Trust Management Towards Service-Oriented Applications

Yan Wang
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
yanwang@ics.mq.edu.au

Kwei-Jay Lin
Dept of EE and CS
University of California, Irvine
Irvine, CA 92697, USA
klin@uci.edu

Duncan S. Wong
Department of Computer Science
City University of Hong Kong
Hong Kong
duncan@cityu.edu.hk

Vijay Varadharajan
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
vijay@ics.mq.edu.au

Abstract

In Service-Oriented Computing (SOC) environments, service clients interact with service providers for services or transactions. From the point view of service clients, the trust status of a service provider is a critical issue to consider, particularly when the service provider is unknown to them. Typically, the trust evaluation is based on the feedback on the service quality provided by service clients. In this paper, we first present a trust management framework that is event-driven and rule-based. In this framework, trust computation is based on formulae. But rules are defined to determine which formula to use and what arguments to use, according to the event occurred during the transaction or service. In addition, we propose some trust evaluation metrics and a formula for trust computation. The formula is designed to be adaptable to different application domains by setting suitable arguments. Particularly, the proposed model addresses the incremental characteristics of trust establishment process. Furthermore, we propose a fuzzy logic based approach for determining reputation ranks that particularly differentiates new service providers and old (long-existing) ones. This is further incentive to new service providers and penalize poor quality services from service providers. Finally, a set of empirical studies has been conducted to study the properties of the proposed approaches, and the method to control the trust changes in both trust increment and decrement cases. The proposed framework is adaptable for different domains and complex trust evaluation systems.

1 Introduction

In the Service-Oriented Computing (SOC) field, a variety of e-services across various domains can be provided to service clients in a loosely-coupled environment via various technologies (such as Web services technology [1]). This makes trust evaluation/management a very challenging and critical issue to the fast developing service-oriented applications.

The trust issue has been actively pursued in Peer-to-Peer networks (P2P). In general, P2P networks are used for information-sharing systems, such as Napster [3]. In such systems, each peer can act as a client or a server at the same time. Being a serving party, the peer can provide some files to the community. Other peers can retrieve information with interest and download from trustworthy peers [6] providing complete files. Thus, in such an environment, it is quite natural for a client peer to doubt the trust status of serving peers prior to any download actions in order to find the right peer to interact.

In e-commerce environments, the trust issue is more prominent as neither a buyer nor a seller is willing to be cheated. eBay [1] enables a simple reputation system which is one of the earliest systems in practice. At eBay, after each transaction, the buyer can rate the seller based on the quality of the transaction and the seller's service. The calculated reputation value (termed as the positive feedback rate) is public and useful for other potential buyers to make a decision prior to a transaction.

In P2P, E-Commerce and SOC fields, there are some common features with respect to the study of trust evaluation. First, the trust status of a seller or service provider is important to a buyer or a service client. A trust management mechanism is necessary for trust request broadcast, trust data collection, and trust computation. Second, each rating is provided by a buyer or service client posterior to a transaction.

On the other hand, there are some differences in all fields. First, the difference exists in the trust management organization. In general, in P2P environments, it advocates that the networks work without any central management. Therefore, in P2P trust evaluation, a typical process is that each peer can rate another peer after an interaction/transaction. This is the local rating. When a certain peer (referred to as *requesting peer*) is willing to know the trust status of a target peer (say peer X), it can send requests to other peers. A peer with interaction history with X can respond to this request with its ratings. This peer is referred to as a *responding peer* or a *recommending peer* as its ratings become recommendations when they are sent to the requesting peer. In contrast, in SOC environments, a central management server can be set up for trust management. Service clients can report their ratings to the central server as transaction feedback posterior to transactions [13]. So is the e-commerce field (e.g., eBay). In addition, in P2P trust evaluations, in general, it is the requesting peer to compute the final trust value according to its trust metrics and preferences. However, in SOC trust evaluation, it is more feasible for the central trust management server(s) to compute the trust values and respond them *as services* to requesting clients.

Therefore, though there are some common features and issues in both P2P and SOC trust evaluation, due to the diversity of services, more complex mechanisms should be studied, which can be more encompassing (note: we take the e-commerce transaction as one of diverse services). In these studies, the first concern is the SOC-oriented trust management architecture. Traditionally, in most trust evaluation models, a binary or numerical rating system is adopted and a formula is proposed for trust computation. This is simple and may be effective enough. But in SOC environments, as there are a variety of service providers and service clients across different domains, each domain may have its own *policy* to come up with an evaluation. Additionally, in trust evaluation, according to the transaction history

and the quality of recent transactions, new trust values can be derived. Particularly, in a negative case, where an undesirable service happened (e.g., a bad quality service or a fraud service), corresponding penalty should be determined in the trust calculation. The penalty varies from event to event, from party to party, from policy to policy, and from domain to domain. So is the positive case where the increment may vary. In most existing studies [6, 12, 24, 13, 14, 21], the trust computation relies on predefined formulae only. This is simple but might not be *adaptable* enough to reflect appropriate trust variations in response to events and policies in domains.

Additionally, given the same trust value of two service providers, does it mean that they have the same reputation level in terms of service quality? In this case, do we need to differentiate them considering other factors?

In this paper, we present a trust evaluation framework, a trust evaluation formula, and a fuzzy reputation rank evaluation model. The proposed architecture is rule-based and event-driven. The domain-specific rules are categorized corresponding to different events. Namely, an event can trigger a corresponding rule or a set of rules. Rules are maintained in rule base operated by the rule owners. The proposed framework also adopts formulae for trust computation and we also advocate defining as less formulae as possible to enable a simple and efficient system. Rules are used for determining which formula to use, and what the arguments are when applying a formula. The proposed fuzzy approach addresses more concerns on differentiating new service providers and old (long existing) ones. Namely, with the same trust value, a new service provider may have a better reputation rank than the old one under some circumstances. This work is based on our initial work in [20] and extends it

This paper is organized as follows. In Section 2, we review some existing studies. Section 3 presents the rule-based and event-driven trust management framework. In Section 4, we discuss some trust evaluation metrics and propose a formula-based method for trust computation. In Section 5, we present the fuzzy reputation rank model. Some empirical study results are illustrated in Section 6. In Section 7, we conclude our work.

2 Related Work

Trust issue is important in e-commerce environments. eBay [1] is a typical Customer-to-Customer (C2C) web site enabling the trust management mechanism. At eBay, after each transaction, the buyer can give a feedback to the seller according to the service quality of the seller. The feedback (or rating) is stored by eBay (a central management architecture). The feedback can be "positive", "neutral" or "negative". eBay calculates the feedback score $S = P - N$, where P is the number of positive feedback left by members and N is the number of negative feedback from members and displays $S's$ value on the web page. This is a simple trust management system providing valuable reputation information to buyers.

Trust evaluation is also considered as an important issue in Peer-to-Peer information sharing networks as a client peer needs to know prior to download actions which serving peer can provide complete files. In [6], Damiani *et al* proposed an approach for evaluating the reputation of peers through distributed polling algorithm before downloading any in-

formation. The approach adopts a binary rating system and it is based on the Gnutella [2] query broadcasting method using TTL limit. EigenTrust [12] also adopts a binary rating system and aims to collect the *local trust values* of all peers to calculate the *global trust value* of a given peer. Some other earlier studies also adopted the binary rating system, such as [24]. In [14], Marti *et al* proposed a voting reputation system that collects responses from other peers on a given peer. The final reputation value is calculated combining the values returned by responding peers and the requesting peer's experience with the given peer.

As pointed in [25], binary ratings work pretty well for file sharing systems where a file is either the definitive correct version or is wrong, but cannot accurately model richer services such as web services and e-commerce, where a boolean may not adequately represent a peer's experience of the quality of service (QoS) with other peers, e.g., the quality of products the peer sends and the expected delivery time [25]. In most later studies on trust evaluation (e.g., [25, 21, 22]), a numeral rating system is adopted, where, for example, the rating is a value in the range of $[0, 1]$. Such a system is more suitable for complex applications, such as e-commerce or service-oriented applications while binary ratings work pretty well for file sharing systems where a file is either the definitive correct version or is wrong.

In the literature, trust issue also caused much attention which is not bound to the P2P networks only. In [16], Sabater and Sierra proposed a model discussing the trust development between groups. When calculating the trust from individual A to individual B, it takes a few factors into account, e.g. the interaction between A and B, the evaluation of A's group to B and B's group, and A's evaluation to B's group. In [10], some models for the propagation of trust and distrust are proposed. In [7], Griffiths proposed a multi-dimensional trust model which allows the agents to model the trustworthiness of others according to various criteria. In [5], Cheng and Friedman propose a model to identify the attack by creating multiple identities to earn reputation, which is more typical in web citing systems. In [13], Lin *et al* propose a method of reputation-based trust evaluation in service-oriented environments based on the proposed architecture consisting of distributed trust management brokers. In [19], Vu *et al* propose a model to evaluate and rank the trust and reputation of QoS-based services, which is valuable for service search and selection. In [11], it addresses the trust issue from the perspective of multi-agent systems where in addition to trust evaluation, motivations of agents, the influence and dependency relationships among them should also be taken into account. In [8], the proposed method addresses the decay of trust values wherein over time trust values may become outdated if the experiences that gave rise to them are no longer relevant. A decay function is proposed for depicting how trust values decay.

However, in the context of service-oriented trust management, some issues remain open and solutions are being expected.

1. How to design the trust management? The architecture should incorporate the multi-domain features of service-oriented applications. Meanwhile, efficient methods should be employed for delivering results with both flexibility and efficiency. Currently, most studies devise formulae only for trust evaluation. This is simple and efficient. But how to extend them to adapt to diverse applications should be a concern for the formula

design.

2. In existing studies, few work addresses well how to depict the characteristics of *long-term and incremental* trust establishment process. Namely, in any service-oriented system, a new service provider may join anytime. How does it obtain an objective trust value? This is critical particularly for a trust management system. Otherwise, the trust system may not deliver objective results or is vulnerable to attacks.

   For example, in the studies in [15, 24], the methods to determine a trust value are similar. In a certain period, if there are x positive feedbacks and y negative feedbacks, the trust value is $T = \frac{x}{x+y} \in [0, 1]$ or $T = \frac{x-y}{x+y} \in [-1, 1]$. Thus, neglecting the binary rating problem, if A just serves in 2 transactions and gets two positive feedbacks, and B serves in 100 transactions and gets 99 positive feedbacks and 1 negative feedback, $T_A = 1$ is better than $T_B = 0.99$ or $T_B = 0.98$. In another case, if A is new and B is old (long-existing). Both of them receive 90% positive feedbacks, does it mean they have the same quality of service and thus the same reputation level? Similar cases can be found in other areas. For example, assume that the most famous brand X (e.g., Coca Cola) has taken 20 or more years to build up its high-level reputation in the field. A new brand Y in the same field emerges and earns much attentions in a few years by similar quality and cheaper price. This is helpful to build up its reputation. But generally it may be pretty difficult for Y to become as strongly reputable as X in a short time.

   In all the above cases, as most existing methods do not depict the trust establishment process, the delivered trust results cannot reflect the trust status of service providers. The work by Wang et al in [21] has somewhat similar purpose. But it takes both positive events and negative events equally. In [26], Zacharia and Maes propose a model for trust evaluation. It assigns a low initial value to a new party and calculate its new trust values based on ratings in later transactions, where later ratings are given higher weights. However, in both above studies, when depicting the trust establishment process, more concerns should be considered, such as the trust increment and decrement, the scale of penalty, etc. In Section 4.2, we will discusses more detailed metrics for trust evaluation.

3. In a good trust management system, new service providers and old service providers should be further differentiated when determining their service reputation ranks. Incentive mechanisms should be considered in evaluation metrics. This is critical to protect and encourage new service providers with good services and penalize old service providers with "bad" reputation and "poor" service quality.

   For example, if new service provider A and old service provider B have the same trust values, $T_A = T_B = 0.7$, A can be assigned a better reputation rank than B.

3 Trust Management Architecture

In this section, we present an event-driven and rule-based architecture for trust management. This architecture inherits the features of formula-based trust evaluation method and is suitable for service-oriented applications.

The proposed architecture is based on a centralized management server (like eBay). Service clients or buyers can report to the trust management authority after transactions. The trust management server manages the portfolio data of service providers and service clients as well as the trust data of service providers.

The proposed architecture is different from the one in [19], which adopts a decentralized (Peer-to-Peer) architecture. A central management architecture occurs less communication cost following a trust request. In the P2P-based architecture, it doesn't have the cost to set up separate servers. But once a peer needs to know the trust status of a service provider, in general, it (a requesting peer) has to broadcast a request to other peers. Hereafter, the requesting peer will collect some volume of trust data for computing the trust result locally. This process has to repeat whenever a peer wants to know the trust status of a target peer. Thus it is costly in terms of network communication. The decentralized architecture also has a problem as when every time the requesting peer broadcasts the request, it is not likely for all peers, who have the transaction history with the target service provider under investigation, to be online and respond. In contrast, in a centralized management architecture, the requesting client can simply communicate with the central trust management server, which stores trust history data, compute the trust value accordingly and respond to the clients.

We assume the centralized trust management architecture is independent of any service provider. This makes it realistic to apply unified evaluation approach to the same domain of applications. In addition, it ensures that the trust computation can be completed on relatively complete trust data, which is difficult in a decentralized environment as we discussed above. Certainly, the centralized architecture is subject to a single point failure due to the scalability problem, which is less risky in the decentralized environment.

As a matter of fact, our focus in the architecture design is the *rule-based* and *event-driven* architecture. In our proposed framework, we assume there are a set of software agents with corresponding tasks. There are several databases storing the data of service providers, clients and trust data respectively. The framework serves for the trust management in service-oriented environments, which is an integrated system combining both rules and formulae for trust computation. These rules are used to define the policy of the trust management authority and categorize events that may occur in transactions. The policy can reflect the nature of the domain. Thus the proposed framework is adaptable to various domains of service-oriented applications.

3.1 Authority Interface Agent

This is the agent interacting with clients. It receives the request from a client and send it to the Event Management Agent in order to start the trust evaluation process. Finally this agent sends the result to clients.

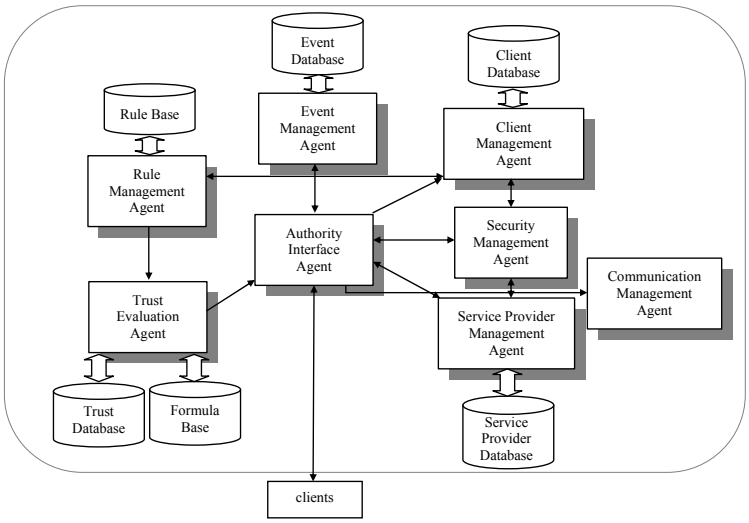

Figure 1: Trust Management Authority

3.2 Client Management Agent and Service Provider Management Agent

The two agents manage clients and service providers respectively and manage corresponding databases. Both service providers and service clients should be registered users.

3.3 Security Management Agent

The Security Management Agent manages the security mechanisms of the trust management authority. It is also responsible of the authentication of service providers and service clients. Alternatively, it can cooperate with other servers to build up a Kerberos-like distributed authentication architecture [18]. We will not discuss this issue as it is out of the scope of this paper.

3.4 Event Management Agent

This agent is responsible of categorizing the events in transactions in a domain and managing the event database. These events are first categorized as *positive* and *negative* events. In the proposed framework, a service client or a buyer reports the feedback together with the event occurred. In particular, if the event is a *negative* one, the agent will communicate with the Rule Management Agent to determine the the penalty in trust evaluation. Namely, negative events should be further categorized. Each category includes events with the same nature and corresponds to the same degree of penalty in trust evaluation.

3.5 Rule Management Agent

This agent manages the rules for trust evaluation in the domain. There rules reflect the nature of the trust management in the domain. Rules are stored in the rule database. They can be inserted and updated. For example, when the feedback is a positive for a satisfactory service, the corresponding formula and arguments for trust evaluation should be determined. Likewise, if the feedback is negative with a unsatisfactory event, the corresponding formula and arguments for penalty will be selected, which thereafter are sent to the Trust Evaluation Agent.

As we introduced above, domain-specific rules are predefined to depict the policy of trust managements. Basically there are three categories.

1. rules for positive events;

   In trust evaluation, in general, positive event may lead to increments in trust evaluation. Rules are used to determine which positive event can lead to increment and the scale of the increment.

2. rules for negative events;

   Rules are predefined to determine the penalty of negative events. For severely negative event, the new trust value can be set to 0.

3. extra rules.

   Extra rules are necessary. They vary from context and system design. For example, in this work, we adopt fuzzy logic to determine reputation ranks. Fuzzy rules (see Table 1) are defined to determine the reputation scores and the final reputation rank.

3.6 Trust Evaluation Agent

This agent is responsible of evaluating the target service provider as requested by a service client. Trust data is stored in the Trust Database, which is operated by this agent. The agent is also responsible of managing the formula base.

3.7 Trust Management Process

In general, a trust evaluation request is sent to the Authority Interface Agent, which transfers it to the Event Management Agent. The reported event, positive or negative, triggers the process of trust evaluation. The Event Management Agent communicates with the Rule Management Agent about the event. Corresponding rules will be determined for the formula and argument selection. These are sent to the Trust Evaluation Agent, which computes the trust result based on both history trust data and the reported feedback. The result is sent to the client via the Interface Agent.

4 Trust Evaluation

In this section, we will study a trust evaluation method. Though a complex application oriented trust management framework has been proposed, it doesn't indicate that there should be as more formulae as possible. A good formula can incorporate both positive and negative cases with arguments adjusted flexibly to adapt to domains and respond to events.

4.1 Trust Rating System

Here we assume a trust result calculated by the trust management authority is a numerical value in the range of $[0, 1]$, where 1 means the most trustworthy and 0 implies the worst reputation. The rating given by a client or a buyer is also a numerical value in the range of $[0, 1]$. For a client A, if it has an interaction with a service provider B, it can give a local trust rating $R_{A \to B}^{(k)} \in [0, 1]$ for the interaction occurred at time period $t^{(k)}$. The value can be calculated considering multiple aspects of the quality of the service provided by B [23, 19, 16, 8]. These aspects exist in the concerns of the availability of services, the efficiency of service or product delivery, and the conformance of the service or product with the advertisement. As discussed in [25], this rating system is more appropriate for service-oriented environments, instead of file-sharing systems.

4.2 Trust Evaluation Metrics

A good trust evaluation system should be a fair system to reflect the trust status of different parties according to the quality of services or transactions. Namely, the trust result difference should reflect the service quality difference of sellers or service providers. This is radically important for buyers or service clients to make decisions when a set of potential sellers or service providers are available.

In general, reputation establishment in the service field is a long period process with the accumulation of good ratings. The whole process can be roughly divided into three periods. The length of each period depends on the initial trust value and quality of services.

Period 1 - Initial Reputation Establishment: Initially, a new service provider or seller has no reputation. Thus it can attract clients by offering good, even extremely good services. But meanwhile, the reputation improvement is not quick as the reputation improvement relies on long-term good services. That is, in this period, the trust may improve quickly, but the trust value is not in the relatively high level.

Period 2 - Reputation Improvement: After the first period improvement, the trust level has reached to a medium/good level. During this period, the trust improves relatively slowly before reaching the highest trust level.

Period 3 - Reputation Stabilization: After the accumulation of good services for some periods, the trust value maintains in the high level. Thus the improvement is not significant.

This is the last period. It may be the longest period for a good service provider if it can always maintain its good service quality. But it doesn't mean once a service provider enters this period, it can stay in it and maintain a high reputation level always. If some negative events occurred, its reputation can drop to a lower level and hence enter prior periods.

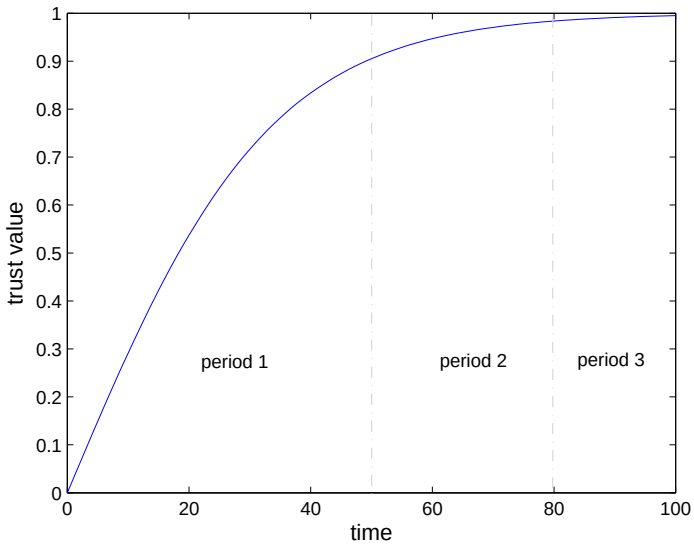

Figure 2: Trust Improvement Curve ($formula(2)$ $\alpha = 3$ $\beta = 1$)

The three periods are depicted in Figure 2. The plotted function is as follows, which is a transformation of Hyperbolic Tangent - $tanh(x)$.

$$y = \frac{e^{3x} - e^{-3x}}{e^{3x} + e^{-3x}} \qquad where \quad x \in [0, +\infty) \ and \ y \in [0, 1] \tag{1}$$

Actually there may be more than one function that can depict the above features of the trust establishment process. We simply select one of them. In late discussions, we can also observe that even for a selected function, the curve can be different to some extent if different arguments are selected (refer to Section 6.1 and Section 6.4). But basically they reflect the same curve change trend.

In the above discussion, we assume each rating obtained is very good (e.g., R=1.0) leading to the curve plotted in Figure 2. In order to show the long period of trust improvement, we set the maximal time to 100 units. But in practice, this can be set by arguments. For instance, formula (1) is an instance of the generalized formula (2). Such a function like formula (2) is referred to as a *basic curve function*, which depicts the trust establishment process features and will also be used for determining the scale of trust updates in later

section. In Section 6.1, we study the selection of arguments and the impact on function curve.

$$bf(x) = \frac{e^{\alpha x} - e^{-\alpha x}}{(e^{\alpha x} + e^{-\alpha x})\beta} \qquad (\alpha \geq 1 \quad \beta \geq 1) \tag{2}$$

In Figure 2, the x-axis stands for time. Each unit is a time period. In this work, we consider 'abstract' time unit, which can also be termed as *round*. It may be a transaction, an hour, a day, a week or a month, which is application dependable. In addition, there may be some negative cases that should lead to penalty in trust evaluation. Prior to presenting the detailed trust evaluation formula, we first propose a set of trust evaluation principles as follows.

Principle 1: In general, the initial trust value of a party should be set to a low value;

This principle, to some extent, enforces the party being evaluated to provide honest and good services to customers in order to obtain good trust ratings and upgrade its trust level.

In addition, this can prevent a malicious service provider from cheating clients easily. Otherwise, if a new service provider is assigned a high trust value, it can cheat clients then disappear. Afterwards, it can join the community again with a new id then keep cheating (e.g. it is free to join some web applications, such as eBay and Facebook (facebook.com)). The trust management system may not radically prevent online cheating. But it should be *costly* to cheat clients, i.e., it takes a *long period* of good services to obtain a high level reputation, with which the cheating may easily happen.

From the perspective of business strategy, for good new service providers, they can attract customers by offering low prices and high quality services aiming at accumulating the reputation level after a certain period.

Principle 2: The low initial trust value is reasonable but it may lead to less customers of new service providers for a period. Thus incentive mechanism should be designed to from the trust management perspective to encourage the reputation improvement of new service providers. This feature will be addressed in Section 4.3 and Section 5.

Principle 3: A good reputation is the result of accumulative good services and behaviors, and consequent good ratings in a relatively long period;

Namely, as we have discussed in Section 2, the trust improvement is a long process, which indicates the difficulty of trust upgrade. The characteristics of the three-period trust establishment process are also analyzed above. The process may be different from domain to domain. In a formula-based trust evaluation system, arguments can be used to control the curve.

Principle 4: The trust evaluation system should be punitive to dishonest and unsatisfactory services. In a good trust evaluation system, it is relatively easier to downgrade the trust value than upgrading trust values.

In contrast to the trust level upgrade, dishonest services and unsatisfactory services will lead to penalty and trust level downgrade. In addition, given two services with the same nature (e.g., the same transaction amount) and different feedback (i.e. one satisfactory and one unsatisfactory), the trust level increment should be not greater than the trust level decrement. In particular, a dishonest transaction will lead to severe penalty (e.g., reset the trust value to 0).

A trust evaluation system is radically an incentive system to encourage honest and satisfactory services or transactions, and penalize dishonest and unsatisfied services or transactions. For instance, after having a honest and satisfactory transaction, the seller can get a trust value increment of 0.01. On the contrary, if the transaction is dissatisfactory, the seller will get a decrement of 0.2 or more.

Principle 5: The new trust evaluation result should reflect the service history.

This principle indicates the trust improvement is history-oriented. Hence, the new trust value should result from reputation history and the service quality in the current round.

Principle 6: Trust computation can be based on formulae. But rules can be defined for the selection of formulae and arguments.

In different domains, there may be different policies for trust evaluation. Thus, though some formulae can be applied in different domains, arguments may be different. In addition, in the case of penalty, the decrement is determined by arguments, which are selected by rules according to the nature of negative events.

4.3 Trust Evaluation Method

Definition 1: Let $T_x^{(k)}$ denote the trust value of target service provider X at current time period $t^{(k)}$, and $R_x^{(k+1)}$ is the rating of X at time period $t^{(k+1)}$. $\Delta = R_x^{(k+1)} - T_x^{(k)}$. *The trust value of X at time period $t^{(k+1)}$* is

$$T_x^{(k+1)} = \begin{cases} min(1, \quad T_x^{(k)} + \theta \cdot \Delta) & \text{if } \Delta \geq 0 \\ max(0, \quad T_x^{(k)} + \theta \cdot \Delta) & \text{if } \Delta < 0 \end{cases} \tag{3}$$

where

$0 \leq \theta < 1$ is the *impact factor* determining the impact of recent change (i.e. Δ) on the trust calculation;

Formula (3) yields a value in the range of $[0, 1]$. To obtain the trust value in period $t^{(k+1)}$, the trust value in period $t^{(k)}$ is used. Thus, the new trust value results from the old trust value and the latest rating. This follows *Principle 5* proposed in Section 4.2.

To calculate θ, we propose an impact factor function based on the basic curve function $bf(x)$.

Definition 2: If the current trust value is $T_x^{(k)}$, the *impact factor function* θ can be

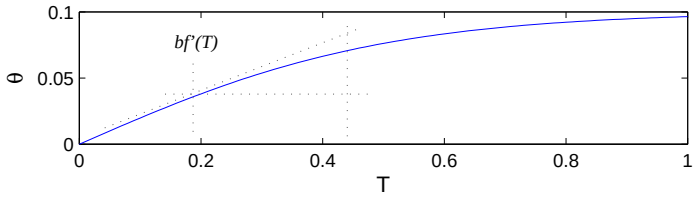

Figure 3: Basic Curve Function (*Formula* (6))

$$\theta = \lambda \cdot bf'(T_x^{(k)}) \tag{4}$$

where $\lambda > 0$ is the scale control factor; $bf'(x) \geq 0$ is the derivative of the basic curve function $bf(x)$.

Definition 3: We define the *scale control factor* $\lambda^{(k+1)}$ as follows.

$$\lambda^{(k+1)} = \begin{cases} \lambda_+ \leq 1 & \text{if } \Delta \geq 0 \\ \lambda_- \geq 1 & \text{if } \Delta < 0 \end{cases} \tag{5}$$

According to formulae (3) and (4), in equation (5), when $\Delta \geq 0$, there will be an increment in the trust calculation. Namely, $T_x^{(k+1)} \geq T_x^{(k)}$. In this case, normally $\lambda^{(k+1)} = \lambda_+ = 1$. Thus the increment will be $bf'(T_x^{(k)}) \cdot \Delta$. When $\Delta < 0$, there will be a decrement in the trust calculation. By default, we can set $\lambda_- > 1$. As $\lambda_+ \leq \lambda_-$, assuming the same $|\Delta|$, the decrement $bf'(T_x^{(k)}) \cdot |\Delta| \cdot \lambda_-$ is no less than the increment $bf'(T_x^{(k)}) \cdot |\Delta| \cdot \lambda_+$. This indicates that it is not easier to improve the trust value than worsening it. Therefore, generally it takes longer time to reach a high level trust value (e.g., 0.95) than drop from a high level to a low level. This follows *Principle 4* in Section 4.2. The value of λ is determined by the pre-defined rules (refer to Section 3). In addition, when a severely negative event happened, $\lambda_- \geq 2$ will be applied for decrement cases. This also indicates a harder trust improvement process.

Regarding the basic curve function, formula (6) - a specific case of formula (2), where $\alpha = 2$ and $\beta = 20$, is plotted in Figure 3.

$$bf(x) = \frac{e^{2x} - e^{-2x}}{(e^{2x} + e^{-2x}) * 10} \tag{6}$$

Furthermore, the derivative function of formula (2) is:

$$bf'(x) = \frac{\alpha}{\beta} - \frac{\alpha(e^{\alpha x} - e^{-\alpha x})^2}{\beta(e^{\alpha x} + e^{-\alpha x})^2} \quad (\alpha \geq 1 \quad \beta \geq 5) \tag{7}$$

An example of formula (7) is plotted in Figure 4, where $\alpha = 2$ and $\beta = 20$.

According to the above proposed formulae, some properties are listed as follows.

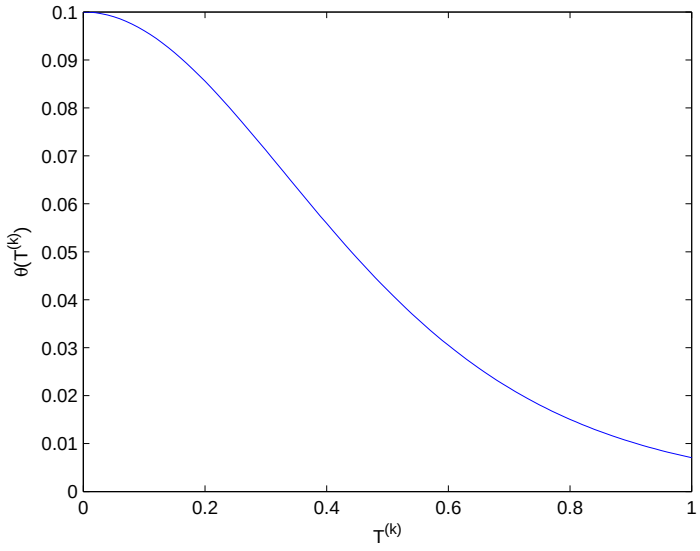

Figure 4: Impact Factor Function (*Formula* (7), $\alpha = 2$, $\beta = 20$, $\lambda = 1$)

Property 1: θ is in reverse proportion to the trust value $T^{(k)}$ (assuming $\lambda = 1$).

This indicates an incentive mechanism (*Principle 2*). If $T_A^{(k)} < T_B^{(k)}$ for parties A and B, given the same $\Delta > 0$, $\theta_A \cdot \Delta > \theta_B \cdot \Delta$. This is incentive to parties with low trust values and can reflect the characteristics of the three-period trust establishment process.

Property 2: The maximal of the impact factor is 0.1 (assuming $\lambda = 1$).

According to Formula (7) and Figure 4, $\theta_{max} = 0.1$ when $T = 0$. Namely, in this case, it is quite incentive for the service provider (probably new) to improve its trust level. But meanwhile, the trust improvement is a long process (refer to *Principle 3*). Thus, even if $T^{(0)} = 0$ and $R^{(1)} = 1$, the increment is $\theta * \Delta = 0.1 * 1 = 0.1$. Thus the new trust value is $T^{(1)}$=0.1, which is still in the low level.

Property 3: The curve of functions and the trend of trust value changes can be controlled by setting *system arguments* - α, β and λ. This follows *Principle 6*.

So far, as we have analyzed, principles 2 to 6 are being followed. Detailed empirical studies are conducted and the results are illustrated in Section 6.

5 Fuzziness of Reputation Values

In the above approach, based on ratings, the trust value of a service provider can be calculated and be provided to a client requesting it. But from the point of view of a client, it

may be difficult to understand the meaning of a trust value. For example, if the trust value of service provider A is $T_A = 1.0$, it is easy to understand that its service quality is very good. However, if $T_A = 0.81$, it is not likely to know the trust level. In this case, linguistic terms can provide clearer information to clients. These terms may be "very low", "low", "moderate", "high" and "very high". In the literature, there are some relevant works for trust evaluations using linguistic terms. In an early work in [4], the trust values are represented by some linguistic terms, such as "very good", "good", "bad" and "very bad". Some approaches adopt Fuzzy Logic [27] for the fuzzification of reputation ranks [17, 9]. However, our purpose is not only to provide clients with semantics of delivered reputation result, but also further differentiate the reputation levels of new service providers and old (long-existing) service providers so as to provide clearer reputation information to clients.

In some cases, a numerical trust value may not thoroughly reflect the trust status of a service provider though we believe the proposed approach can depict the characteristics of the trust establishment process. For example, if there are two service providers A and B and $T_A = T_B = 0.6$ are provided to a client, does it mean A and B have the same reputation level? If yes, which level are they in? In particular, if we assume A is a new service provider and B is not, do they have the same reputation level? In our proposed approach in Section 4, a new service provider is assigned an initial trust value which is low. Due to the long period of trust establishment (*Principle 3*), even if the service quality of a new service provider is quite good, its trust level is low in a certain period. Thus $T_A = 0.6$ indicates an acceptable level as A is new. For B, there are further two cases. Firstly, B struggles to reach the level of $T = 0.6$ in a longer period and maintains it. Secondly, B reaches a higher level. But due to some negative events, it drops to the level of $T = 0.6$. In both cases, $T_B = 0.6$ implies a less reputable or less reliable situation than A. Hence, when we are designing a trust management system, we should differentiate service providers and provide information to clients as clearly as possible.

To categorize the trust rank, we set up 5 fuzzy sets [27]. They are "very low", "low", "moderate", "high" and "very high" respectively. The membership function of each fuzzy set is designed, and are illustrated in Figure 5, where T_i $(i = 1, \ldots, 10)$ is the *system argument* that can be set for different applications. For the membership function of "very low" fuzzy set, for example, we set

$$\mu_{very\ low}(T) = \begin{cases} \frac{T_2 - T}{T_2} & \text{if } 0 \leq T \leq T_2 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

For other membership functions, we set

$$\mu_{low}(T) = \begin{cases} \frac{T - T_1}{T_3 - T_1} & \text{if } T_1 \leq T \leq T_3 \\ \frac{T_5 - T}{T_5 - T_3} & \text{if } T_3 < T \leq T_5 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

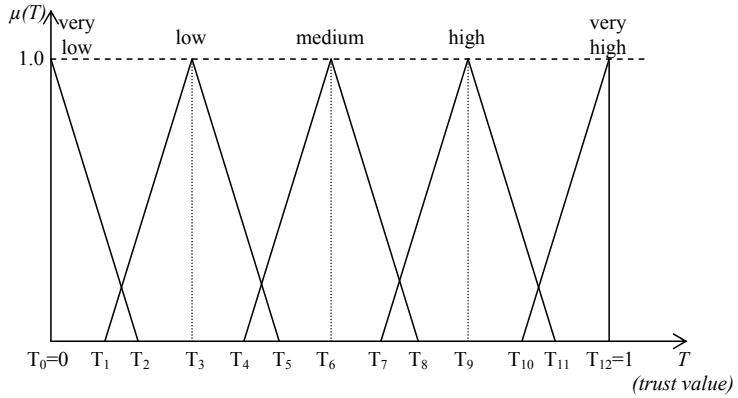

Figure 5: Member Functions of Fuzzy Sets for Trust Values

$$\mu_{medium}(T) = \begin{cases} \frac{T-T_4}{T_6-T_4} & \text{if } T_4 \leq T \leq T_6 \\ \frac{T_8-T}{T_8-T_6} & \text{if } T_6 < T \leq T_8 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$\mu_{high}(T) = \begin{cases} \frac{T-T_7}{T_9-T_7} & \text{if } T_7 \leq T \leq T_9 \\ \frac{T_{11}-T}{T_{11}-T_9} & \text{if } T_9 < T \leq T_{11} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$\mu_{very\ high}(T) = \begin{cases} \frac{T-T_{10}}{1-T_{10}} & \text{if } T_{10} \leq T \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Furthermore, as we discussed above, the service period of a service provider is also a concern. Here, we divide it into five fuzzy sets. They are "very new", "new", "medium", "old" and "very old". The membership functions of service period are illustrated in Figure 6. Some of them as listed as follows.

$$\lambda_{very\ new}(t) = \begin{cases} \frac{t_2-t}{t_2} & \text{if } 0 \leq t \leq t_2 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

$$\lambda_{new}(t) = \begin{cases} \frac{t-t_1}{t_3-t_1} & \text{if } t_1 \leq t \leq t_3 \\ \frac{t_5-t}{t_5-t_3} & \text{if } t_3 \leq t \leq t_5 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

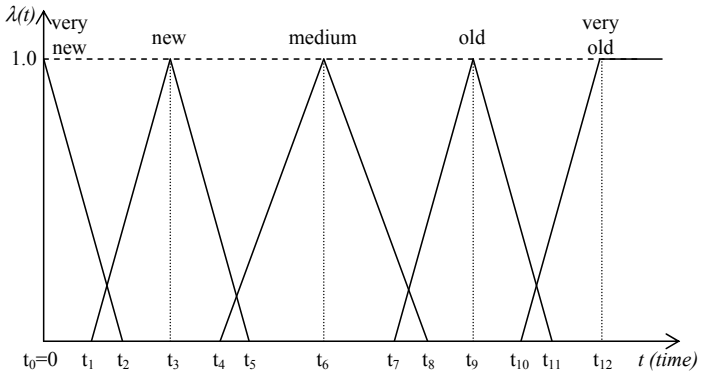

Figure 6: Member Functions of Service History Period

...

$$\lambda_{very\ old}(t) = \begin{cases} \frac{t-t_{10}}{1-t_{10}} & \text{if } t_{10} \leq t \leq 1 \\ \\ 1 & \text{otherwise} \end{cases} \tag{15}$$

Given the fuzzy sets of trust value T and that of service period t, we can design fuzzy rules to determine the reputation rank of a service provider.

In general, a fuzzy rule has the format as follows:

IF T is "L_1" AND t is "L_2" THEN the reputation is "L_3".

In the above fuzzy rule, L_i is a linguistic term. Regarding term "L_3", we design five fuzzy sets as "low as a new service provider","medium as new" (i.e. medium reputation level for a new service provider), "good as new", "very good as new", "extremely low", "very low", "low", "medium","good" and "very good".

t \ T	very low	low	medium	high	very high
very new	low as new (2́)	medium as new (3́)	good as new (4́)	very good as new (5́)	very good as new (5́)
new	low as new (2́)	medium as new (3́)	good as new (4́)	very good as new (5́)	very good as new (5́)
medium	low (2̀)	low (2̀)	medium (3̀)	good (4̀)	very good (5̀)
old	very low (1̀)	low (2̀)	medium (3̀)	good (4̀)	very good (5̀)
very old	extremely low (0̀)	very low (1̀)	low (2̀)	good (4̀)	very good (5̀)

Table 1: Fuzzy Rules for Reputation Ranks

All fuzzy rules are listed in Table 1.

In Table 1, the final reputation rank is determined by both the length of service period and current trust value. The purpose is to distinguish service providers with the same trust level but different service periods. For example, there are two rules as follows.

IF T is "*low*" AND t is "*new*" THEN the reputation rank is "*medium as new*", and
IF T is "*low*" AND t is "*old*" THEN the reputation rank is "*low*".

Assume there are two service providers A and B with the same trust level as "low". But A is "new" and B is "old" in terms of the length of service period. Thus A gets a "medium as new" reputation rank, which indicates its reputation level is not in the high level but it is a new service provider. In contrast, B gets a "low" reputation rank. Another case is with "very low" trust level. If A is "new", it obtains "low as new" rank. But B obtains "extremely low" rank if it is an "old" service provider. For these examples, we can see that by applying the proposed fuzzy rules, it is incentive to new service providers (*Principle 2*) and punitive to old service providers with low trust levels (*Principle 4*).

In Table 1, we also give the score for each reputation rank. In order to differentiate the service period, we label a score with state. Namely, a rank score is represented as a vector with two elements:

$$rank\ score\ =\ <value,\ state>$$

To represent the separate element, we can use functions $sta(\cdot)$ and $val(\cdot)$ to represent the state and the value of a rank score respectively.

To give a simpler presentation, we use $\acute{s}$ to indicate a value s of state "new" and use $\grave{s}$ to represent a value of s with an "old" state. Namely, $\acute{2}\ =\ <2,\ new>$ indicates $val(\acute{2}) = 2$ and $sta(\acute{2}) =$ "*new*". Likewise, $\grave{2}\ =\ <2,\ old>$ indicates $val(\grave{2}) = 2$ and $sta(\grave{2}) =$ "*old*".

With fuzzy rules, a case may exist that multiple rules in Table 1 can be activated. In order to have a fixed reputation rank, we designed a *satisfaction degree function* ϕ to determine which rule can be activated.

Definition 4: Let $T \in [0,1]$ and $t \in [0,1]$ denote the trust value and service period respectively. Let $\lambda_i(t)$ and $\mu_j(T)$ denote the membership functions of trust value and the membership function of service period. Given a service provider A with T and t, the *satisfaction degree function* is

$$\phi_{i-j}(t,T) = \lambda_i(t) * \mu_j(T) \tag{16}$$

Based on Definition 4, given T and t, if $\phi_{i-j}(t,T) > 0$, it indicates the rule in row i and column j in Table 1 is activated. In case that multiple rules are satisfied, the satisfaction degree of each rule is calculated, based on which the final reputation rank can be determined.

Definition 5: Let R_{ij} demote the rule with rank score S_{i-j} satisfied by T and t. $\phi_{i-j} \in \Phi = \{\phi_{i-j} > 0\}$ is the corresponding satisfaction degree. When multiple rules are activated, the final reputation score value $val(S)$ is

$$val(S) = \frac{\sum_{\phi_{i-j}\in\Phi} \phi_{i-j} * val(S_{i-j})}{\sum_{\phi_{i-j}\in\Phi} \phi_{i-j}} \tag{17}$$

When aggregating scores, the value can be determined by Definition 5. Its state is determined by the proportion of each state.

Definition 6: Let $val(S) = val(\acute{r}) + val(\grave{r})$, where $val(\acute{r})$ is the sum of scores of new state and $val(\grave{r})$ is the sum of scores old state. Then

rank score	new service provider	old service provider
4.75 ≤ val(S) ≤ 5		
4.5 ≤val(S) < 4.75		
4 ≤ val(S) < 4.5		
3.5 ≤ val(S) < 4		
3 ≤ val(S) < 3.5		
2.5 ≤ val(S) < 3		
2 ≤ val(S) < 2.5		
1.5 ≤ val(S) < 2		
1 ≤ val(S) < 1.5		

Table 2: Final Reputation Ranks

$$sta(S) = \begin{cases} new & \text{if } val(\acute{r}) \geq val(\grave{r}) \\ old & \text{if } val(\acute{r}) < val(\grave{r}) \end{cases} \tag{18}$$

For example, if service provider A has a trust value in both "low" and "medium" fuzzy sets, and the service period is in both "new" and "medium" fuzzy sets. Thus A satisfies four rules (see Table 1).

1. IF T is "low" AND t is "new" THEN the reputation score is $\acute{3}$.
2. IF T is "low" AND t is "medium" THEN the reputation score is $\acute{2}$.
3. IF T is "medium" AND t is "new" THEN the reputation score is $\acute{4}$.
4. IF T is "medium" AND t is "medium" THEN the reputation score is $\grave{3}$.

Hence, as $val(\acute{3}) + val(\acute{4}) = 7 > val(\acute{2}) + val(\grave{3}) = 5$ and $\frac{3+2+4+3}{4} = 3$ (assuming ϕ_{i-j} is the same for four rules), the final score is $\acute{3}$.

When aggregating reputation ranks, another case should also be considered. If the final fuzzy reputation score is not an integer, it should be particularly *rounded*. Here we aim to provide the service clients with the following ranking system, where score 5 is represented as *five stars*. We use the star format as it is equivalent to the perception in human society and the fuzzy rank will be finally presented to human users (say, via a Web page). To give more precise ranks, *half star* is allowed. Thus, if the final score is 4.7, it is rounded as 4.5 while 4.9 is rounded to 5. In order to represent the state of a rank, the system differentiate ranks of new and old service providers by *green stars* and *red stars* respectively. The score range and the representation of ranks are illustrated in Table 2.

The fizziness of reputation values can offer clearer semantics to clients and inform the relative position of a service provider with respect to its service quality. But the crisp trust value can leave precision to clients particularly when a client need to select one from

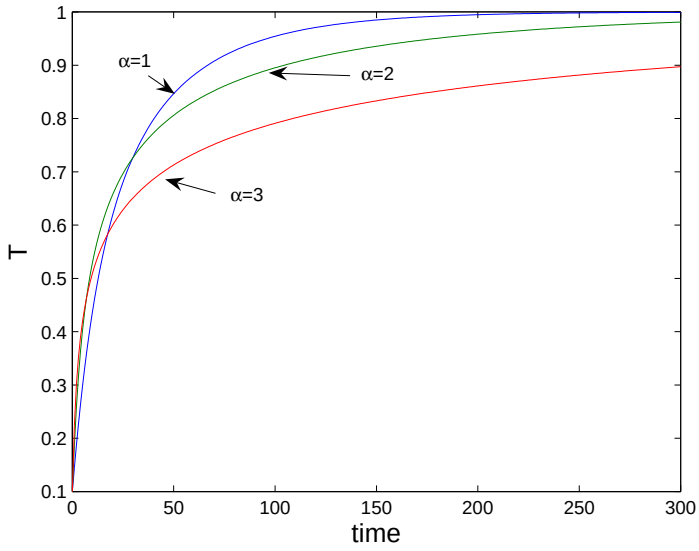

Figure 7: Study 1 with different α values ($\beta = 20$ $\lambda_+ = 1$)

two service providers with the same rank score (i.e., the same rank value and rank state). Therefore, a solution is to offer clients the combined information in a vector as follows and the decision-making by the client is dependable on the client's preference on either crisp value or fuzzy rank.

$$< trust\ value,\ \ rank\ score,\ \ rank\ state > \tag{19}$$

In addition, in the above discussion, we finally divide service provider into "new" and "old" categories. Given the framework, more categories can be introduced if necessary. But superabundant categories may leave confusion to customers and hence should be avoided.

6 Empirical Study

To further study the properties of our proposed approaches, we have conducted a set of empirical studies. In this section, we illustrate the results and analyze them. For trust computation, we adopt formula (3), where formula (4) is used as the impact factor function and formula (7) is used as the basic curve function.

6.1 Study 1 - Argument Selection

In this section, we study how to control the scale of trust changes. Basically, it can be controlled by setting different values of α and β. We first set $\beta = 20$, $T^{(0)} = 0.1$, $R^{(k+1)}$=1, $\lambda_+ = 1$, and α is set to 1, 2 and 3 respectively. The result is plotted in Figure 7. We can

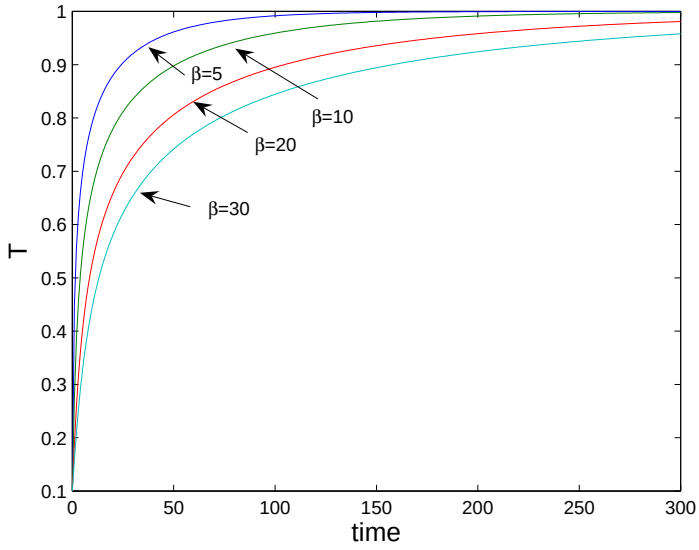

Figure 8: Study 1 with different β values ($\alpha = 2$ $\lambda_+ = 1$)

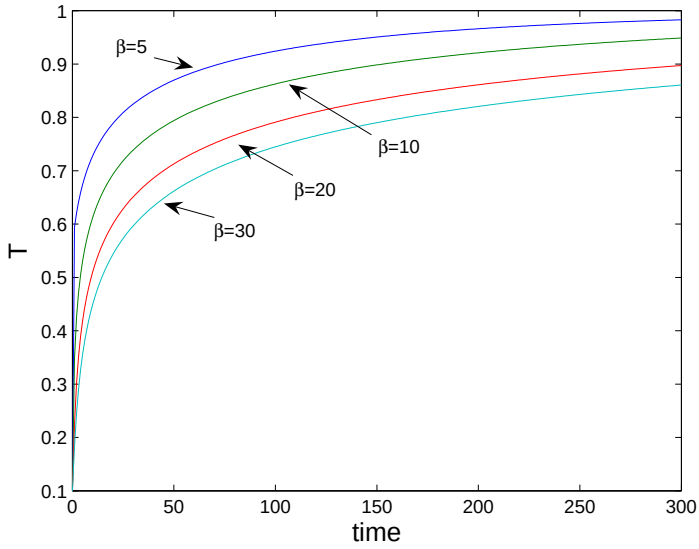

Figure 9: Study 1 with different β values($\alpha = 3$ $\lambda_+ = 1$)

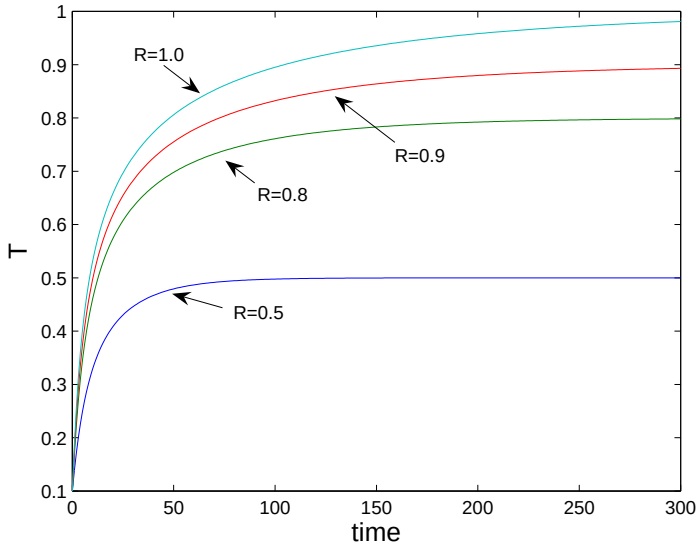

Figure 10: Study 2 ($\alpha = 2$ $\beta = 20$ $\lambda_+ = 1$)

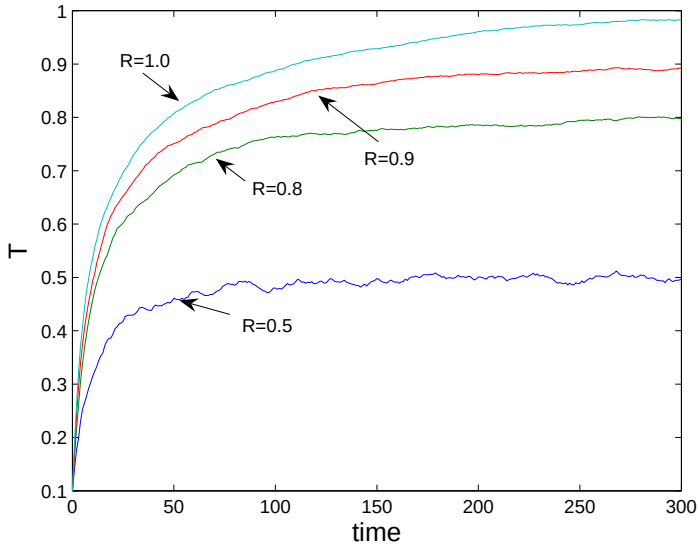

Figure 11: Study 2 ($\alpha = 2$ $\beta = 20$ $\lambda_+ = 1$ $|\epsilon| \leq 0.1$)

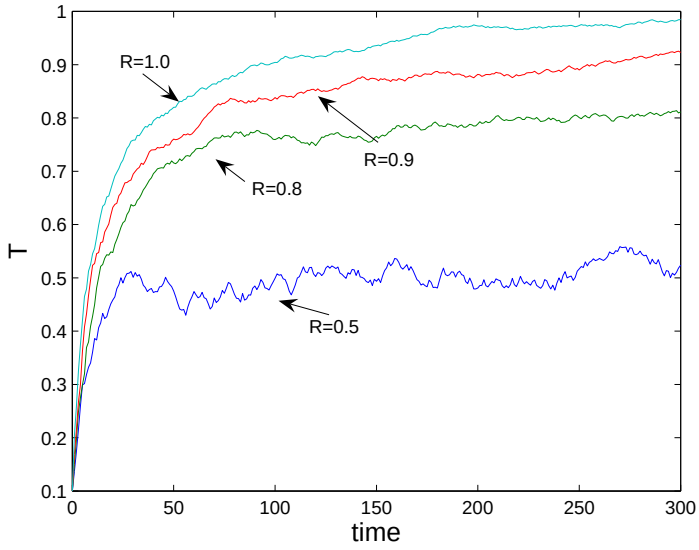

Figure 12: Study 2 ($\alpha = 2$ $\beta = 20$ $\lambda_+ = 1$ $|\epsilon| \leq 0.3$)

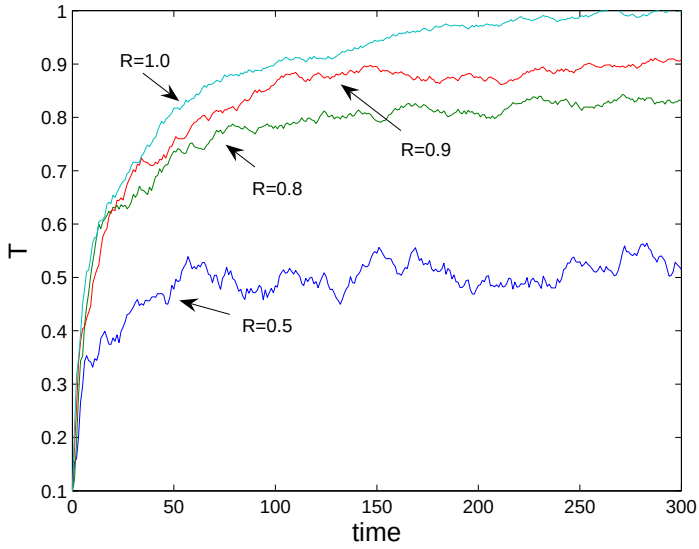

Figure 13: Study 2 ($\alpha = 2$ $\beta = 20$ $\lambda_+ = 1$ $|\epsilon| \leq 0.5$)

observe that the larger the value of α, the stricter the trust evaluation. That is, in case of a larger α value, it takes longer to reach a high trust level.

In the second case, we set $\alpha = 2$ and set β to 5, 10, 20 and 30 respectively. The result is plotted in Figure 8. Another case is plotted in Figure 9, where α is changed to 3. We can observe that the larger the value of β, the stricter the trust evaluation. Again, the result shows the proposed method is flexible to diverse applications by setting appropriate arguments. For example, if it is suitable for a domain to reach a high trust level in less than 50 time units, $\alpha = 2$ and $\beta = 5$ may be the right arguments. If stricter policy should be applied and 150 to 200 time units are suitable, β can be changed to 10 or a larger value.

In a word, smaller α and β values can lead to quicker trust improvement while larger α and β values lead to slower changes. Therefore, based on our proposed framework, α and β are part of the system parameters. Their values can be determined by predefined rules according to the domain and applications requirements.

6.2 Study 2 - Trust Establishment

In this section, we study how a new service provider establishes its trust level. The initial trust value is set to $T^{(0)} = 0.1$. We assume there are 4 service providers. They are S_1, S_2, S_3 and S_4. In each round, they get static ratings of 0.5, 0.8, 0.9 and 1.0 respectively.

The results are plotted in Figure 10. We can observe that the trust value of each party increases incrementally from 0.1. For any two parties A and B, $T_A^{(k+1)} > T_B^{(k+1)}$ if $R_A^{(k+1)} > R_B^{(k+1)}$. This is rational as it is incentive to better services and better ratings. In addition, when the trust value is static in each period, e.g., $R^{(k+1)} = a \in [0, 1]$, the final trust value is approaching a. Namely,

$$lim_{t^{(k)} \to \infty} T^{(k)} = a \tag{20}$$

Furthermore, we also studied the cases where $R^{(k+1)}$ is not static. Different from the above study, $R^{(k+1)}$ is approximately equal to a value, with some deviations. In Figure 11, $R_{(k+1)}$ is approximately 0.5, 0.8, 0.9 or 1.0 respectively for four service providers. The deviation is $|\epsilon| \leq 0.1$. Similarly, in the studies illustrated in Figures 12 and 13, we change the deviation ϵ to $|\epsilon| \leq 0.3$ and $|\epsilon| \leq 0.5$ respectively. From all three studies, we can draw the conclusion that when $R^{(k+1)} \approx a \in [0, 1]$, the final trust value is approaching a. Namely,

$$lim_{t^{(k)} \to \infty} T^{(k)} \approx a \tag{21}$$

Furthermore, we set $T^{(0)} = 0.1$ for all service providers. It follows *Principle 1* discussed in Section 4.2. It is particularly important for new service providers preventing from cheating clients without any cost as we observed in this study that it takes a long process for a service provider to reach a high reputation level even if it is service quality is always very good (say, S_3 and S_4). Additionally, we set the same initial trust value to all service providers aiming at comparing them with different service quality from the same starting point. However, this doesn't mean that each new service provider will be assigned the same initial trust value. The initial trust value can be determined according to the prior service history if any. For example, a new teacher joins a university. The initial trust value for him/her can be a medium or good one (e.g., 0.5 or 0.7) if the new teacher has obtained a good teaching

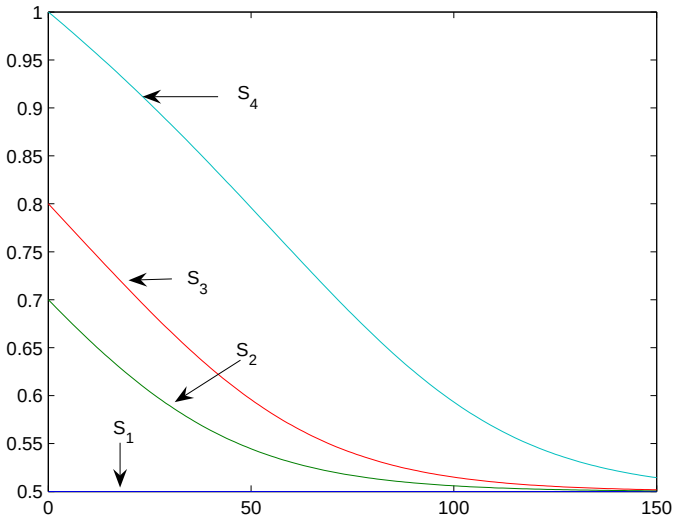

Figure 14: Study 3 ($\alpha = 2$ $\beta = 20$ $\lambda_- = 1$ $R^{(k+1)} = 0.5$)

reputation in another university (assumed to be a good one). Thus it will be much easier for him/her to reach to a high trust level.

6.3 Study 3 - Trust Decrement

In this section, we study how a service provider's trust value drops. The basic curve function is based on formula (2), where $\alpha = 2$ and $\beta = 20$. We assume there are four service providers S_1, S_2, S_3 and S_4. Their initial trust values (i.e. $T^{(0)}$) are 0.5, 0.8, 0.9 and 1.0 respectively. We also assume static $R^{(k+1)}$ and $\lambda_- = 1$.

We first set $R^{(k+1)} = 0.5$. The results are plotted in Figure 14. We can observe that for parties S_2, S_3 and S_4, the trust value drops from $T^{(0)}$ approaching 0.5 after some periods. Similarly, we can observe the same trust value change trends when $R^{(k+1)} = 0.3$, or 0, plotted in Figures 15 and 16 respectively.

Note that in real applications, $R^{(k+1)} = 0$ may be associated with severely negative events (say a fraud) that can lead to an action of setting $T^{(k+1)}$ to a low value (e.g., 0.1) directly yielded by the predefined rules or some other penalty (e.g., the service provider is removed from the system).

However, we can draw the same conclusion as Study 2. Namely, when the trust value is static in each period, i.e. $R^{(k+1)} = a \in [0, 1]$, the final trust value is approaching a:

$$lim_{t^{(k)} \to \infty} T^{(k)} = a \tag{22}$$

Similarly, we also studied the cases with random $R^{(k+1)} \approx 0.5$. We set the deviation ϵ to 0.1 and 0.3 respectively. The results are plotted in Figures 17 and 18, from which we can

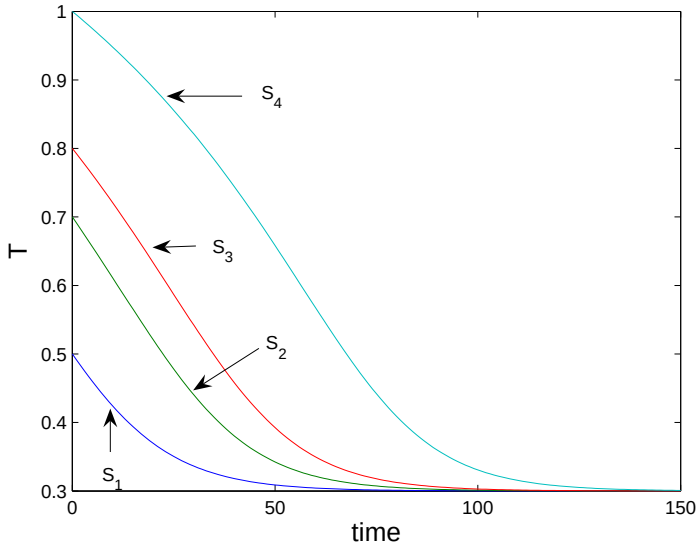

Figure 15: Study 3 ($\alpha = 2$ $\beta = 20$ $\lambda_{-} = 1$ $R^{(k+1)} = 0.3$)

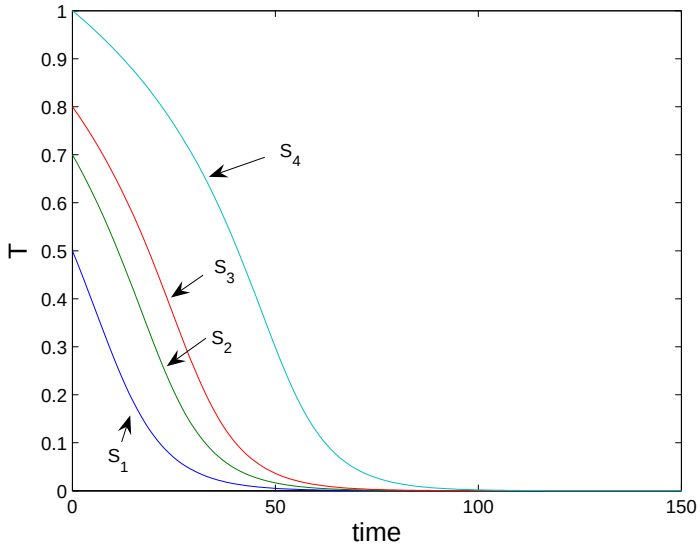

Figure 16: Study 3 ($\alpha = 2$ $\beta = 20$ $\lambda_{-} = 1$ $R^{(k+1)} = 0$)

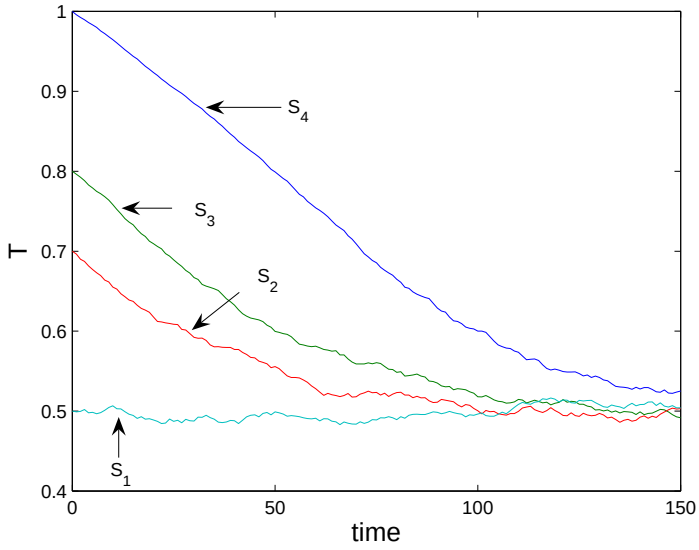

Figure 17: Study 3 ($\alpha = 2$ $\beta = 20$ $\lambda_- = 1$ $R^{(k+1)} = 0.5$ $|\epsilon| \leq 0.1$)

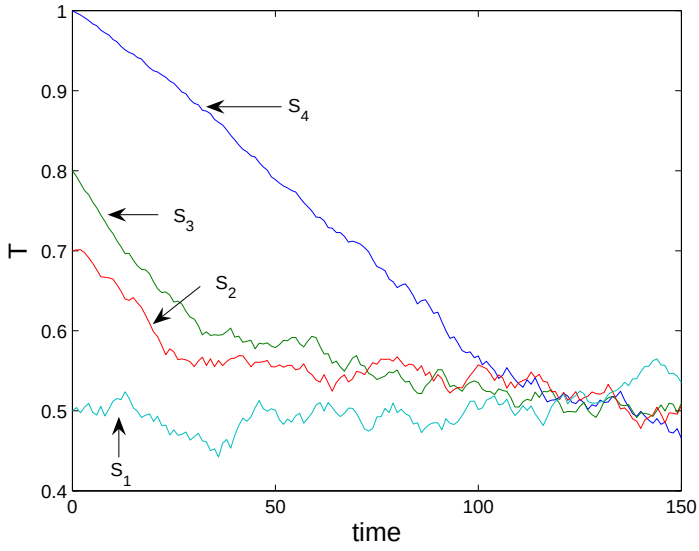

Figure 18: Study 3 ($\alpha = 2$ $\beta = 20$ $\lambda_- = 1$ $R^{(k+1)} = 0.5$ $|\epsilon| \leq 0.3$)

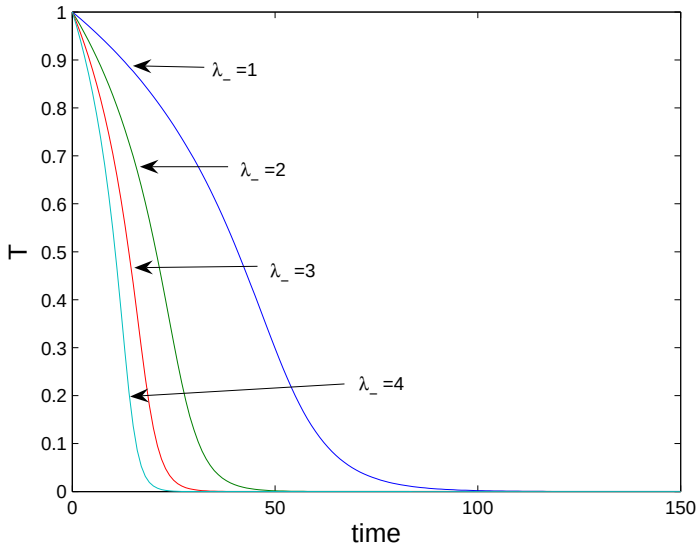

Figure 19: Study 4 ($\alpha = 2$ $\beta = 20$)

draw the conclusion that when $R^{(k+1)} \approx a \in [0, 1]$, the final trust value is approaching a. Namely,

$$lim_{t^{(k)} \to \infty} T^{(k)} \approx a \tag{23}$$

6.4 Study 4 - The Selection of λ_- Value

In this section, we study how the trust value drops with different λ_- values (refer to formulae (3), (4) and (7)). The initial trust value is set to 1.0. In each period, $R^{(k+1)} = 0$. λ_- is set to 1, 2, 3 and 4 respectively. Results are plotted in Figure 19. Unsurprisingly, a bigger λ_- value results in more decrement. In real applications, as we can adopt the rule-based framework, rules should be predefined to determine the λ_- value corresponding to the negative event category.

6.5 Study 5 - A Simulation

In this study, we compare several service providers with different service periods and different trust levels. We aim to compare their reputation ranks delivered by the proposed model.

In this study, there are 5 service providers. They are S_1, S_2, S_3, S_4 and S_5. S_1, S_2 and S_3 join in round (time unit) 0. $R_{S_1}^{(k+1)} \approx 0.95$. $R_{S_2}^{(k+1)} \approx 0.8$. S_3 is a service provider with "very good" service quality as $R_{S_2}^{(k+1)} \approx 0.9$ before round 100. At round 100, we assume S_3's trust value was set to 0.2 due to a severely negative event, after which its service quality

remains at the level of $R_{S_3}^{(k+1)} \approx 0.6$. Joining in round 150, S_4 is a new service provider reaching the level of $R_{S_4}^{(k+1)} \approx 0.95$. S_5 also joins at round 150. Its services remain at the level $R_{S_5}^{(k+1)} \approx 0.6$. Their trust values are illustrated in Figure 20.

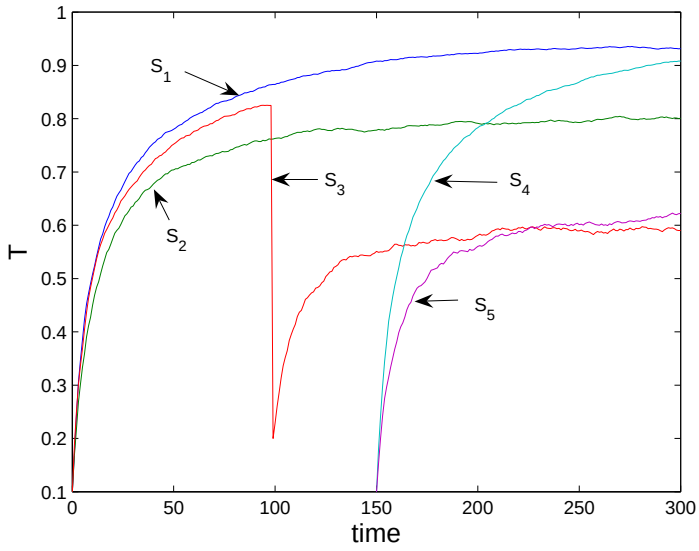

Figure 20: Study 5 ($\alpha = 2$ $\beta = 20$)

Given the above situation, we compare their trust value changes and reputation rank changes in different rounds.

In round 100 (refer to Figure 20, Tables 1 and 3), when S_4 and S_5 haven't joined, S_1, S_2 and S_3 are "medium" in terms of service period. Both S_1 and S_2 have reached the "high" trust level as $T_{S_1} \approx 0.85$ and $T_{S_2} \approx 0.75$. Thus they get the "good" reputation rank (4 red stars) with the rank score of $\grave{4}$ (for the sake of simplification, we wouldn't list the arguments of T_i and t_i for the membership functions of fuzzy sets). In contrast, $T_{S_3} \approx 0.2$. Thus it get the " very low" trust level and the "low" reputation rank with rank score $\grave{2}$ (2 red stars).

In round 160, when S_4 and S_5 newly join, S_2 remains the same reputation rank while S_1 reaches the "very good" reputation rank as its trust value $T_{S_1} \approx 0.93$ becomes "very good". For S_3, though its trust value changes from 0.2 to 0.6, its reputation rank improves to "medium" as it is now an "old" service provider and its trust value becomes "medium" with $T_{S_3} \approx 0.6$. In contrast, in this round, though $T_{S_4} \approx 0.4$ and $T_{S_5} \approx 0.3$ are in the "low" level, S_4 and S_5 get 3 green stars as they are "very new" service providers. This result is quite incentive to new service providers (*Principle 2*).

In round 200 (refer to Figure 20, Tables 1 and 4), the reputation ranks are not changed with S_1, S_2 and S_3. Both S_4 and S_5 improve their trust levels to $T_{S_4} \approx 0.8$ and $T_{S_5} \approx 0.55$ respectively. Thus they get 5 green stars and 4 green stars respectively as "new" service providers.

	Round=100			Round=160		
	trust	service period	Reputation Rank	trust	service period	Reputation Rank
S_1	high (0.85)	medium	good ($\grave{4}$)	very high (0.93)	old	very good ($\grave{5}$)
S_2	high (0.75)	medium	good ($\grave{4}$)	high (0.8)	old	good ($\grave{4}$)
S_3	very low (0.2)	medium	low ($\grave{2}$)	medium (0.6)	old	medium ($\grave{3}$)
S_4	N/A	N/A	N/A	low (0.4)	very new	medium as new ($\acute{3}$)
S_5	N/A	N/A	N/A	low (0.3)	very new	medium as new ($\acute{3}$)

Table 3: Study 5 - Result for Round 100 and Round 160

In round 300, both S_1 and S_2 remain the reputation rank unchanged while S_3 becomes worse. S_3 keeps its trust level as "medium" ($T_{S_3} \approx 0.6$). But it gets a "low" not "medium" reputation rank as it is now "very old" in terms of service period. This is to penalize old service providers with bad service quality. For S_4, it remains 5 stars but they are red stars as S_4 becomes an "old" service provider now. In contrast, S_5 obtains 3 red stars, not 4 green stars as it becomes "old" and it didn't improve its services to "good" or "very good" trust level.

	Round=200			Round=300		
	trust	service period	Reputation Rank	trust	service period	Reputation Rank
S_1	very high (0.95)	very old	very good ($\grave{5}$)	high high (0.95)	very old	very good ($\grave{5}$)
S_2	high (0.8)	very old	good ($\grave{4}$)	high (0.8)	very old	good ($\grave{4}$)
S_3	medium (0.6)	very old	medium ($\grave{3}$)	medium (0.6)	very old	low ($\grave{2}$)
S_4	high (0.8)	new	very good as new ($\acute{5}$)	very high (0.9)	old	very good ($\grave{5}$)
S_5	high (0.55)	new	good as new ($\acute{4}$)	medium (0.6)	old	medium ($\grave{3}$)

Table 4: Study 5 - Result for Round 200 and Round 300

From the above study, we can observe some features of the proposed fuzzy model.

1. It is incentive to good service providers with good service quality for a long service period.

   For example, due to the good service quality, both S_1 and S_4 obtain 5 stars after some periods in round 300.

2. Additionally, it is incentive to new service providers (*Principle 2*).

   For example, in round 160, though the trust values of S_4 and S_5 are worse than S_3, they all get the "medium" reputation rank as S_4 and S_5 are "very new". In another example in round 200, though S_5 is in the same ("medium") trust level as S_3, it obtains 4 stars as it is "new service provider". In contrast, S_3 gets 2 stars.

3. On the other hand, it penalizes poor service quality particularly for "old" service providers.

For example, as S_3 remains $T_{S_3} \approx 0.6$ - a "medium" trust level in rounds 160, 200 and 300, its reputation rank drops from "medium" (3 stars) in rounds 160 and 200 to "low" (2 stars) in round 300.

4. Furthermore, the proposed model can differentiate services from "old" service providers and "new" service providers by two sets of colorful ranking stars.

   This aims to provide clearer information to service clients and prevent some new service providers from cheating clients after obtaining a good reputation rank. Namely, (1) a 5-red-star service provider is more reliable than a 5-green-star service provider; (2) a 5-green-star service provider should maintain its good service quality level so as to obtain 5 *red* stars. S_4 is the typical example (refer to Figure 4). Due to its good services, it gets 5 green stars in round 200 when it is "new". But at that moment, S_4 is slightly less comparable than S_1 who obtains 5 red stars already. Finally, as S_4 can maintain constantly the good service level, it obtains 5 red stars in round 300 when it becomes "old".

7 Conclusions

In this paper, we have proposed a trust evaluation model and a fuzzy logic based model for determining the reputation rank for service providers. This work has some advances in the following aspects. We believe they are critical to a trust management service authority with a large pool of both service providers and service clients.

1. In this paper, we present a framework for complex service oriented trust management, which is rule-based and event-driven. A new method based on a formula is also proposed for trust evaluation. It is adaptable to different applications by using different *system arguments* - α, β and λ.

2. This work addresses how to depict the trust establishment process. The designed trust evaluation formula adopts a generic approach, using a predefined basic curve function to determine the impact control function. It can not only reflect the nature of long-term and incremental trust establishment process, but also provide incentive measures to new service providers and service providers with "low" trust level.

3. This work adopts fuzzy logic to further take the service period length into account when determining the reputation ranks. The specific two colorful sets of reputation ranks can not only deliver clearer information to service clients, but also differentiate new service providers and old ones. This is a further incentive mechanism to encourage good services from service providers, protect new service providers, and penalize "poor" service quality from a long-existing service providers. The design flexibility is also provided to adapt to the features of different service application domains.

In addition, both analytical and empirical studies have been conducted for illustrating the properties of the proposed method.

For future work, detailed event category and rules should be studied. We will also study the distributed trust management architecture with the proposed rule-based and event-driven properties, where the collaboration of trust management authorities will be one of the research focuses. In addition, the dynamic assignment of initial trust values mentioned in Section 6.2 will be studied in detail.

References

[1] *eBay*. http://www.eBay.com/.

[2] *GNutella*. http://www.gnutella.com/.

[3] *Napster*. http://www.napster.com/.

[4] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *HICSS*, pages 9–17, 2000.

[5] A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In *SIGCOMM05 Workshops*, 2005.

[6] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation based approach for choosing reliable resources in peertopeer networks. In *Proceedings of ACM CCS'02*, pages 207–216, Washington DC, USA, November 2002.

[7] N. Griffiths. Task delegation using experience-based multidimensional trust. In *Proceedings of the 4th International Joint Conference on Autonomous Agents in Multi-Agent Systems (AAMAS-05)*, pages 489–496, 2005.

[8] N. Griffiths. Enhancing peer-to-peer collaboration using trust. *Expert Syst. Appl.*, 31(4):849–858, 2006.

[9] N. Griffiths, K.-M. Chao, and M. Younas. Fuzzy trust for peer-to-peer systems. In *ICDCS Workshops*, page 73, 2006.

[10] R. V. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW*, pages 403–412, 2004.

[11] T. Huynh, N. Jennings, and N. Shadbolt. An integrated trust and reputation model for open multi-agent systems. *Autonomous agents and multiagent systems*, 13(2):119–154, 2006.

[12] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th International WWW Conference*, Budapest, Hungary, May 2003.

[13] K.-J. Lin, H. Lu, T. Yu, and C. en Tai. A reputation and trust management broker framework for web applications. In *Proceedings of The 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05)*, pages 262–269, March 2005.

[14] S. Marti and H. Garcia-Molina. Limited reputation sharing in P2P systems. In *Proceedings of ACM EC'04*, pages 91–101, New York, USA, May 2004.

[15] I. Ray and S. Chakraborty. A vector model of trust for developing trustworthy systems. In *Proceedings of ESORICS 2004*, volume LNCS 3193, Springer-Verlag, pages 260–275, 2004.

[16] J. Sabater and C. Sierra. REGRET: A reputation model for gregarious societies. In *Proceedings of the First International Joint Conference on Autonomous Agents in Multi-Agent Systems (AAMAS-01)*, pages 475–482, 2002.

[17] S. Song, K. Hwang, R. Zhou, and Y.-K. Kwok. Trusted p2p transactions with fuzzy reputation aggregation. *IEEE Internet Computing*, 9(6):24–34, 2005.

[18] J. G. Steiner, C. Neuman, and J. I. Schiller. Kerberos: An authentication service for open network systems. In *Proceedings of the USENIX Winter 1988 Technical Conference*, 1988.

[19] L.-H. Vu, M. Hauswirth, and K. Aberer. Qos-based service selection and ranking with trust and reputation management. In *Proceedings of 13th International Conference on Cooperative Information Systems (CoopIS 2005)*, Oct 31-Nov 4 2005.

[20] Y. Wang, K.-J. Lin, D. S. Wong, and V. Varadharajan. The design of a rule-based and event-driven trust management framework. In *The IEEE International Conference on e-Business Engineering (ICEBE 2007)*, pages 97–104, Hong Kong, October 2007.

[21] Y. Wang and V. Varadharajan. Interaction trust evaluation in decentralized environments. In K. Bauknecht, M. Bichler, and B. Pröll, editors, *Proceedings of 5th International Conference on Electronic Commerce and Web Technologies (EC-Web'04)*, volume LNCS 3182, Springer-Verlag, pages 144–153, Zaragoza, Spain, August-September 2004.

[22] Y. Wang and V. Varadharajan. T*rust*^{2}: Developing trust in peer-to-peer environments. In *Proceedings of 2005 IEEE International Conference on Services Computing (SCC 2005)*, pages 24–31, Orlando, Florida, USA, July 2005.

[23] Y. Wang and V. Varadharajan. Two-phase peer evaluation in P2P e-commerce environments. In *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Serivce (EEE-05)*, pages 654–657, Hong Kong, China, March 29-April 1, 2005.

[24] L. Xiong and L. Liu. PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Trans. on Knowlege and Data Engineering*, 16(7):843–857, 2004.

[25] B. Yu, M. P. Singh, and K. Sycara. Developing trust in large-scale peer-to-peer systems. In *Proceedings of 2004 IEEE First Symposium on Multi-Agent Security and Survivability*, pages 1–10, August 2004.

[26] G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence Journal*, 9:881–908, 2000.

[27] L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.