant to the requesting servant.

In a P2P system, because there is no centralized organization to mediate exchanges, so the entities, called peers, share their resources or finish the transactions independently. But the intermediate architectures also exist in P2P networks, such as the FastTrack architecture used in KaZaA[1]. There are nodes and supernodes in the FastTrack architecture. The supernodes act as directory servers during the search phase by keeping tracks of other nodes and supernodes that are logged onto the network.

1 www.kazza.com

Instead of focusing on online transactions, most reputation systems for P2P network are proposed to help users to find the reliable servants and high quality resources. Compared with the centralized networks, the P2P networks provide an easy and scalable way to share tremendous amount of resources with a low cost of maintenance [17]. To encourage resource sharing among peers and combat malicious peer behaviors, reputation management is essential for peers to assess the trustworthiness of others and to selectively interact with more reputable ones [21]. Without an efficient reputation management facility, peers will have little incentive to contribute their computing or bandwidth resources [22]. According to [22], there are six key issues that should be considered to design a cost-effective P2P reputation system: high accuracy, fast convergence speed, low overhead, adaptive to peer dynamics, robust to malicious peers and scalability.

Many E-auction companies, such as eBay[2], Amazon[3] and Yahoo[4], provide platforms and reputation systems to facilitate online transactions. The growth of online transactions also occurs in Peer-to-Peer (P2P) environments. Peers can participant asynchronously and perform the transactions point-to-point anonymously. Although P2P networks provide a platform where peers can communicate with each others easily, some problems have appeared and need to be solved urgently. Firstly, some malicious participants may distribute virus programs and attack others' computers. The peers may hesitate to interact with unknown peers due to the concern of receiving corrupted or poisoned files or being exploited by malware [23]. Secondly, some participants may cheat others in transactions, such as receiving moneys without returning items. These cases may happen more frequently in P2P systems than on eBay or in other centralized organizations, because peers are hard to know others' reputation and there is no third party to refrain from peers' bad behaviors. Comparing to transact on e-auction websites, peers will take higher risk in transactions in P2P network.

2 www.ebay.com

3 www.amazon.com

4 www.yahoo.com

Some reputation systems have already been introduced in P2P network to solve these problems. The reputation systems for P2P network will help peers to choose a capable and reliable partner to interact with.

2.5 Current Reputation Systems in Business

eBay is the most popular online auction website that allows buyers to bid for the items posted by the sellers from all over the world. eBay's reputation system is based on its Feedback Forum and the summation or average of ratings method. In the feedback forum, buyers and sellers rate each other as positive (1), negative (-1), or neutral (0) after transactions. They also have the rights to leave positive or negative comments about their transaction partners. The feedback forum is a centralized reputation system, where eBay collects all the ratings from the forum and computes the reputation values for each user. A user's total reputation score is the sum of positive ratings minus the sum of negative ratings and the total amounts of positive, negative and neutral ratings for the user are also provided as important reputation information. eBay's reputation system is successful, but it still has some shortcomings that can be misleading. Firstly, according to Resnick and Zeckhauser's research [16], about 99% of all the ratings on eBay are positive, less than 1% is negative and less than 0.5% is neutral. They also found that there is a high correlation between ratings by buyers and sellers, suggesting that eBay's users are reciprocal and retaliatory. Secondly, a participant with 100 positive and 50 negative ratings should less reputable than the one with 50 positive and no negative ratings, but they have the same reputation scores on eBay. Thirdly, although participants can only rate each other after the completion of a transaction, it still has the possibility to boost someone's reputation by fake transactions. Despite its drawbacks, the eBay reputation system is one of eBay's successful factors and has a strong positive impact on its marketplace.

Different from eBay's reputation system, some websites provide production reviews that help users to make better purchase decisions. Amazon is an online bookstore which allows members to write book reviews. The reviews consist of prose text and a rating in the range 1 to 5 stars and the average of all ratings is presented as a book's "reputation". Users, who are not members, can vote on reviews as being helpful or not helpful and the results of the vote are also displayed with the reviews. Based on the total numbers of helpful votes each reviewer has received, Amazon determines each reviewer's rank as Top 1000, Top 500, Top 100, Top 50, Top 10 or #1 Reviewer.

In most developing countries, people do not want to participant in online transactions even when reputation systems are used in most e-auction or E-business websites. The escrow mechanism becomes the only way that people dare to participant in online transactions and is very popular in some countries (e.g. China). The third party appeared in escrow is responsible for mediating transactions as a custodian and a temporary bank. Sellers will not get the money until buyers receive the items and give a feedback. The money will be kept temporarily at the third party. Escrow is an effective mechanism, because it can remove most possibility of deceiving behaviors. Even though escrow is effective in B2B (business to business) and C2C (customers to customers) online transactions, there are three drawbacks when people use it. The first one is the high operation cost by third parties. In order to attract users, the e-auction websites need to spend a lot of money to establish their own escrow mechanisms or corporate with the companies that provide escrow services. The second drawback is its complex procedure, which impairs the convenience of the use of the Internet. As third parties will get the feedback from buyers until they receive the items, sellers need to wait a long time to get the money and finish the transactions. The Last one is its low availability and portability, which constrains the areas available to make transactions. Nowadays escrow has a place in the e-market, only because the reputation systems are not

good enough in some situations but it will be replaced by the mature reputation systems in the future.

Chapter 3

Literature Review on Reputation Systems

3.1 Current Reputation Systems in the Literature

Many researchers are working on reputation systems. Some of their proposed solutions have issued the features of portable reputation systems.

3.1.1 Portable Reputation with EgoSphere

EgoSphere [1] is such a system that reputations built on one website can be used elsewhere. In order to collect reputation data from other websites, it suggests a robot style design called EgoSphere Webproxy that is similar to the web crawler robots used by search engines to index the web. The webproxy running on users' PCs at client side monitors their communications with the EgoSphere-supported sites. Whenever users request a webpage from one of those sites, the webproxy will fetch the ratings, recommendations or any other reputation evidence in HTML files and send them to EgoSphere Reputation Database. Because one webproxy cannot collect complete information about a specific user, the EgoSphere reputation database will collate the evidence gathered from many webproxies into a unified view of the user's reputations at different sites. Offering users the portable reputation service, the EgoSphere Reputation Exchange module calculates the transferred reputation value with a single linear regression algorithm. The module works as a bridge between different websites. It computes the correlation coefficient of two sties and how much reputation should transfer from one to the other. For instance, if two sites provide similar services and the correlation coefficient is close to 1, most of the user's reputation built at one site can be transferred to the other and vice versa.

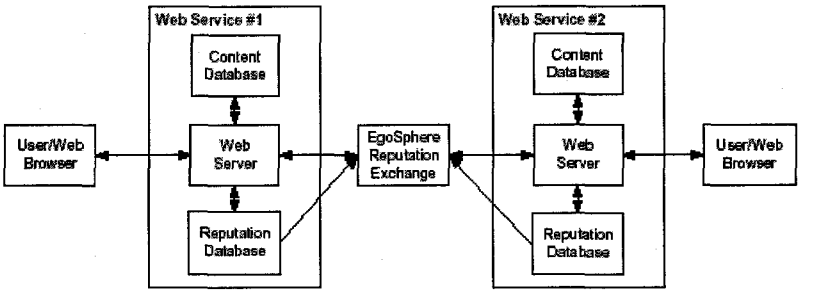

Figure 3.1 Basic Idea of EgoSphere [1]

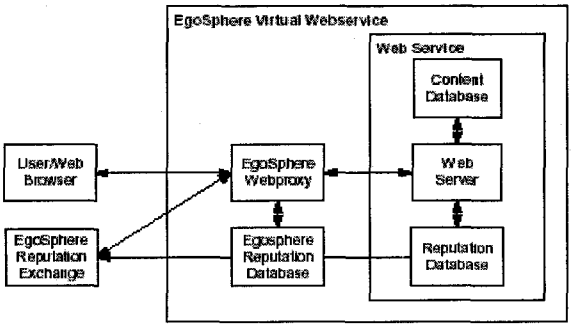

Figure 3.2 EgoSphere Web Services [1]

Figure 3.2 shows EgoSphere uses a web proxy on the client-side to annotate HTML responses from the web severs with EgoSphere information, and to gather reputation system.

The webproxy, the reputation database and the reputation exchange module consist of EgoSphere's centralized architecture that provides portable reputation service for users. However, this system has two major drawbacks. First, the webproxy struggles to gather a user's complete reputation information, as it's not practical to monitor every single request from the user. Second, if a user wants to transfer his/her reputations from multiple websites to one, the exchange module can only deal with the reputation data from two websites at a time, which highly increases the processing workload of the system.

3.1.2 A Security Architecture for Reputation Systems

Ismail et al. [2] proposed a security architecture for reputation system which is built with a collection and certificate authority module. Before the rater and the ratee use this system, they need Token Issue Module to finish registration. The rater submits the feedback about a ratee based on transactions they made to the collection and certification authority module, where reputation value with the feedback will be calculated and a certificate for the ratee will be issued. Before the ratee shows his/her certificate to the partner in a transaction, the certificate will be sent to the relying party module to be validated which helps users make a decision whether to proceed in a transaction with him/her or not. This certificate mechanism presents the portability of feedback in this centralized system. However, the system has a noticeable shortcoming. Despite their own functions, those modules have no connections between each other and users have to send requests to only one of the modules at a time, which makes the process a little cumbersome. We can simplify the architecture composing a unified module by combining the token issuer module, the relying party module, and the collection and certificate authority module, so that users can register and validate their certificates in only one process before transactions.

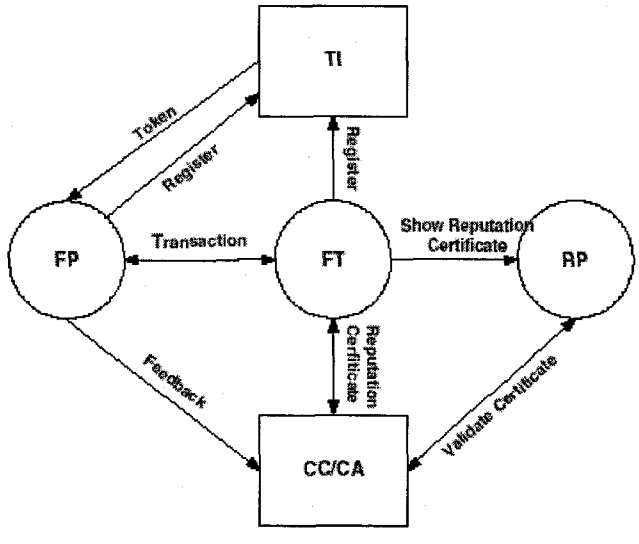

Figure 3.3 Abstract View of Security [2]

FT	A feedback target is the entity who is being evaluated and gained the reputation rating based on the feedback given by a feedback provider.
FP	A feedback provider is the entity who provides a feedback about FT based on transactions made between FP and FT.
RP	A relying party is the entity who uses the reputation rating produced to make a decision whether to proceed in a transaction with FT or not.
TI	The token issuer manages the registration of FP and FT, records transactions made between FP and FT, signs the token to produce a legitimate token.
CC/CA	The collection centre/certificate authority collects the valid feedbacks and uses them to calculate reputation rating and then issue the certificates for FT.

Table 3.1 Entities and symbols used in figure 3 [2]

Table 3.1 shows the entities in Figure 3.3. Figure 3.3 depicts the abstract view of the security architecture for reputation systems.

3.1.3 SMART: A Small World Based Reputation System for MANETs

SMART [3] is a reputation system that aggregates and distributes current peers' reputation in a MANET network. The request agent that is responsible for gahtering data is the core part in the architecture. For the sake of evaluating trust, it sends the query of collecting peers' recommendations to the network, or receives the rating for a previously transaction from the network. The system cannot only collect information itself but also receive the information submitted by peers. All the collected information about the peers' reputation is stored in the data collection agent. When peers are about to send the reputation data before transactions, a module (Path Manager) will find one or multiple paths to the destination and get the trust ratings of every peer in the path from the network trust repository, the source of reputation information that can be updated by Trust Manager and Feedback Agent. The trust manager is used to scan new interior information in the data collection agent and the feedback agent will monitor exterior information in the trust repository. Each part in the system is connected by Contacts Manager and Path Manager, and the former is in charge of the inner connection, while the latter is responsible for maintaining connection with exterior.

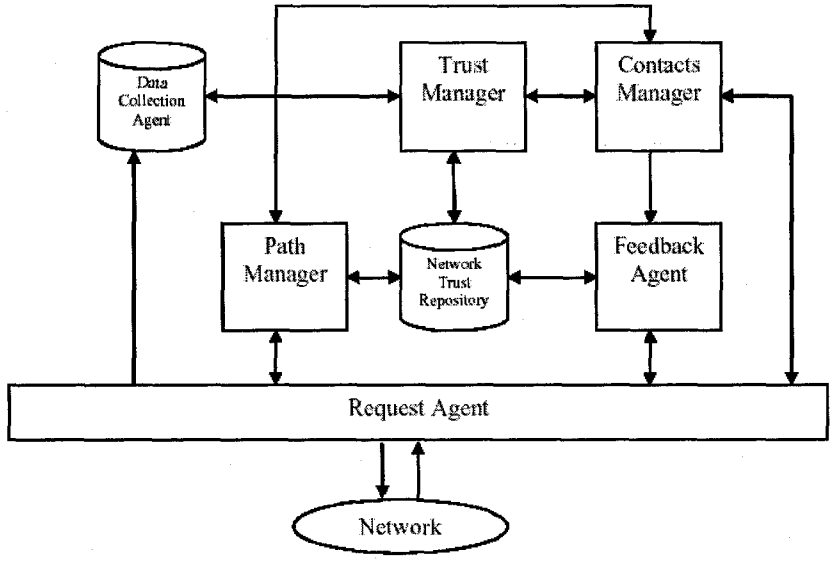

Figure 3.4 System Architecture of SMART [3]

This is a well designed architecture that the interior and exterior tasks are performed by different modules. The collected data and the temporary data from the network are stored in different databases. The trust manager and the feedback agent deal with interior and exterior information separately. This kind of design can keep the system running fleetly and efficiently.

3.1.4 A Reputation-Based Approach to Preserving Privacy in Web Services

Rezgui et al. [4] proposed a reputation system that aims at automating the process of privacy enforcement in the online environment. The architecture has three main components: Reputation Manager, Probing Agents, and Service Wrappers. When a user submits a request to the reputation manager asking for another one's reputation value, it will collect, compute reputation information, store the obtained value in the reputation repository, and send the result back to the user. In order to avoid frequently collecting the same reputation

information of one peer and leading to a huge traffic at the reputation manager, the probing agents are responsible of permanently monitoring the services that provide reputation information and submitting the new collected information to the reputation manager. These agents built in the reputation manager are anonymous to Web Services being monitored. They conduct the original idea of the tracking module in our architecture. The probing agent is an efficient way to get the latest reputation information. A service wrapper is a piece of software improving the security issues and handling all messages received from users. Before sending a message, a user submits the message to the wrapper stalled on his/her computer. Then the wrapper sends a request to the reputation manager to inquire about the receiver's reputation. Based on the response, the wrapper may forward or cancel sending the message. The service wrapper will prevent the user from communicating with the disreputable users and provide a security environment.

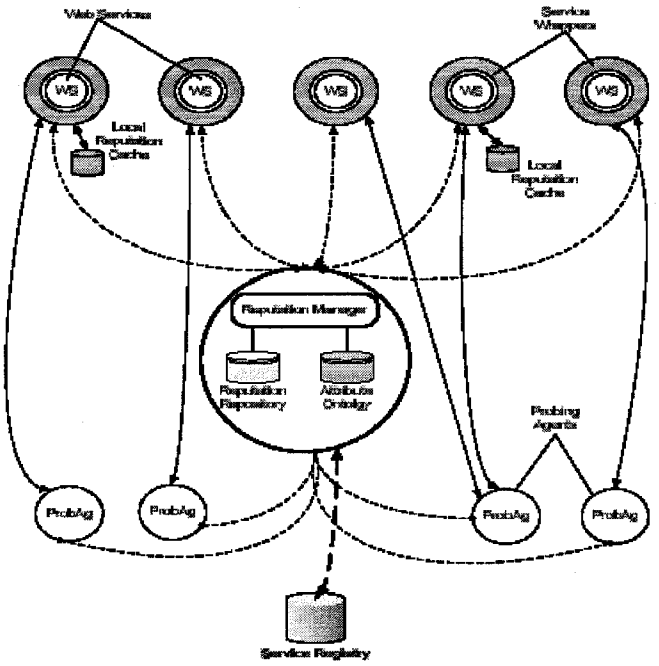

Figure 3.5 Architecture of Reputation Management System [4]

3.1.5 PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities

Li et al. proposed [5] the PeerTrust which is used to evaluate and submit reputation information in P2P network. It is built in a distributed manner that does not need a third party to appear in the architecture. Every peer has a trust manager, a small database and a data locator in their PCs to cope with reputation information. The trust manager will firstly collect a target peer's reputation information from the P2P network, and then compute the trust value. If the peer wants to recommend this target peer, it is also responsible for submitting feedback through the data locator. The data locator is used to find the information destinations both in the collection and submission process. There is no central database in the architecture, and the data is stored in a distributed manner. Peers store reputation information in their PCs and share their own or collected information with the help of the data locator.

In PeerTrust, the trust evaluation process is executed in a decentralized manner at each peer. Different from centralized reputation systems, it does not need a central third party to compute and distribute each peer's trust value. This allows the system sufficiently leverages each peer's resource and ability. A peer obtains another peer's trust information from the rest of the peers in P2P network and computes the trust value by his/her Trust Manager. This well designed distributed reputation system still has some drawbacks. Firstly, the distributed information storage method will need peers to save their reputation information in their PCs which means once they lose the information by accident, it will be hard to get them back comparing to centralized reputation systems. The other problem is the second- or third-hand reputation information is usually be collected in P2P network, so the credibility of the information is hard to be evaluated.

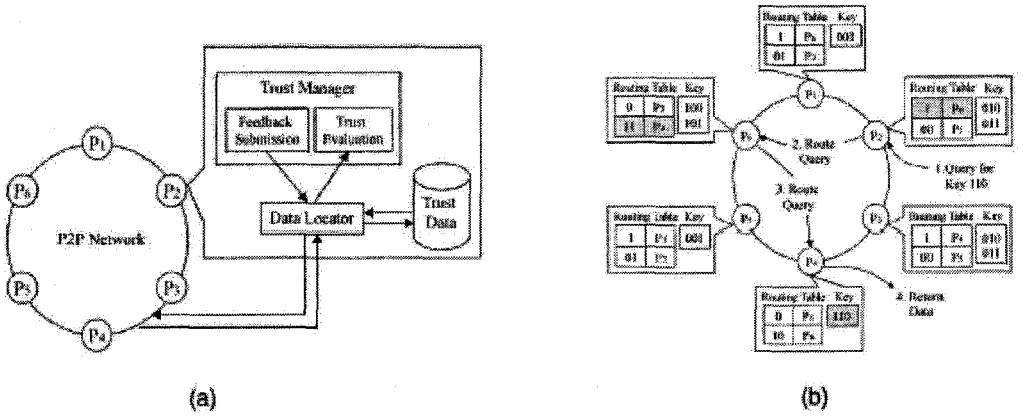

Figure 3.6 PeerTrust system architecture (a)System architecture (b)Data location[5]

3.1.6 An Efficient Distributed Reputation Scheme for P2P Systems

Liau et al. [6] proposed a distributed reputation system for P2P network. This system is based on a reputation certificate called RCert which contains one peer's rating information collected from previous transactions with other peers. A peer first uses resource discovery mechanism to find the service or product provider. All the peers that have the resources or productions will send reply messages with their reputation certificates to this peer. The peer will make decision after evaluating all the reputation certificates, and then send an acknowledgement with his/her digital signature to the chosen provider. After this, a TimeStamp will be sent from the provider to the peer. The TimeStamp consists of provider's digital signature and the time value on provider's PC. After verifying the signature and the time by using the public key of provider, the peer will start transaction with the provider. Once they finish the transaction, this peer will update the provider's reputation certificate by adding the ratings and the TimeStamp, and then a new certificate will be sent back to the provider to be presented in next transaction.

The TimeStamp is a new idea and very important in this system. First, it records raters' and ratees' digital signature which will avoid malicious ratings. Second, it contains the time values of each transaction which can force peers to use their latest certificates. Since some

peers got the bad ratings in past transactions, they probably want to hide this bad reputation information by using the older copies of reputation certificates. To solve this problem, the authors proposed the RCertPX protocol to keep tracking the latest timestamp in peers' reputation certificates because the RCert protocol cannot prevent the peers from using older copies of reputation certificates. The TimeStamp can also be used to verify the validity of the reputation certificate. It does not need to ask for the privacy key of the holder to verify the certificate. This can be done by contacting the rater who provides the last TimeStamp in the reputation certificate. If the rater returns a TimeStamp with the same time value appeared in the certificat, the certificate is valid. If the last rater is not available, the peer can also try to contact the preceding raters until there is an available one.

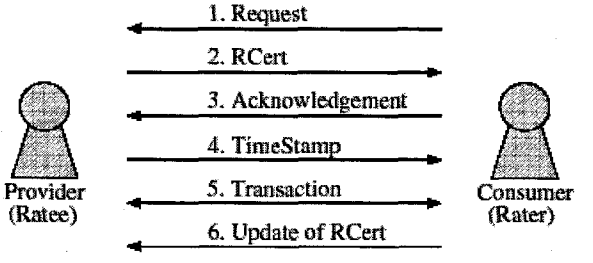

Figure 3.7 RCertP Protocol [6]

In addition, the RCertPX system is an extendibility of the RCertP system. Its architecture is shown as below:

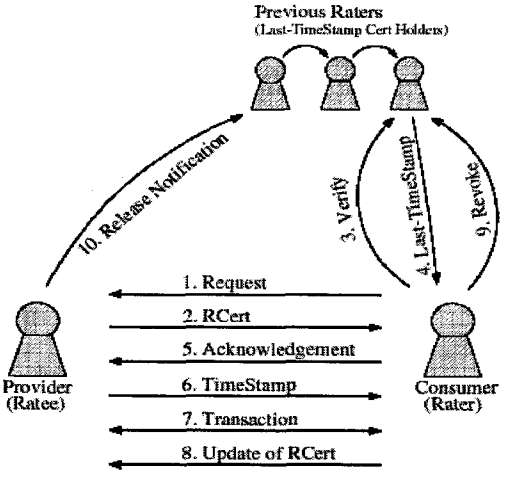

Figure 3.8 RCertPX Protocol [6]

3.1.7 Architecture and Algorithms for a Distributed Reputation System

The recommendation system proposed by Kinateder and Rothermel [7] help users to collect the recommendations of the productions they are interested in. A user that needs the reputation information about a target production will use the recommendation system to formulate a query for recommendations about this production. The query will be sent to the system and then responses matching the query will be received by the system. The system is also able to condense the data, accumulate the recommendations and finally present the results to users. The main advantage of this system is it can collect the all kinds of recommendations like binary ratings, percentage values, textual reviews or any combination of the former ones, and then organized these recommendations by types in the report. Some factors and methods are considered in the system to ensure the quality of recommendations such as the adding confidence value to the ratings, considering the time of recommendations' creation and using the digital signature to avoid one person to give many recommendations with different pseudonyms. Although it is a powerful collection system, without a integration method to deal with different types of feedback, all collected information are listed in one report and the system do not give a total reputation score for each production. Users need to take a lot of time to read all the reviews to make a purchase decision.

3.1.8 Trusted P2P Transactions with Fuzzy Reputation Aggregation

Different to the recommendation system discussed above, some reputation systems only collect one type of reputation information or only collect the information from one specific community. For instance, the Fuzzy Trust System proposed by Song et al. [8] is based on a distributed architecture for peers' reputation evaluation and dissemination in P2P networks, and it only collects information from eBay to test their system. This system uses Fuzzy logic to solve some uncertain parameters appeared in trust evaluation. The system is divided

into two main portions: one is the reputation aggregation system; the other is the reputation evaluation and dissemination system. The reputation aggregation system will generate peers' local trust score with Fuzzy inference mechanism. Following this, the reputation evaluation and dissemination system will collect local trust score from all peers as one of the weights to calculate one peer's global reputation which can be thought as his/her total reputation value. In the global reputation aggregation process, there are three variables used as weights: peer's reputation (local trust scores), the transaction date, and the transaction amount. Not only can the reputation information be disseminated in this system, the system also provide a platform for consumer product exchange, distributed file sharing, security information sharing and processing cycle sharing. All these processes occur in this centralized architecture. The global reputation calculated by peers' local trust scores and transaction information is one main reason that the system has the specifications of portable reputation systems. By adding a portable reputation mechanism (e.g. reputation card) in the architecture, it will be a successful portable reputation system.

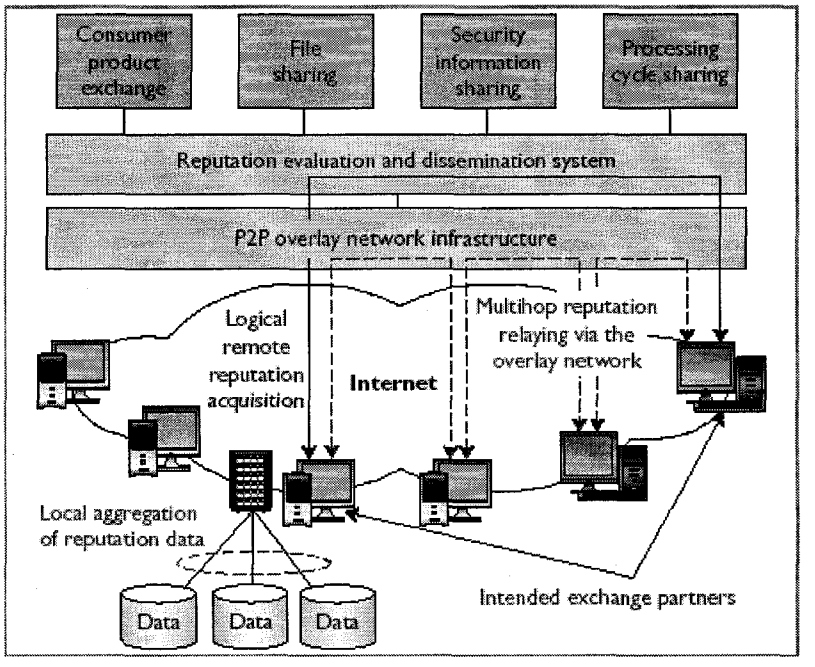

Figure 3.9 A Typical Peer Reputation Aggregation System [8]

3.2 Classification and Discussion

Most of the existing reputation systems are centrally controlled by third parties and prevalently used, whereas distributed reputation systems for P2P networks are relatively new. Some researchers working in this field have proposed some possible solutions that make users benefit from the portability of online reputations in distributed or P2P networks.

3.2.1 Comparison and Classification

A1—A8 are used to present the eight architectures (3.1.1-3.1.8) discussed before. The content of compare is based on the findings and observation of the eight approaches. The following table compares these architectures directly and provides a probable classification for them.

Attribute / Approach	Type	Core Components	Data Storage Method	Feedback	*Characteristic*
A1	Centralized	EgoSphere Web Service	Collected information in central database	Reputation list organized by types	*Webproxy*
A2	Centralized	TI, RP and CC/CA	Members' transaction record and collected information	Reputation Certificate	*Portable reputation certificate*
A3	Distributed	Request Agent and Trust Manager	Stored across the network in a distributed manner	Calculated reputation value	*Contacts Manager and Path Manager*
A4	Centralized	Reputation Manager	Collected information in central database	Calculated reputation value	*Probing Agents and Service Wrappers*
A5	Distributed	Trust Manager	*Stored across the network in a* distributed manner	Calculated reputation value	*Do not need database, Data Locator*
A6	Distributed	RCert.	No database, update information in client-side	Reputation Certificate	*Portable reputation certificate, TimeStamp, all activities occur in client-side*
A7	Distributed	Recommend system	Just collect data and do not save them	Reputation list organized by types	*Organization of different types' reputation*
A8	Integration	Reputation aggregation system, trust score dissemination system	Collected information in central database	Calculated reputation value	*Second hand trust information are considered*
ORAS Plus	*Integration*	*Collection Module Aggregation Module Mapping module*	*Collected information in central database*	*Reputation Card Calculated Value*	*Collect information from different websites Portable reputation card*

Table 3.2 Compare of different approaches

These architectures can be classified by different criteria as follow:

(1) Type:

Centralized: A1 A2 A4

Distributed: A3 A5 A6 A7

Integration: A8, ORAS Plus

(2) Data Storage Method:

Use central database: A1 A2 A4 A8, ORAS Plus

Do not use database: A3 A5 A6 A7

(3) Reorganized Feedback:

Calculated reputation value: A3 A4 A5 A8, ORAS Plus

Reputation Certificate: A2 A6, ORAS Plus

Reputation list organized by types: A1 A7

In distributed environments such as P2P communities, a distributed reputation system easily conducts the portability of reputation. However, centralized and integrated reputation systems can also provide a portable feature. EgoSphere [1], for example, is based on a centralized architecture but provides portable reputation service for users. The system comprises the functions of data gathering, data exchange, data transfer and update. EgoSphere uses a webproxy on the client-side computers to gather reputation information and then send them back to its web server. Users could get the information they want in EgoSphere and make their existing reputations portable.

Despite central databases are often appeared in centralized architectures, a portable reputation system with a distributed architecture can also use central databases to enhance the safety of data storage. Moreover, the types of reorganized feedback are the important factors of portable reputation mechanism. Most reputation systems with the portable

characteristic adopt reputation certificates as reorganized feedback. The certificate is an efficient way to facility the portable reputation mechanism.

As for other architectures in the literature, ORAS Plus implements some of their advantages such as the use of reputation certifications. However, it extends the function of the reputation certification by making it portable. The most novel aspect of ORAS is that it uses a collection module to gather information from different communities. Different from the feedback provided by other systems, users can either obtain a calculated value or the original (raw) reputation data by using the collection and aggregation module. More factors are considered in the aggregation process such as the transaction time and the transaction amount. Furthermore, members can get the reputation cards and build their own profiles on ORAS Plus. ORAS Plus was set up in a flexible way in that it was managed centrally and used dispersedly. Users' reputation information is stored and managed centrally, and their reputation cards will be updated automatically.

The following parts will analyze and compare different types of architectures, reputation collection methods, data storage methods and types of reorganized feedbacks.

3.2.2 Different Types of Architectures

The architecture of reputation systems can be classified in three categories: Centralized, Distributed, and Integrated.

Reputation systems are helpful in fostering strangers' level of trust. However, most current reputation systems are based on completely centralized architectures. They require users to explicitly make and reveal the ratings about others, which are not accepted by many users. Such ratings would often be made strategically and may not reflect someone's trustworthiness [24]. For instance, as mentioned before, eBay's ratings are almost positive

and there are privities between buyers and sellers. Reputation systems are also facing other drawbacks. The first one is concerning the storage of the reputation data. The reputation information is stored in a huge database and hard to be managed by third parties, especially when the website has many transactions going on in any moment. Increasing the number of interacting participants requires more storage and processing capacity from the system, it may become the bottleneck for the system, and this may increase the costs for the service provider. Also, the central party can easily access to all the reputation information, which can sometimes be misused for data mining [25]. Finally, the central party has the possibility to lose the user's information without anyone noticing. The second drawback is high cost. Since the centralized systems need third parties to build a platform for peers, they also need a lot of money and manpower for marketing and operation management. For instance, eBay's annual net revenues were 5.96 billion dollars in 2006, but its total operation expenses were as high as 3.28 billion dollars according to the eBay company's 2006 financial results [10]. The last but not the least, a critical shortcoming of current centralized reputation systems is that they are generally bounded to a specific website; as a result the lack of portability stands out. A user with a good reputation built at Site A is unable to take advantage of that reputation at Site B, even though the two sites offer exactly the same services.

The centralized reputation system also has many benefits. Firstly, compared to the database in a distributed reputation system with users concerning about losing reputation data stored in their own computers, the centralized database is a safer way to keep the reputation information by the reliable third party. Secondly, the main reason that centralized reputation systems are so popular is that they can be powerful enough to well play the role of a trusted third party which is more trustworthy than the individual recommenders. For example, a person is selling an item both on eBay and in a P2P community. Most users would probably do the transaction with him/her on eBay due to its authority as a famous online marketplace.

In a distributed reputation system, a third party doesn't exist in the architecture. There is not a specific component for submitting ratings or obtaining reputation scores of others. Peers can summit ratings anywhere based on their direct transaction experience with other community members in the distributed system. A user who considers transacting with a given target party must find the distributed stores or try to obtain ratings from as many community members as possible who have had direct experience with that target party [5]. The distributed systems will get rid of those shortcomings of centralized reputation systems such as huge databases and high cost. Their major advantage is providing a portable method for users to use their reputations. That makes it possible for users to use the same reputation information in different communities, not only the one that was earned. The scalability will not be a problem for distributed reputation systems because when new participants join the community, the total storage and processing capacity of the network will increase. Another different issue is that it is more difficult to influence the process of trust estimations because there is no central authority performing any calculations and each participant makes their own trustworthiness judgments themselves [25].

However, most distributed reputation systems are designed for P2P networks where the scope of online transactions is very limited, and people therefore barely participate in transactions. Another problem is the security of the reputation data. Since users have to store and manage their own reputation information, it will be very hard to get the information back if it is lost. The last problem is regarding identity. In some cases it may be difficult to identify the data controllers in distributed systems. Different peers can easily use the same reputation value to do transactions because there is no password or any other authentication method to identify the users in P2P networks.

The integrated reputation systems are specially designed combining the features of centralized and distributed architectures. An integrated architecture is usually controlled by

a third party like centralized reputation systems, but it also offers users the distributed environment to consume its service. Although the reputation information is collected and managed by the third party, users can share it in a distributed manner. For instance, Song et al. [8] proposed the Fuzzy Trust System based on a distributed architecture for peers' reputation evaluation and dissemination in P2P networks, whereas the system still has a central database to save and manage the information. For users, it can be thought as a distributed system even though it is integrated with the features of centralized architectures.

3.2.3 Reputation Collection Methods

On the basis of the previous researches, it is not difficult to find that a portable reputation system should have a powerful collection module in addition to a portable reputation mechanism. Some reputation system cannot provide portable reputation system because they only collect information from limited resources. A portable reputation system should have the ability to find all the reputation information about users though the Internet. In order to do this, designers need to build a collection module that can connect to all mayor reputation systems or websites and also an aggregation or organization module to process different types of information they got from the internet. According to different reputation collecting methods, the collection module could be classified in three categories:

(1) Collecting information from other reputation systems. The portable reputation provider will connect to the web servers of the reputation sources (e.g. eBay) and collect users' reputation information. This collecting method saves plenty of time comparing to the other two methods below, but sometimes it need the authority of the website before colleting.

(2) Monitoring and gathering information from each user. The portable reputation provider monitors users' reputation-related interactions at different sites and collects

their reputation information. Egosphere's [1] webproxy is a typical example of such method. However, since it is difficult to monitor every activity of a user, the system may not have enough information for judging his/her reputation.

(3) Obtaining information both from the users' ratings and other websites. Users will submit feedback, and the reputation source websites offer their reputation data directly to the portable reputation provider. It seems to be the best way to get all the reputation information, but this scheme requires fully corporation from most of the websites and it may cost the provider a lot of money to do so.

3.2.4 Data Storage Methods

Different architectures have different methods of storing data. The central databases are always applied in centralized architectures, but in some distributed reputation systems users do not need to save data or they could save the data in their own computers. Decentralized data management can be achieved using a DHT-based approach like a P-Grid [27]. In PeerTrust [5], where a P-Grid is used, the trust data about a peer i is stored at designated peers that are located by hashing a unique ID of peer i to a data key. An alternative way to use a structured storage mechanism is to have each peer maintain trust values in its local storage. However, it faces the problem that peers cannot find the all the information about a target peer when someone who has the information is offline. Another common approach is to store reputation data in a certificate [6] which contains the reputation ratings assigned to the service provider and is digitally signed by the rater (using its private key) to ensure data integrity.

One of the popular P2P communities is Kazza which provides a completely distributed peer-to-peer file sharing service. A peer who considers transacting with a target peer must try to obtain ratings from as many community members as possible who have had direct

experience with that target peer. Thus, peers can share their reputation information in Kazaa without going to a specific site to look for the information, and save the reputation information in their computers. In order to help peers to communicate with each other, the supernodes are used in Kazza to act as directory servers during the search phase by keeping tracks of other nodes. Some researchers also proposed some distributed reputation systems that do not use central databases. For instance, Li and Liu [5] proposed a dynamic P2P trust model in which each peer in the P2P network has a trust manager responsible for feedback submission and trust evaluation. There is no central database in the system. Trust data needed to compute the trust measure for peers is stored across the network in a distributed manner. The update and submission processes will be finished at client side, and peers can manage and submit their reputation information themselves. M. Kinateder and K. Rothermel [7] also proposed a recommendation system that does not require to save any data. The system only collects the information and submits it in the reputation lists organized by reputation types.

The methods of storing data usually determine the types of the architecture. If central databases are used in a system, the system is usually designed on a centralized architecture. The types of architecture also affect the methods of storing data. For instance, centralized systems usually need databases while distributed ones may not.

3.2.5 Different Types of Reorganized Feedback

Not only the participants' ratings after transactions are considered as feedback, but also the information provided by reputation systems to show one participant's reputation value is another kind of feedback. Before researchers design the architectures of reputation systems, they should be clear what kind of feedback users can get from the systems as querying others' reputation values. Not all reputation systems need users to build their reputation. Instead, some of them can collect users' existing reputation information and organize it as

different kinds of feedback. Here we call this kind of feedback reorganized feedback. The reorganized feedback provided by reputation systems can be classified in three types: (1) Calculated reputation values, (2) Reputation lists organized by reputation types, and (3) Reputation Certificates.

After collecting, those reputation systems calculate the total reputation values based on the collected information. Most of them will only do the calculation if the reputations are from the same rating mechanisms. For example, the PeerTrust [5] and the FuzzyTrust system [8] collect reputation evidences (e.g. positive or negative ratings, transaction times and transaction amounts) in a P2P community and then put them into their algorithms. Through their systems, users can get any peer's total reputation value in the community.

However, not every reputation system works in the same way. When those systems get different types of reputation, it is hard to set one standard for the different rating mechanisms. A list of the reputations therefore will be a substitute in this case. After collecting different kinds of reputation information, the system can classify them by types and make a list without calculation. M. Kinateder and K. Rothermel [7] proposed a production recommendation system that provides the reputation lists organized by different types of reputation. Various types of rating information could be included in the list, like binary ratings (e.g. we recommend the target, or not), percentage values (e.g. the product has a quality of 90%), multiple attribute-rating pairs, or textual reviews.

The reputation certificates containing users' reputation information from different websites or communities are similar to the calculated reputation values and the reputation lists but users can take them anywhere in online environment in addition to letting others know about their reputations by showing the reputation certificates. More and more reputation systems adopt this novel idea. Liau et al. [6] designed a completely decentralized reputation system

for P2P networks based on such a certificate called RCert, a document residing on a participating P2P node and can be taken anywhere by the holders when considering to transact with each other. It consists of information ratings collected from previous transactions with other peers. Ismail et al. [2] also designed their reputation system in a similar way to benefit users from portable reputation services.

Chapter 4

Methodology

According to Wikipedia[5], Methodology is "the analysis of the principles of methods, rules, and postulates employed by a discipline. It is a collection of theories, concepts or ideas and a comparative study of different approaches."

4.1 Service-Oriented Architecture (SOA)

The design of ORAS Plus is based on the method of Service-Oriented Architecture (SOA). SOA is a design for linking business and computational resources (principally organizations, applications and data) on demand to achieve the desired results for service consumers (which can be end users or other services). OASIS (the Organization for the Advancement of Structured Information Standards) defines SOA as the following:

Definition: A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.

SOA is an architecture that relies on service-orientation as its fundamental design principle. Service-oriented architecture is an architectural style where new or existing functionalities are grouped into atomic services. These services communicate with each other by passing data from one service to another, or by coordinating an activity between one or more services. In an SOA environment independent services can be accessed without knowledge of their underlying platform implementation [43].

5 www.wikipedia.com

A typical Service-oriented Architecture consists of the interaction among service providers, service requesters and service brokers. The interactions in service-oriented architecture are shown in the following figure.

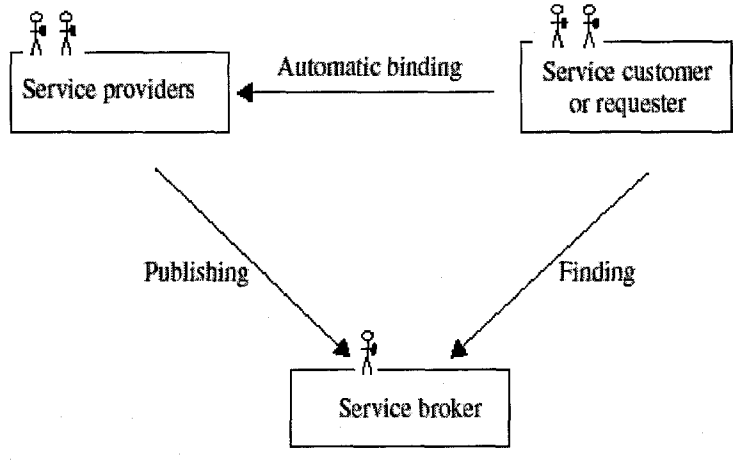

Figure 4.1 Interactions in Service-Oriented Architecture [29]

The service provider creates Web Services and possibly publishes its interface and access information to the service registry. Service providers must make their services deliverable, described and published by registering with the service broker or making them available to service requesters directly [30].

Service requesters may be either machines or humans. They try to discover the desired services by sending the requests to service brokers or the service providers directly. The service requestor or Web Services client locates entries in the broker registry using various find operations and then binds to the service provider in order to invoke one of its Web Services.

Service brokers provide central repositories and directories for both service providers to publish services and service requesters to find resources. Service brokers are responsible for making the Web Services interface and implementation access information available to any potential service requestor.

According to [29], the Service-oriented Environment should have the following eight characteristics:

(1) The multiple channels of sales, marketing, purchasing and information inquiries;

(2) A collaborative approach between sellers, buyers, users and service providers;

(3) The high connectivity and the ability electronic handling of information, data and documents;

(4) The strong information infrastructure that extends beyond the original physical individuals and businesses;

(5) A platform provides products, services, end-user interaction and utilization of information services;

(6) It is self-organization and reconfiguration to meet dynamic business needs;

(7) It could capture business intelligence through trust, reputation and smart information sharing;

(8) It could value-added consumer relationships, customer service and strengthened small-medium businesses.

Many network infrastructures can be used in service-oriented architecture, such as the client-server network, the peer-to-peer network, the grid network and the mobile network. The ORAS Plus is designed to be used in the client-server network, where one computer acts as the server responsible for communicating parties and others act as clients that will consume the service provided by the server.

4.2 SOA Technologies

Web Services could be used to implement a service-oriented architecture. A major focus of Web Services is to make functional building blocks accessible over standard Internet protocols that are independent from platforms and programming languages. These services can be new applications or just wrapped around existing legacy systems to make them network-enabled. A service can rely on another service to achieve its goals.

SOA is built on Web Services standards. Web Services are middleware technology that offers standard communication interfaces that allow ease of communication between heterogeneous applications over the distributed network environment. They provide inter-application operability, inter-organization collaboration and business integration to achieve wide commercial objectives [29].

Service-oriented architecture is often defined as services exposed using the Web Services Protocol Stack. The base level of Web Services standards relevant to SOA includes the following:

XML stands for Extensible Markup Language. It is designed to describe data and to focus on what data is.

HTTP is a request/response protocol between clients and servers used to transfer or convey information.

According to w3schools.com, SOAP (Simple Object Access Protocol) is a simple XML based protocol to let applications exchange information over HTTP. It is a protocol for accessing a Web Service and lets an agent or application invoke another agent or application by using an XML message over the Internet [33].

According to w3schools.com, WSDL (Web Services Description Language) is an XML-based language for describing Web Services and how to access them. Not only service providers use WSDL to describe their Web Services, but also service requesters use it to describe what services they are looking for.

UDDI (Web Services Description Language) provides universal common description of Web Services and allows search, query, discovery or the ability to locate the services and enables run-time or automated dynamic binding or integration between the service providers and service requester [33].

The process that a service requester tries to find a Web Service by using above technologies is shown in Figure 4.2.

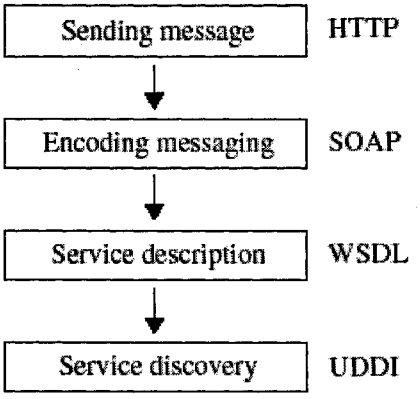

Figure 4.2 Web Service protocol [29]

4.3 Why and How We Use SOA in ORAS Plus

Web applications are used to handle the communication between service providers and consumers in e-business, whereas Web Services can make one automatically invoke the applications running in other businesses by using a service-oriented architecture. In other words, companies can easily link their web applications or services with their partners and customers. Enterprise architects believe that SOA can help businesses respond more quickly

and cost-effectively to changing market conditions [44]. Web Services are not focusing on replace the traditional infrastructures. They are complementary in both traditional interactions and e-business mechanisms. Web Services are one main mechanism which represents a significant advance in the e-business of the future [31].

ORAS Plus has its own Web Services that connects to other websites' Web Services and is accessible to any user. The Web Services provide reputation search function by obtaining data from other Web Services over the Internet. Under the HTTP and SOAP protocols, the data are transferred in XML files.

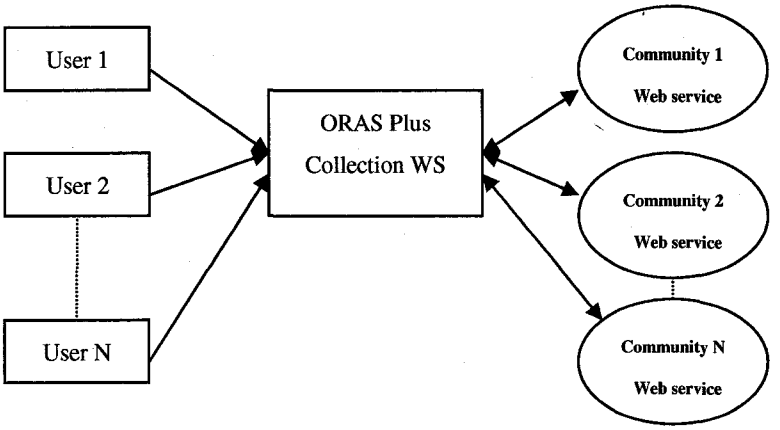

Figure 4.3 ORAS Plus Web Service protocol

Not only the collection module can be published as a Web Service, but the aggregation module which contains the algorithms also could be publish as a Web Service to make more blocks in our system accessible by users and other Web Services. By designing the system under SOA protocol, it make more modules in the system available to users and other Web Services which ensures that every function could be used efficiently and the services created properly represent the business view. At the heart of SOA planning is the process of defining architectures for the use of information in support of the business, and the plan for implementing those architectures.

Chapter 5
Proposed Solution

In this section, we propose a portable reputation system (ORAS Plus) with a centralized approach allowing users to search others' online reputation and take their own existing reputation built in one service to another. To achieve this goal, the system will gather reputation data from different reputation systems by using the collection module and have the ability to automatically organize and update the reputation data. We use a calculation engine [40] to integrate collected information and provide the global reputation for each user. After getting users' global reputation values, we use a reputation card mechanism, RepCard, to make users' reputation portable. Although the system will be controlled centrally, once obtaining their reputation profiles users can use them everywhere. From the user's aspect, this system can be considered distributed.

If ORAS Plus is wanted to be used in business to generate revenue, it will be faced with the legal responsibility under common law. Because user authentication on the Internet is difficult, the accuracy of the collected information is hard to be evaluated which means users may be misled by inaccurate information provided by our system. Most reputation systems face the same legal problems, such as eBay which warns users that it does not control the information provided by other users, some of which may be inaccurate or deceptive. An online reputation system provider may be able to reduce its exposure to liability using a contractual exclusion clause [45]. The ORAS User Agreement should be used in case it is deployed for general use by the public. Users must agree to assume the risks that others may be dishonest. Because unpredicted situation may occur, the User Agreement can not get rid of all the legal liability ORAS will face in the future. As the reputation data is collected or bought from other websites, ORAS Plus will not be responsible for the accuracy of the

collected information. But it should be responsible for the legal problems that might be raised by the use of RepCard.

5.1 Assumptions

(1) A user has unique online identity. As we all know, the online identity has become a big issue in online environments. Identities are hard to control and most likely distinguished from one website to another. However, by far there are many services offered to solve the issue such as OpenID[6], ClaimID[7], Opinity[8], Microsoft's CardSpace, and we hope it will be solved in the future by using new technologies such as scanning fingerprint. Since we focus on the problem of the portability of reputation systems, we suppose that a user has unique online identities when using our system.

(2) Web Services of reputation data from other reputation web sites are available to us or the public. We assume that we could obtain the information about one's reputations from different reputation systems but no other information else with regard to privacy etc.

(3) Other websites allow our members to post the RepCard. The RepCard, the core design of our portable reputation system, is merely a chunk of HTML code to the users at the client-side. Without the permissions from such online marketplace websites as eBay, Amazon auction, or Yahoo! Auctions, members will be unable to post the code on those sites to make their reputation portable.

---

6 www.openid.net

7 www.claimid.com

8 www.opinity.com

5.2 Portable Reputation Mechanism

On the basis of the previous study of portability for reputation systems, a user profile badge called RepCard is introduced by our team member (Zousong Duan) to address the portability issue of the reputation system. As we mentioned before, the reorganized feedback provided by reputation systems can be classified in three types: (1) Calculated reputation values, (2) Reputation lists organized by reputation types, and (3) Reputation Certificates. The RepCard is a kind of reputation certificates that can be post on any website. A reputation certificate has its advantages comparing to the other two methods. A reputation certificate can be easily designed to have the portable characteristic and is more feasible to be used in business. As the certificate is used as one user's online ID, it is possible to integrate new technology in the future to solve the identity issue.

A RepCard, generated by a server-side program, is actually a chunk of HTML code that can be included on web pages, bolgs, or in one's email as a signature. When others click on the RepCard, they will be linked to the profile page of a member where the detailed information about his/her reputations at different sites is listed.

Figure 5.1 RepCard

As showed in the figure, users' names, total numbers of transactions and global reputation values will be showed on the RepCard.

In order to build this reputation card mechanism in ORAS Plus, some problems need to be solved first. Who can use the reputation card? How to manage and update the information on the reputation card after users post them on websites? In order to provide more service for our members, only the members of ORAS Plus can get the reputation card (RepCard). We use the verification module to verify whether a user is our member. All members' information will be saved in the member database and the information on the RepCard is directly connected with the member database. That means once the information in the database is updated, the RepCard will display the new information about users. All the reputation cards are generated in RepCard Module. The tracking module is introduced to update the information in the member database as frequently as users wish. The tracking module will get members' names from the database and sent them to the collection module to recollect the information. With the use of the tracking module, our members do not need to go back to our system to get the new RepCards.

The RepCard is a new method to address the online identity issue compared to the PKI (public key infrastucture) which issues legal identity certificates for users. We do not adopt PKI in ORAS Plus, because it is not usually used on the public Internet and can not efficiently solve the problems inherent to recent reputation systems such as when users cheat other users by using different pseudonyms. From technical aspect, the RepCard cannot solve these problems about online identities, but it has a good effect on restricting users' online behavior. A user who has many usernames on the Internet may do transactions with one username in order to get a RepCard containing high reputation values. Also, a user with the same username on different websites will not dare to deceive others, because the bad reputation information on any website will be displayed in the RepCard. More details about the online identity issues and technical aspects of RepCards are mentioned in Duan's thesis [46].

5.3 Calculation Engine

As different types of feedback may be collected from different websites and the reputation lists reorganized by reputation types are not easy for users to judge somebody's reputation quickly, we use a calculation engine to integrate them and make a global reputation value for each user. The calculation engine is working as ORAS (Online Reputation Aggregation System) [40] proposed by Hui Li. In order to get participant's global reputation, ORAS will calculate participants' local reputation value first. The algorithms and parameters used in the calculation are showed as follow:

$$P_i(r_n) = \sum_{k=1}^{I(i)} f_{ik}(r_n) * \frac{W_{ik}}{\sum_{m=1}^{I(i)} W_{im}} \quad \text{if } f_{ik} = r_n \text{ then } f_{ik}(r_n) \text{ is 1; otherwise } f_{ik}(r_n) \text{is 0.} \tag{1}$$

$$W_{ik} = CR_{ik} * CF_{ik}. \tag{2}$$

$$CF_{ik} = a * T_{ik} + b * S_{ik} \qquad a, b \in [0,1] \,\&\, a + b = 1. \tag{3}$$

$$R_i = \sum_{n=1}^{N} P_i(r_n) * NumVal(r_n) \tag{4}$$

Table 5.1 Algorithms in Local Reputation Calculation [40]

where

$P_i(r_n)$ = the estimated probability that ratee i will act as r_n in the future

r_n = discrete feedback such as "excellent", "good", "average", "bad", and "very bad"

N = the number of possible values of discrete feedback

$I(i)$ = the total number of transactions

$f_{ik}(r_n)$ = 1 if ratee i's feedback value for transaction k is r_n ; 0 otherwise

f_{ik} = a ratee i's feedback value for transaction k

W_{ik} = the aggregation weight for ratee i's feedback value for transaction k

CR_{ik} = the credibility of the rater who rated ratee i for transaction k. Note that ratee i can be rated many times by the same rater, however we consider the rater's reputation at the moment transaction k is performed

CF_{ik} = the context factor for ratee i's feedback value for transaction k

T_{ik} = the time context factor for ratee i's feedback value for transaction k

S_{ik} = the size context factor for ratee i's feedback value for transaction k

$NumVal(r_n)$ = the numerical value corresponding to the discrete feedback r_n

There are five inputs that the calculation engine will need to calculate the local reputation value: total transaction numbers (I), transaction time (T), transaction amount (S), rater's credibility (CR) and feedback value (f). We assume that all the data we need could be collected. After collection, T, S, CR and f will be normalized to a number between 0 and 1 for calculation. The normalization process will be performed by the mapping module according to the mapping tables in the mapping database. The mapping tables will be addressed in the Implementation section. All these factors are used to calculate users' local reputation values. Also, the probabilities in formula 1 are important, because it would give the querying agent a better view about the ratee's possible future actions.

Users have to choose the weights (a and b) they want to put on the transaction time and transaction amount. For example, if a user only wants to consider transaction time as a context factor, then she/he would assign 1 to a; and 0 to b. More recent transactions are more likely to reflect the current behavior of the ratee, thereby the feedback for larger and more recent transactions can be assigned more weight. Also, the size of the transaction can be considered in order to avoid the situation where a user behaves honestly for small

transactions and dishonestly for large transactions. It is hard to get each rater's credibility, but it is a very important factor to evaluate the quality of the feedback. Feedback from a rater with a high reputation value will be considered more valuable and given more weight. Total Number of Transactions is also an important parameter and would help avoid the situation that a peer may increase his/her trust value by simply increasing his/her transaction volume. The average amount of satisfaction a ratee received for each transaction will better reflect his/her trustworthiness.

The final global reputation value will be calculated as the following algorithm:

$$GR_i = \sum_{j=1}^{I(j)} R_{ij} * \frac{W_j}{\sum_{m=1}^{I(j)} W_m} \qquad [40]$$

where

R_{ij} = local reputation for ratee i within community j as computed before

W_j = the aggregation weight for community j

$I(j)$ = the number of communities considered

GR is participants' global reputation value which will be showed in RepCard.

After having a calculation engine, the next step is to build it in the architecture. The calculation engine is showed as the aggregation module in ORAS Plus. In order to consume this calculation engine, the architecture of ORAS Plus will build a collection module to gather all the values needed in the calculation. The collection module should have the ability to find all the reputation information about a target participant. The collection module will talk to the Web Services of different reputation systems directly to get the information and then save them for mapping. The mapping module is another important part for the architecture to work with the calculation engine. As different reputation systems

have different rating mechanisms, different types of feedback values and raters' credibility values may be obtained by the collection module. In Hui's [40] calculation method, all the input (transaction time, transaction amount, feedback values and raters' credibility) should be the value between 0 and 1, so the mapping module will take the collected information from the raw database where the collection module saves data and finish the mapping process according to the mapping table in the mapping database. After transfer all the values as required, new values will be sent to the aggregation module for calculations. The whole process can be described as the following figure:

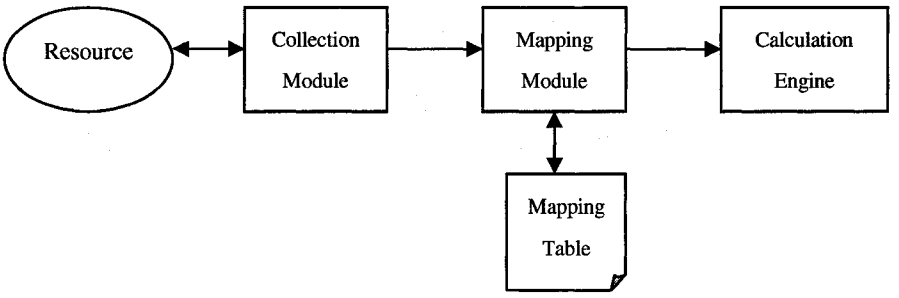

Figure 5.2 Integration process of Calculation Module

5.4 Architecture Design

Based on the design of ORAS (Online Reputation Aggregation System) [40], we proposed a centralized architecture for a portable reputation system that integrates the reputation card mechanism and the calculation engine [40], the ORAS Plus. The components are illustrated in figure 5.3.

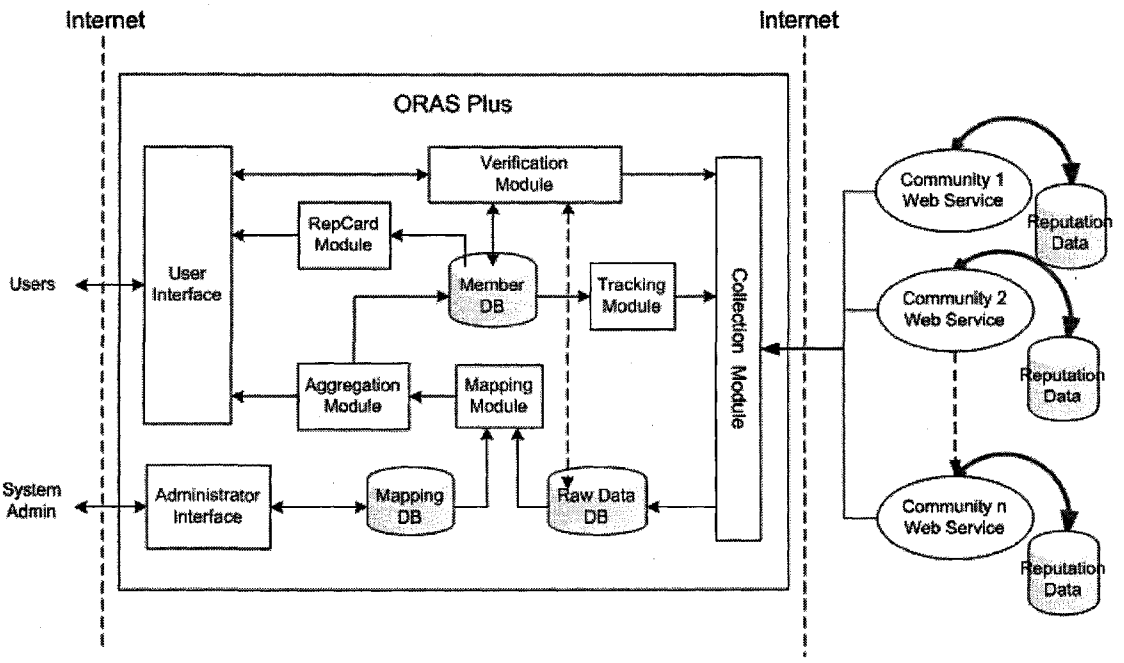

Figure 5.3 Architecture of the Portable Reputation System

In Fig5.3, the RepCard module is designed and implemented by my teammate Duan. The aggregation module is designed and implemented by my teammate Hui. The rest part of the architecture are designed and implemented by the author.

5.5 Components

User Interface

The user interface is a couple of web applications and pages that divides users into two types: regular users and members. Regular users are those who use our system to query for the reputation information about a specific target, and they also can search the reputation information of a group of members with regard of certain criteria (e.g. reputable members with high global reputation values) in Member Database. Regular user can register to become members. Members cannot only search others' reputation, but also have the right to get a reputation certificate after collecting their own reputation values. They also have the privilege to manage their reputation profiles and RepCards.

Administrator Interface

System administrators have the privilege to decide which online communities with reputation systems should be included in ORAS Plus as cooperate partners, classify the corresponding rating mechanism that different communities use, and convert these mechanisms into a universal one through the criteria defined in Mapping Module. After converting, a new mapping table will be generated in the mapping database and ready for use in the mapping process.

RepCard Module

The RepCard module creates a RepCard for each member based on his/her local reputation information gathered by Collection Module and a chunk of HTML code that is used to post the RepCard on other websites. A member' global reputation values will be shown on his/her RepCard after the calculation in Aggregation Module, and s/he can easily copy and post the chunk of HTML code that links to the RepCard on web pages or blogs. When clicking on the RepCard, other Web users will be linked to an up-to-date detailed reputation profile of the member.

Verification Module

The verification module will check whether the queried individual is a member of ORAS Plus or whether s/he has raw reputation data in Raw Data DB. If s/he is a member, the information will be obtained from Member DB and then the result will be returned to her/him. If s/he is not a member but already has raw transaction data stored in Raw Data DB, the system will return the result after mapping and computing the data in Mapping Module and Aggregation Module. In the case that s/he is not a member and doesn't have raw transaction data in the database, the query request will be forwarded to the collection module.

This design derives from the consideration of reducing the workload of the collection module and the money that may cost in the future. Because the tracking module will ensure the reputation information in the database is up-to-date, there is no need to recollect one user's information in a short period and the make more workload for the collection module. Another reason to use this module is that we assume that other websites will give the information free, but if not, we have to buy the information from other websites. That means every time we search and get someone's information from other website, we need to pay for that. Reducing the times of collection will reduce the cost of the system.

Collection Module

The collection module is responsible for collecting the local reputation information of an individual via the Web Services of different communities. The returned information is saved in the raw database. The collection module worked as a searching engine is the core part of our architecture.

Mapping Module

This module converts the local reputation data gathered from different communities into universal formats. For example, some of the reputation systems use an average rating mechanism on a scale of 1-5, while others may use it on a scale of 1-10. Mapping Module standardizes those scales and other criteria and makes them one universal mechanism with a scale of 0-1.

Aggregation Module

Aggregation Module [40] computes the reputation data received from Mapping Module with given algorithms, aggregates the data, and displays the result in a user understandable

format such as a queried individual's globe reputation value, the total number of online interactions s/he has ever performed, and the amount totally involved.

Tracking Module

The tracking module is responsible for updating members' RepCards posted on websites. This module is used to track and receive the latest local reputation information of members. As the information on the reputation cards is connected with the information in the member database, renewing the information on the reputation cards can be achieved by updating the information in the member database. Tracking module will automatically send requests to the member database to get the names of the members who need to update their reputation cards according to the frequency that members indicated. Then it will re-collect members' local reputation information in different online communities automatically to keep the members' reputation information up-to-date, send the gathered information to Mapping Module and Aggregation Module, and then update the reputation values and other data in Member Database, as well as members' reputation profiles and their RepCards. Since some members do not have interactions very often, Tracking Module is set to re-collect their reputation information weekly by default. However, ORAS Plus gives the choices to members so they can choose how often their reputation information is updated when registering and the time they choose will be saved. When one user's information is requested by another user, the information in the database will also be updated.

Member Database

The member database is responsible for storing member's reputation profiles, including local reputation values for different online communities, the global reputation values computed in Aggregation Module and etc.

Mapping Database

This database stores the names and rating mechanisms of different reputation sources and the feedback mapping tables for converting different types of feedback into a universal format that is used in Mapping Module.

Raw Database

All the raw data collected by Collection Module will be saved in this database. Before the calculation, Mapping Module will get the raw information from this database.

5.6 How the system works

ORAS Plus works basically as a reputation searching engine. Every time users ask for a target individual's reputation values (could be themselves') through the interface, the request will be sent to the verification process. Then the verification module will check the target's username in the member database. If the information is already in the database, it will be sent back to the user interface directly. If not, the request will be sent to the collection module that will collect the target individual's existing reputations from different websites or communities. The collected reputation information will be saved in the raw database. The mapping module will get the data from the raw database and finish the mapping process according to the mapping table in the mapping database. After that, the required data will be sent to the aggregation module which calculates the global reputation values for users and users can choose their preferred weights to be put on the transaction time, transaction value and different websites to calculate their local and global reputation values. If the target individule is our member, the calculated data will be saved in the member database, and then the reputation card, RepCard, will be generated from the RepCard module. Members can see their usernames, total transaction numbers and the global reputation values on the RepCards. If members think their global reputation values are good enough to promote their online transactions, they can post the cards on blogs and web pages. All they need to do is copy and post a chunk of HTML code on the blogs and

web pages. In order to keep users' reputation values on the RepCards up-to-date, the tracking module will update the information in the member database as frequently as they wish. The tracking module will firstly get the member's name from the member database and send it to the collection module. The recollection and recalculation process will be done to update one member's information, and then the information on the RepCard will be updated as well.

Chapter 6

Implementation and Testing

The entire implementation is completed by three people. The aggregation module and the RepCard module will be respectively implemented by our team members, Hui Li [40] and Zousong Duan. Therefore, the details of the implementation of the two modules will not be introduced in this thesis.

6.1 System Requirement

We implemented our system in NetBeans, a free development environment, and host three Web Services by using Tomcat. Our software requirements for the implementation are summarized in Table 6.1.

Software	Version	Description
NetBeans	5.5	Java classes and jsp servlets are compiled and executed here. Publishing Web Services.
Apache Tomcat	5.5.17	Web server for hosting Web Services and web applications.
MySql	5.0	To build database and manage members' information.
Internet explorer	6.0	To display results from the system and facilitate users' inputs.

Table 6.1 System Requirements

6.2 Implementation

We built two projects with NetBeans. One is used to demonstrate the Web Services provided by partner websites such as eBay, Yahoo! Auctions and Amazon. The other is the ORAS Plus system that will consume the Web Services and implement the proposed Web applications. In order to collect the data we need, we built three Web Services to simulate as eBay, Yahoo and Amazon in the first project. We also built three databases called eBayDB, YahooDB and AmazonDB that allow the Web Services to access. According to the assumptions before, all the three databases should have all the information we need to calculate users' global reputation values. Each user has a form in the database and each row of the form record users' information in each transaction.

trid	trTime	trSize	feedback	raterCr
1	05/17/2005	17.00	0	20
2	03/06/2006	23.99	1	63

Table 6.2 Examples of eBayDB

Each Web Service has the function to get the data from their database as a major part of the collecting process. For example, eBay's Web Service uses the following codes to get information from eBayDB.

```
Connection con = null;
String driverName = "com.mysql.jdbc.Driver";
String url = "jdbc:mysql://localhost:3306/";
String db = "ebaydb";
String dbuser = "root";
String pass = "7";
String sqlquery = "SELECT * FROM " + userid;
try{
    Class.forName(driverName);
    con = DriverManager.getConnection(url+db, dbuser, pass);
} catch(Exception e) {
    System.out.println(e);
}
try{
     PreparedStatement psmt = con.prepareStatement(sqlquery);
     ResultSet rs = psmt.executeQuery();
}
```

After getting the information from the database, all the data related to one user will be put in one txt file and then the file will be saved in our local hard driver. These data in the txt files will be invoked by Mapping Module in next step. The following codes are used to generate the txt file and write data in it.

```
try {
    FileWriter fw = new FileWriter(f);
    BufferedWriter bw = new BufferedWriter(fw);
    while(rs.next()) {
  bw.write(rs.getString(1)+""+rs.getString(2)+""+rs.getString(3)+""+rs.getString(4)+" "+rs.getString(5));
        bw.newLine();
      }
        bw.close();
        fw.close();
      } catch(IOException ioe) {
      }
```

The other project is our ORAS Plus system that will accept users' queries and consume the Web Services. All the files or classes are connected as follow:

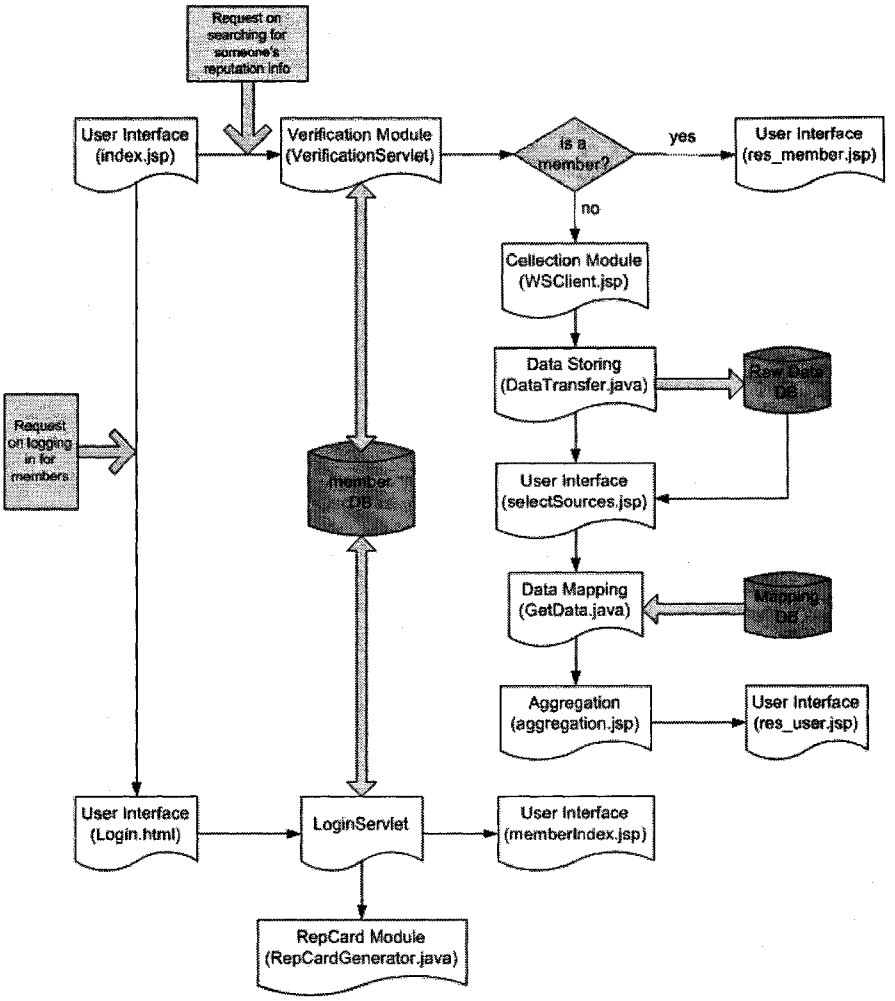

Figure 6.1 System Flow

The searching request about a target user will be sent to a Java servlet called Verificationservlet first, it will check whether the target user is our member or not. This servlet will verify his/her identity according to the names in the member database.

```
if(!user.equals("")){
   String driver = "com.mysql.jdbc.Driver";
   try{
      Class.forName(driver);
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/memberdb", "root", "7");
      String sqlquery = "SELECT * FROM member WHERE memberid = '" + user +"'";
      // execute database query//
      PreparedStatement psmt = con.prepareStatement(sqlquery);
      ResultSet rs = psmt.executeQuery();
      // if he/she is a member, forward this name to res_member.jsp //
      if(rs.next()) {
            rs.close();
            psmt.close();
            con.close();
            forward("/res_member.jsp?memberid="+user,request,response);
       }
      // if not a member, forward this name to WSClient.jsp for collection //
       else {
              rs.close();
              psmt.close();
              con.close();
              forward("/WSClient.jsp?userid="+user,request,response);
              }
   } catch(Exception e){
   }
}
```

If the given user is our member, the res_member.jsp file will get the reputation information from the member database and display them in a web page.

If not, the WSClient.jsp file will talk to the three Web Services we built to ask for the information. This is the first step of the collecting process. Combining the searching process in the Web Services, these two processes consist of the collection module. The following codes are used to connect the three Web Services from the client side.

```
try {
dbwsClient.amazon.AmazonDBWSService aservice = new dbwsClient.amazon.AmazonDBWSService();
dbwsClient.amazon.AmazonDBWS aport = aservice.getAmazonDBWSPort();

dbwsClient.ebay.EbayDBWSService eservice = new dbwsClient.ebay.EbayDBWSService();
dbwsClient.ebay.EbayDBWS eport = eservice.getEbayDBWSPort();

dbwsClient.yahoo.YahooDBWSService yservice = new dbwsClient.yahoo.YahooDBWSService();
dbwsClient.yahoo.YahooDBWS yport = yservice.getYahooDBWSPort();

 aport.getInfo(user);
    eport.getInfo(user);
    yport.getInfo(user);
    } catch (Exception ex) {}
```

The mapping process is another major part of the implementation. All the collected data are saved in the txt files first. Later, the data will be taken from these files and saved into the raw database to prepare for the mapping process. There are four collected values need to be mapped: feedback values, rater's credibility, transaction time and transaction amount. All

these four values mentioned above is the separate information in each transaction. The following figure is an example of the raw database.

id	userid	source	trid	feedback	raterCr	Trtime	Trsize
1	bob	ebay	1	1	45	2006/04/05	40.50
2	mike	yahoo	1	-1	22	2007/01/22	33.99

Table 6.3 Example of Raw Database

Here we use each transaction's feedback values as an example to show the mapping process. One of the mapping tables in the mapping database are showed in Table 6.4.

id	ratingMechanism	oldFeedback	newFeedback
1	text	very good	1
2	text	good	0.75
3	text	average	0.5
4	text	poor	0.25
5	text	very poor	0

Table 6.4 Mapping Table for Feedback Values

This table is used to map two types of rating mechanisms. One is the five-star average mechanism, and the other is the text rating mechanism. All the collected original feedbacks will be transferred to the new feedbacks as the requirement of the table. GetData.java is used to get the old feedback values from the raw database and transfer them into new feedback values according to the table. The method is in the following codes:

```
String fb = temp[1];
// transfer old feedback into new feedback according to mapping table //
String sqlquery = "SELECT method.newFeedback FROM method INNER JOIN source ON
source.ratingMechanism=method.ratingMechanism WHERE source.sourceName='" + source + "' AND
method.oldFeedback ='" + fb + "'";
        PreparedStatement psmt = con.prepareStatement(sqlquery);
        ResultSet rs = psmt.executeQuery();
   // save the value in arraylist //
      if (rs.next()) {
      td.setFeedback(rs.getDouble("newFeedback"));
                  }
      td.setTrId(Integer.parseInt(temp[0]));
      td.setRaterCr(Double.parseDouble(temp[2]));
      td.setTrTime(temp[3]);
      td.setTrSize(Double.parseDouble(temp[4]));
      td.setSource(source);

      arrlst.add(td);
      psmt.close();
                  }
          con.close();
         } catch(Exception e) {
          System.out.println(e);
           }
           br.close();
           fr.close();
   } catch(IOException ioe)
```

As new reputation systems will corporate with our system in the future, more types of rating mechanisms for mapping need to be added in the mapping tables by administrators. AddSrcServlet.java, a java servlet, is used to add the new standard from the administrator interface into the mapping table. For instance, once new information is added in the first row, the AddSrcServlet.java file will excuse the following program to add one new standard in the mapping table.

```
if (!of1.equals(null)) {
 double nf1 = Double.parseDouble(request.getParameter("nf1"));
 String sql1 = "INSERT INTO method (ratingMechanism,oldFeedback,newFeedback) VALUES ('" + rm
+"','"+ of1 +"','"+ nf1 + "')";
            PreparedStatement psmt1 = con.prepareStatement(sql1);
            psmt1.executeUpdate();
            psmt1.close();
}
```

After mapping all the data, the aggregation module will start working. The output of the aggregation module is users' local reputation and global reputation. For our members, the calculated data will be saved in member database.

memberid	globalRep	totalTr	totalAmount	localRep_ebay	localRep_yahoo	localRep_amazone	lastUpdate
jay	0.7	45	5600	0.6	0.7	0.2	2006/03/12
helen	0.8	6	500	0.3	0.5	0.7	2007/08/20

Table 6.5 A Table in Member Database

The global reputation values will be displayed in the reputation cards by the RepCard module. After members post their reputation cards on the websites, tracking module will be in charge of the updating work. As the information on the reputation cards is connected with the information in the member database, renew the information on the reputation cards can be achieved by updating the information in the member database. Tracking module will automatically send requests to the member database to get the names of the members who need to update their reputation cards according to the frequency that members required before. There is a timer in the tracking module that records the update intervals that members chose in the registration, so how frequently one member want to update his/her reputation card is recorded in the tracking module. The rest of the update process is to use the collection module, the mapping module and the aggregation module to recollect and recalculate the data.

6.3 Testing and Scenarios

We test our system in three steps, as there are three roles which involved in the system. We act as an administrator, a regular user and a member to test ORAS Plus.

Scenario 1: Administrators

The administrators of ORAS Plus can add new standard into the mapping table, when a new rating mechanism comes out. Here we will try to add a new reputation information source, eBay, in the mapping table to test our system. Figure 6.2, Figure 6.3 and Figure 6.4 show the administrator page and the mapping database before adding a new reputation source.

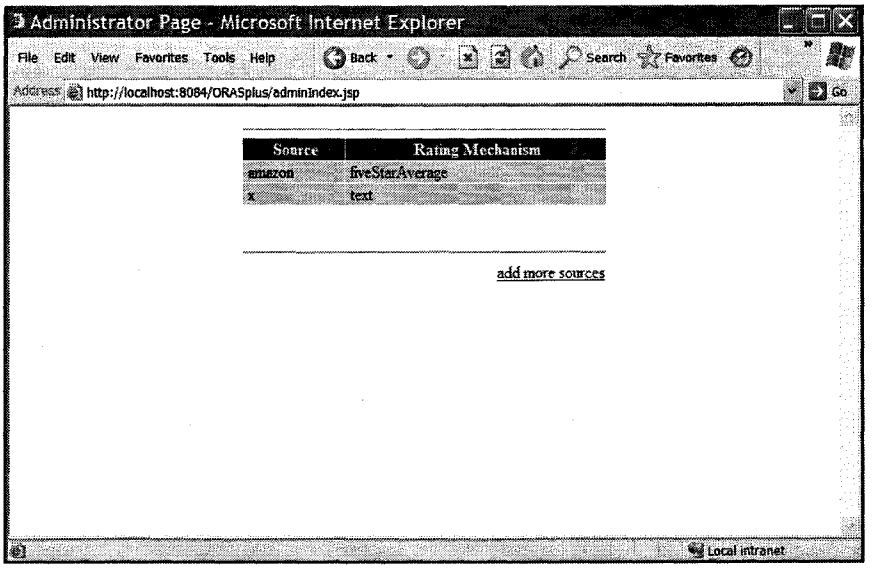

Figure 6.2 Existing Reputation Sources

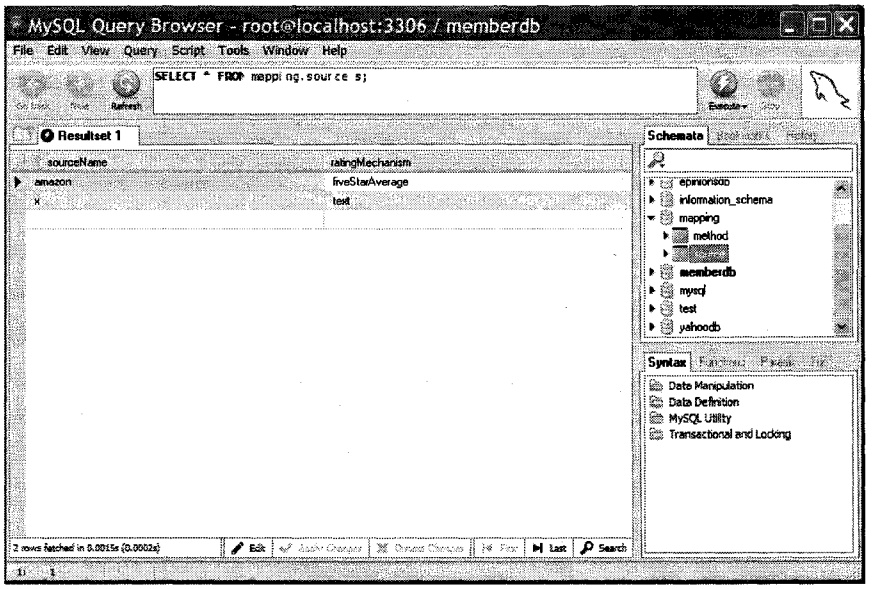

Figure 6.3 Original Reputation Sources in the Database

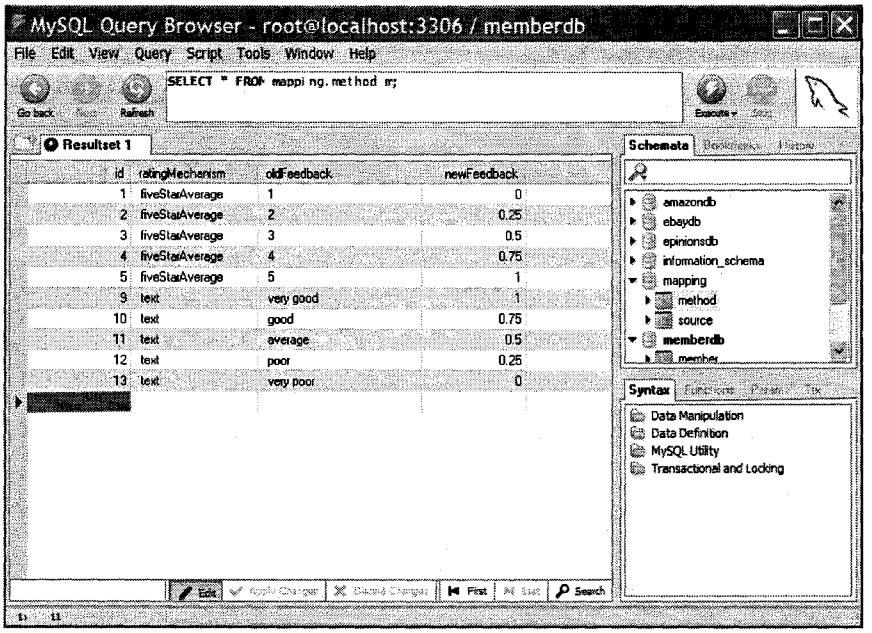

Figure 6.4 Original mapping tables in the Database

Figure 6.5 is the page used for administrators to add new reputation sources. After adding all the feedback values that may appear in the new rating mechanism, the administrators need to tell the system the mapped value for each situation. The mapped values should always between 0 and 1.

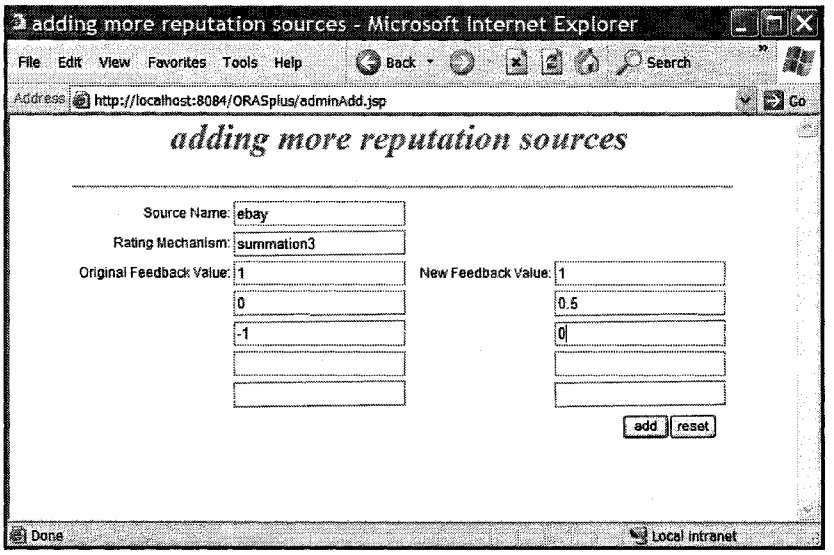

Figure 6.5 JSP Page for Administrators Adding More Reputation Sources

After adding eBay's rating mechanism and mapping information in ORAS Plus, the changes in the mapping database are showed in Figure 6.6, Figure 6.7 and Figure 6.8.

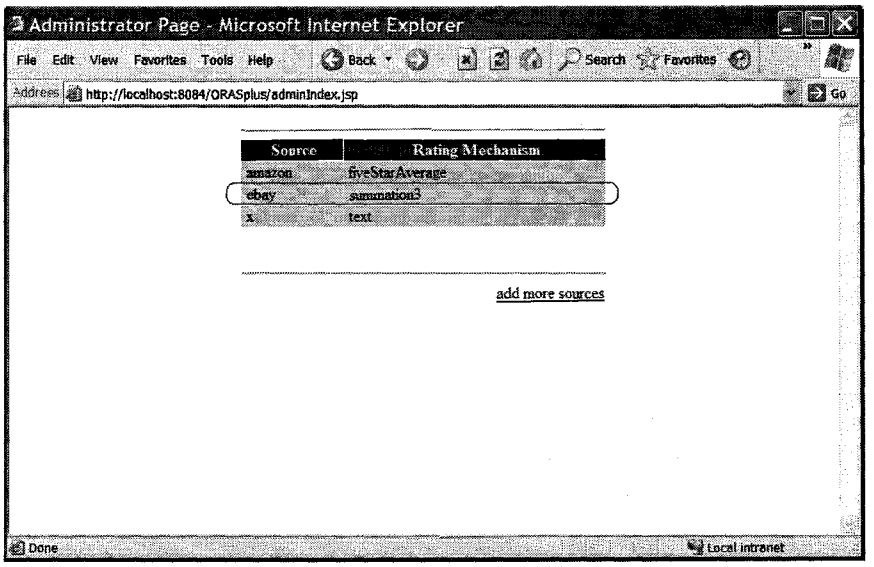

Figure 6.6 JSP Page for Administrators after Adding a New Reputation Source

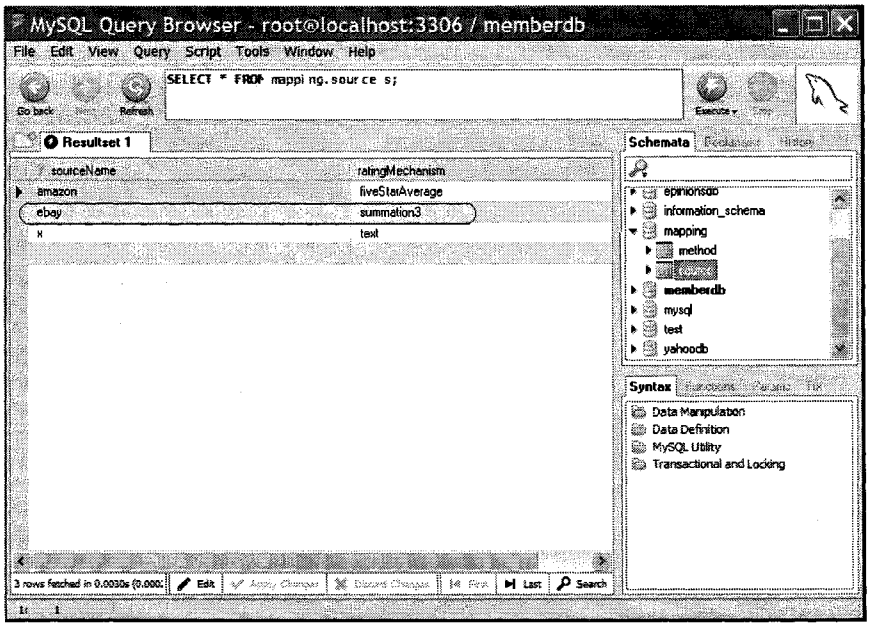

Figure 6.7 Reputation Sources in the Database after Adding a New One

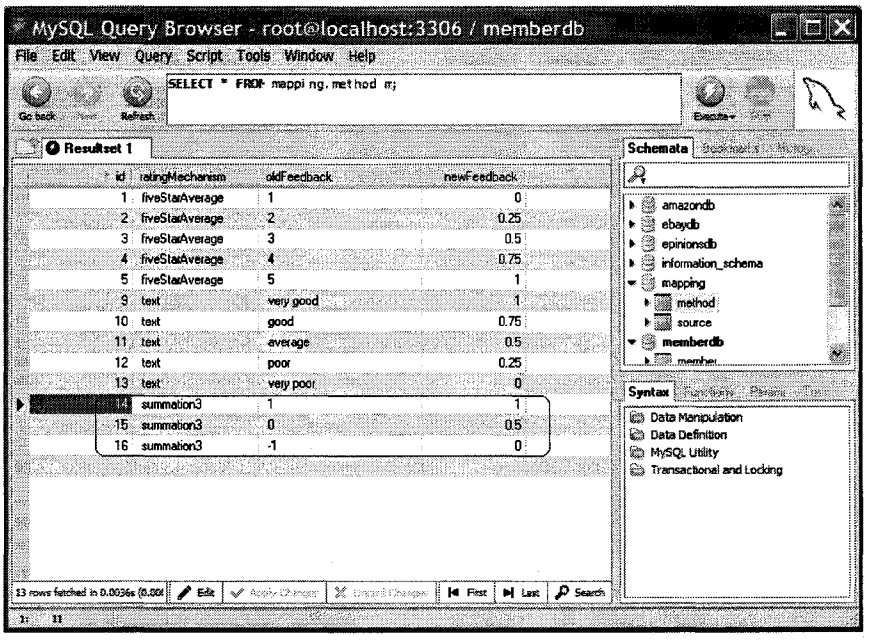

Figure 6.8 New Mapping Tables in the Database

Scenario 2: Regular Users

Regular Users will use our system to search the online reputation of a stranger. All they need to do is tell the name that they want to search to our system. For instance, we type bob in the user interface, a jsp page called index.jsp.

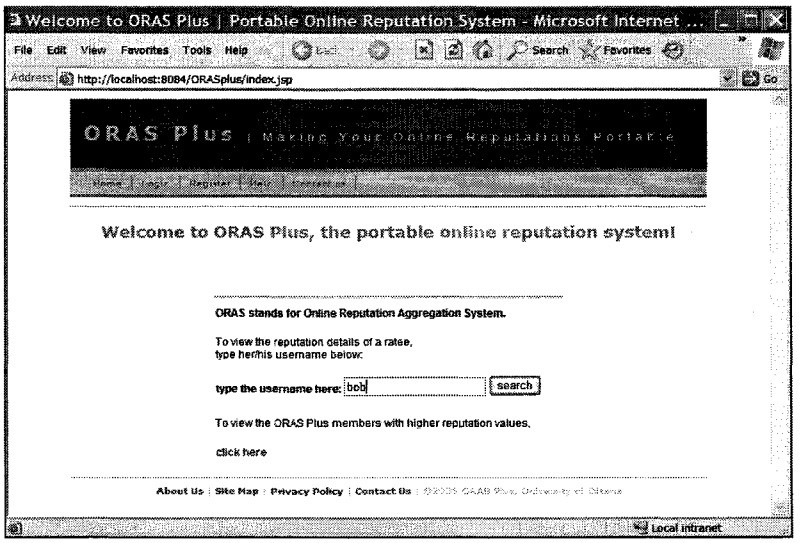

Figure 6.9 User Interface with Username Bob

The searching request will first forward to the verification servlet. As bob is not our member, the serlevt will forward this request to the collection module. In the collection module, our client that connects to all the websites that corporate with us will talk to their Web Services to get the information. After the three Web Services that we built to test our system get bob's reputation information from their databases, the text files that contain bob's reputation information at these communities will be generated in our local hard drive. The DataTransfer.java file will get the data from these text files and save them in the raw database for mapping.

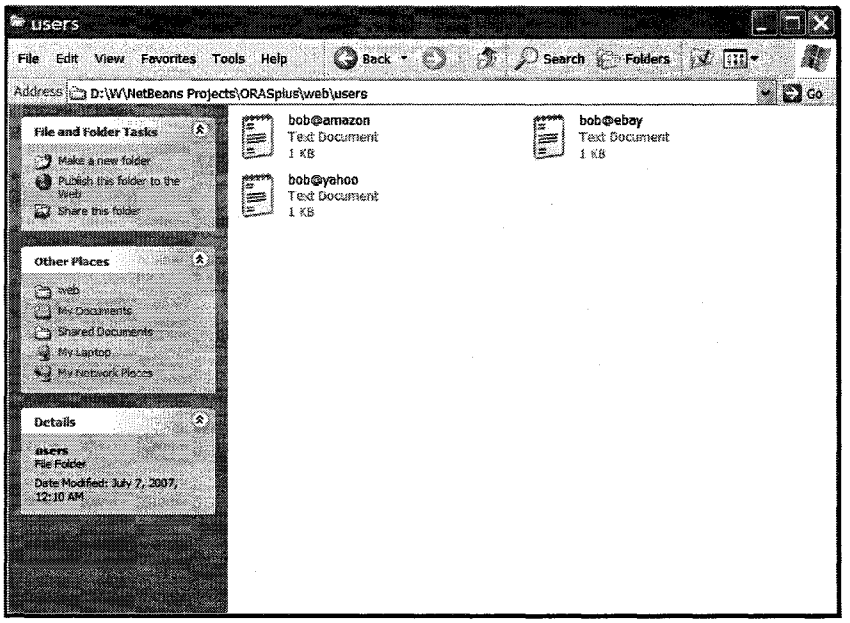

Figure 6.10 Generated Text Files Containing Bob's Reputation Information at Different Online Communities

After the mapping process, the aggregation module will start working to calculate bob's global reputation and the results will be displayed to users as the following figure. There is no information found on Yahoo, so the bob's Yahoo local reputation value is 0.

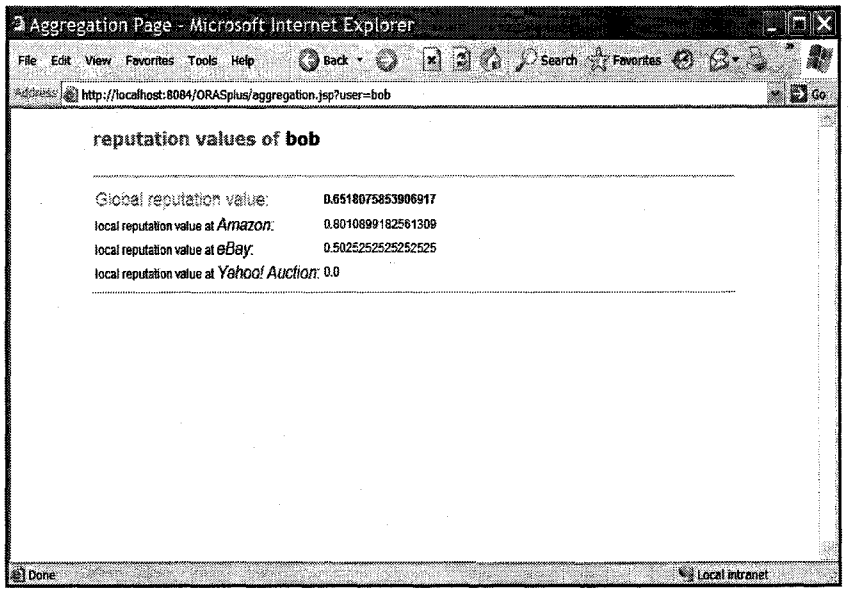

Figure 6.11 bob's local and global reputation value

Scenario 3:

When we search our member's reputation in ORAS Plus, instead of sending request to the collection module, the verification module will get the information from the member database. When we type Helen, one of our members, in the interface in Figure 6.12, the results from the database are showed in Figure 6.13

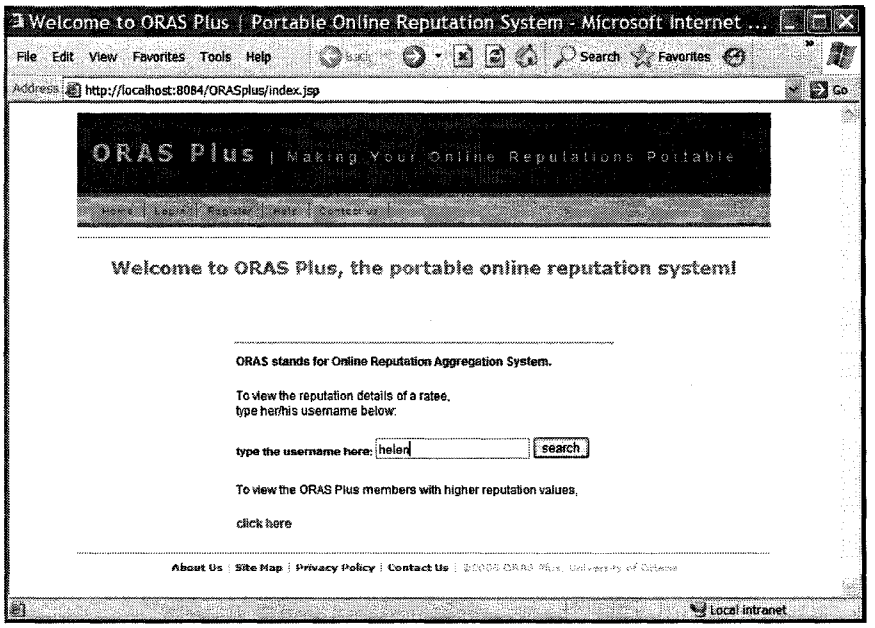

Figure 6.12 User Interface with Username Helen

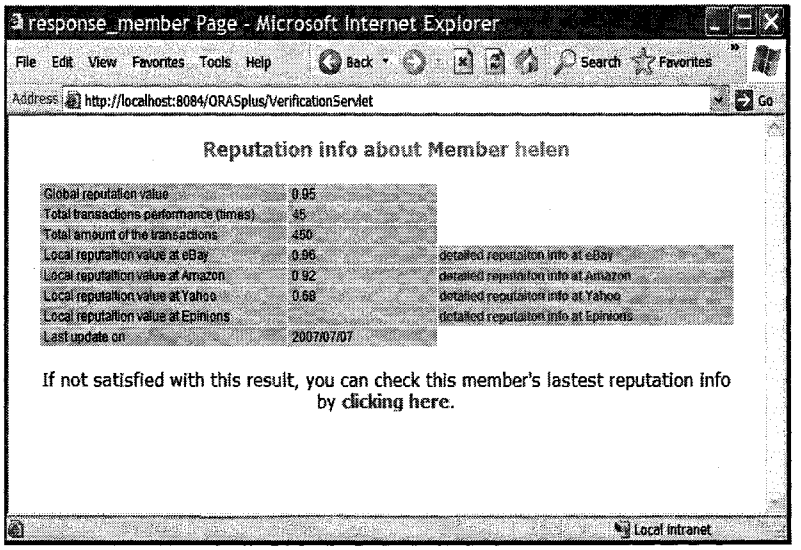

Figure 6.13 Helen's Local and Global Reputation

Users can search for the members with higher reputation values in ORAS Plus. The result will return in a declining rank of global reputation values. This is illustrated in Figure 6.14 below.

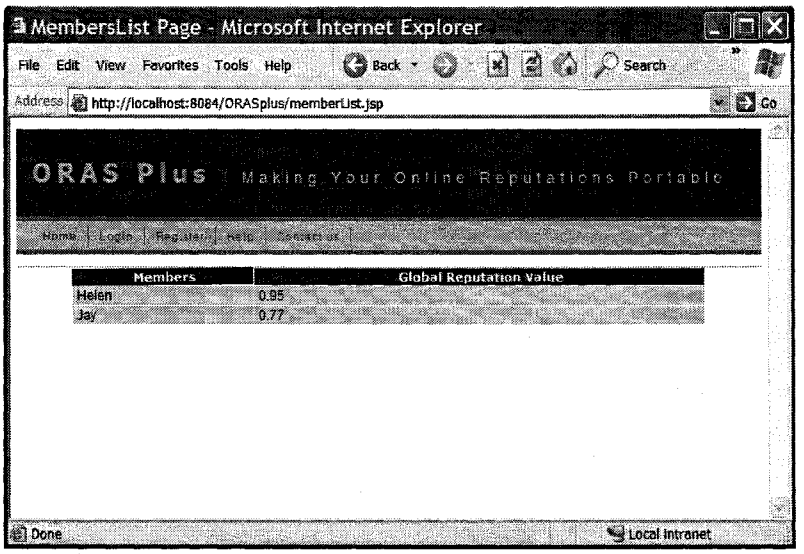

Figure 6.14 Members in a Rank of Global Reputation Value

A member can sign in ORAS Plus to see his/her profile and RepCard. For instance, Jay, one of our members, can post his RepCard on his blog to promote his reputation. The process is described in the following figures.

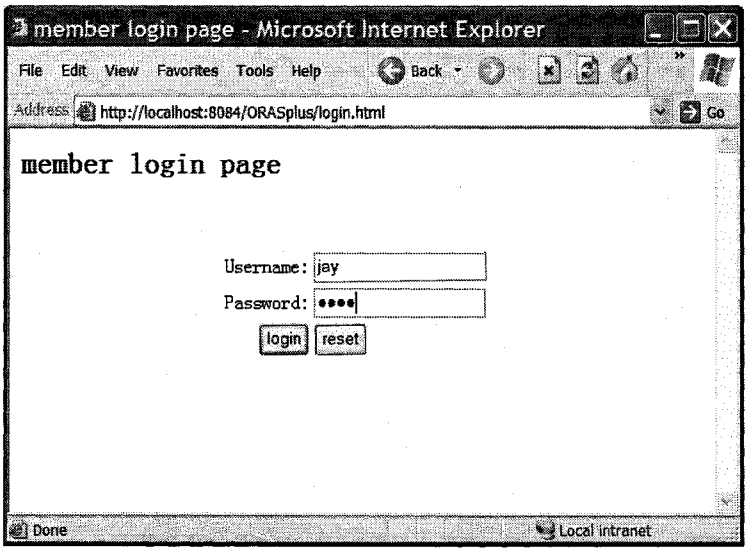

Figure 6.15 Login Page

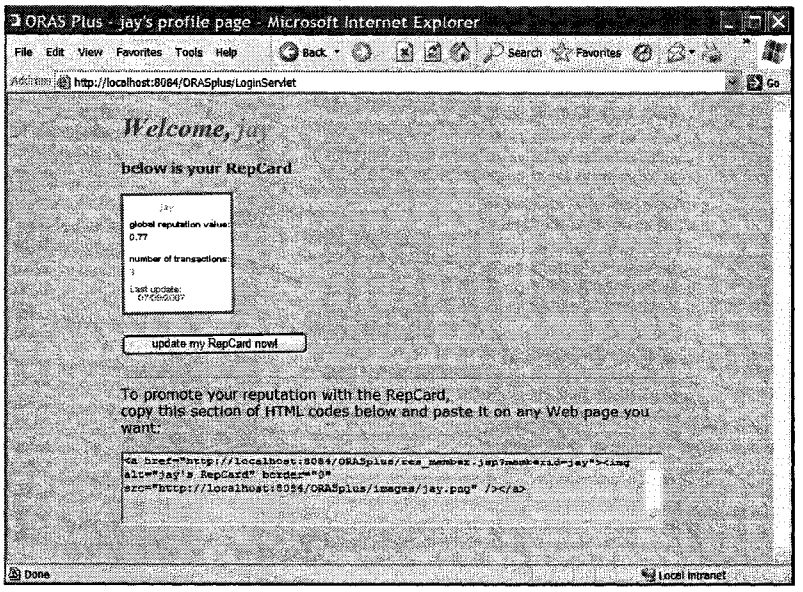

Figure 6.16 Jay's Profile Page

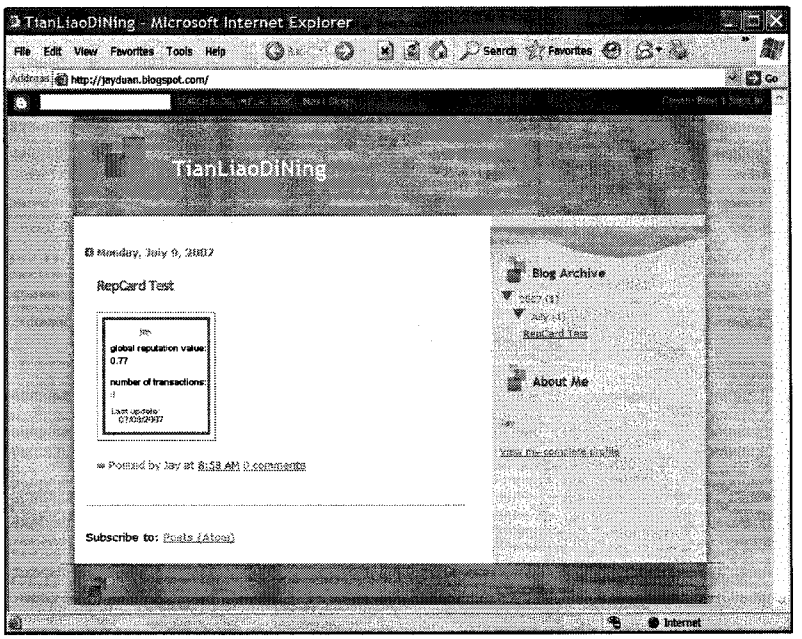

Figure 6.17 Posted RepCard on a Blog

When someone clicks on Jay's RepCard, the details about Jay's reputation will be displayed in a new web page.

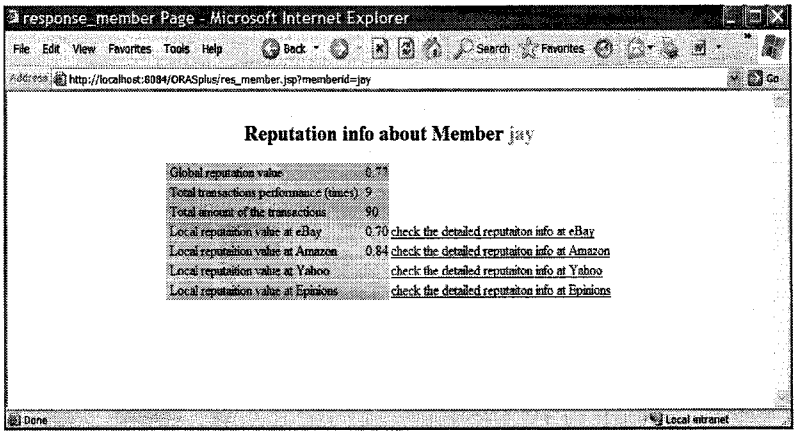

Figure 6.18 Reputation Information Page of Member Jay

Chapter 7

Conclusion

7.1 Contribution

(1) The first contribution is that the thesis analyzed current reputation architectures appeared in the literature and made a classification from the architecture aspect. It compared the two current popular reputation architectures, centralized architecture and distributed architecture, and found out the advantages and the shortcomings of the two methods. After that, this thesis discussed the three main aspects of designing reputation architecture (reputation collection methods, data storage methods and methods of reorganizing collected reputation data) and found out their relationship to the portable reputation systems.

(2) The second contribution of the thesis is that it proposed a reputation architecture (ORAS Plus) which basically solved the drawbacks appeared in current reputation systems by integrating a computation engine and a portable reputation mechanism. ORAS Plus can collect participants' existing reputation from the whole Internet, calculate the global reputation for each participant and generate the reputation card which can be used anywhere. The design of the architecture absorbed the advantages of the centralized architecture and the distributed architecture. For our users, they can get the benefits of distributed reputation systems from this centralized architecture design. ORAS Plus which worked as a reputation search engine and a reputation card provider is a reputation system that is possible to be used in business in the future.

(3) The third contribution of the thesis is that it implemented and tested a proof of concept prototype of the proposed architecture using Web Services, HTML, java servlet and other

web technologies. The result of the test showed the implemented system had achieved the prospective goals of the portable reputation system.

7.2 Limitation and Future Work

Although ORAS Plus solved some shortcomings appeared in current reputation systems such as making users' reputation portable and one user's reputation information all over the Internet could be searched by ORAS Plus, it still has some limitations.

First, it assumed that all the participants will use one unique identity, but one participant may have many User IDs on different websites. As online identity is a big issue, the verification module should have the ability to verify participant's identity by using new technologies or methods in the future. And if all users' identities could be obtained or managed by a third party, ORAS Plus could be implemented without worrying about the identity issues. This could also be achieved by using ORAS Plus within a company's setting to evaluate employees.

Second, the architecture and final results from the aggregation module should be evaluated. The system should be tested when many users are involved at the same period. The final results (global reputation values) should be evaluated and analyzed to help users well understand the meaning of the reputation values.

Another limitation is that we can only collect information from the websites that want to cooperate with us. If ORAS Plus will appear as a business website in the future, it will be very difficult to cooperate with all existing reputation systems. An efficient business module which could make other reputation systems want to cooperate with ORAS Plus should be introduced in the future.

From the architectural perspective, ORAS Plus could have more modules to provide more services to users, such as searching products' or websites' reputation. Providing the reputation of products or websites instead of human users would get around some of the online identity problems. As the users of P2P networks are growing, ORAS Plus could also be designed in a distributed manner in order to work in P2P network. A distributed manner will replace the use of a central database. ORAS Plus could be designed as a software that will be installed on users' PCs or cell phones. All the information will be saved in their PCs or cell phones, and users can make their unique reputation cards by adding their photos on cards. The software will not only have the ability to update the reputation card but also to find the persons and items with high reputation all over the world.

References

1 Keith Bonawitz , Chaitra Chandrasekhar and Rui Viana, Portable Reputations with EgoSphere , 2004 Available at: http://pdos.csail.mit.edu/6.824-2004/reports/bonawitz.pdf

2 Roslan Ismail, Colin Boyd, Audun Josang, and Selywn Russell, A Security Architecture for Reputation Systems. In 4th International Conference, EC-Web Prague, Czech Republic, September 2-5, 2003.

3 Vishal Sankhla, SMART: A Small World based Reputation System for MANETs. Technologies for E-Services, December 2004.

4 Abdelmounaam Rezgui, Athman Bouguettaya, and Zaki Malik, A Reputation-Based Approach to Preserving Privacy in Web Services. Springer Berlin / Heidelberg vol.2819, 2003.

5 Li Xiong, Ling Liu, PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities , IEEE Transaction on Knowledge and Data Engineering, 16(7), July 2004.

6 Chu Yee Liau, Xuan Zhou, Stephane Bressan, and Kian-Lee Tan, Efficient Distributed Reputation Scheme for Peer-to-Peer Systems. Springer, (26): p. 54—63, 2003.

7 Michael Kinateder and Kurt Rothermel, Architecture and Algorithms for a Distributed Reputation System , Trust Management, 2003.

8 Shanshan Song, Kai Hwang, and Runfang Zhou, Trusted P2P Transactions with Fuzzy Reputation Aggregation. IEEE Internet Computing, 9(6): p. 24 – 34, December 2005.

9 Austin Gilbert, Ajith Abraham and Marcin Paprzycki, A System for Ensuring Data Integrity in Grid Environments , In Proceedings of the International Conference on Information Technology, 2004.

10 eBAY INC. Announces Fourth Quarter and Full Year 2006 Financial Results Available at: http://www.ebay.com

11 Audun Jøsang, R.I., & Colin Boyd, A Survey of Trust and Reputation Systems for Online service Provision. Decision Support Systems, 43(2): p. 618-644, 2006.

12 O.E.Williamson and Calculativeness, Trust and Economic Organization, Journal of Law and Economics, 36:453.486, April 1993.

13 D.H. McKnight and N.L. Chervany, The Meanings of Trust, Technical Report Working Paper Series, University of Minnesota, Management Systems Research Center, April 1996.

14 Gambetta. Can We Trust Trust? In D. Gambetta, editor, Trust: Making and Breaking Cooperative Relations, pages 213.238. Basil Blackwell. Oxford, 1990.

15 S. Tadelis. Firm Reputation with Hidden Information, Economic Theory, 21(2):635-651, 2003.

16 P. Resnick, R. Zeckhauser, R. Friedman, and K. Kuwabara. Reputation Systems, Communications of the ACM, 43(12):45.48, December 2000.

17 Chuang Yu, Reputation Propagation between Decentralized P2P Environments, Seminar on Internetworking, April 26th 2005.

18 Tobias MAHLER and Thomas OLSEN, Reputation Systems and Data Protection Law, eAdoption and the Knowledge Economy: Issues, Applications, Case Studies, part 1, ISBN 1 58603 470 7, IOS Press, pp. 180-187, Amsterdam 2004.

19 Keser, C., Experimental games for the design of reputation management systems, IBM Systems Journal, Vol. 42, No. 3, pp. 498–506, 2003.

20 Milojicic D. S., Kalogeraki V. and Lukose R., Peer-to-Peer Computing, Tech Report: HPL-2002-57.

21 M.Yang, Y. Dai and X. Li, "Bring Reputation System to Social Network in the Maze P2P File-Sharing System", The IEEE 2006 International Symposium on Collaborative Technologies and Systems (CTS 2006), May 14-17, 2006, Las Vegas, USA.

22 Runfang Zhou and Kai Hwang, PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing, Transactions on Parallel and Distributed Systems, June 21st, 2006.

23 D. Hughes, G. Coulson, and J. Walkerdine, Free Riding on Gnutella Revisited: The Bell Tolls?, IEEE Distributed Systems Online, Volume 6, June 2005.

24 Bin Yu and M. P. Singh, Distributed Reputation Management for Electronic Commerce, Computational Intelligence, Volume 18, Number 4, 2002.

25 Katri Ylitalo and Yki Kortesniemi, Privacy in Distributed Reputation Management, Security and Privacy for Emerging Areas in Communication Networks, P63-71, Sep. 2005.

26 G. Swamynathan, Reputation Management in Decentralized Networks Major Area Examination: Report,

27 K. Aberer and Z. Despotovic, Managing trust in a Peer-2-Peer information system, In H. Paques, L. Liu, and D. Grossman, editors, Proceedings of the Tenth International Conference on Information and Knowledge Management, New York, ACM Press, pages 310-317, Nov. 5-10 2001,.

28 Y. Wang and J. Vassileva, Trust and Reputation Model in Peer-to-Peer Networks, In: Proc. of the 3rd Int'l Conf. on Peer-to-Peer Computing, IEEE Press, P150-157, 2003.

29 Chang, E., Hussain, F., & Dillion T., Trust and Reputation for Service-Oriented Environments: Technologies for Building Business Intelligence and Consumer Confidence, John Wiley & Sons Ltd, 2006.

30 Roy J. & Ramanujan A., Understanding web services, IT Professional, vol. 3, no. 6, pp. 69 –73, 2001.

31 Cruz S.M M.S., Campos M.L.M., Pires P.F. & Campos L.M, Monitoring e- business web services usage through a log based architecture, in Proceedings of the 2004 IEEE

International Conference on Web Services (ICWS' WS'04), San Diego, pp. 61 –69, July 6–9, 2004.

32 Ba S. & Pavlou P., Evidence of the Effect of Trust Building Technology in Electronic Markets: Price Premiums and Buyer Behavior, MIS Quarterly, 26(3): 243–268, 2002.

33 Hess D., Simple Object Access Protocol (SOAP) and Web Services: An Introduction, Gartner, 2002.

34 Hussain F., Chang E. & Dillon T.S., (2004) Factors of Trust rust that influence Trustworthiness in Peer-to-Peer (P2P) based e-commerce, Proceedings of the International Workshop of Business and Information, Taipei, China, 2004.

35 Hussain F., Chang E. & Dillon T.S., Taxonomy of Trust Relationships in Peer-to-Peer (P2P) Communication, Proceedings of the Second International Workshop on Security in Information Systems, Porto, Portugal, pp. 99 –103, 2004.

36 Mui L., Mohtashem M. & Halberstadt A., A computational Model of Trust and Reputation, 2002.
Available: http://www.cnn.com/2002/WORLD/europe/10/04/world.cities/

37 Sabater J. & Sierra C., REGRET: A Reputation Model for Gregarious Societies, 2003.
Available:http://citeseer.nj.nec.com/cache/papers/cs/22333/http:zSzzSzwww.iiia.csi.eszSzReportsSz2000zSz2000-06.pdf/sabater00regret.pdf

38 Abdul-Rahman A. & Hailes S., Supporting Trust rust in Virtual Communities, 2000.
Available:http://citeseer.nj.nec.com/cache/papers/cs/10496/http:zSzzSzwww-dept.cs.ucl.ac.u

kzSzcgi-binzSzstaffzSzF.AbdulRahmanzSzpapers.plzQzhicss33.pdf/abdul-rahman00supporting.pdf

39 Mamdani, E. and Assilian, S., An experiment linguistic synthesis with a fuzzy logic controller, Int. J. Man Mach. Stud., vol. 7, pp. 1–13, 1975

40 Li, H., Benyoucef, M., & Bochmann, G.V., A Personalized Online Reputation Aggregation System, Working Paper, University of Ottawa, Canada, 2007.

41 Takagi, T. and Sugeno, M., Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Syst. Man. Cybernet., vol. 15, SMC-IS, no. 1, pp. 116–131, 1985.

42 P. Resnick, R. Zeckhauser, Trust Among Strangers in Internet Transactions:Empirical Analysis of eBay's Reputation System , Working Paper for the NBER Workshop on Empirical Studies of Electronic Commerce, 2001.

43 Channabasavaiah, Holley and Tuggle, Migrating to a service-oriented architecture, IBM DeveloperWorks, Dec 16th 2003.

44 Christopher Koch, A New Blueprint for The Enterprise, CIO Magazine, Mar 1st 2005.

45 M. Benyoucef, J. Chandler, K. el-Khaitb, G. von Bochmann, C. Adams, Trust in E-Services, ZDea Group Publishing, Chapter 4, P84-107, 2007.

46 Z. Duan, Online Reputation Sytems, Master's thesis, Ottawa University, Canada, 2007