A Configurable Online Reputation Aggregation System

Hui Li

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

School of Information Technology and Engineering
Faculty of Engineering

University of Ottawa
Ottawa, Ontario K1N6N5, Canada

Abstract

Online reputation systems are currently receiving increased attention while online interactions are flourishing. However, they lack one important feature: globality. Users are allowed to build a reputation within one online community, and sometimes several reputations within several independent online communities, but each reputation is only valid within the corresponding community. Moreover, such reputation is usually aggregated by the provider of the online reputation system, giving the querying agent no say in the process. This thesis presents a novel solution to this problem. We conduct a literature review on existing trust and reputation models and classify these models using proper criteria. We introduce an online reputation system that collects reputation information about a ratee from several online communities and allows for this information to be aggregated according to the inquiring agent's own requirements. We propose a configurable aggregation method for local and global reputation based on a discrete statistical model, taking into account several factors and parameters that qualify the reputation. We also implement a prototype of the proposed reputation computation model.

Dedication

I dedicate this thesis to my beloved parents, my father Peiquan Li and my mother Xianghong Du for their encouragements and support.

Acknowledgement

This research could not have been completed without the support of many people. In the past two years, there have always been someone prepared to help me. I apologize to those I may miss - a collective Thanks to all. Firstly, I would like to express my appreciation to my supervisor, Dr. Morad Benyoucef, and my co-supervisor, Dr. Gregor v. Bochmann, for their continuous encouragement, helpful advice and considerable patience during the completion of this thesis. Secondly, I would like to thank University of Ottawa for funding this interfaculty research. Thirdly, I would also like to thank my colleagues Zhuosong Duan and Bo Zhang for their help and moral support. The discussions with them have inspired me a lot. Fourthly, I would like to thank my parents for standing by me through the many trials and decisions of my educational career. Finally, I greatly appreciate the support and encouragement from my boyfriend Yu Zhang.

Table of Content

List of Tables

List of Figures

Chapter 1

Introduction

The Internet has been very beneficial to daily life by providing vast information and convenient services. For instance, electronic mail reduces the message sending time and makes communication much easier. Online shopping makes it possible to purchase at home. Search engines can get the relevant information immediately. Moreover, the Internet has enabled the proliferation of online business and interpersonal interactions between individuals who have never interacted before. Usually, these interactions are not completed without a certain concern given that private information as well as the exchange of money and goods are involved. Will the items be sent on time after the payment? Is it safe to provide credit card and other private personal information? Is the information from the website reliable? Is this person trustworthy enough to make a deal? Users are usually in the situation where they need to trust complete strangers to make online interactions. A mechanism is therefore needed to build trust among strangers to interact online. There are many mechanisms for fostering online trust, among them reputation systems which rely on feedback from members of an online community to assess the trustworthiness of an individual. The feedback is collected every time when there is an interaction with the individual in question. In a centralized solution, for instance, the feedback is then aggregated into a reputation value and made available to all members of the community.

1.1 Motivation of the Research

There are several difficulties and challenges facing online reputation systems. One that is often mentioned is the fact that a user may change his/her name or identity to escape from a bad reputation. This and other difficulties and challenges are being addressed by the research community with more or less success but they are beyond the scope of this thesis. The reader is referred to [1] for more insight. The issue we are addressing in this thesis is the lack of globality: it is actually difficult to exchange reputation data between different reputation systems [2]. A member of the eBay community, for instance, cannot use his/her reputation outside the eBay community - hence our choice of the name "*local reputation*". Globality is an important and useful feature for online reputation systems. With globality, a user who has a good reputation in one system could use his/her reputation in other systems - hence our choice of the name "*global reputation*". However, except for few timid initiatives, the globality of online reputation has not yet been seriously addressed.

As an intermediate step towards globality, reputation data from different online communities could be aggregated. However, aggregating feedback from different communities is a challenge. Each community calculates reputation differently according to different ontologies of reputation. For example, a rating value on eBay ranges from -1 to 1. Other online communities use ratings between 0 and 5 and may also include textual comments. In order to aggregate reputation data from various communities, we need a common reputation model into which the data can be translated.

1.2 Research Objective

The objectives of the research can be described as follows:

(1) To study online trust, reputation and reputation systems and analyze their relationships.
(2) To conduct a literature review on the current computation models for online trust and reputation.
(3) To design a computation model for the Online Reputation Aggregation System (ORAS) with a set of rating attributes for the aggregation formalism of the computation model.

(4) To implement a prototype of a configurable online reputation aggregation system as an intermediate step towards a global reputation.

1.3 Thesis Organization

The rest of this thesis is organized in the following manner. Chapter 2 discusses the fundamental concepts of trust, reputation and reputation systems. Chapter 3 focuses on the literature review and the classification of important research on trust or reputation computation models. Our design of a configurable online reputation aggregation system is detailed in Chapter 4. Chapter 5 presents the implementation of the proposed system and screenshots from an interaction scenario. Finally, in Chapter 6 we conclude and discuss future research perspectives.

Chapter 2

Trust and Reputation Concepts

Trust and reputation have been important research topics in many fields. This chapter reviews some existing studies on trust, reputation and reputation systems.

2.1 Trust and Online Trust Concepts

People usually suffer from information overload and increased uncertainty in relation to their day-to-day activities in modern life. People can manage these complexities by depending on trust, which is the precondition for the occurrence of online commercial and interpersonal interactions. What is trust? Some studies provide the following definitions.

2.1.1 Definition of Trust

Trust is quite challenging to define because it manifests itself in many different forms in different contexts. Almost every aspect of our everyday lives is based on some form of trust. For example, when we purchase a product from a shop, we may choose to buy a certain brand name because we have found it to be good and trustworthy through past experiences or because it has a good reputation for being widely trusted. We choose to purchase in this store because we trust that this store sells authorized products. While purchasing, we provide our credit card and personal contact information since we trust that this store will keep customers' information confidential. Trust is defined as "a relationship of reliance"[1]. Trust can be treated as a social phenomenon or a complex notion [3]. Moreover, trust is the "perception of the degree to which an exchange partner will fulfill

[1] http://en.wikipedia.org/wiki/Trust

their transactional obligations in situations characterized by risk or uncertainty" [4]. In other words, trust provides a certain degree of security before taking action with transaction partners despite incomplete information and uncertainty.

According to Wang et al. [5], there are two likely reasons for the occurrence of a multiple definition of trust in the literature. First, trust is an abstract concept and is often used interchangeably with credibility, reliability, or confidence. Second, trust is a multi-faceted concept that incorporates cognitive, emotional, and behavioral dimensions.

Research often considers trust from different points of view, such as *social* and *psychological* [6]. From a social point of view, trust is an important factor to keep society operating efficiently. For instance, *dispositional trust* [3] and *general trust* [7] take trust as the player's social roles. From a psychological point of view, trust is subjective since different people may have different degrees of satisfaction toward a transaction. Marsh's *situation trust* belongs to this view (this will be explained later in Section 3.1.1).

Gambetta [8] provides a definition of trust based on probability concepts: *"trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent will perform a particular action, both before [we] can monitor such action (or independently of his capacity of ever to be able to monitor it) and in a context in which it affects [our] own action."* The term "subjective probability" shows that different degrees of trust are not objective; however, they depend on the trusting agents' subjective perception.

In computer science literature, Marsh [7] is among the first to introduce a computational model for trust which integrates many aspects of trust taken from sociology and psychology. This abstract model separates trust into three different aspects: basic, general, and situational. *Basic trust* is the initial trust which an entity may have without relying on neither recommendations nor his/her direct experiences and is derived from agents' past experiences through their entire life. *General trust* is not relative to any specific situation.

Situation trust occurs between agents in specific situations.

Elofson [9] gives the following definition of trust based on observations: "*trust is the outcome of observations leading to the belief that the actions of another may be relied upon, without explicit guarantee, to achieve a goal in a risky situation.*" Elofson also states that trust is dynamic and can be developed over time as the outcome of a series of observations.

Abdul-Rahman et al. [3] divide trust into *direct* and *recommender* trust. While direct trust comes from direct experience, recommender trust can be derived from word-of-mouth recommendations. Four degrees for direct trust are defined as follows: "very trustworthy", "trustworthy", "untrustworthy" and "very untrustworthy". The authors discuss three types of trust in virtual communities as well. *Interpersonal trust* is the direct trust one agent has in another agent. Therefore, it is agent and context specific. *System or impersonal trust* depends on how the agent perceives the institution or system in which s/he is participating, and is not based on any property of the trustee. *Dispositional trust* is the participant's general trusting attitude.

Olsson [10] states that trust is a "relationship between actors" and is a useful concept when collaborative actions need to be performed. Olsson also introduces *individual trust* and *collective trust*. *Individual trust*, which is a fundamental model of trusting behaviour, involves a single agent's trusting behaviour. An agent adapts a behaviour based on his/her own experience from interactions with other agents. *Collective trust* is established not only based on an agent's own experience, but also based on the trust that other agents in the same community have established. Collective trust is similar to recommender trust which has been mentioned before.

Trust can also be classified into *reliability trust* and *decision trust* [11]. Reliability trust is "*the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends*" [11]. Decision trust on the other hand 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*" [11].

Benyoucef mentions four kinds of online trust: consumer-to-consumer (C2C), business-to-consumer (B2C), business-to-business (B2B), and individual-to-individual (i2i) trust [12]. In commercial interactions, C2C occurs when one consumer needs to trust another consumer. B2C trust is that a consumer needs to trust an online business. B2B trust is that one business needs to trust another business. i2i trust exists in interpersonal interactions; these are non-commercial exchanges between individuals.

2.1.2 Characteristics of Trust and Online Trust

Trust is often discussed in the online context. Online trust shares similar characteristics with those of offline trust. However, there are some important distinctions in an online environment. To facilitate a better understanding of the nature of trust in an online context, four characteristics of trust identified by Wang et al. [5] are listed below:

(1) *Trustor and Trustee:* in any trusting relationship, there must be a trusting party (trustor) that tries to trust another party (trustee). These two parties can be individuals or organizations. In online situations, a trustor may be a user who is browsing an e-commerce web site and the trustee in this case may be the e-commerce website itself. In reality, the trustee is the merchant that the website represents. A trustee can also be another user who wants to sell a product online (i.e., on eBay). Trust is based on the evaluation about the ability of the trustee to satisfy the needs of the trustor and the degree of trust that the trustor places on the trustee.

(2) *Vulnerability*: trust is only needed in an environment that is uncertain and risky. In online situations, merchants or individuals may behave in an unpredictable manner

because of the high complexity and anonymity associated with e-commerce transactions. Consumers are vulnerable. They may be cheated and their financial information could be stolen when they are transacting online.

(3) *Produced actions*: trust leads to actions. For instance, you lend money to your friend since you trust that s/he will return it to you later. Consumers would provide credit card and personal information during a purchase if they trust online merchants.

(4) *Subjective matter*: trust is not an objective matter but a *subjective degree of belief* about trustees [3]. It is affected by individual differences and situational factors. Different people could have different degrees of trust towards different trustees in different scenarios. For example, two best friends would have different past experiences with the same product, so one of them trusts this product and another does not.

In addition, trust is *dynamic* and *non-monotonic* [3]. The degree of trust will later be increased or decreased because of additional evidence or experience.

2.2 Reputation Concepts

One of the many ways to foster trust in online interactions is through collecting and managing information about interacting parties' past behaviour. This information can then be aggregated in order to come up with a summary evaluation, which is called "reputation". Reputation is a variable which depends on feedback about an interacting party's past behaviour given by others and it affects the interacting party's future payoffs [1].

2.2.1 Definition of Reputation

According to the Concise Oxford dictionary, reputation is "*what is generally said or believed about a person's or thing's character or standing*". Trust and reputation are used interchangeably in some of the existing literature, since the use of reputation information may foster trust. However, it is clear that there are important differences between the two concepts. Reputation is globally visible to all members of a community. People can trust others based on their good reputation, while they can also trust someone with a bad reputation because they have a certain knowledge of that individual through past direct experience or referral relationships. Hence, reputation is "*a collective measure of trustworthiness (in the sense of reliability) based on the referrals or ratings from members in a community*", while trust is "*a personal and subjective phenomenon*" [11].

Furthermore, according to Alfarez et al., a reputation is "*an expectation about an agent's behaviour based on information about or observation of its past behaviour*" [3]. Querying agents need to rely on other sources of information, such as word-of-mouth information, to evaluate a given agent's reputation when reputation resources are limited.

Change et al. [13] state that the reputation of a *trusted agent* is an aggregated reputation value that is recommended by all *third party recommendation agents* weighted by the *trustworthiness of the recommendation agents*, the *trustworthiness of the opinion* and the ranking of the 1st, 2nd and 3rd hand opinions.

In our research, reputation is an expectation about an agent's behaviour based on information about or observation of its past behaviour. We calculate the estimated probability for each possible outcome based on information about past transactions in our reputation model which will be introduced in Section 4.2.

2.2.2 Ontology of Reputation

Change et al. [13] also propose an *Advanced Reputation Ontology* based on the definition mentioned at the end of the previous section. The graphical view of an *advanced reputation* is shown below:

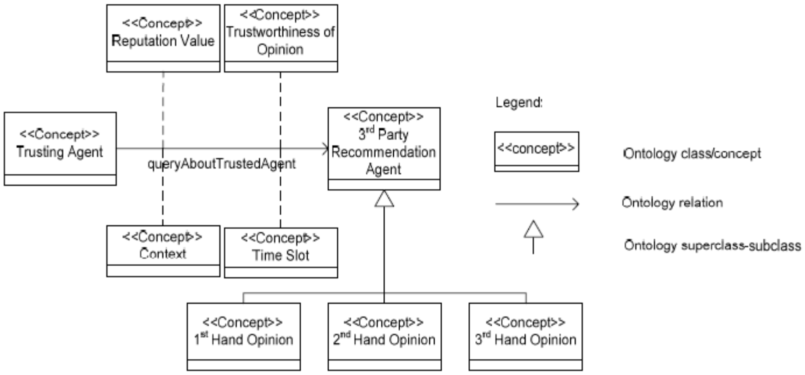


Figure 1: Ontology of an advanced reputation of the trusted agent [13]

The reputation value of the *trusted agent* is an aggregated trust value obtained from *recommendation agents* [13]. The *trusting agent* is an agent who poses a query, asks for feedback, requests recommendation or seeks references about the trusted agent [14]. The *recommendation agent* is an agent that can provide the recommendation to the trusting agent or respond to a reputation query [14]. Context and time slot factors are important when aggregating ratings, since transactions may be different from each other. For example, a user’s honesty on small transactions does not guarantee that s/he would do honest transactions on large ones because of a possibility of a great profit from being dishonest on large transactions. In this case, the size of the transaction should be considered when aggregating the feedback. Due to the consistency of actions, the latest feedback on transactions could be more useful than old ones. Moreover, first-hand opinion is obtained from direct experiences which are not mediated by any third party; second-hand opinion is obtained indirectly from references; and third-hand opinion refers to public opinion [14].

2.2.3 Characteristics of Reputation

Until recently reputation has been considered as a one-dimensional value. In other words, individuals are perceived to have one overall reputation score. Zacharia [15] put forward the concept of reputation as a multidimensional value. An individual may have a very high reputation in one domain, while having a low reputation in another [16]. For example, an individual may have a good reputation for selling quality products, while having bad reputations for late delivery and for high prices.

Moreover, reputation can be seen as a social product or social process [17]. Seen as a social product, it is an opinion or agreement on certain reputation entities. In this case, individuals, groups of individuals, and organizations would have a reputation as well as some other abstract concepts such as services and activities. Seen as a process, it is the flow of information from one agent to another agent and would influence the interpersonal interactions.

Reputation information is generalized by combining personal opinions and opinions from others for the same reputation subject [3]. Casare et al. [17] state that the information sources for reputation can be classified as primary sources or secondary sources, according to whether the information is obtained by direct or indirect interactions. Primary reputation is obtained from direct interactions or observations of those interactions. Secondary reputation is obtained from others' opinions [17]. These reputations can be summarized as several different types of measurements, such as a number, a percentage, a word or an expression. The type of measurement is chosen according to the different levels of detail about the reputation. For example, "good" or "bad" can only describe reputation approximately.

Four distinct types of agents are involved when evaluating the reputation of an agent [17]. They are:

(1) *Evaluators*: these are agents who can develop an evaluation or evaluative belief about other agents, including individuals, groups, organizations, etc. The information used by evaluators can be direct experiences with the targets or through third parties.

(2) *Targets*: these are agents that play the role of the evaluation object.

(3) *Beneficiaries*: these are individuals, groups, organizations, etc. who benefit from the evaluation.

(4) *Propagators*: these are third parties that can propagate the reputation information to

other agents who need the information, usually beneficiaries. In order to propagate the reputation information, a functional ontology of reputation needs to be defined.

2.3 Reputation Systems

Online reputation systems are community tools that "*collect, distribute, and aggregate feedback about participants' past behaviour*" [2]. They help people decide whom to trust, encourage trustworthy behaviour, and discourage unskilled or dishonest participations. Reputation systems are related to "collaborative filtering systems" [11]. Malaga [16] emphasizes the meaning of "collaborative filtering" given by Goldberg et al. [18] that people collaborate to help each other perform filtering action by recording their reaction toward documents they read. These systems allow users to provide evaluations about a target user after the completion of a transaction. The target usually includes a product, a piece of information and a service provider such as an individual, a company and an organization. The results of the evaluation are aggregated into a final reputation score, which can help other users to decide whether or not to transact with this target user in the future. It in turn provides an incentive for good behaviour in the online environment.

According to Resnick et al. [2], reputation systems need to have the following three properties to operate:

(1) *Longevity of agents*: agents are long lived, which means that it should be impossible for an agent to change his/her identity or pseudonym to erase the records about his/her past behaviours. Without longevity, agents can erase their bad reputation scores easily, so new reputation scores may not reflect their real reputation status.

(2) *Protocol of ratings*: reputation systems need to have a certain protocol by which ratings about current interactions are captured and distributed.

(3) *Usability of reputation system*: ratings about past interactions must be useful to guide certain decisions or actions. There is no reason for reputation systems to exist without any usability.

2.3.1 Existing Online Reputation Systems

Individuals usually need to trust complete strangers online in order to conclude a commercial or interpersonal deal. Online reputation systems play multiple roles in determining the reputation of agents. As mentioned earlier, they usually collect, aggregate and disseminate reputation information about agents. Online reputation systems are emerging as "*a promising alternative to more traditional trust building mechanisms, such as branding and formal contracting, in settings where the latter may be ineffective or prohibitively expensive* " [19]. Reputation systems are already being used in many online commercial applications, such as eBay, Amazon's zShops[2], Yahoo!Auction[3], ePinions[4] and Bizrate[5]. EBay, founded as an antique and collectibles auction site in 1995, is a pioneer in applying the concept of reputation systems. In just a few years, eBay grew from a small online company to the world's largest online auction site. In April 2004, eBay had 94.9 million registered users globally[6]. In January 2005, there were more than 125 million eBay users worldwide[7]. And by the end of 2006, the site had 220 million registered users[8]. Some believe that eBay's success is due to its reputation system. A brief overview of the major online entities offering some form of reputation system is given below.

(1) *eBay*: it allows participants to rate each other using a three point scale rating which consists of +1,0,or -1 for positive, neutral, or negative; and a text comment. All the feedback values are then added together (i.e., aggregated) to form one overall reputation value to be consulted by members of the eBay community. Users can choose a specific time slot for a target, such as 30 days, 60 days or 90 days.

(2) *Amazon's zShops*: users are provided a 1-to-5 rating scale and a set of measures such as

[2] www.zshops.com/
[3] auctions.yahoo.com/
[4] www.epinions.com/
[5] www.bitrate.com/
[6] www.cbsnews.com/stories/2004/04/13/tech/main611598.shtml?source=search_story
[7] www.cbsnews.com/stories/2002/10/30/60II/main527542.shtml
[8] news.com.com/2100-7349_3-6165628.html

"product quality", "fairness", etc. The system attaches an average score to a specific identity. Users, including non-members, can post evaluation on reviews as being "helpful" or "not helpful".

(3) *Yahoo!Auction*: users use a three point scale similar to the one on eBay to rate each other. Yahoo! US and Canada sites will be retiring on June 16, 2007.

(4) *ePinions*: it offers product, merchant, review and reviewer ratings and tries to categorize products, merchants, etc. Reviews consist of 1-to-5 ratings and text ratings on many aspects such as *ease of ordering*, *customer service*, *on-time delivery* etc. Reviews can be rated as "not helpful", "somewhat helpful", "helpful" or "very helpful". The exact method for calculating reputation scores is unknown.

(5) *Bizrate*: consumers are asked to rate site navigation, selection, prices, shopping options and their satisfaction with the shopping experience. If a merchant gets a sufficient number of positive ratings over a period of time, it can be granted a *Customer Certification*. This merchant can display the BizRate Customer Certificated seal on its website. BizRate also runs a product review service. Users, including non-members, can post evaluation on reviews as being "helpful", "not helpful" or "off topic". Reviews are ordered according to the ratio of helpfulness over total evaluations, where the reviews with the highest ratio are listed first.

Malaga [16] identifies the following six major problems with current online reputation systems:

(1) *Inaccurate equations*: some existing reputation management systems use equations which cannot accurately reflect the ratees' reputation. For example, eBay uses simple summation to calculate reputation scores, and so does Yahoo! Auction and Auction Universe. On eBay, a user who has had 100 positive ratings will have the same reputation score as a person who has had 300 positive and 200 negative ones.

(2) *Barrier to entry*: users usually start with a reputation of zero or a very low reputation score. This would be a barrier for them to enter into the market, since many users would not deal with the ones with low reputation scores.

(3) *No Incentive to rate*: there is no incentive for users to rate transactions. This would lead to insufficient ratings hence an accurate reputation score.

(4) *Inability to filter or search*: some reputation systems face information overload problems. EBay does provide the ability to search for a particular item, but it does not include reputation as a search criterion. For instance, an eBay user might search for a Canon digital camera from a seller who has a reputation score above 50. The ability of filter and search by reputation would definitely improve the efficiency and usability of reputation systems.

(5) *Categorization*: until recently reputation has been considered one-dimensional in most reputation systems. Actually, entities may have many reputations. As mentioned earlier, for example, a person would have a high reputation on product quality, while having a low reputation delivering on time.

(6) *Unlimited memory*: many reputation management systems have unlimited memories. In that case, they would use all transactions to calculate an overall reputation. Some sites only consider the most recent ratings; however, it makes it impossible for users to know about the past behaviour of the target user.

2.3.2 Classification of Reputation Systems

Reputation systems can be classified into negative, positive and hybrid reputation systems [20]. A *negative reputation system* gathers and distributes feedback on untrustworthy participants to discourage their negative behaviour; while a *positive reputation system* rewards participants with a history of honest behaviour and the trustworthiness of a person is evaluated by the total number of cooperative transactions during his/her past experiences [20]. In a *hybrid reputation system*, both positive and negative behaviours are taken into account. In such a case, participants start with neutral reputation values, then points are taken away as a punishment for bad behaviour or added as a reward for good behaviour [21]. EBay's feedback forum is an example of a hybrid reputation system [11].

Additionally, Josang [11] defines two main types of reputation systems according to their network architectures called centralised and distributed reputation systems. The architecture determines how ratings and reputation scores are communicated between users [11]. Most existing reputation systems are centralized, but distributed reputation systems have attracted many researchers' attention recently. In centralised reputation systems, the performance of a given user is rated by others in the community according to their direct experiences. A central authority, also called reputation center, is used to collect all ratings, derive a reputation score for each user and make this reputation score publicly available to members of the community. Thereafter, users can decide whether or not to transact with a particular user depending on his/her reputation score. Figure 2 below shows a general framework for a centralized reputation system.

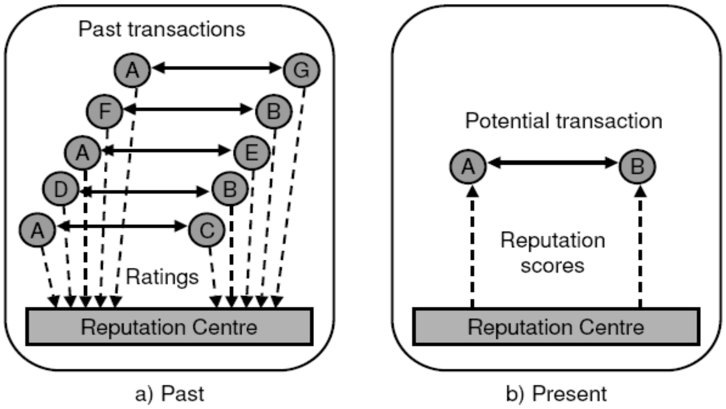


Figure 2: General framework for a centralized reputation system [11]

A and B denote transaction partners, who have had a transaction history in the past, and want to perform a transaction at present according to the partner's reputation scores. EBay's reputation system is a typical centralized reputation system.

A centralized reputation system is helpful in fostering trust among strangers. However, most existing centralized reputation systems suffer from some limitations. Yu [22] points out some of them. The central authority requires users to explicitly make and publicly reveal their ratings of others, which could not be acceptable to many users. Furthermore, a central storage is needed to keep the reputation information, which could be very costly. Additionally, users might not want their private actions to be tracked by a *central point* [23].

A distributed reputation system is better suited than a centralised reputation system when there is no centralised function in the system [11]. In a distributed reputation system, there is no large, recognizable central location for submitting ratings or obtaining reputation scores of others. According to Josang [11], in a distributed reputation system, there can be *distributed stores* where users can submit their ratings, or each user can record his/her opinion, and provide this information in response to querying users. A querying user who wants to transact with a target user, needs to either find the *distributed stores* or try to gather direct opinions from as many previous transaction partners as possible. In centralized reputation systems, the central authority calculates users' reputation score; instead, in distributed ones, querying users compute the reputation score based on the gathered ratings, and possibly on additional information. Figure 3below shows a general framework for a distributed reputation system.

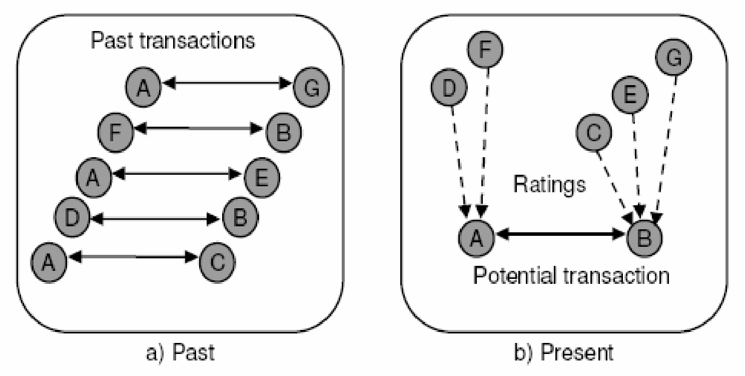


Figure 3: General framework for a distributed reputation system [11]

Chapter 3
Current Computation Models for Online Trust and Reputation: A Classification of Approaches

Various computation models for online reputation systems have appeared in the last few years, each one with its own characteristics and using different computational formalisms. This chapter covers a selection of computational trust and reputation models that represent a good sample of the current research and classifies them based on characteristics that would be beneficial for our future design.

3.1 Related Work

The following sections will review and comment on some of the existing computation models for online trust and reputation. Their classification will be given in Section 3.2.

3.1.1 S. Marsh's Trust Model

Marsh has separated trust into three different aspects: *basic*, *general*, and *situational* in his computation trust model [7]:

(1) *Basic trust*: it is the initial trust which is derived from a user's past experience. Marsh uses T_x to represent the basic trust of entity x having a value between -1 and 1(not including 1). A value of +1 would represent a blind trust which is not accepted here.

(2) *General trust*: this value simply represents general trust a user has in another and is not under any specific situation. $T_x(y)$ is used to represent the amount of trust entity x has in entity y. This value is in the interval [-1, 1). A value of 0 means x has no trust in y. A value of −1 would mean a negative trust.

(3) *Situational trust*: entities usually find themselves in different situations. Further, different situations would result in different values of trust even for the same person. Situation trust, denoted by $T_x(y, a)^t$, is the trust that one entity has in another in a specific situation. The basic formula for situational trust is:

$$T_x(y,a)^t = U_x(a)^t \times I_x(a)^t \times estimate\left(T_x(y)^t\right). \tag{3.1}$$

An entity x wants to evaluate entity y's trust under situation a at time t. $U_x(\alpha)^t$ represents the utility x gains from situation α at time t. $I_x(\alpha)^t$ is the importance of the situation α for agent x at time t. *estimate(*$T_x(y)^t$) is the estimate of general trust after taking into account all possible relevant data with respect to $T_x(y,\alpha)$ in the past. For example, if t is the current time, x will aggregate all situations $T_x(y, \sigma)^T$, where $\theta < T < t$ (θ and t define the temporal interval that the agent is considering) and σ is similar or identical to the present situation α. Only the experiences within this interval will be considered for the aggregation.

Three statistical methods are used to define *estimate(*$T_x(y)^t$): the *maximum*, the *minimum* and the *mean*. The optimistic method always selects the maximum trust value from the range of experiences s/he has had; the pessimistic method uses the minimum trust value from the experience; the realistic method calculates the estimate value as an average using the formula:

$$estimate\left(T_x(y)\right) = \frac{1}{|A|}\sum_{a \in A} T_x(y). \tag{3.2}$$

A is the set of all situations which are similar to the present situation α available in the consideration interval.

By evaluating the trust value of another entity, the evaluating entity would know whether s/he is trustworthy to transact or not in general or under certain situation. Marsh's model is one of the earliest trust models; therefore it has been very beneficial to later researches studying trust.

3.1.2 Sporas and Histos

Zacharia et al. [15] propose two reputation mechanisms called Sporas and Histos. In Sporas [15, 24], new users start with a minimum reputation value. And the reputation value of a given user should be higher than the reputation of a new user at any given time. Only the most recent rating between two users is considered, if they have interacted more than once. Moreover, for users with very high reputation values, their ratings change much less after each update than for users with low reputation values. This approach is similar to credit card history schemes[9]. For example, if a user enters the system with low reputation at the beginning, s/he will not suffer forever from the early poor reputation when s/he improves her/his behavior later. Similarly, a user with high reputation will not ruin his/her reputation for a long time because of being unreliable once. This model encourages new users' entries and takes the reputation of raters into consideration when updating the reputation value of a ratee.

Histos [15] is a more configurable model and can deal with direct information and witness information. In Histos, a directed graph is used to represent the pair-wise ratings in the system, where nodes represent agents and edges represent the most recent reputation rating given by one user to another with the direction pointing toward the rated user [15]. If there is a connected path from A_1 to $A_{13,}$ we can compute a more configurable reputation value for $A_{13.}$

[9] Fair Isaak Co. http://www.fairisaac.com

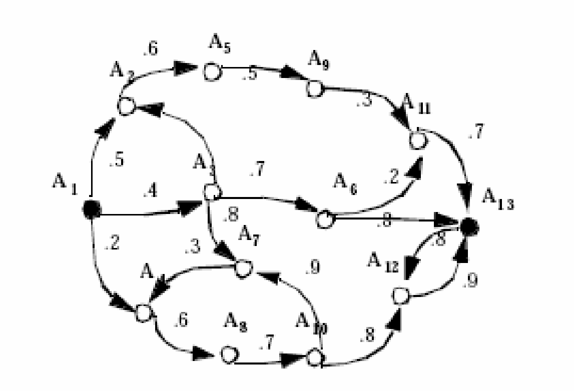


Figure 4: A directed graph representing the rating paths between user A1 and A13 [15]

The system uses a Breadth-First-Search[10] algorithm to find all paths connecting A_1 to A_{13} that are of length less than or equal to some limit. Then the reputation of A_{13} is calculated recursively as a weighted mean of the rating values that users in the previous level (i.e., $A_{11,}$ $A_{6,}A_{12}$) gave to A_{13} [15]. The weights are the reputations of the raters and the number of paths is limited in this model. In the base case of the recursion where the path length is 1, the ratee has been rated directly by the root of the graph. In this case, the reputation value of the ratee is equal to the rating value rated by the root of the graph.

Different from Sporas, the reputation value in Histos is a subjective property assigned particularly by each user. In this model, the most recent experience of direct interactions will be evaluated. The reputation value will not depend on context. The advantage of this model relies on its use of witness information. The shortcoming of this model is the use of reputation value as the reliability of a witness which is not always true in ordinary life. For example, a user with a good reputation could be an unreliable witness and provide dishonest ratings for his/her competitor.

3.1.3 The Trust-Reputation Model of A.-Rahman and Hailes

Abdul-Rahman and Hailes [3] propose a model based on subjective probability which deals exclusively with beliefs about the trustworthiness of users based on the user's col-

[10] http://en.wikipedia.org/wiki/Breadth-first_search

lected statistics on 1) *direct experience* and 2) *recommendation* from others. The trustworthiness of a user (rater or ratee) is evaluated as “very trustworthy” (vt), “trustworthy” (t), “untrustworthy” (u) and “very untrustworthy” (vu). To evaluate the reputation of a user in a certain context, we need to obtain a tuple with the number of past experiences in each trust degree. For direct interactions, the trust value is equal to the degree that corresponds to the maximum value in the tuple. For instance, if an associated 4-tuple represented by $s = (s_{vt}, s_t, s_u, s_{vu})\text{is}(0,3,0,2)$, the trust of this user will be *t* (trustworthy) which corresponds to the second position in the tuple. If there is more than one maximum value in the tuple, then the trust value will be assigned an *uncertainty* value. The authors point out three possible situations, which are “*mostly good and some bad*”, “*mostly bad and some good*”, and “*equal amount of good and bad*”.

In this model, the recommended trust degree can be adjusted according to the *semantic distance*. The *semantic distance* is used to represent the trust degree for giving *recommendation* about others. This concept is introduced as follows. After an interaction with the ratee, the querying agent has his/her own perception of that ratee. The difference between the rater’s and querying agent’s perceptions is called *semantic distance* which can be used to adjust future recommendations from the rater [3]. For instance, let us assume that rater *a* recommends to querying agent *q* that ratee *i* is “very trustworthy”; however, *q*’s perception about *i* (after *q* and *i* interacted with each other) is only “trustworthy”; then the future recommendation from *a* will be downgraded by one rank. The rating provided by a rater is downgraded / upgraded according to the semantic distance.

This trust model is based on real-world social trust characteristics and uses a word-of-mouth reputation mechanism. The special point of this model is that it evaluates the trust on the recommendation given by the rater. Further, this recommendation will be adjusted according to the difference between the recommendation of the rater and the direct perception of the querying agent gotten from past experiences.

3.1.4 Malaga’s Web-based Reputation Management System

Ross A. Malaga [16] examines current approaches used in web-based reputation systems and outlines six main problems with them. These six problems are inaccurate equations, barrier to entry, no incentive to rate, inability to search, no categorization and unlimited memory which are already discussed in Section 2.3.1. The author developed an equation which can solve some of these problems. The equation is as follows [16]:

$$R_{it} = \frac{\sum_{k=1}^{t-1} R_{ik} \times w \times R_k + s + v \times F_{tot}}{t-1} \quad . \tag{3.3}$$

where R_{it} is an individual *i*’s reputation at transaction *t*, R_{ik} is an individual *i*’s reputation rating given by rater *k* for a specific transaction which is the previous transaction before transaction *t*, R_k is rater *k*’s reputation score for the specific rating R_{ik}, *w* is a constant used to determine how much weight to give to a rater’s reputation, *s* is a starting reputation score which can encourage new users’ entry with meaningful reputation scores, F_{tot} is the total number of times an individual has rated others and *v* is a constant used to determine how much weight to give to an individual’s rating of others. Taking F_{tot} into account will provide the incentives for users to rate.

In our evaluation, this model solves some problems of current web-based reputation systems; however it does not specify the values of *s*, constant *w* and *v*. Additionally, a user can theoretically build a very high reputation just by rating others. Therefore, the amount of reputation that can be gained by rating others must be limited. How the initial reputation should be determined is another concern. This method does not consider the time slot factor of evaluation.

3.1.5 Yu and Singh’s Evidential Model

Yu and Singh [25] have proposed an evidential model for distributed reputation systems. In large and open distributed reputation systems, to find trustworthy transaction partners, collaboration with witnesses is a good way to check other users' past behaviour. A witness can provide two kinds of information when it is queried about a ratee [25]. If the witness knows or has had transactions with the ratee, it will return the information about it directly. Otherwise, it will return referrals of other witnesses who could provide useful information to the querying agent. These referrals can either provide the useful information or provide new referrals within the desired limit. The set of referral chains generated is a *TrustNet* similar to that used in the Histos model [15]. The information stored by a user about direct interactions is a set of values that reflect the quality of service (*QoS*). Only the most recent experiences are considered when calculating the *QoS*.

In this evidential model, two possible outcomes are assumed, which are trustworthy (T_A) or not trustworthy ($\neg T_A$), and separate beliefs are being kept about whether A is trustworthy or not, denoted by m (T_A) and m ($\neg T_A$). The reputation score R of an agent A is then defined as [25]:

$$R(A) = m(T_A) - m(\neg T_A) \quad \text{where } m(T_A), m(\neg T_A) \in [0,1] \qquad \textbf{(3.4)}$$

$$R(A) \in [-1,1]$$

This evidential model ignores the role of underlying beliefs and takes testimonies into account [25]. Each querying agent defines upper and lower thresholds that define the frontier between trustworthy and non-trustworthy agents. The ratings provided by raters are belief measures determined as a function of A's past history of behaviours with raters as trustworthy or not trustworthy using predefined threshold values as described above [25]. These belief measures from different witnesses are then aggregated using Dempster-shafer theory[11] and the resulting beliefs are put into Formula 3.4 to compute the reputation score.

This evidential model uses the mathematical theory of evidence to represent and propagate the ratings about target users. When evaluating the trustworthiness of a user, if direct

[11] Dempster-Shafer theory is a mathematical theory of evidence based on *belief functions* and *plausible reasoning*, which is used to combine separate pieces of information (evidence) to calculate the probability of an event. The theory was developed by Arthur P. Dempster and Glenn Shafer.

information is available then, local direct past experience with the target user is the only source to be considered. The drawback of this model is that it does not combine direct information with witness information. This model works well in distributed reputation systems.

3.1.6 EigenTrust

Sepandar D.Kamvar et al. [26] proposed the EigenTrust model which computes a trust value through the iterative aggregation along with transitive chains until the trust values for all members of the community converge to stable values within a peer-to-peer system.

In order to compute a trust score, a local trust value is calculated first, and then normalized. The local trust s_{ij} is defined as [26]:

$$s_{ij} = sat(i,j) - unsat(i,j) \quad . \tag{3.5}$$

Each peer *i* can store the number of satisfactory transactions it has had with peer *j*, *sat(i,j)* and the number of unsatisfactory transactions it has had with peer *j*, *unsat(i, j)*. A normalized local trust value, $c_{ij,}$ is defined below:

$$c_{ij} = \frac{\max(s_{ij},0)}{\sum_j \max(s_{ij},0)} \quad . \tag{3.6}$$

The Max() function takes the bigger value between s_{ij} and 0. For example, if s_{ij} is smaller than 0, $\max(s_{ij},0)$ will return 0 and c_{ij} will become 0. By using formula 3.6, all negative local trust values are normalized into 0 which is the smallest normalized local trust value. This ensures that all values will be between 0 and 1. If $\sum_j \max(s_{ij}) = 0$, then c_{ij} is not defined.

Local trust values need to be aggregated. The challenge for reputation systems in a distributed environment is how to aggregate the local trust values without a centralized storage and management facility [26]. The local trust is aggregated using the following formula.

$$t_{ik} = \sum_{j} c_{ij} c_{jk} \tag{3.7}$$

Where t_{ik} represents the trust that peer i places in peer k based on asking his/her friends. A distributed EigenTrust algorithm is introduced in this model and shown below [26]:

$$t_i^{(k+1)} = (1-a)\left(c_{1i} t_1^{(k)} + \ldots + c_{ni} t_n^{(k)}\right) + a p_i \ . \tag{3.8}$$

Where a is the probability assigned to different situations, c is the normalized local trust score, t is the aggregated local trust value. If peer i is an inactive peer, s/he either does not have transactions with other peers or assigns a zero score to all other peers, c_{ij} will be undefined and set to p_i. Some set of peers P are known to be trusted. p_i is defined as $1/|P|$ if $i \in P$, and 0 otherwise.

The EigenTrust model addressed some practical issues, such as a priori notions of trust, inactive peers, and malicious collectives. This model presents a distributed eigenTrust algorithm where all peers in the network cooperate to compute and store the global trust vector. The trust or reputation is computed by the transitive iteration through looped or arbitrarily long chains. The sum of the reputation or trust value is not required to be constant.

3.1.7 The PeerTrust Model of Xiong and Liu

Li Xiong et al. [27] present PeerTrust, a reputation-based trust supporting framework, which includes a trust model for comparing the trustworthiness of peers based on a transaction-based feedback system. In peer-to-peer (P2P) online communities, members can communicate directly with one another to exchange information, distribute tasks, or execute transactions [27]. The unique characteristic of the PeerTrust model is the identification of five important factors for evaluating the trustworthiness of a peer in a P2P e-commerce community.

A peer's trustworthiness is defined by an evaluation of the peer on past transactions with other peers. Five important factors for such evaluation are defined as follows [27]: (1)*Feedback in Terms of Amount of Satisfaction*, (2)*Number of Transactions*, (3)*Credibility of Feedback*, (4)*Transaction Context Factor*, and (5)*Community Context Factor*.

By combing these factors, a general trust metric is formalized as follows [27]:

$$T(u) = \alpha * \sum_{i=1}^{I(u)} S(u,i) * Cr(p(u,i)) * TF(u,i) + \beta * CF(u) \quad . \tag{3.9}$$

Let *T(u)* be the trust value of peer *u* during the given period, *I(u)* be the total number of transactions performed by peer *u* during the given period, *p(u,i)* be the other participating peer in peer *u*'s *i*th transaction, *S(u,i)* be the normalized amount of satisfaction peer *u* receives from *p(u,i)* in its *i*th transaction, *Cr(p(u,i))* be the credibility of the feedback submitted by *p(u,i)*, *TF(u,i)* be the adaptive transaction context factor for peer *u*'s *i*th transaction, *CF(u)* be the adaptive community context factor for peer *u* during the given period. The parameters α and β are used to assign different weights to the feedback-based evaluation and community context according to different situations. Li Xiong also provided different ways to calculate the credibility of feedback. One is using the trust value of a peer recursively as its credibility measure. Another one is for a peer *w* to use a personalized similarity measure to rate the credibility of another peer *v* through *w*'s personalized experience.

Peer-to-peer reputation-based trust is a popular trust model. The trust metric of this model is complete and mature compared to other trust models, but it does not provide a way to calculate the adaptive context factor. How to incorporate transaction size, importance, and time slot are not addressed.

3.1.8 Yamamoto's Reputation Model for Online Consumer-to-Consumer (C2C) Markets

Yamamoto et al. [28] introduce a reputation model inspired by an agent-based approach and they describe C2C online transactions within the framework of the Prisoner's Dilemma[12]. The situation in some C2C online transactions can be representative of the Prisoner's Dilemma. For example, in a C2C online transaction (e.g., online auction) there are usually two players/participants who can be cooperative by being honest and truthful or defective by being dishonest. In this model, two possible transaction histories are considered, namely cooperative (i.e., positive) history $|T^i_C,t|$ and defective (i.e., negative) history $|T^i_D, t|$. This model is based on the distinction between positive, negative and hybrid reputation systems as discussed in Section 2.3.2. The reputation of user *i* can be calculated using the following equation [28]:

$$R^i_t = a|T^i_C,t| - (1-a)\,|T^i_D, t|, \text{ where } a \in [0, 1]\ . \qquad \textbf{(3.10)}$$

The value of *a* determines which reputation system is considered: when *a* equals 1, a positive reputation system is considered; when *a* equals 0, a negative reputation system is considered; and when *a* is between 0 and 1, a mixed system between positive and negative systems, also called hybrid system, will be considered. For example, eBay's feedback forum (a hybrid system) uses a value of 0.5 for *a*. In eBay, the reputation of a user is evaluated by the sum of negative feedback and positive feedback.

In general, negative reputation systems are not sufficiently effective in online transactions because of the lack of enforcing unique identies. That will make it hard to distinguish the difference between cooperative and non-cooperative users. This model does not consider certain factors when aggregating feedback, such as time slot and transaction size.

[12] In game theory, the prisoner's dilemma is a type of non-zero-sum game in which two players may each "cooperate" with or "defect" (i.e., betray) the other player. In this game, as in all game theory, the only concern of each individual player ("prisoner") is maximizing his/her own payoff, without any concern for the other player's payoff.[Source: Wikipedia]

3.1.9 Shi's Trust Model with Statistical Foundation

Jianqiang Shi et al. [29] developed a statistical model to compute trust based on self-experience and the recommendations from raters. The authors assume that the space of possible outcomes of transactions is finite (for instance "excellent", "very good", "good" and "bad") and that N transactions have been observed for the same ratee by the querying agent or other raters. Assuming that ratee b will perform in a similar manner in the future, one can predict the probability of the different outcomes for future transactions using the formula [29]:

$$T_b(o) = (\text{number of times that the observed outcome was equal to } o) \,/\, N. \qquad \textbf{(3.11)}$$

$T_b(o)$ is the probability that a future transaction with ratee b will lead to an outcome o. Clearly, the sum of the values $T_b(o)$ over all values of o yields the value one. $T_b(o)$ is also called the "trust that ratee b will provide an outcome o".

An incremental trust update formula can be used instead of keeping in memory all previous transaction outcomes [29]. The current trust $T_b(o)$ (for each value of o) and the number of observations to date are kept in memory, and after a new transaction yielding outcome o was observed, the trust values and N will be updated as follows:

$$\begin{aligned} T_b(o) &= (T_b(o)*N + 1) / (N+1). \\ T_b(o') &= (T_b(o')*N) / (N+1) \text{ for } o' \text{ different from } o. \\ N &= N+1. \end{aligned} \qquad \textbf{(3.12)}$$

The standard error (SE) of experimental outcomes is used in this model to obtain a desirable accuracy and a desirable confidence level with a minimum number of experiences [29]. For an outcome with a score of 0 or 1 for negative and positive (Bernoulli distribution), the SE of the estimated proportion p is given by: $SE = [p(1-p)/N]^{1/2}$ where p is the proportion obtaining a score of 1, and N is the sample observation size. This SE is calculated as the standard deviation of the range of possible estimate values. The SE reaches its maximum value when p = 0.5, so the worst case occurs when 50% ratings are positive and 50% are negative.

This model was extended to discuss the *quality of recommendation* using Bayesian statistics and Weighted Majority Algorithm in [30]. This trust model is based on a statistical foundation which is intuitive and useful in many practical situations. The action of interest usually cannot be predicted exactly, so a stochastic process can better describe the behaviors of the target user. Therefore, statistical foundation certainly adds value to this model when evaluating the trust of the target user.

3.1.10 Song's FuzzyTrust Model

Shanshan Song and et al. [31] designed the FuzzyTrust prototype system which uses *fuzzy-logic inference* rules for evaluating peer reputation in P2P transactions. The reputation is calculated by performing two major inference steps: *local-score calculation* and *global reputation aggregation*. (a) performs fuzzy logic inferences to determine the local trust scores and (b) uses accumulated local scores for weight inference in global reputation aggregation (See Figure 5).

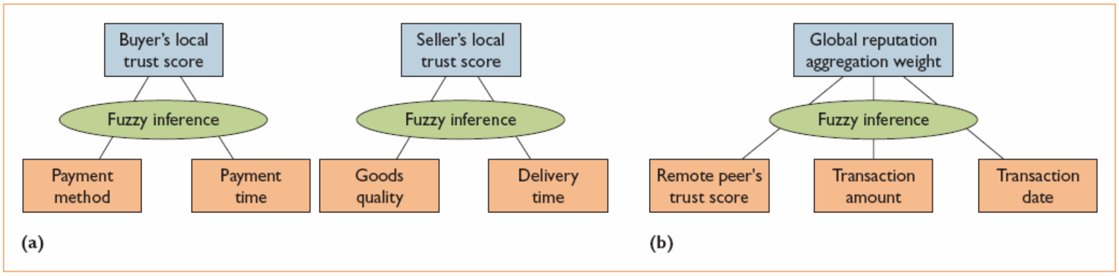


Figure 5: The FuzzyTrust system [31]

In FuzzyTrust, peers perform *fuzzy inference* on local parameters to generate the local scores. There are some local parameters, such as payment method and time, goods quality, delivery time and so on [31]. After local trust scores from all peers are calculated, the FuzzyTrust system aggregates them to produce a global reputation for each peer. The aggregation weights are determined using the peer's reputation, the transaction date and the transaction amount. Five frequently used fuzzy inference rules are listed below [31]:

(1) If the transaction amount is very high and the transaction time is new (i.e., recent), then the aggregation weight is very large.

(2) If the transaction amount is very low or the transaction time is very old, then the aggregation weight is small.

(3) If a peer's reputation is good and the transaction amount is high, then the aggregation weight is very large.

(4) If a peer's reputation is good and the transaction amount is low, then the aggregation weight is medium.

(5) If a peer's reputation is bad, then the aggregation weight is very small.

The formula to calculate the global reputation is [31]:

$$R_i = \sum_{j \in S} \left[\frac{w_j}{\sum_{j \in S} w_j} t_{ji} \right] = \frac{\sum_{j \in S} w_j t_{ji}}{\sum_{j \in S} w_j} \qquad \textbf{(3.13)}$$

where R_i is the global reputation of peer i, S is the set of peers with whom peer i has conducted transactions, t_{ji} is the local trust score of peer i rated by peer j, and w_j is the aggregation weight of t_{ji} which is inferred through the fuzzy inference rules.

The advantage of the FuzzyTrust model is the ability to handle imprecise or uncertain information collected from the raters adaptively in order to design an efficient reputation aggregation system. However, the model cannot specify local parameters, such as the transaction amount, the rater's trust score, the transaction date and the aggregation weight quantitatively. This would lead to the inaccuracy of the reputation value.

3.2 Classification

Trust and reputation can be analyzed and classified from many different perspectives. In this section we propose a set of aspects by which we classify the current computational trust and reputation models in a clear manner. The focus of this thesis is on computational models; therefore, the classification aspects have been selected considering the special characteristics of these models, such as computational principle, parameters of computational formalisms, *information sources*, reputation scope and *model's granularity* [32]. The following paragraphs explain these aspects and present the classification summary (see Table 2).

3.2.1 Computational Principle

Based on various computational principles, reputation models can be classified into summation or average, discrete or continuous, the credibility of rater and underlying models. These models are explained in the following.

3.2.1.1 Summation or Average of Ratings

Summation is a simple calculation method which sums all feedback values together, no matter whether positive, neutral or negative. This is the principle used in eBay's feedback forum. The advantage here is that users can easily understand the calculation principle; however, this feedback-summation-only metric is flawed [11, 16, 27]. As an illustration, a user who has 10 positive feedback points out of 10 transactions would have the same reputation value as a user who has 20 positive feedback points and 10 negative feedback points out of 30 transactions. In fact, the first user has a 100 percent positive feedback and the other has only a 67 percent positive feedback. According to Dellarocas [33], binary (positive, negative) reputation mechanisms are not known to function well.

Average rating is more accurate compared to simple *summation* since the number of transactions is taken into consideration to avoid the situation where users increase their reputation by simply increasing their transaction volume. A more precise way is to use a *weighted average of ratings*, where the weight of a rating can be determined by attributes such as the rater's credibility for rating others, the size of the transaction (e.g., its monetary value), the time of the transaction, etc. [13, 16, 27]. Evidently, the ratings provided by users with higher credibility should be weighted more than those from users with lower credibility. In addition, transactions may differ from one another based on the transaction context. For instance, the ratings associated with high value transactions should be weighted more than those associated with low value ones and more recent ratings should be weighted more than older ratings.

Average of ratings represents the best estimation of past ratings about a ratee; however, each rating could have an error associated with it. It is necessary to know the expected error of the estimation. As described in Section 3.1.9, the expected error can be computed as the standard deviation of the set of measurement values.

3.2.1.2 Discrete or Continuous Trust Models

Raters often provide *discrete* feedback instead of *continuous* one when the qualitative trust representation is sufficient to show the trustworthiness of a ratee. In Manchala's trust model [34], for instance, a transaction is evaluated as "excellent", "good", "normal", "bad" and "worst", and the transaction value (i.e., size) is evaluated as "micro", "small", "medium", "high" and "expensive". In Abdul-Rahman and Hailes' model [3], the trust-worthiness of an agent (rater or ratee) is evaluated as "very trustworthy", "trustworthy", "untrustworthy" and "very untrustworthy". Shi's statistical model [29] takes discrete outcomes such as "good" and "bad" as inputs to calculate the probability of a certain outcome using Bayesian theory. The *a posteriori* (i.e., the updated) reputation value is computed by aggregating the *a priori* (i.e., previous) reputation value with new ratings [11]. It is worth mentioning that one shortcoming of *discrete models* is that they are not as precise as *continuous models* since "*heuristics mechanisms like lookup tables must be used*" [11] to convert feedback values into their numeric equivalent. Discrete models are not rich enough to specify more subtle differences in characteristics. For example, an agent X is judged to be "Trustworthy" and another agent Y is more trustworthy, but not enough to be "Very Trustworthy". What trust degree should we give to Y? Lookup tables depend on agents' own judgment; therefore they are not precise.

Trust can also be represented in a quantitative manner by a *continuous* real value within lower and upper bounds [29]. Marsh [7] uses a continuous value over the range $[-1,1)$ to represent trust, where -1 is the lowest trust value and 1 is the highest trust value. It is clear that the higher the trust value, the more trustworthy the ratee. However, continuous representation of trust could bring ambiguity to models. The use of continuous values is very subjective to each rater. For example, we would not be sure about whether a trust value of 0.6 is "high", "low" or "average". By averaging the discrete ratings of a ratee, his/her reputation becomes a continuous value.

3.2.1.3 The Credibility of Raters

In trust systems, the quality of recommendation is not guaranteed, since malicious raters could give unfair recommendations. As stated in [15, 16, 27, 31, 35], feedback from peers with higher credibility should be weighted more than feedback from those with lower credibility since raters with lower credibility are more likely to submit dishonest or misleading feedback.

Some models ignore this problem. Some models manage the credibility information of raters, that is, they take the rater's credibility into account when calculating the average reputation. Jianqiang Shi et al. [30] for instance use data analysis and machine learning techniques to detect unfair recommendations. In the PeerTrust system [27], one solution to this problem is using a function of their reputation within the community, therefore reputable raters are considered more credible, thus their ratings are weighted more. A model that considers the raters' credibility is more reliable than one that does not.

Semantic Distance can be used to evaluate the trustworthiness of a rater, as described in Section 3.1.3. *Similarity* is another form of *Semantic Distance.* The credibility of a rater can be defined as the feedback similarity between this rater and another rater over a common group of ratees with whom they have interacted before as described in the Peer-Trust model [27].

*Flow model*s also provide a solution to this problem. *Flow models* are systems that compute trust or reputation by transitive interaction through looped or arbitrarily long chains. Some flow models assume a constant reputation weight for the whole community, so the members of community can only increase their reputation by decreasing those of others [11]. However, flow models do not always require the sum of reputation values to be constant. Along the transitive chains, these values need to converge to stable ones [11].

3.2.1.4 Underlying Models

By underlying models we mean the methods (mathematical, statistical, etc.) for representing reputation within online reputation systems such as *Belief models*, *Fuzzy models* and *Statistical models*.

Belief models use belief theory which is a method related to probability theory[13], but where the sum of probabilities over all possible outcomes does not necessarily add up to one, and the remaining probability is interpreted as uncertainty [11]. Usually a belief or trust metric, which can express one's belief in trust, need to be determined first. For example, in Yu's model [25], separate beliefs about whether a user is trustworthy (T_A) or untrustworthy(T_A) are kept.

In *fuzzy models*, trust and reputation can be represented as linguistically fuzzy concepts. *Fuzzy logic* is needed to provide rules to describe the degree of a user's trustworthiness. For example, considering the quality of product and delivery time for a seller, one can set a fuzzy logic as follows: a seller is good if the quality of product is good and the delivery time is fast; and a seller is bad if the quality of product is bad or the delivery time is slow. Fuzzy models have the ability to handle uncertainty, fuzziness, and incomplete information.

[13] Probability theory is the branch of mathematics concerned with analysis of random phenomena.

In *Statistical models,* trust and reputation can be computed and updated using statistical methods. Average models as described in Section 3.2.1.1 belong to this category, since average models use the mean of past ratings to predict the future behavior of a ratee. Bayesian models are statistical models too. They compute reputation values by statistical updating of beta probability density functions (PDF)[14]. They take binary ratings as input (i.e., positive or negative) and put these inputs in the form of the beta PDF parameter tuple (α, β) where α represents the amount of positive rating and β represents the amount of negative rating. How does the updating of reputation score work? The main idea is that the *a posteriori* (i.e., the updated) reputation value is computed by aggregating the *a priori* (i.e., previous) reputation value with new ratings [11]. Bayesian models provide a theoretical basis for computing reputation scores, but Bayesian theory may be too complex for people to understand. Jianqiang Shi et al. [29] build trust by estimating the probabilistic future behaviour of a ratee from past experience in their statistical trust model. Bayesian models are precise models, but the standard deviation is still a good approximation to calculate the expected error.

3.2.2 Parameters of Computational Formalisms

There are some parameters that need to be considered while calculating reputation values. These parameters include *rating attributes* such as *reputation values*, *credibility of raters*, and *context factor of transactions*. Further, the number of transactions can be included to compute average the ratings of the ratees. Different computational models may require different types of parameters. For example, in Yamamoto et al.'s model [28], the number of cooperative and non-cooperative actions are used instead of real reputation values. The different parameters are detailed below.

[14] In mathematics, a probability density function (pdf) is a function that represents a probability distribution in terms of integrals.

(1) *Feedback Value:* This is an essential parameter in reputation models - also called rating, trust value, feedback and recommendation. This value is typically given by a rater as feedback on the ratee for a single transaction. It reflects how well this peer performs during the online interaction. Reputation systems can differ in their feedback representation formats, which could be discrete or continuous; numerical, textual, or both. Furthermore, some systems use feedback values alone to aggregate a user's reputation without considering any other attributes (e.g., eBay's feedback forum only sums up the feedback values).

(2) *Information on Rater Credibility*: As described in Section 3.2.1.3, evaluating the credibility of a rater is important for calculating the reputation of a ratee, but determining the credibility of a rater is a challenge.

(3) *Context factors*: Transactions may be different from one another. Various transaction attributes such as the size and time of the transaction can be considered; thereby the feedback for larger and more recent transactions can be assigned more weight. More recent transactions are more likely to reflect the current behavior of the ratee [13, 15, 31]. Also, the size of the transaction [31] can be considered in order to avoid the situation where a user behaves honestly for small transactions and dishonestly for large transactions.

(4) *Total Number of Transactions*: This parameter 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 as described in Section 3.2.1.1.

3.2.3 Information Sources

Trust and reputation, derived from the opinions about a target agent, can be classified based on the origin of these opinions. The study of Sara Casare et al. [17] emphasizes that beliefs can be obtained from several sources, such as *direct experiences*, *received information* and s*ocial prejudice*. Further, those sources can be classified into *primary sources* and *secondary sources* based on whether the information is obtained by direct and indirect interactions, as discussed in Section 2.2.3. According to Jordi Sabater et al. [32], *direct experiences* and *witness information* are the traditional information sources used by current computational trust and reputation models. It is more reliable to take account of both *direct experiences* and *witness information*; however it will make computation models more complex.

(1) *Primary Sources*: *Primary information* is obtained from a *primary source,* also called *direct experience*. This is the most relevant and reliable information source for reputation systems. The *primary source* of information not only corresponds to those agents who have direct interaction with their partners, but also those who observe the actions of others [17]. Lik Mui [36] and Jordi Sabater et al. [32] also distinguish two types of *direct experiences*. A common one is based on the direct interaction with the partner, and another type is based on the observed interaction of other members of the community.

(2) *Secondary Source* : *secondary information* can be gathered indirectly from a *secondary source*, also called *second-hand evidence* [36]. Moreover, as mentioned in [32], *witness information*, which is also called *word-of-mouth* or *indirect information*, is the information that comes from other members of the community. This information can be based either on their direct experiences or it can be gathered from others. Witness information is abundant compared with direct information, but it is complex for models to use.

3.2.4 Reputation Scope

Reputation could be distinguished according to the manner in which reputation information is employed, in a *local* or a *global* way, called *reputation scope* in [17]. According to [32], trust and reputation can either be seen as a *global property* shared by all the observers in the community or as a *subjective property* maintained as an individual belief. In a global scope, all members in the community, who have past experiences with a target agent, can contribute to form the reputation value of that agent. These values are publicly available and unique to all members in the community and updated every time a group member provides a new evaluation. In a subjective scope, each member in the community evaluates a target agent independently, so the reputation value will be assigned a personalized value according to agents' different points of view. Different members may have different direct experiences or witness information with the target agent. In this case, we cannot say the reputation value of an individual *x*; however, we need to say the reputation value of an individual *x* from the point of view of *y*.

Most online reputation management systems take reputation values as a global property. For example, eBay's *feedback forum* provides *feedback profiles* of sellers/buyers publicly to the users. The shortcoming of taking reputation values as a global property is that these values lack personalization. Models usually set a standard evaluation metric when treating these values in a global context; however, different individuals may have different degrees of perception. Sometimes, what is bad for me could be acceptable for others. For example, late delivery would be OK for me, but not acceptable for those who need the product to be delivered on time. The global reputation values are not helpful for users to solve subjective affairs.

For the later case, in a subjective scope, each agent uses its personal experiences and information from personal inferences to build the reputation values of each member in the community. These models are suitable for medium and small size environments where it is possible to establish strong links among members [32].

3.2.5 Model's Granularity

According to Jordi Sabater et al. [32], the model's granularity refers to whether the reputation value is context dependent or not. As mentioned in Section 2.2.3, an individual may have a very high reputation in one domain, while having a low reputation in another. For example, we could trust a chef to recommend a delicious dish, but we may not trust him/her to suggest a medicine. For that reason, the reputation value will be more meaningful with a certain context added. However, adding the ability to deal with multi-context to models would result in high cost in terms of complexity. In a single-context model, a single reputation value is assigned to each transaction partner without taking account of a specific context. In a multi-context model, different reputation values are associated to several contexts at a time for a single partner. Most of models consider reputation values as one-dimensional, while there are few of them that pay attention to the multi-contextual characteristic of reputation.

3.2.6 Classification Summary

Earlier in this chapter, we reviewed the related work on a selection of important computation models on trust and reputation and emphasized their characteristics. We introduced some classification categories earlier, such as computational principle, parameters of computational formalisms, information sources, reputation scope and model's granularity which are used to classify these models. As shown in Table 2, the most widely used computational principle is the weighted average of ratings. We also can conclude that most of models we investigated are not context-dependent. Moreover, we note that the reputation values are usually computed based on primary information sources. Secondary information would help to obtain more accurate reputation values, but it is usually discarded because of the complexity of the models. Please note that reputation models consider reputation as subjective or global, so we treat the reputation scope as a characteristic of the reputation models. The abbreviations used in the Table 2 are presented in Table 1 below.

Computation Model	S	Summation
	A	Average of Ratings
	D	Discrete
	C	Continuous
	NRC	No Rater Credibility
	MCI	Manage Credibility Information
	SD	Semantic Distance
	F	Flow
	B	Belief
	Fu	Fuzzy
	St	Statistical
Information Source	P	Primary
	S	Secondary

Table 1: The Abbreviated Notation

Models	Computational Principle	Parameters	Information Sources	Reputation Scope	Model’s Granularity (Context-dependent)
S.Marsh [7]	A, C,MCI and St	utility querying agent gains from the situation, importance of the situation, estimate of general trust	P	Subjective	Yes
Sporas [15]	A, C, MCI and St	Ratings, reputation of user giving the rating, number of ratings, range of reputation value	S	Global	No
Histos [15]	C and F	Ratings, reputation of user giving the rating, number of connected paths (number of transactions?)	P and S	Subjective	No
A.Rahman & Hailes [3]	D and SD	Recommended trust degree (trust value), recommender trust degree, recommender weight	P, S	Subjective	Yes
Malaga [16]	A, C,MCI and St	Reputation value, weight of rater, starting reputation score, weight for rating others, number of transaction	P	Global	No
Yu & Singh [25]	C, F and B	Past history of behaviours as trustworthy and untrustworthy	P, S	Subjective	No
Kamvar et al. [26]	C and F,	Local trust score, trust score for pre-trust peer	P	Global	No
Xiong & Liu [27]	A, C,SD and St	Reputation value, credibility of feedback, transaction context factor(size, time, importance), community context factor(weight of rating others), number of transaction	P	Global	No
Yamamoto et al. [28]	S, D and NRC	Cooperative and non-cooperative actions	P, S	Global	No
Shi, Bochmann & Adams [29, 30]	A, D, SD [30] and St	Number of observations	P S [30]	Subjective	Yes
Song et al. [31]	A, Fu, C,MCI and St	Local trust score, fuzzy inference for local aggregation weights, number of transactions.	P	Global	No

Table 2: A Selection of Computational Models Comparisons

Chapter 4
Model of an Online Reputation Aggregation System (ORAS)

In addition to the study and classification of major reputation models in the literature, the second purpose of this thesis is to design and develop a configurable reputation aggregation system and its computation model to aggregate reputation data for a specific ratee from various online communities.

Basically, we can identify three types of entities involved in a reputation model: (1) the *querying agent*, who is the user who wants to know whether a given user (the ratee) can be trusted; (2) the *ratee*, who is rated by others on his/her past behaviour; and (3) the *rater*, also called recommender, who is the user who provides information about the ratee, usually after having had some transactions with the ratee. In trust models, raters provide recommendations to the querying agent about a ratee according to certain criteria.

4.1 Assumptions

In order to make our design practical, it is necessary to make the following two assumptions.

(1) *Unique Identity*: A user's identity is long lived and unique when using our system. In real life, user identities are typically different from one online community to another making it hard or impossible to track a user's reputation across various communities. Validating the user's online identity becomes a huge issue for online reputation systems. Until now online businesses have been trying to provide solutions to solve this

issue such as Opinity, OpenID[15] for blogs, ClaimID[16](OpenID enabled), Identity 2.0[17], Microsoft's CardSpace[18], Liberty Alliance[19], etc. Since the focus of this thesis is on the computational model of reputation, we assume that users utilize unique identities across all communities when using our system. Moreover, we assume that it is hard or impossible for users to erase their reputation information. We believe that our assumption is realistic in view of the efforts being spent to devise a solution that links user identity with a unique biometric[20] feature using biometric systems[21] such as fingerprint sensors.

(2) *Available Reputation Data through Web Services*: We assume that our system could get an individual's (i.e., ratee's) raw reputation data from all corresponding online communities by discovering their corresponding web services, binding to them, requesting the reputation data, and obtaining it through the Internet without human intervention. Again, this is a realistic assumption knowing that big online businesses such as Amazon.com have been making some of their popular services accessible to distant software applications through web services. EBay has already given access to its data for research purposes. It is also realistic to assume that online communities may charge for granting access to their reputation data.

4.2 Design of the Computation Model

After reviewing the existing trust or reputation models, we decided to propose a discrete reputation model in which the reputation is calculated as the weighted average of obtained

[15] www.openID.net

[16] www.claimID.com

[17] http://identity20.com/

[18] http://cardspace.netfx3.com/content/introduction.aspx

[19] http://www.projectliberty.org/

[20] A measurable physical characteristic or personal behavioral trait used to recognize the identity of an enrollee or verify a claimed identity. Aug. 2007. Retrieved from http://www.eliminatepasswords.com/www/BioTerms.html

[21] An automated system capable of capturing a biometric sample from an end user, extracting biometric data from the sample, comparing the data with one or more reference templates, deciding on how well they match, and indicating whether or not an identification or verification of identity has been achieved. Aug. 2007. Retrieved from http://www.eliminatepasswords.com/www/BioTerms.html

ratings taking the credibility information into account. Reputation values in our model are used globally, not in a context-dependent manner. The computation is performed in two phases: (1) aggregate the local reputation based on the raw reputation data for every online community to which the ratee belongs; (2) compute the global reputation based on all local reputations calculated in Step (1). This computation model for reputation aggregation is described in our working paper [37] and will be explained in the following section. Please note that online communities would have different formats of feedback values and their computational models would be different.

4.2.1 Local Reputation Aggregation

As mentioned earlier, a ratee's local reputation is linked to one single community (e.g., the reputation maintained for a seller on eBay is considered local to the eBay community). If a community maintains an online reputation system, then the ratee is rated every time s/he transacts within that community. Note that we are not interested in the aggregated reputation value as provided by the community's reputation system but rather in the "*attributes*" of all ratings which form the raw reputation data. According to assumption (2) in Section 4.1 above, all raw reputation data is from the corresponding online communities. Online communities may charge a fee for granting access to their reputation data. We assume that the reputation system keeps the following attributes about each and every rating: (1) the feedback value provided by the rater, (2) information on rater credibility, and (3) the context attributes which could include the time of the transaction, and the size (e.g., the monetary value) of the transaction. Moreover, the total number of transactions is an important parameter to consider. This raw data is transferred to our system, and then aggregated into a local reputation value for the ratee within the corresponding community using the computation model described later in this section.

Normalization of Reputation Inputs

In order to apply our computation model, the attributes that serve in the aggregation need to be normalized first. Reputation systems maintained by different communities use different formats to represent these attributes. In order to aggregate them, it is necessary to normalize them into the same scale of numerical values using mapping tables (see an example in Table 4) or conversion formulas as proposed in [26]. Most models normalize feedback values between 0 and 1. The reason we choose this manner is as follows: (1) a 0 to 1 value is good enough to reflect the real reputation situation; (2) reputation values should be positive; (3) normalization needs to follow one rule which is that each feedback should have monotone increase weights. For example, on eBay, normalizing feedback values -1, 0, 1 within the range [0, 1] would yield the numerical values 1, 0.5 and 0 or 1, 0.5 and 0.25, respectively. Reputation values, which are numerical values greater than zero, can be normalized as follows:

$$f_{ik} = \frac{feedback}{f_{\max}} \tag{4.1}$$

Where $f_{\max}$ corresponds to the equivalent value of full satisfaction in the given reputation system. For example, in Amazon.com auctions, a value from zero to five is used. If a user gets 4.5 as feedback, the normalized feedback value will be 0.9 (4.5 / 5 = 0.9). Word expression feedback format can also be represented using numerical values. For instance, *good*, *neutral*, and *bad* can be transformed to 1, 0.5 and 0, while *excellent*, *good*, *average*, *bad* and *very bad* can be normalized into 1, 0.75, 0.5, 0.25, and 0.

In our model, the credibility of a rater is obtained from external sources where the credibility of a rater has already been determined. We do not care about how the credibility of the rater is computed. After we get the credibility of a rater, we will normalize it into the 0 to 1 scale.

Furthermore, a time window should be determined before we collect the feedback information. Since more recent transactions are more likely to reflect the current behavior of the ratee, the recent feedback would be rated higher. We want to measure how recent the feedback is. The data we receive is time stamped. For example, eBay keeps the date and time of transactions. In our system, the starting point for the time window is determined by the querying agent and the end point is the present time. The time factor is calculated using the formula below:

$$T_{ik} = \frac{D_{ik}}{D_{timewindow}} \tag{4.2}$$

where D_{ik} is the number of days from the starting point of the time window to the date of the rated transaction, and $D_{timewindow}$ is the number of days from the starting point of the time window to the present.

For example, *i* is rated on Oct 07, 2005. The starting point is Jan. 01, 2000 and the present time is May 3, 2007. The time context factor is calculated as follows:

$$T_{ik} = \frac{(5-0)*365+(10-1)*30+(7-1)}{(7-0)*365+(5-1)*30+(3-1)} = \frac{2101}{2677} = 0.78 \ . \tag{4.3}$$

As mentioned earlier, the feedback for larger value transactions can be assigned more weight than those for lower value transactions. The weights for different sizes are stored in the importance lookup table. This will be discussed further in Section 5.1.

Local Aggregation

We decided to follow an approach inspired by Jianqiang Shi et al. [29]. In order to represent discrete reputation better, they propose a stochastic trust model which is based on the assumption that the ratee behaves like a stochastic process, and the reputation value represents the expectation that the ratee will act accordingly in the future (as explained in Section 3.1.9). Our approach calculates the estimated probability of each possible distinct outcome (e.g. “Positive”, “Neutral”, or “Negative”; “Excellent”, “Good”, “Bad”, or “Very Bad”) for the action of the ratee taking into account the different rating attributes introduced earlier (see Formulas 4.4, 4.5, 4.6). We then take the average of these values together with the corresponding numerical value (representing that outcome) (see Formula 4.7). The aggregated reputation of ratee *i* denoted by R_i is calculated using the following formulas:

$$P_i(o) = \sum_{k=1, f_{ik}=o}^{I(i)} \frac{W_{ik}}{\sum_{m=1}^{I(i)} W_{im}} \tag{4.4}$$

$$W_{ik} = CR_{ik} * CF_{ik} \tag{4.5}$$

$$CF_{ik} = \frac{a}{a+b} * T_{ik} + \frac{b}{a+b} * S_{ik} \qquad a, b \in [0,1]. \tag{4.6}$$

$$R_i = \sum_{o \in O} P_i(o) * NumVal(o) \tag{4.7}$$

where

$P_i(o)$ = the estimated probability that ratee *i* will provide the outcome *o* in the future

O = the set of possible outcomes, such as “excellent”, “good”, “average”, “bad”, and “very bad”

I(i) = the total number of transactions

f_{ik} = 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(o) = the numerical value corresponding to the outcome *o* (table lookup)

Note that the context factor is determined by both transaction time and size. Our model sums them together with weights *a* and *b.* The “configurable” system allows the querying agent to enter values for *a* and *b*. For example, if the querying agent only wants to consider transaction time (size) as a context factor, then he/she would assign 1 to *a (b)*; and 0 to *b (a)*. Note that the context factor cannot be 0. If the querying agent wants to consider transaction time and size as equally important, then he/she would assign 1 to *a* and 1 to *b*.

Example of Local Aggregation

For an illustration, consider the example of a ratee *i* on the eBay community who has been rated 10 times (i.e., I(i) = 10) possibly more than once by the same rater. Table 3 shows the 10 feedback values f_{ik}, and the aggregation weights W_{ik} for each feedback value. Table 4 shows the mapping table used to convert discrete feedback values into numerical values.

k	f_{ik}	W_{ik}
1	Positive (1)	1
2	Neutral (0)	0.5
3	Negative (-1)	0.5
4	Positive (1)	1
5	Neutral (0)	1
6	Negative (-1)	1
7	Positive (1)	1
8	Positive (1)	0.7
9	Neutral (0)	1
10	Positive (1)	0.3

Table 3: Feedback values and aggregation

Discrete feedback	***Numerical values***
Positive (1)	1
Neutral (0)	0.5
Negative (-1)	0

Table 4: EBay's feedback mapping table

The estimated probability that a ratee will behave as "Positive", "Neutral" or "Negative" in future transactions can be calculated as follows:

$$P_i(Positive) = \sum_{k=1, f_{ik}=Positive}^{10} \frac{W_{ik}}{\sum_{m=1}^{10} W_{im}} = \frac{1+1+1+0.7+0.3}{8} = \frac{1}{2} \quad \textbf{(4.8)}$$

$$P_i(Neutral) = \sum_{k=1, f_{ik}=Neutral}^{10} \frac{W_{ik}}{\sum_{m=1}^{10} W_{im}} = \frac{0.5+1+1}{8} = \frac{5}{16} \quad \textbf{(4.9)}$$

$$P_i(Negative) = \sum_{k=1, f_{ik}=Negative}^{10} \frac{W_{ik}}{\sum_{m=1}^{10} W_{im}} = \frac{0.5+1}{8} = \frac{3}{16} \quad \textbf{(4.10)}$$

These probabilities would give the querying agent a better view about the ratee's future action. The local reputation of ratee *i* then has a value of 0.65625 as calculated below:

$$R_i = \sum_{o \in O} P_i(o) * NumVal(o) = \frac{1}{2} * 1 + \frac{5}{16} * 0.5 + \frac{3}{16} * 0 = \frac{21}{32} = 0.65625 \quad \textbf{(4.11)}$$

4.2.2 Global Reputation Aggregation

After the local reputations have been calculated for every online community (see Section 4.2.1), they need to be aggregated to produce the ratee's global reputation across the communities where he/she has transacted. Our system allows querying agents to decide which communities to include in the calculation as well as their corresponding weights (global aggregation weights). As one "configurable" feature, querying agents can enter this information through the user interface. For example, if the querying agent considers the eBay community to be more reliable, then he/she can assign a large weight to its local reputation. Moreover, the querying agent can discard the communities he/she does not consider to be reliable. The global reputation (GR_i) is calculated as follows:

$$GR_i = \sum_{j=1}^{I(j)} R_{ij} * \frac{W_j}{\sum_{m=1}^{I(j)} W_m} \quad \textbf{(4.12)}$$

where R_{ij} is the local reputation for ratee *i* within community *j* as computed in 4.2.1, W_j is the aggregation weight for community *j,* and *I(j)* = the number of communities considered.

After the querying agent transacts with the ratee (assuming the local and global reputation values were sufficiently encouraging to the querying agent to engage in a transaction with the ratee), he/she has the option to use the system to provide his/her own feedback on the ratee. Based on this feedback, the system can upgrade or downgrade the aggregation weights of the various communities accordingly. For instance, if the local reputation from the eBay community is similar to the feedback entered by the querying agent, then the system assigns a larger weight to the eBay community for future sessions. This is the "semantic distance" concept as discussed in [30]. The same illustrative example given in Section 3.1.9 could apply here by replacing ratee *i* with community *j*.

Chapter 5

Implementation and Results

This thesis is part of a bigger project dealing with portable reputation. My colleagues Duan and Zhang [38] have implemented the *collection module* used to obtain raw reputation data from *external databases* through web services, the *mapping module* to normalize the reputation information and the *administrator interface* as part of a portable online reputation system. We have designed the architecture of our Online Reputation Aggregation System (ORAS) and implemented the *aggregation module* (see Figure 6) used to calculate "local" reputation values for each online community using the reputation data obtained from local *reputation information database* and to compute an aggregated "global" reputation. We have also implemented the *user interface* to collect the personal preferences of querying agents, such as the aggregation weights for each online community, "attributes" of ratings and their weights.

In this chapter, a prototype implementation of the ORAS computation model is described and sample results are shown as screenshots. The user interface is also implemented to illustrate the "configurable" characteristic of ORAS. Every online community manages its own reputation system and makes it available to its users in two ways: (1) the usual way via a web browser; (2) and through a web services interface that enables software applications to access the reputation data contained in their databases. Online communities describe their reputation-related web services using WSDL (Web Service Description Language) and publish them. The system discovers them via a UDDI (Universal Description, Discovery and Integration) registry. The communication between ORAS and the web services is done with the SOAP (Simple Object Access Protocol) protocol. The web services in question are maintained by the online communities on their server(s) and are connected to their backend applications such as database servers.

5.1 The Architecture Design

In the following, we are going to describe the general architecture of our ORAS. As described in Section 3.2.4, most reputation models lack the personalization of the reputation value. It is assumed that something is good for you if it is good for me. We note that ORAS is a “configurable” system, which means that it gives querying agents the opportunity to be involved by letting them choose the online communities, the rating attributes and their weights. ORAS is composed of the following main components, as shown in Figure 7:

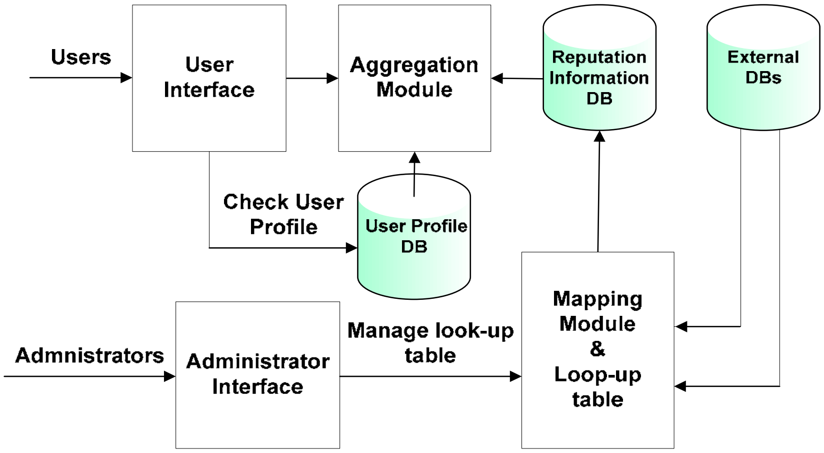


Figure 6: The Architecture of ORAS

User Interface

In its simplest form, the User Interface is used by querying agents to register, enter the identity of the ratee to be looked up, select (from a list returned by ORAS) the online communities they would like to consider for the ratee in question, enter the aggregation weight for each community, the aggregation weights for the rating attributes.

Administrator Interface

ORAS can be managed through the Administrator Interface. The Administrator role can be used, for instance, to modify Lookup Tables used by the Mapping Module, to update conversion parameters, add or remove online reputation sources to or from the reputation source list, etc.

Aggregation Module

The Aggregation Module implements the algorithm for computing reputation values. It aggregates the reputation data received from the Mapping Module and displays the results to the querying users in an understandable format.

Mapping Module

The Mapping Module normalizes raw reputation data gathered from different communities into a common form using Lookup Tables. Reputation systems differ from each other in their feedback format. For example, some reputation systems use a summation mechanism on a scale of positive, negative and neutral, while others use an average rating mechanism on a scale of 1-5. The Mapping Module standardizes these rating scales and other evaluation factors such as the credibility of the rater, time, and size factor into one universal mechanism with a scale of 0 to 1, as explained in Section 4.2.1.

Lookup Table for the weight of the size attribute

Lookup tables store the scales used to normalize aggregation inputs such as the size lookup table. The system administrator has the ability to change the scale of lookup tables. As an illustration, a size scale lookup table, which stores the weight of the size attribute, can look something like the following table:

Size	≤10 $	10 <a≤100$	100<a≤500$	500<a≤1000$	> >1000$
Weight	0.4	0.6	0.8	0.9	1

Table 5: An example of size lookup table

Lookup tables can be modified by the administrator from time to time to better reflect the reality of the moment. For example, since product prices usually increase year after year, the corresponding lookup table needs to change consequently.

Reputation Information Database

There is one relation stored in this database, which is the table of transaction details. Each row in the table stores the detailed information about one transaction. Each transaction has the following attributes: ratee’s username, feedback value, the credibility of the rater, the transaction time, transaction size and reputation source. The *ratee’s username*, which is input by the user when he/she is querying our system, identifies the target user we are going to evaluate. The *Feedback value* is the normalized amount of satisfaction a ratee obtained from a rater based on this transaction as described in Section 3.2.1. The *credibility of the rater* is obtained from the online community. The *reputation source* shows that detailed transaction information is in this online community. The credibility of the rater, transaction time and size are useful attributes as mentioned in Section 3.2.1.

5.2 The Development Environment of the Implementation

In order to implement the computation model of ORAS, we used NetBean IDE 5.5 as our Integrated Development Environment, MySQL 5.0 to build a reputation information database and Apache Tomcat 5.5 as our web server.

NetBean IDE 5.5[22] is a free, open-source Integrated Development Environment (IDE) for the Java programming language, a modular, standards-based application platform used to build any kind of applications. NetBeans IDE 5.5 can maximize productivity for building robust, standard-compliant Java applications including professional cross-platform Java desktop, enterprise, web and mobile applications.

MySQL[23] is a popular open source, free Database Management System that offers reliable, cross-platform performance for most database needs. MySQL 5.0 is a fast, multi-threaded, multi-user and robust SQL (Structured Query Language) database server. It is available for Linux, Windows, Solaris, Mac OS X, FreeBSD, HP-UX, IBM AIX 5L and a range of other operating systems.

[22] www.netbeans.org/index.html

[23] http://www.mysql.com/

Apache Tomcat [24] is a web container or application server developed at the Apache Software Foundation (ASF), used in the implementation for the Java Servlet and JavaServer Pages (JSP) technologies developed by Sun Microsystems. Tomcat provides an environment for Java code to run in cooperation with a web server. It powers numerous web applications in various areas.

5.3 Implementation of the Computation Model

Using NetBeans IDE 5.5, we created a Java Application Programming Interface (API), *repuAggregator*, to calculate the local reputation for each community. The method *LocalRep()* in *repuAggregator* takes some parameters including the *ratee's username*, the values of check boxes for the credibility of the rater and transaction context, *the weights for transaction time* and *size* and the *reputation source*. If the value of the check box for the credibility of rater is "ON", we need to consider it during the calculation process. Similarly, we need to consider the transaction context (i.e., time and size) and their weights, if the value of its check box is "ON".

A Java class *getRecord* is created for obtaining the normalized reputation information of a specific ratee from the *reputation information database*. In class *getRecord,* there is a method *getTransInfo()* used to achieve the connection to the database. This method is invoked by the method *LocalRep()* in class *repuAggregator* and takes ratee's username and reputation source as parameters.

Another Java class, *TransDetail*, is responsible for temporarily storing the returned transaction data corresponding to the structure of transaction detail table in *reputation information database*, The attributes of this class include ratee's username, feedback value, the credibility of rater, transaction time, transaction size and reputation source. *Set* and *Get* methods for each attribute are also included in this class. The global reputation is calcu-

[24] http://tomcat.apache.org/

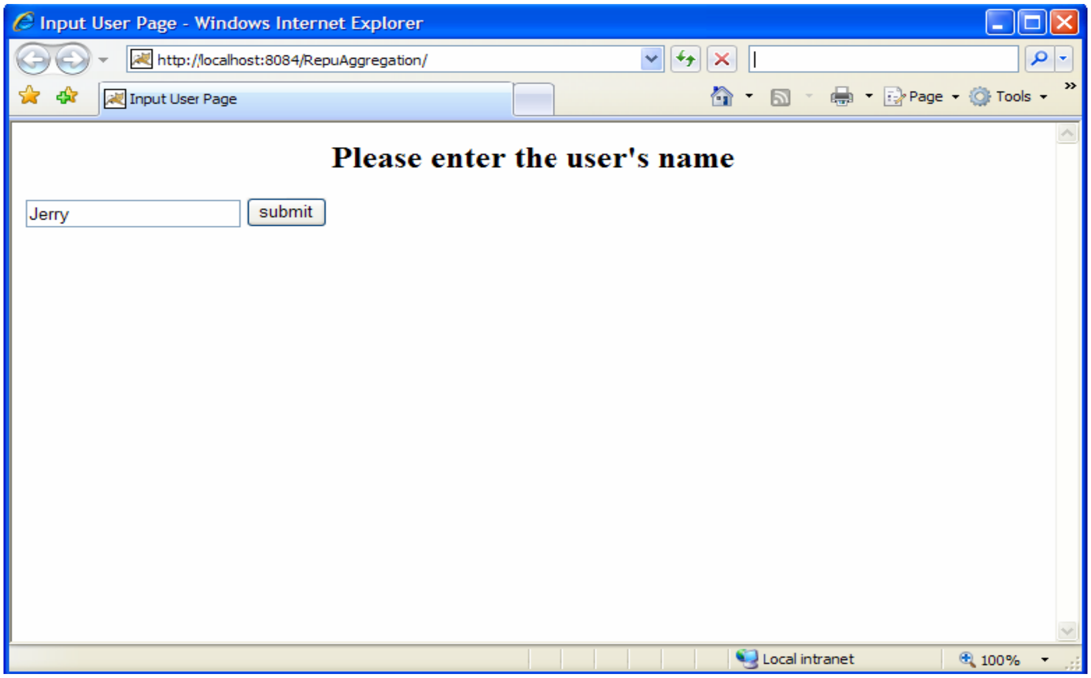
lated in a JSP file called *Aggregation*.

In the prototype, we select some dummy online communities that we called dummy eBay, dummy Amazon's zshop and dummy Yahoo! Auction as our online reputation sources. Dummy eBay and Yahoo use three ratings such as *positive*, *neutral* and *negative* to represent the trustworthiness of ratees in their systems. Dummy Amazon uses five ratings which are *very good*, *good*, *average*, *bad* and *very bad*.

5.4 Interaction Scenario

The following interaction scenario with screenshots should give a clear view about how ORAS works. Please note that the online communities (eBay, Amazon and Yahoo!Auction) we used in this scenario are not the real businesses but dummy ones with similar reputation representation schemes.

A querying agent *Jeff* wants to check the reputation information of a specific user, called *Jerry*. *Jeff* queries ORAS about *Jerry.* First, *Jeff* needs to enter *Jerry*'s username on the index.jsp, as shown in Figure 7. After *Jeff* clicks the Submit button, ORAS queries all reputation systems maintained by the communities where *Jerry* is active. ORAS has access to the *reputation information database*, shown in Figure 8, to check whether there are reputation information records about *Jerry*. The reputation information stored in *reputation information database* is normalized using the *Mapping Module* implemented by Duan and Zhang [38].

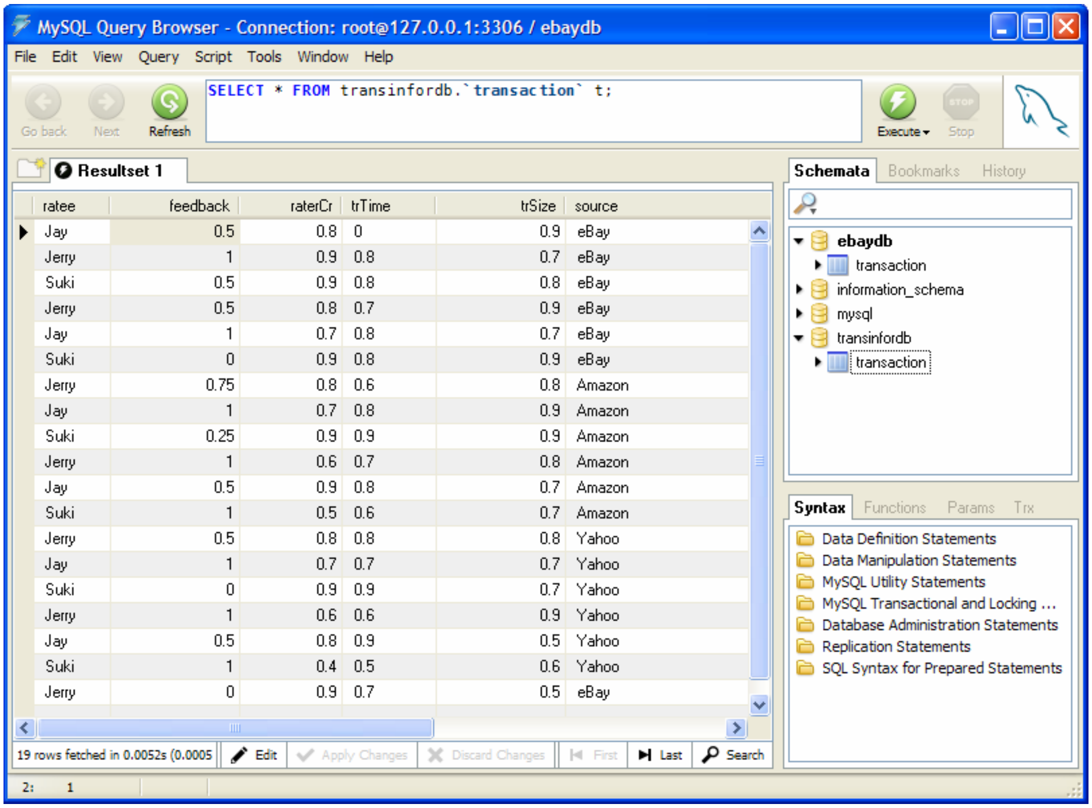


Input User Page - Windows Internet Explorer
http://localhost:8084/RepuAggregation/
Input User Page
Page
Tools
Local intranet
100%

Please enter the user's name

Jerry [submit]

Figure 7: JSP Page - index.jsp with Username Jerry



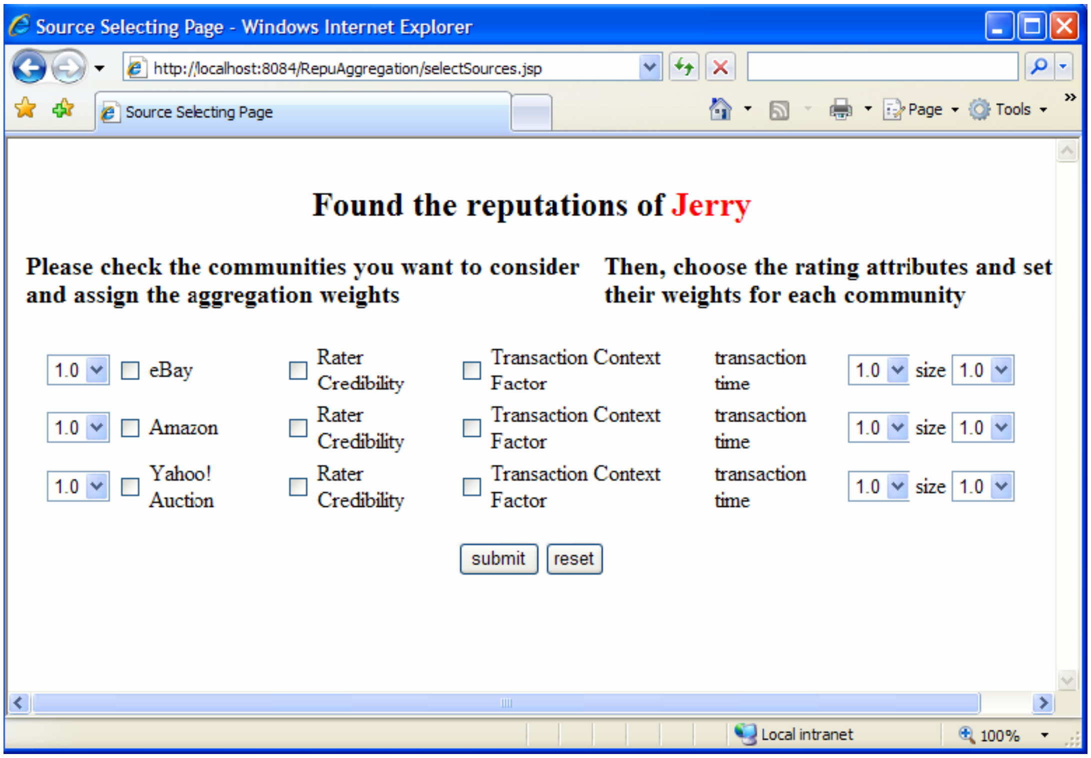
```
SELECT * FROM transinfordb.`transaction` t;
```

ratee	feedback	raterCr	trTime	trSize	source
Jay	0.5	0.8	0	0.9	eBay
Jerry	1	0.9	0.8	0.7	eBay
Suki	0.5	0.9	0.8	0.8	eBay
Jerry	0.5	0.8	0.7	0.9	eBay
Jay	1	0.7	0.8	0.7	eBay
Suki	0	0.9	0.8	0.9	eBay
Jerry	0.75	0.8	0.6	0.8	Amazon
Jay	1	0.7	0.8	0.9	Amazon
Suki	0.25	0.9	0.9	0.9	Amazon
Jerry	1	0.6	0.7	0.8	Amazon
Jay	0.5	0.9	0.8	0.7	Amazon
Suki	1	0.5	0.6	0.7	Amazon
Jerry	0.5	0.8	0.8	0.8	Yahoo
Jay	1	0.7	0.7	0.7	Yahoo
Suki	0	0.9	0.9	0.7	Yahoo
Jerry	1	0.6	0.6	0.9	Yahoo
Jay	0.5	0.8	0.9	0.5	Yahoo
Suki	1	0.4	0.5	0.6	Yahoo
Jerry	0	0.9	0.7	0.5	eBay

19 rows fetched in 0.0052s (0.0005

Figure 8: Reputation Information Database

A user interface, Figure 9, is displayed to *Jeff* who selects the communities he/she wants to consider and assigns aggregation weights to them. An aggregation weight is a decimal value between 0 and 1. The highest weights should be assigned to the communities believed to be the most accurate in reflecting the real reputation of the ratee. For instance in Figure 9, the list of online communities returned to *Jeff* includes dummy eBay, Amazon and Yahoo! Auction. It means that there are transaction information records about *Jerry* found in these online communities.


Source Selecting Page - Windows Internet Explorer
http://localhost:8084/RepuAggregation/selectSources.jsp
Source Selecting Page
Page
Tools
submit
reset
Local intranet
100%

Found the reputations of Jerry

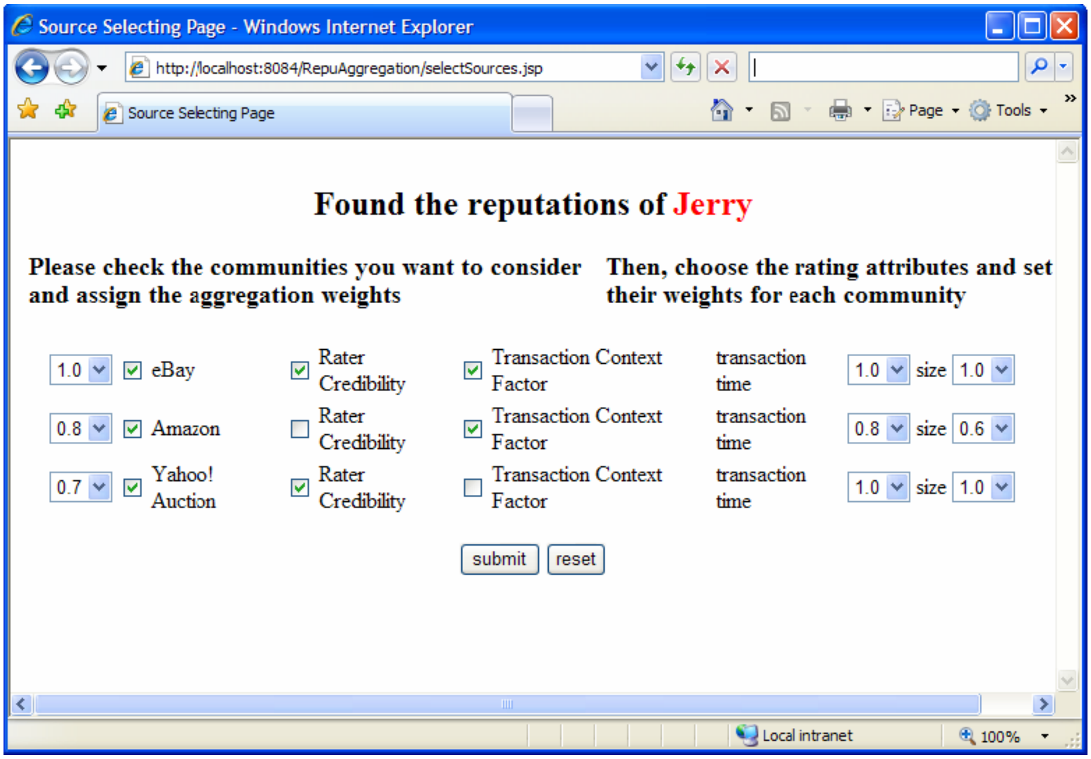
Please check the communities you want to consider and assign the aggregation weights

Then, choose the rating attributes and set their weights for each community

1.0	□ eBay	□ Rater Credibility	□ Transaction Context Factor	transaction time	1.0	size	1.0
1.0	□ Amazon	□ Rater Credibility	□ Transaction Context Factor	transaction time	1.0	size	1.0
1.0	□ Yahoo! Auction	□ Rater Credibility	□ Transaction Context Factor	transaction time	1.0	size	1.0

Figure 9: JSP page- selectSources.jsp

Jeff needs to select the communities he/she wants to consider. If *Jeff* fails to choose any community, an error message is displayed. For every selected community, *Jeff* chooses the rating attributes by checking the corresponding check boxes and sets their weights for Jerry, as shown in Figure 10. In our scenario, the global aggregation weights for dummy eBay, Amazon and Yahoo! Auction are assigned 1, 0.8 and 0.7. Furthermore, for dummy eBay, the credibility of rater and transaction context factor are chosen with the same value of a and b in Formula 4.6, representing the weights assigned to transaction Time Factor and Size Factor, respectively. For dummy Amazon, the transaction context factor is chosen with the weights 0.8 for transaction time and 0.6 for size. For dummy Yahoo! Auction, the credibility of rater is selected, which means that the credibility of rater needs to be considered when calculating the local reputation value of *Jerry* for Yahoo! Auction. This is shown in Figure 10.


Source Selecting Page - Windows Internet Explorer
http://localhost:8084/RepuAggregation/selectSources.jsp
Source Selecting Page
Page
Tools
submit
reset
Local intranet
100%

Found the reputations of Jerry

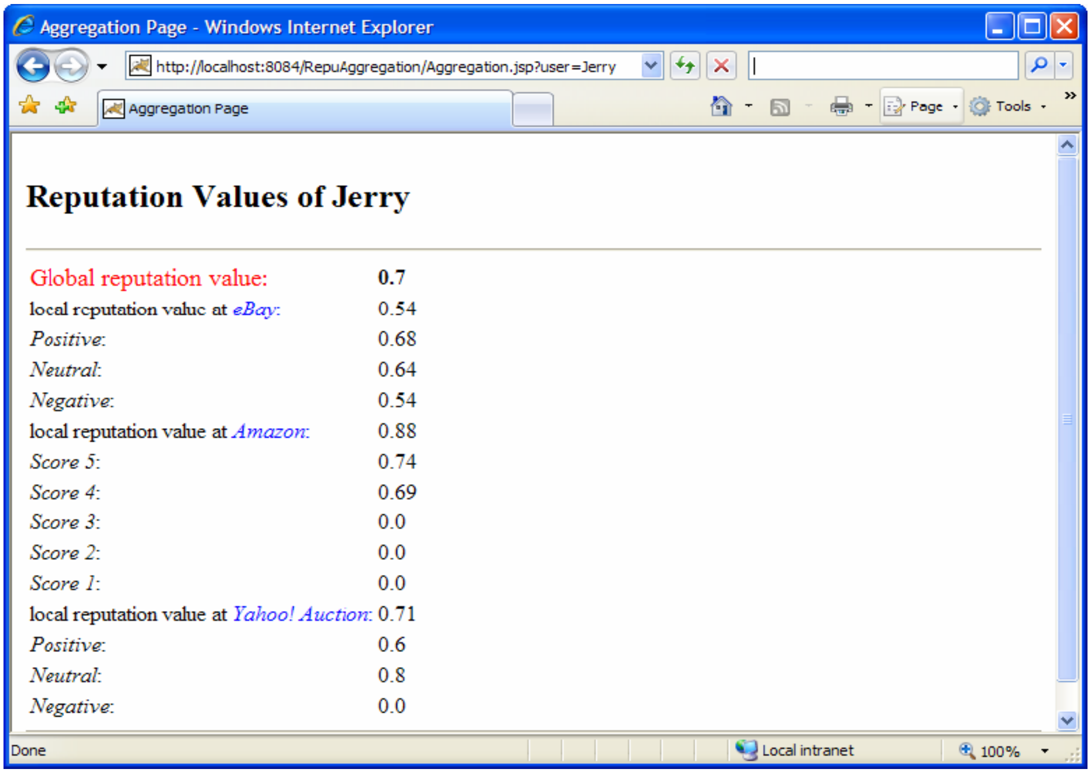
Please check the communities you want to consider and assign the aggregation weights

Then, choose the rating attributes and set their weights for each community

1.0	☑ eBay	☑ Rater Credibility	☑ Transaction Context Factor	transaction time	1.0	size	1.0
0.8	☑ Amazon	□ Rater Credibility	☑ Transaction Context Factor	transaction time	0.8	size	0.6
0.7	☑ Yahoo! Auction	☑ Rater Credibility	□ Transaction Context Factor	transaction time	1.0	size	1.0

Figure 10: JSP page- selectSources.jsp shows weights input for Jerry

After ORAS obtains the reputation data about *Jerry* from the *reputation information database*, ORAS computes the reputation values for *Jerry* then, an output screen, Figure 11, is returned to *Jeff*. On the output screen, the global reputation calculated by ORAS for *Jerry* is displayed at the top. The screen also indicates *Jerry*'s local reputation value and the weights corresponding to each possible outcome for each community. *Jeff* can get an idea about *Jerry's* reputation by viewing these reputation values calculated by ORAS.


Aggregation Page - Windows Internet Explorer
http://localhost:8084/RepuAggregation/Aggregation.jsp?user=Jerry
Aggregation Page
Page
Tools
Done
Local intranet
100%

Reputation Values of Jerry

Global reputation value:	**0.7**
local reputation value at *eBay*:	0.54
Positive:	0.68
Neutral:	0.64
Negative:	0.54
local reputation value at *Amazon*:	0.88
Score 5:	0.74
Score 4:	0.69
Score 3:	0.0
Score 2:	0.0
Score 1:	0.0
local reputation value at *Yahoo! Auction*:	0.71
Positive:	0.6
Neutral:	0.8
Negative:	0.0

Figure 11: JSP page- Aggregation.jsp

5.5 Verification of Result

To verify the results obtained by our implementation, we consider the interaction scenario of the ratee *Jerry* described in the previous section.

On the dummy eBay community, he has been rated three times (i.e., I(i) = 3) possibly more than once by the same rater. The querying agent *Jeff* chose the credibility of rater and assigned “1” to both *a* and *b* representing the weights for transaction Time Factor and Size Factor. On the Amazon community, he has been rated two times (i.e., I(i) = 2). *Jeff* chose the weight “0.8” for transaction time and the weight “0.6” for size factor. On the Yahoo! Auction community, he has been rated two times (i.e., I(i) = 2). *Jeff* considered the credibility of rater only. Table 6, 8 and 10 show the feedback values, the weight for the credibility of rater, and the weight for transaction context factor including time and size factors for each feedback value. Aggregation weights W_{ik} are calculated using Formula 4.5 and 4.6. Table 7, 9 and 11 show the mapping table used to convert discrete feedback values into numerical values.

k	*f_{ik}*	*CR_{ik}*	*T_{ik}*	*S_{ik}*	*W_{ik}*
1	Positive (1)	0.9	0.8	0.7	0.68
2	Neutral (0)	0.8	0.7	0.9	0.64
3	Negative (-1)	0.9	0.7	0.5	0.54

Table 6: Reputation information for Jerry at dummy eBay

Discrete feedback	***Numerical values***
Positive (1)	1
Neutral (0)	0.5
Negative (-1)	0

Table 7: Dummy eBay's feedback mapping table

The local reputation value for *Jerry* within the dummy eBay community is calculated as

$$R_i = \sum_{k=1, f_{ik}=Positive}^{3} \frac{W_{ik}}{\sum_{m=1}^{3} W_{im}} * NumVal(Positive) + \sum_{k=1, f_{ik}=Neutral}^{3} \frac{W_{ik}}{\sum_{m=1}^{3} W_{im}} * NumVal(Neutral) \tag{5.1}$$

$$+ \sum_{k=1, f_{ik}=Negative}^{3} \frac{W_{ik}}{\sum_{m=1}^{3} W_{im}} * NumVal(Negative)$$

$$= \frac{0.68}{0.68+0.64+0.54} * 1 + \frac{0.64}{0.68+0.64+0.54} * 0.5 + 0 \approx 0.54$$

formula 5.1. *Jerry* has a local reputation value of 0.54 within the dummy eBay community which is the same as the one shown in Figure 11. Similarly, the local reputation for *Jerry* within the Amazon community and Yahoo! Auction are 0.88 and 0.71 using the data shown in Table 8, 9, 10 and 11 below.

k	f_{ik}	T_{ik}	S_{ik}	W_{ik}
1	5	0.7	0.8	0.74
2	4	0.6	0.8	0.69

Table 8: Reputation Information for Jerry at dummy Amazon

Discrete feedback	*Numerical values*
5	1
4	0.75
3	0.5
2	0.25
1	0

Table 9: Dummy Amazon's feedback mapping table

k	f_{ik}	CR_{ik}	W_{ik}
1	Positive (1)	0.6	0.6
2	Neutral (0)	0.8	0.8

Table 10: Reputation information for Jerry at dummy Yahoo! Auction

Discrete feedback	*Numerical values*
Positive (1)	1
Neutral (0)	0.5
Negative (-1)	0

Table 11: Dummy Yahoo! Auction 's feedback mapping table

The querying agent *Jeff* chose the aggregation weights 1, 0.8 and 0.7 for dummy eBay, Amazon and Yahoo! Auction, so the global reputation value is calculated as follows:

$$GR_i = \sum_{j=1}^{3} R_{ij} * \frac{W_j}{\sum_{m=1}^{3} W_m} = 1 * \frac{0.54}{0.54 + 0.88 + 0.71} \tag{5.2}$$

$$+ 0.8 * \frac{0.88}{0.54 + 0.88 + 0.71} + 0.7 * \frac{0.71}{0.54 + 0.88 + 0.71}$$

$$\approx 0.7$$

We can see clearly that the global reputation we calculated here is the same as the one obtained from our system. After verifying the result, we can conclude that our implementation of the computation model for ORAS is successful.

Chapter 6

Conclusion and Research Perspectives

6.1 Conclusions

In this thesis, we provided a classification of existing computation models on trust and reputation after we reviewed the related literature. We also presented a novel solution to the problem of globality of online reputation. Users who build a reputation in one community are unable to transfer it to another community. In view of the importance that reputation systems are gaining as a way of fostering trust in online business and interpersonal interactions, we believe globality to be an important feature. Our approach to achieve it is to gather raw rating data about a ratee from various online communities instead of an aggregated reputation score (i.e., the reputation score like in eBay), aggregate the data from a given community into what we call a local reputation, then aggregate all local reputation values into a global reputation. The aggregation is based on options and weights which are selected by the querying agent according to his/her personal requirements. Our computation algorithm used in local aggregation is a novel aggregation method based on a discrete statistical model; it takes into account several factors and parameters that qualify the reputation, and it works in a novel framework in which reputation information is aggregated from different online resources.

The contributions of this thesis are listed below:

(1) A literature review on a selection of current computation models for online trust or reputation and a classification of these models using proper characteristics

(2) A computation model for the Online Reputation Aggregation System (ORAS) and a set of rating attributes for the aggregation formalism in the computation model

(3) A prototype implementation of the ORAS computation model based on the designed architecture and verification of the computation model using an example

To our knowledge, our reputation aggregation system is the first one to let the querying agent choose their target communities, aggregation factors and corresponding weights. It is a “Configurable” online reputation aggregation system. In the existing literature, there is no standard ontology for calculating reputation and propagating reputation information between various reputation systems. In other words, most of the reputation systems lack “globality”. Our system is a good start. Furthermore, it calculates the estimated probabilities of future actions for a given ratee taking into account several reputation factors.

We did not implement any form of security for ORAS, which is beyond the objective of the thesis. User identity is a huge issue for establishing trust and securing interoperability online as stated in Section 4.1. We simply assume that the user identity is unique when using our system, which is currently not a realistic assumption.

6.2 Proposals for Future Work

ORAS is still in an initial stage; and several extensions are envisaged for this work, among them: (1) considering reputation to be multidimensional where a ratee can be rated on more than one issue (as a seller rated by product quality, service, etc.; as a buyer rated by payment time, etc.); (2) considering other factors in the aggregation of local reputation; (3) investigating other ways to calculate the credibility of raters; (4) evaluating our system; and (5) addressing issues in terms of reliability, reusability and security.

As we described in Sections 2.2.3 and 3.2.5, reputation should be multidimensional. Our model can be extended to a multi-context model by changing the single scale of reputation into a vector scale. Each dimension of the vector would represent a reputation value under a certain context. In this case, we need to redesign our computation model. In order for a multidimensional model to work, there must be rating data available for different contexts.

Evaluating the credibility of a rater is not a straightforward process. Some existing approaches suggest concepts such as *semantic distance* [3] and *personalized similarity measure* [27] for dealing with this matter. Better methods to evaluate the credibility of a rater could be developed in the future.

In order to evaluate our system, we need to make it available for use in the real world online by a large number of users for a longer period of time. After that we would study all the data gathered by the system to draw conclusions on the efficiency, reliability and usability of our solution

References

1. Dellarocas, C., *The Digitization of Word-of-Mouth: Promise and Challenges of Online Reputation Systems.* Management Science, 2003. **49**(10): p. 1407-1424.
2. Resnick, P., Zeckhauser,R., Friedman, E. , Kuwabara, K. , *Reputation Systems.* Communications of the ACM, December 2000. **12**: p. 45-48.
3. Abdul-Rahman, A., Hailes,S. *Supporting Trust in Virtual Communities*. in *Proceedings of the 33rd Annual Hawaii International Conference on System Science*. 2000. Hawaii.
4. Bailey, B.P., Gurak, L. J., & Konstan, J. A. *Trust in Cyberspace*. 2003 [cited 2007 June 10]; Available from: http://www.digitalenterprise.org.
5. Wang, Y.D., Emurian, H. H., *An Overview of Online Trust: Concepts, Elements and Implications.* Computers in Human Behaviour, 2005. **21**(1): p. 105-125.
6. Yuan, S., & Sung, H. , *A Learning-Enabled Integrative Trust Model for E-Markets.* Applied Artificial Intelligence, 2004. **18**: p. 69-95.
7. Marsh, S.P., *Formalising Trust as a Computational Concept*. April 1994, Ph.D. Thesis,University of Sterling.
8. Gambetta, D., *Can We Trust Trust?* Trust: Making and Breaking Cooperative Relations, ed. D. Gambetta. 1988, Oxford: Basil Blackwell. 213-237.
9. Elofson, G., *Developing Trust with Intelligent Agents: An Exploratory Study.* In Proceedings of the first International Workshop on Trust, 1998: p. 125-139.
10. Olsson, O. *Trust in eCommerce- the Ontological Status of Trust*. in *ECOM-02-Electronic Commerce-Theory and Applications*. 2002. Gdansk.
11. Josang, A., Ismail, R. , Boyd, C. , *A Survey of Trust and Reputation Systems for Online Service Provision.* Decision Support Systems, March 2007. **42**(2): p. 618-644.
12. Benyoucef, M., *Reputation Systems for Fostering Online Trust*. 2005, School of Management,University of Ottawa.
13. Change, E., Hussain, F. K. , Dillon,T. , *Reputation Ontology for Reputation Systems.* OTM Workshops,LNCS 3762, 2005: p. 957-966.
14. Change, E., Hussain, F. K. , and Dillon,T., *Trust and Reputation for Service-Oriented Environments: Technologies for Building Business Intelligence and Consumer Confidence*. 2006: John Wiley & Sons, Ltd.
15. Zacharia, G., Moukas, A. , Maes, P., *Collaborative Reputation Mechanisms in Electronic Marketplaces.* Proceedings of the 32nd Hawaii International Conference on System Science, 1999. **8**: p. 8026.
16. A.Malaga, R., *Web-Based Reputation Management Systems: Problems and Suggested Solutions.* Electronic Commerce Research, 2001: p. 403-417.
17. Casare, S., Sichman,J. , *Towards a Functional Ontology of Reputation.* proceedings of the fourth international joint conference on Autonomous agents and multiagent systems 2005: p. 505-511.
18. Goldberg, D.e.a., *Using collaborative filtering to weave an information tapestry.*

Communications of the ACM, 1992. **35**(12): p. 61-70.

19. Dellarocas, C., & Resnick, P. *Online Reputation Mechanisms: A Roadmap for Future Research - A Summary Report*. in *First Interdisciplinary Symposium on Online Reputation Mechanisms*. 2003. Cambridge, MA, USA.

20. Kollock, P., *The Production of Trust in Online Markets*. Advances in Group Processes, ed. E.J. Lawler, Macy, M., Thyne, S., and Walker, H.A. . Vol. 16. 1999: CT: JAI Press.

21. Gilbert, A., Abraham, A., Paprzycki, M., *A System for Ensuring Data Integrity in Grid Environments.* Proceedings of the International Conference on Information Technology: Coding and Computing(ITCC'04), 2004. **1**: p. 435-439.

22. Yu, B., Singh, P. Munindar, *Distributed reputation management for electronic commerce.* First International Joint Conference on Autonomous Agents and Multiagent Systems, 2002.

23. Dutta, D., Goel, A., Govindan, R.,& Zhang, H. *The Design of A Distributed Rating Scheme for Peer-to-peer System*. in *Ecnomics of P2P systems*. 2003. Berkeley,CA.

24. Zacharia, G., Maes,P., *Trust management through reputation mechanisms.* Applied Artificial Intelligence, 2002. **14**: p. 881-908.

25. Yu, B., Singh, P. Munindar, *An Evidential Model of Distributed Reputation Management.* In Proceedings of the first Int.Joint Conference on Autonomous Agents & Multiagent Systems(AMAS),ACM, July 2002: p. 294-301.

26. Kamvar, S.D., Schlosser, M. T., Garcia-Molina, H., *The Eigentrust algorithm for Reputation Management in P2P Networks* Proceedings of the 12th International Conference on World Wide Web May 2003: p. 640-651.

27. Xiong, L., Liu, L. , *PeerTrust:Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities.* IEEE Transaction on Knowledge and Data Engineering, July 2004. **16**(7): p. 843-857.

28. Yamamoto, H., Ishida,K., Ohta,T., *Modeling Reputation Management System on Online C2C Market.* Computational&Mathematical Organization Theory, 2004. **10**: p. 165-178.

29. Shi, J., Bochmann,G. v., Adams,C. , *A Trust Model with Statistical Foundation*. Processings of IFIP Workshop on Formal Aspects in Security and Trust, ed. M. T. Dimitrakos , F. Aug. 2004, Toulouse: Springer. 145-158.

30. Shi, J., Bochmann,G. v. , Adams,C. , *Dealing with recommendations in a statistical trust model.* Processings of AAMAS Workshop on Trust in Agent Societies, July, 2005: p. 144-155.

31. Song, S., Hwang, K. , Zhou, R. , Kwok, Y. , *Trusted P2P Transactions with Fuzzy Reputation Aggregation.* IEEE Computer Society, November, December 2005: p. 24-34.

32. Sabater, J., Sierra,C. , *Review on Computational trust and reputation models.* Artificial Intelligence Review, September 2005. **24**(1): p. 33-60.

33. Dellarocas, C., *Analyzing the Economic Efficiency of Ebay-Like Online Reputation Reporting Mechanisms.* Proceedings of theThird ACM Conference on Electronic Commerce, 2001: p. 171-179.

34. Manchala, D.W., *Trust Metrics, Models and Protocols for Electronic Commerce*

Transactions. Processings of the 18th International Conference on Distributed Computing Systems, 1998: p. 312-321.

35. Zhou, R., Hwang, K. , *Power Trust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer computing.* IEEE Transactions on Parallel and Distributed Systems, June 21, 2006. **18**: p. 460-473.

36. Mui, L., *Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks*, in *Department of Electrical Engineering and Computer Science*. 2002, Ph.D.Thesis, Massachusetts Institute of Technology

37. Li, H., Benyoucef, M., & Bochmann, G.v., *A Personalized Online Reputation Aggregation System*. 2007, Working Paper 07-28, University of Ottawa, Canada.

38. Duan, Z., Zhang, B. & Benyoucef, M. *Portable Online Reputation Systems*. in *Networking and Electronic Commerce Research Conference (NAEC 2007)*. 2007. Riva Del Garda, Italy.