Elsevier Editorial System(tm) for Expert Systems With Applications
Manuscript Draft

Manuscript Number: ESWA-D-12-00647

Title: On the Analysis of Reputation for Agent-based Web Services

Article Type: Full Length Article

Keywords: Agent-based web services, Reputation

Corresponding Author: Dr. Jamal Bentahar, Ph.D.

Corresponding Author's Institution: Concordia University

First Author: Jamal Bentahar, Ph.D.

Order of Authors: Jamal Bentahar, Ph.D.; Babak Khosravifar, Ph.D.; Mohamed Adel Serhani; Mahsa Alishahi, M.Sc.

Highlights

- Theoretical analysis of reputation-based infrastructure for agent-based web services.
- Computation of incentives and penalties to make the system components trustful.
- Extensive simulation study of multiple scenarios confirming the theoretical findings.

On the Analysis of Reputation for Agent-based Web Services

Jamal Bentahar, Babak Khosravifar, Mohamed Adel Serhani, and Mahsa Alishahi

Concordia Institute For Information Systems Engineering, Concordia University, Canada

Abstract

Maintaining sound reputation requires robust control and investigation. In this paper, we analyze a reputation mechanism that objectively maintains accurate reputation evaluation of selfish agent-based web services. In the proposed framework, web services are ranked using their reputation as result of provided feedback reflecting consumers' satisfaction. However, selfish web services may alter their public reputation level by managing to get fake feedback. We investigate the payoffs of different scenarios by focusing on the issues that discourage web services to act maliciously. We also analyze the details of the proposed mechanism by discussing simulation and empirical results that fully depict the system parameters and show the feasibility of the proposed approach.

Keywords:
Agent-based web services, Reputation.

1. Introduction

Web services are deployed to maintain continuous interactions between loosely coupled applications. Abstracting web services using knowledge-empowered agents will benefit them from flexible and intelligent interactions that those agents are able to manage [8, 22]. However, because agents are autonomous and selfish, a major issue in agent-based settings is reputation, which is a significant factor that regulates the process of service selection [6]. During recent years, extensive

Email address: bentahar@encs.concordia, b_khosr@encs.concordia.ca, serhanim@uaeu.ac.ae, mahsa_alishahi@yahoo.com (Jamal Bentahar, Babak Khosravifar, Mohamed Adel Serhani, and Mahsa Alishahi)

work has been done to address reputation in virtual teams, multi-agent systems, and service environments [4], [10], [12], [16], [18]. Many of the proposed models are based on data collected from different sources that are considered reliable. However, this might not be the case in many concrete situations.

There is a different point of view in addressing reputation, which is maintaining an incentive-based sound reputation mechanism [23, 24]. In this perspective, the ideal case is the situation in which rational agents have incentives to act such that ultimately the whole environment turns into a truthful network of agents. Maintaining this environment requirers designing a sound reputation framework with some defined characteristics that establish incentives and penalties. The concept of sound reputation assessment is being considered in very few attempts. This paper aims to advance the state-of-the-art by addressing this open issue. A detailed literature review will be given in the discussion section.

The objective of this paper is to propose a reputation mechanism by building on and extending a collusion-resistant reputation framework we proposed in [14], in which a game-theoretic analysis to maintain accurate reputation assessment for agent-based web service systems has been developed. In this reputation assessment framework, web services are ranked using users' feedback posted with respect to the quality and satisfaction of the received service. The goal is to investigate the payoffs obtained through different situations and propose solutions that allow building collusion-resistant reputation mechanism. In this paper, we extend this framework by expanding the reputation management, considering more collusion scenarios, and providing more theoretical and simulation results and analysis. Moreover, we discuss in detail the system implementation and simulation environment. More details regarding the contributions of this paper are provided in the following subsection.

Contributions. In this paper, we consider agent-based[1] web services and address the aforementioned problems by providing accurate reputation assessment in open environments in which web services are selfish and utility maximizers. The reputation is accurately assessed mainly as result of incentives provided to participating agents in order to act truthfully and avoid malicious actions. We aim to advance the state-of-the-art by analyzing the system's parameters. We investigate the incentives to cheat that malicious web services can have and incentives to act truthfully while being aware of the possible penalties assigned by a special

[1]We assume that web services are abstracted and associated with agents able to reason, interact with others and make decisions.

agent called *controller agent*. In fact, we theoretically and empirically analyze the obtained payoffs according to the agent's followed strategy (i.e. acting truthfully or maliciously). In our simulations, we discuss the obtained results and elaborate on the outcome of different strategies that participants (or players) might choose. We investigate incentives for web services to act truthfully and identify the state that is socially acceptable for all the participants.

Organization. The rest of the paper is organized as follows. In Section 2, we explain some preliminaries we use in the framework. Section 3 introduces the reputation mechanism and its components. In Section 4, the possible alterations on reputation are discussed and then theoretically analyzed in Section 5. In our analysis, incentives that encourage agents act truthfully are investigated. Section 6 provides detailed experimental analysis and discussions that are inspired by the theoretical results presented in Section 5. The obtained results show the potential of the proposed framework in maintaining collusion-resistant reputation mechanism. Section 7 discusses related work and Section 8 concludes the paper and identifies possible future work.

2. Preliminaries

Service Consumers are intelligent agents that continuously seek for services provided by some other agents. Each service consumer agent is equipped with a purchase mechanism that facilitates its request initiation process. Moreover, this mechanism analyzes the received quality of service (QoS) and generates corresponding feedback. Each service consumer c holds an acceptable quality threshold QT_c that is compared against the received QoS to decide about posting positive or negative feedback. Service consumer agents rationally follow their predefined goal, which is obtaining the most satisfactory QoS over time. However, some of them could be encouraged by some web services to temporarily support them by reporting false feedback, which could be temporarily compatible with the goals of these service consumer agents. This issue will be discussed in details later in this paper.

Web Services are agent-based services engaged in answering service consumers' requests. As mentioned earlier, web services might initiate some collusion with consumers that might be beneficial for both parties. Each web service agent i is equipped with a selling mechanism enabling the agent to approach its predefined goal. This goal is to have a maximum reputation (which results in maximum market share). The reputation R_i of the agent i is a value, which is computed as result of feedback aggregation kept in the feedback file and is su-

pervised by the controller agent (both the feedback file and controller agent are explained later as preliminaries). Each web service agent holds parameters regarding quality of service Q_i, and market share M_i that are used by the selling mechanism to reach the predefined goal.

Feedback File is used in the proposed system to gather posted feedback from service consumers. Consumers' feedback are aggregated to reflect the total credibility of web services. The feedback file is required to be supervised against malicious actions maintained by some selfish agents in the environment (selfish consumer agents and web services). Malicious actions are actions that violate the feedback file by posting some false feedback resulting in falsely reputation increase of some web services.

Controller Agent Cg is the assigned agent that takes the feedback file under surveillance. Cg is responsible of removing false feedback that support particular web services. Cg is equipped with an investigation mechanism enabling the agent to investigate the recent feedback aggregated in the feedback file and recognize the faked ones by investigating further actions of the benefitted web service. In general, Cg might fail to accurately detect fake feedback (false negative error) or similarly might recognize truthful feedback as fake (false positive error). Therefore, Cg holds a parameter regarding its accuracy A_{Cg}. Obviously, the controller agent's goal is to maximize its accuracy level. To this end, this agent would not act very tough while supervising the feedback file. Following this harsh attitude, Cg generates high ratio of false negatives, which result in decreasing its accuracy level. In general, the controller agent requires a reasoning mechanism to analyze web services behavior, maintain sound feedback file, and discard fake posted feedback.

Figure 1 illustrates the links among the different entities in the proposed framework. Consumer agents take the initiative by looking for services using a service selection mechanism. These agents might contact previously known web services (red direct link in the figure) or refer to the controller agent to get updated with the most recent reputation ranking of web services (indirect link and blue discontinue arrows in the figure). The red direct link is stronger compared to the one referred by others. However, over time the consumer agent gets to know new and high quality web service agents. Once the web service is selected (i.e. the request is sent) and the corresponding service is provided, the consumer agent posts a feedback to the feedback file through the service selection mechanism. The controller agent updates the reputation ranking by aggregating the accumulated feedback. In this process, active web services would ask the controller agent for advertisement, which means they require to be considered in the reputation ranking provided to

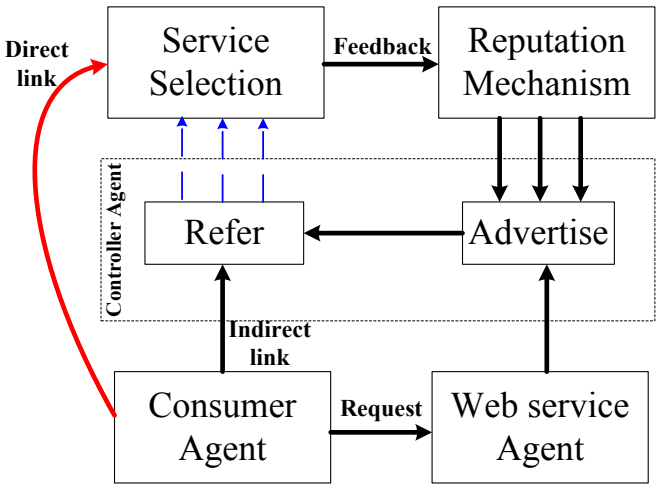

Figure 1: Architecture of the proposed framework

the consumer agents.

3. Reputation Mechanism

The reputation mechanism enables service consumers to evaluate the credibility of web services they want to invoke. In this system, Cg updates its surveillance algorithm and web services learn from their surrounding environment to make good decisions. The main result of this paper is that over time, agent-based web services will get encouraged to act truthfully and discouraged to increase self reputation level with fake feedback. In the assessment process, there are key factors that we need to measure from the feedback. These factors, which reflect the health of a typical web service i are [3]: quality (Q_i), and market share (M_i).

In the rest of this part, we explain each factor, then, we formalize the reputation of a typical web service as aggregation of these factors.

3.1. Reputation Parameters

Quality Q_i is used to measure the mean rate that is given to web service i representing its quality in handling the users' requests in a timely fashion. Q_i is computed by collecting all the rates given to the web service to be evaluated. For simplicity reasons, but without affecting the main results of this paper, we consider discrete feedback having the form $(+)$ for positive and $(-)$ for negative feedback. Let $\mathcal{P}_i$ be the set of positive feedback a web service i has received and

$\mathcal{T}_i$ be the set of all the feedback i has received since published in the web. Thus, the acceptance factor would be simply computed in Equation 1.

$$Q_i = \frac{|\mathcal{P}_i|}{|\mathcal{T}_i|} \tag{1}$$

where $|\mathcal{P}_i|$ and $|\mathcal{T}_i|$ are the cardinality of $\mathcal{P}_i$ and $\mathcal{T}_i$ respectively.

Time Discount. In the trivial way of calculating Q_i in Equation 1, only the number of positive feedback is compared with the total number of feedback. This calculation is not highly efficient when the environment is equipped with selfish agents that dynamically change their behaviors. We need then to consider the interactions history in a more effective way by giving more importance to the recent information. This can be done using a timely relevance function. In this paper, we consider the following function similar to the one used in [7] and [13]: $e^{-\lambda \Delta t_k}$, where Δt_k is the time difference between the current time t and feedback k submission time t_k and λ ($\lambda \in [0, 1]$) is the recency scaling factor (i.e. scaling time values). Therefore, $e^{-\lambda \Delta t_k}$ is a weighted feedback. Consequently, the quality factor Q_i of web service i can be measured as shown in Equation 2.

$$Q_i = \frac{\int_{k \in \mathcal{P}_i} e^{-\lambda \Delta t_k} dt_k}{\int_{k \in \mathcal{T}_i} e^{-\lambda \Delta t_k} dt_k} = \frac{\int_{k \in \mathcal{P}_i} e^{-\lambda (t - t_k)} dt_k}{\int_{k \in \mathcal{T}_i} e^{-\lambda (t - t_k)} dt_k}$$

$$\Rightarrow Q_i = \frac{\frac{1}{\lambda} e^{-\lambda t} e^{\lambda t_k}|_{k \in \mathcal{P}_i}}{\frac{1}{\lambda} e^{-\lambda t} e^{\lambda t_k}|_{k \in \mathcal{T}_i}} \tag{2}$$

We notice that:

$$\lim_{|\mathcal{P}_i| \to \infty} \int_{k \in \mathcal{P}_i} e^{-\lambda \Delta t_k} dt_k = \int_0^{\infty} e^{-\lambda \Delta t_k} dt_k = \frac{1}{\lambda}$$

Consequently:

$$\lim_{\substack{|\mathcal{P}_i| \to \infty \\ |\mathcal{T}_i| \to \infty}} Q_i = \frac{\frac{1}{\lambda}}{\frac{1}{\lambda}} = 1$$

Intuitively, this means when the number of (positive) feedback is huge, the quality converges towards 1, which reflects the popularity of the concerned web service.

Market Share M_i is a parameter that indicates the extent to which the web service is active in the providers' network. This basically affects the popularity property of the web service in the sense that high service load together with high provided quality bring higher number of consumers because the web service is considered successful. In the proposed reputation mechanism, a successful web service is the one that receives high number of positive feedback, which brings high request number in the future. Equation 3 defines the market share for web service i, which satisfies the popularity property. In this equation, the numerator represents the total feedback received for i, whereas the denominator is the integrated value for all recorded feedback ($\mathcal{G}$) for all active web services controlled by Cg. As in Equation 2, the time discount is also considered.

$$M_i = \frac{\int_{k\in\mathcal{T}_i} e^{-\lambda\Delta t_k} dt_k}{\int_{k\in\mathcal{G}} e^{-\lambda\Delta t_k} dt_k} = \frac{\int_{k\in\mathcal{T}_i} e^{-\lambda(t-t_k)} dt_k}{\int_{k\in\mathcal{G}} e^{-\lambda(t-t_k)} dt_k}$$

$$\Rightarrow M_i = \frac{1}{\frac{1}{\lambda} e^{-\lambda t} e^{\lambda t_k}|_{k\in\mathcal{G}-\mathcal{T}_i}} \tag{3}$$

where $\mathcal{G} - \mathcal{T}_i \neq \emptyset$

3.2. Reputation Assessment

Taking the aforementioned parameters into account, we propose the estimated total reputation for web service i that is crucial for its selection process and overall survival in the environment. First, we weight each parameter with a coefficient ($\beta_1 + \beta_2 = 1$). The value of each coefficient reflects the importance of the associated parameter. Therefore, we obtain the estimated reputation value r_i regarding web service i in Equation 4. The reputation value r_i is only deduced from the feedback posted on the feedback file. However, at some point this value might not be the one that is publicly announced to the service consumers. These agents refer to the controller agent for the most accurate information regarding web services' reputation value and use the obtained value as a measure of reliability.

$$r_i = \beta_1 Q_i + \beta_2 M_i \tag{4}$$

In the proposed reputation mechanism, Cg is dedicated to manage the reputation assessment and make it sound. Therefore, on top of the rates that a web service i receives from collecting the consumers' feedback (r_i), Cg is eligible to

offer a rate reflecting its own point of view regarding the web service's reputation. The rate (C_i) that is given by Cg affects web service i's total reputation. If C_i is so low (lower than r_i), that means web service i has a bad-reputed history that might encourage users to avoid the web service agent. If the rate is relatively high (higher than r_i), the consumers rely more on what they have evaluated from the files. Equation 5 gives the formula of computing the total reputation R_i, which is defined so that it satisfies the conservative property. Such a property consists in giving higher weight to the lowest feedback. This is achieved because if $r_i > C_i$, then $\gamma_2 > \gamma_1$ where γ_1 is the weight of r_i and γ_2 is the weight of C_i.

$$R_i = \gamma_1 r_i + \gamma_2 C_i \text{ such that: } \begin{cases} \gamma_2 - \gamma_1 = & r_i - C_i \\ \gamma_1 + \gamma_2 = & 1 \end{cases} \tag{5}$$

4. Reputation Alteration

4.1. Collusion (Web Service Perspective)

In an open environment populated with agents who are aimed to achieve their predefined goals, some agents may choose strategies that only benefit themselves and in general are not good strategies for the whole system. In a multi-agent system of web services and service consumers, selfish web services might desire to increase self reputation to a level that have not been ranked to. The faked reputation level would temporarily bring additional service requests towards the malicious web service. However, the goal of the malicious web service is to keep the faked reputation as much as possible. A web service would collude with some consumer agents to provide continuous positive feedback supporting the agent. These consumer agents have to be encouraged to collaborate in the collusion process by obtaining some privileges, such as low service fee, outstanding QoS, etc.

To discuss the collusion concept in more details, consider web service i, which aims at increasing its quality report Q_i and market share M_i. In a collusion process, the malicious web service i faces a major risk reflected by the rate C_i submitted by the controller agent in the sense that if the malicious action is being detected, C_i would be fairly small reflecting bad history of the web service. The submitted rate via the controller agent affects the reputation value of the web service to some certain extent. In case of acting truthfully, the web service would obtain a better reputation compared to the case where its fake reputation is being recognized and thus, a low rate is submitted. To this end, a malicious web service, that is aiming to increase self reputation level, has a main challenge, which is the

decision of acting maliciously. This means that even though the web service is capable of colluding with some consumers, there might be some reasons that prevent the agent from initiating such an action. Thus, to account for the web service's willingness to act maliciously, we introduce a willingness parameter w_i. In this case, the expected reputation values of acting 1) truthfully ($Exp(R_i|Truth)$) and 2) maliciously ($Exp(R_i|Mal)$) should be compared. A web service is willing to act maliciously when the expected reputation value of colluding is more than the one of acting truthfully. The parameter w_i is set as follows:

$$w_i = 1 \quad if \quad Exp(R_i|Mal) > Exp(R_i|Truth)$$
$$w_i = 0 \quad if \quad Exp(R_i|Mal) \leq Exp(R_i|Truth)$$

The expected values of reputation in different cases are computed in Equations 6 and 7. In Equation 6, q and $1-q$ are respectively probabilities of being detected and ignored by the controller agent. The parameter $\overline{r_i}$ is the altered reputation as a result of collusion and the parameter $\overline{C_i}$ is the rate the controller uses if the collusion is detected. The value of $\overline{r_i}$ is greater than r_i thanks to the submitted faked feedback by the colluding consumers. Likewise, the value of $\overline{C_i}$ is less than C_i as long as the controller agent would penalize more in case of collusion being detected.

$$\begin{aligned} Exp(R_i|Mal) &= q(\gamma_1\overline{r_i} + \gamma_2\overline{C_i}) + (1-q)(\gamma_1\overline{r_i} + \gamma_2 C_i) \\ &\Rightarrow Exp(R_i|Mal) = \gamma_1\overline{r_i} + \gamma_2(q\overline{C_i} + (1-q)C_i) \end{aligned} \tag{6}$$

$$Exp(R_i|Truth) = \gamma_1 r_i + \gamma_2 C_i \tag{7}$$

In general, a normal web service that is acting truthfully, expects the actual reputation level when there is nothing wrong in the feedback file. Later in this paper, we also consider the false positive cases where the truthful action also gets penalized and thus, the expected reputation rank should be updated. The malicious web service also has some other challenges, which are beyond the scope of this paper: 1) when to act maliciously; 2) who to collude with; and 3) how many fake feedback to provide. To be focussed, in this paper we only consider the malicious actions consisting of providing positive feedback. The fact of providing negative feedback, for example a web service can (indirectly) provide continuous negative feedback to a concurrent web service, is also important to be considered in future work.

4.2. Collusion Scenario

The collusion scenario could be initiated by either the consumer agent or web service agent. In this paper, we assume that this procedure is initiated by the malicious web service that is already willing to collude. This means that for the web service, the expected reputation rank with respect to collusion is more than the one of following a truthful action. Before discussing the collusion scenario, we also need to provide some insights regarding the consumer agent. In the proposed framework, the consumer agents are aimed at obtaining the best service quality and therefore, they need to seek for the best reputed web service. In order to find the best web service, the consumer agent is required to refer to the controller agent to obtain the most updated web services' ranks. Otherwise, the consumer has to consider its history of service selection and accordingly, requests the most reliable web service. To this end, on top of the eagerness of high quality service, the consumer agent requires from the controller agent to be updated. Therefore, if the consumer agent accepts the malicious web service's invitation for collusion, the corresponding risk of reaction from the controller agent needs to be taken into account. If the controller agent recognizes the collusion, the recognized consumer would not benefit from the controller agent's services for some certain time, which affects the consumer agent's expected service quality. To be focussed, in our collusion analysis, we skip the details of collusion willingness regarding consumer's point of view and mainly consider the collusion process initiated by the web service and collaborated by the consumer agent. This limitation does not affect the obtained results.

In the collusion scenario, the malicious web service and consumer agent agree on a collusion that brings some benefits to both colluding parties. Web service i gets extra positive feedback that increase its reputation value out of the feedback file. The enhanced reputation value $\overline{r_i}$ is computed in Equation 8.

$$\overline{r_i} = \beta_1 \overline{Q_i} + \beta_2 \overline{M_i} \tag{8}$$
$$where \quad \overline{Q_i} = Q_i(1 + f_Q) \quad and \quad \overline{M_i} = M_i(1 + f_M)$$
$$f_Q = f(|\mathcal{P}_i|, |\mathcal{T}_i|, |F_i|) \quad and \quad f_M = g(|\mathcal{T}_i|, |\mathcal{G}|, |F_i|)$$

In Equation 8, the factors f_Q and f_M respectively represent the update factor regarding web service i's quality and market share parameters. These factors are functions of current status of web service i in the feedback file ($|\mathcal{P}_i|$ denotes the number of positive feedback for i, $|\mathcal{T}_i|$ the number of all feedback for i, $|F_i|$ the number of faked submitted feedback for i, and $|\mathcal{G}|$ the number of all recorded

feedback for all active web services). The functions f and g are monotonically increasing with respect to $|\mathcal{P}_i|$ and $|\mathcal{T}_i|$ respectively (note that $F_i \subseteq \mathcal{P}_i \subseteq \mathcal{T}_i \subseteq \mathcal{T}$). Overall, the evaluated rate of reputation of web service i would be increased after collusion. The colluding consumer obtains higher quality of service with low fees, which exceeds its expectations if it acts truthfully. To this end, if the collusion is not recognized by the controller agent, the web service gains higher reputation value and the colluding consumer obtains better deal.

4.3. Detecting Malicious Actions

In the proposed framework, the controller agent serves as representative of the reputation system. This agent is aimed to seize malicious acts and maintain a sound reputation system. In fact, Cg's challenges are: 1) how cautious to be (how to set the certainty parameter C_i explained earlier, which is proposed by the controller agent to measure the confidence this agent has on web service i); 2) always being careful not to generate false alarms (detections); and 3) setting proper penalties to avoid detection failures [21]. Failing to detect malicious acts leads to false alarms, which are composed of two cases: the case of penalizing truthful agents (web service and consumer) by mistake (false positive), and the case of ignoring malicious agents by mistake (false negative). When a web service i is under investigation by Cg for a possible malicious action, a reputation value during the investigation time is calculated as shown in Equation 5 and denoted by μ_i. This means only the feedback received during this period are considered.

In the penalizing scenario, the controller Cg applies a penalty that affects the penalized web service with respect to its reputation value. Also the colluding consumer is penalized in the sense that it will not profit from the controller's services, for instance in terms of getting updated regarding the most recent reputation ranking. To this end, Cg analyzes the applied penalty to minimize malicious acts in the network. One clue would be applying a relatively high penalty to maintain a strong control over the feedback file. Such (harsh) manner does not necessarily imply a high performance for Cg because penalizing truthful agents imposes negative influence on its accuracy level. Therefore, Cg always looks for an optimum penalty value, which minimizes malicious acts and maximizes self-performance level. To detect malicious actions, Cg is then required to be equipped with a mechanism to analyze the interactions of the web services with the consumers. During the investigation, Cg aims to make the best decisions to update its significance level, which affects the accuracy of the rate C_i. Also, Cg needs to learn from the current penalties the information that is used in further detections. In

our framework, we suggest using the t-statistic as a measurement of error and detection criteria that Cg uses to capture suspected behavior of the web services. Inequation 9 shows this detection criteria where σ_i is the standard deviation of the reputation of i during the investigation period. The threshold ν is set by the controller agent and is application-dependant. The t-statistic is used because the mean and standard deviation of a sample reflecting the investigation time are to be considered, instead of the parameters of the whole periods since the activation of the web service. In fact, this error computes an estimate for the number of standard deviations the given sample (reflecting the behavior of web service i during the investigation time) is from the mean reputation value of i.

$$|\frac{R_i - \mu_i}{\sigma_i}| > \nu \tag{9}$$

4.4. Suspecting Phase

The controller agent initiates a secret suspecting phase about web service i when Inequation 9 is satisfied. In this stage, the behavior of web service i is under closer investigation. The controlled web service is better off doing its best not to get penalized. If the web service did not act according to the raise in its reputation level ($\Delta R_i = R_i - \mu_i$), Cg might penalize the agent for faked feedback. If not, Cg would ignore the investigation and consider the raised reputation level as a normal improvement.

Although Cg uses its history of investigations together with the learned information collected from the environment, always there is a chance of mistake that would cause wrong decision. In general there are four cases: (c_1) the web service acts maliciously and accordingly gets penalized by Cg; (c_2) the web service acts maliciously, but gets ignored by Cg; (c_3) the web services acts truthfully, but gets penalized by Cg; and (c_4) the web service acts truthfully and Cg considers its action normal. Cases (c_1) as true positive and (c_4) as true neutral represent the fair situations. However, cases (c_2) as false negative and (c_3) as false positive are failures, which decrease Cg's performance. In the following, we analyze the scenario for each case and conclude with a general payoff gained by each involved party.

The concept of reputation update is about changing ones reputation level, which influences social opinions. Adversely, the reputation is updated once Cg applies some penalties to detected malicious acts. In general, the feedback file is subject to modifications performed by some non-authorized agents or an authorized controller agent. The interaction between a selfish web service and the controller agent can be modelled as a repeated game over time. The game con-

sists of actions (made by the web service) and reactions (made by Cg). Here we consider the aforementioned four cases and obtain the corresponding payoffs of each case. The obtained reputation value for web service i after web service i's action together with Cg's reaction (which could be estimated by $Exp(R_i|Mal)$ or $Exp(R_i|Truth)$ that are computed in Equations 6 and 7) is denoted OR_i. We use R'_i to denote the actual (or fair) reputation that has to be set for web service i. However the current set value (OR_i) might be different from R'_i because of false positives or negatives. In the rest of this paper, we consider the effect of collusion and penalties on the reputation of web services.

According to the decision made by the controller agent, four outcomes are to be considered and we categorize them as follows: false negative (FN), false positive (FP), true positive (TP), and true neutral (TN). Hereafter, we explain and analyze each one of them.

Malicious Act not Penalized (FN). This is the case where web service i acts maliciously for instance by colluding with some users and Cg does not recognize it. Thus, web service i increases its reputation level. We refer to this improvement as Imp_i. Imp_i is in fact the increased reputation that is obtained by increasing r_i value. We also refer to the assigned penalty value as Pn_i. This value is set by Cg considering the past history of i and is updated through time elapse. Equation 10 gives the corresponding values for the obtained reputation level OR_i and the actual (fair) reputation value R'_i.

$$OR_i = R_i + Imp_i; \quad R'_i = R_i - Pn_i \tag{10}$$

$$OR_i - R'_i = Imp_i + Pn_i = \omega \tag{11}$$

The difference between the actual (fair) and current reputation values reflect the payoff that we can use in our game-theoretic analysis (Equation 11). We use this difference to be able to compare the possible scenarios in terms of reputation level. For simplicity, we set $Imp_i + Pn_i$ to ω. The difference here is positive, which means the web service gets benefit of $+\omega$.

Truthful Act Penalized (FP). This is the case where web service i acts normal, but Cg decides to penalize him. In this case, i would lose its actual reputation value as shown in Equation 12. Equation 13 shows the obtained payoff, which is a negative value in this case. This reflects the fact that web service i loses ω. This basically affects Cg as well in the sense that a wrong decision is being made, so there is a negative effect applied to its accuracy level.

$$OR_i = R_i - Pn_i; \quad R'_i = R_i + Imp_i \tag{12}$$

$$OR_i - R'_i = -\omega \tag{13}$$

Truthful Act not Penalized (TN). This is the ideal case where i acts normal and Cg refuses to penalize. In this case the current reputation is the same as the actual reputation ($OR_i = R'_i$). Thus, the payoff assigned to i is zero ($OR_i - R'_i = \omega = 0$).

Malicious Act Penalized (TP). This is also the fair case where web service i acts maliciously hoping to increase self reputation level. Cg detects the action and thus, applies the penalty. In this case, i loses both the penalty and improvement ($-Pn_i - Imp_i = -\omega$).

In the cases considered here, we also need to discuss the obtained payoff for the consumer and controller agents. However, in this paper we only focus on the controller agent and skip the details of the penalizing procedure regarding consumer agents. Nevertheless, we assume that the penalized user would not be able to get the controller agent's services, for instance receiving information about the reputation ranking of web services. Therefore, the colluding consumer would be also influenced. Regarding the controller agent's payoff, one basic idea that we use in the rest of this paper is to consider the accuracy of Cg in detecting the malicious acts and according to the performed reaction, we set the payoff. Therefore, in the first two cases where the detections are wrong, Cg obtains a negative payoff (say $-\pi$), and in the second two where the decisions are correct, Cg obtains the positive payoff (say $+\pi$). The payoff is not received immediately after Cg's reaction, but after a period of time. The main question is then who is going to pay the controller agent and how? Different scenarios can be applied and in this paper we assume that web services and consumers contribute together to the Cg's payoff by paying a fee to the controller for making the system secure and fairly competitive, which is of a great significance for both the consumers and web services. In such a setting, $-\pi$ means less income for Cg because some web services and users stop paying the fees. Here we analyze the different cases according to the four outcomes discussed earlier.

Malicious Act not Penalized (FN). In this case, some bad web services are get promoted and ranked high. This can quickly be recognized by the competitors (web services) and some users who had previous experiences with those bad web services. Therefore, those competitors and users will refuse paying the controller as the system is no more secure for the users and fairly competitive for honest web services.

Truthful Act Penalized (FP). In this case, some honest web services are unfairly penalized, which make them stop paying the controller. Other honest com-

petitors and some users who know the reputation of the penalized web services will feel the system unfair and insecure. They can consequently decide to stop contributing in the payment of Cg and get its services.

Truthful Act not Penalized (TN). This is the situation where all the web services and users are satisfied as the system seems secure and working correctly, which brings more competition for the benefit of the users. Web services and users will then continue supporting Cg and requesting its services.

Malicious Act Penalized (TP). In this situation, some users and competitors who know the penalized web services will feel satisfied as the system is getting more secure and fairly competitive. This will encourage them to increase the Cg's payment to counterbalance the system against the loss caused by the penalized web services who will probably cease participating in the payment of the controller.

The reason behind this payoff assumption is the fact that we consider the interaction between the web service and controller agent as a repeated game. The repeated game theory brings the concept of learning in detection and penalizing process. Such a repeated game would rationally help web services to obtain experiences from the past interactions with Cg and thus, know whether to act maliciously or truthfully. The objective of the repeated game is to maintain a sound reputation mechanism in which the controller agent is getting stronger in reputation updates, and the web services are discouraged to act maliciously.

5. Game Theoretic Analysis and Simulation

This section is dedicated to analyze the incentives and equilibria of reputation mechanism using the feedback file. Since the challenge is on the reputation (from web service's point of view, either to act maliciously, i.e. fake F or act truthfully, i.e. act normal N) and accuracy of the feedback file (from Cg's point of view), we model the problem as a two-player game. The active web services are of type good S_G or bad S_B ($P[S_G]$ and $P[S_B]$ represent the portion of each in the environment, e.g. 0.7 and 0.3). Web services of type good are more reliable and likely to act honestly, while the bad ones are more likely to act maliciously. The types are labelled with Cg's opinion imposed by web service's general reputation in the system. Let $Pr[N|S_G]$ (resp. $Pr[N|S_B]$) be the probability that a web service of type good (resp. bad) acts normal. In general, Cg's expected value for normal action from a typical web service is:

$$Pr[N] = P[S_G]Pr[N|S_G] + P[S_B]Pr[N|S_B] \tag{14}$$

where $Exp(R_i|Truth)$ and $Exp(R_i|Mal)$ that are computed in Equations 6 and 7 are good estimators of $Pr[N|S_G]$ and $Pr[N|S_B]$ respectively. For instance, the probability that a web service of type good to act truthfully can be estimated to be the expected reputation of this web service given that it acts truthfully.

The set of pure strategies for web services is defined as $st = \{F, N\}$. This chosen strategy imposes the behavior that the web service shows and thus, the controller agent observes after the action is occurred. Cg also chooses between two strategies: penalizing (P) and not penalizing, which means ignoring (I) the event. We consider a payment function χ associated to the sequence of actions performed by web services. The payment mechanism is defined as follows: $\chi : st \times st^{M-1} \mapsto [-\omega, +\omega]$, where M is the number of actions performed during the past and current periods and $-\omega$ and $+\omega$ are explained and computed in equations 11 and 13. Thus, $\chi(O_i, O_{-i})$ represents the assigned payoff to web service i when it selects $O_i \in st$ at current moment and $O_{-i} \in st^{M-1}$ represents its $M-1$ previous chosen strategies during $M-1$ periods. There is a similar payoff function for Cg that assigns values in the range $[-\pi, +\pi]$. In the rest of this section, we start by analyzing the one-shot game, which is then extended to continuous game.

Proposition 1. *In one-shot game, penalizing a fake action is the unique Nash equilibrium.*

Proof. Clearly acting fake by web service i, controller agent Cg would have a best utility if penalizing strategy is chosen rather than ignoring. On the other hand, if Cg chooses to penalize, i would not change its chosen strategy since in both cases i will lose $-\omega$. Consequently, penalizing a fake action is a Nash. Adversely, the normal act by i would let Cg to ignore. However, if the strategy is to ignore (by Cg), the best strategy for i is to act fake. Therefore, there is no Nash in ignoring the normal act. Therefore, the obtained Nash is unique. □

In one-shot game, players only consider the present information and they rationally tend to stay in fake-penalized state. This unique Nash is a good situation for Cg, but not for i. We need to study a socially better situation for both players when they learn the best strategies over time. This can be done by considering the repeated game. If i can estimate the expected payoff with respect to Cg's response, it might prefer acting normal. In fact, this issue is how to make agents (i and Cg) converge to a Pareto-Optimal [1], which is the best situation for both players. We call this situation Pareto-Optimal Socially Superior.

Definition 1. ***Pareto-Optimality.*** *A situation in a game is said to be Pareto-Optimal once there is no other situation that makes at least one player better off without making any other player worse off.*

In the following, we extend the one-shot game to the repeated game over periods of time. Therefore, following different strategies in time intervals will generate the corresponding payoffs to the players. At a given moment, Cg would decide whether to continue or stop investigating. To this end, e_0 is referred to as the case of doing no investigation effort and basically ignoring all actions. Otherwise, the best effort is made by Cg doing investigation. Cg has to decide about a proper strategy and obviously, if it chooses e_0 and i plays fake, the controller agent would lose right away. For simplicity, we analyze the game during fix intervals of time and a strategy of acting in each interval needs to be decided. We apply a weight to each interval to reflect the payoff portion during this interval. For instance, if 2 intervals are considered, μ would be the payoff coefficient for the acts done in $[t_0, t_1]$ and $1 - \mu$ the payoff coefficient for the acts done in $[t_1, t_2]$.

For simplicity and illustration purposes but without loss of generality, we consider the repeated game with two shots. The general case with n shots ($n \geq 2$) will follow. In such a game, web service i as player 1 has to make two decisions over over fake F and act normal N, one in the first decision time spot (weighted by μ), and the other in the second decision time spot (weighted by $1 - \mu$). Since i is the game starter, and Cg initially decides whether to stop or continue the game, we consider two continuous actions that reflect our game the best. An example of these actions is faking the first time spot (denoted here by F^{μ}) and the second time spot ($F^{1-\mu}$), which is denoted by $F^{\mu}F^{1-\mu}$. Therefore, i's set of pure strategies is $A_i = \{F^{\mu}F^{1-\mu}, F^{\mu}N^{1-\mu}, N^{\mu}F^{1-\mu}, N^{\mu}N^{1-\mu}\}$. In n-shot game, the set of pure strategies is: $A_i = \{F^{\mu_1} \dots F^{\mu_n}, F^{\mu_1} \dots N^{\mu_n}, \dots, N^{\mu_1} \dots N^{\mu_n}\}$ where $\sum_{i=1}^{n} \mu_i = 1$. Considering the choice of efforts, Cg's set of pure strategies (penalizing P or ignoring I) is $A_{Cg} = \{e_0, P^{\mu}P^{1-\mu}, P^{\mu}I^{1-\mu}, I^{\mu}P^{1-\mu}, I^{\mu}I^{1-\mu}\}$. Table 1 represents the payoff table of the two players over their chosen strategies. We continue our discussions in the rest of this section on this table.

In this game, the idea is to give the highest possible payoff $+\omega$ to the case in which i decides to fake the most and gets ignored by Cg. The more Cg recognizes the malicious act of i, the highest assigned negative value weighted by the payoff portion of the time spot (μ or $1 - \mu$). For instance, if web service i decides to fake during the first time spot but gets penalized, i's payoff would be $-\mu\omega$, and if it decides to fake again, but gets ignored this time, it will gain $(1 - \mu)\omega$, which makes the final payoff $\chi(O_i, O_{-i}) = (1 - 2\mu)\omega$ (see line 3 column 1 of Table 1).

Table 1: Two-shot game between web service i and controller agent Cg with obtained payoffs

Controller agent Cg \ Web service i	$F^{\mu}F^{1-\mu}$	$F^{\mu}N^{1-\mu}$	$N^{\mu}F^{1-\mu}$	$N^{\mu}N^{1-\mu}$
e_0	ω,-π	μω,-μπ	(1-μ)ω,-(1-μ)π	0,0
$P^{\mu}P^{1-\mu}$	**-ω,π**	-ω,(2μ-1)π	-ω,(1-2μ)π	-ω,-π
$P^{\mu}I^{1-\mu}$	(1-2μ)ω,(2μ-1)π	-μω,π	(1-2μ)ω,-π	-μω,(1-2μ)π
$I^{\mu}P^{1-\mu}$	(2μ-1)ω,(1-2μ)π	(2μ-1)ω,-π	-(1-μ)ω,π	-(1-μ)ω,(2μ-1)π
$I^{\mu}I^{1-\mu}$	ω,-π	μω,(1-2μ)π	(1-μ)ω,(2μ-1)π	0,π

There is a similar payoff assignment for Cg in the sense that its accurate detection is under investigation. For example, a correct detection in the first time spot would bring $+\mu\pi$, and if the second detection is wrong, this first portion will be added to the negative payoff of the second time spot $-(1-\mu)\pi$, which makes the final payoff equal to $(2\mu-1)\pi$ (see line 3 column 1 of Table 1). The crucial key to survive in the environment for both players is to consider the previous events and moves. In the following, we elaborate on different cases while web services do or do not consider Cg's behavior in the game.

Proposition 2. *In repeated game, if i is not aware of Cg's previous chosen strategies, then faking all the time and penalizing all fake actions is the unique Nash equilibrium.*

Proof. (We illustrate the proof for two-shot game from which the general case follows.)
Nash. *It is clear from Table 1 that in both faking intervals, Cg receives the maximum payoff by penalizing both cases. In this case, i would not increase its payoff ($-\omega$) and thus, would not prefer any other strategy. In any other case, by choosing the maximum received payoff for any player, the other player has a better strategy to increase its payoff.*
Uniqueness. *We prove that theis Nash point is the only Nash with respect to the following cases. In the first row of Table 1, there is no Nash because Cg makes no effort, so the maximum received payoff is zero and thus, it can be increased by changing the status. In the third and forth rows, still there is no Nash since in these rows there are choices of P and I in the sense that for any of these choices, i would be better off changing to a strategy that maximizes its assigned payoff. In*

the last row, the payoff assignment is similar to the first one, so that Cg prefers to change its chosen strategy to apply penalty to fake actions. □

We also have the following propositions generalized from the two-shot game. We motivate the fact that if the penalty assigned by Cg is clear, the strategy chosen by i would be different. The proofs are straightforward from the two-shot game as shown in Table 1.

Proposition 3. *In repeated game, if i is not aware of Cg's previous chosen strategies, then faking all the time is dominant strategy for i.*

Proposition 4. *In repeated game, if i is not aware of Cg's accuracy level, then acting normal by i and ignoring by Cg all the time is Pareto-Optimal Socially Superior .*

To analyze the reasons behind encouragement to act truthfully, we need to measure some expected values. In the repeated game, the probability that exactly n normal acts out of M acts are done in the past and current moment ($Pr[n, M]$) can be computed using binomial distribution as follows:

$$Pr[n, M] = \binom{M}{n} Pr[N]^n (1 - Pr[N])^{M-n} \tag{15}$$

where $Pr[N]$ is calculated in Equation 14. We use this probability in measuring the expected cumulative payoff denoted by $V(O_i, O_{-i})$ for web service i in the sense that in the chosen strategies (O_i, O_{-i}) n actions were normal as follows:

$$V(O_i, O_{-i}) = \sum_{n=0}^{M} Pr[n, M] \chi(O_i, O_{-i}) \tag{16}$$

As the objective of a rational web service i is to maximize the expected cumulative payoff, it would select the current strategy O_i^* that maximizes $V(O_i, O_{-i})$:

$$O_i^* = \underset{O_i \in st}{\arg\max}\, V(O_i, O_{-i}) \tag{17}$$

This would be achieved when the following inequality is satisfied:

$$\sum_{n=0}^{M} Pr[n, M] \chi(O_i^*, O_{-i}) > \sum_{n=0}^{M} Pr[n, M] \chi(\overline{O_i^*}, O_{-i}) \tag{18}$$

where $\overline{O_i^*}$ denotes the opposite strategy of O_i^*, which means:

$$V(O_i^*, O_{-i}) > V(\overline{O_i^*}, O_{-i}) \tag{19}$$

Recall that q is the probability of correct recognition via Cg that impacts the strategy that i adopts in the repeated game. Therefore, in the repeated game, these probabilities of Cg are labelled as $q^{t_0}, \ldots, q^{t_M}$, which reflects the evolution of Cg's accuracy over time. Indeed, Cg's accuracy has impact on the expected cumulative payoff that web service i estimates given the penalty and improvement it makes. Therefore, Cg applies such penalty that discourages i to act maliciously.

Proposition 5. *At a given moment t_n, If $Pn_i > \frac{1-2q^{t_n}}{q^{t_n}} Imp_i$, then web service i receives less cumulative payoff $V(O_i, O_{-i})$ if it acts maliciously.*

Proof. To prove this proposition, we can simply assume that all the previous strategies O_{-i} are known as normal, and prove that if the condition $Pn_i > \frac{1-q^{t_n}}{q^{t_n}} Imp_i$ is true, then $O_i^* = N$. As $V(O_i, O_{-i})$ is defined in terms of $\chi(O_i, O_{-i})$ (Equation 16), which in turn is defined in terms of i's reputation, we simply need to prove that if the condition is true, then i will have less reputation value. To do that, we need to prove that:

$$Exp(R_i|N^{\mu_1} \ldots N^{\mu_{n-1}} F^{\mu_n}) < Exp(R_i|N^{\mu_1} \ldots N^{\mu_n})$$

By simple calculation, we expand the expected values to their possible cases together with their probabilities, so we get:

$$\begin{aligned} &Exp(R_i|N^{\mu_1} \ldots N^{\mu_{n-1}} F^{\mu_n}) = \\ &(q^{t_n})(R_i - Imp_i - Pn_i) \\ &+(1 - q^{t_n})(R_i + Imp_i) \end{aligned}$$

$$Exp(R_i|N^{\mu_1} \ldots N^{\mu_n}) = R_i$$

The first equation gives the expected reputation value given that a fake action is made at the moment t_n, and the second one shows the expected reputation value given that no fake action is made. Assuming that $Pn_i > \frac{1-2q^{t_n}}{q^{t_n}} Imp_i$, it is easy to see that:

$$(q^{t_n})(R_i - Imp_i - Pn_i) + (1 - q^{t_n})(R_i + Imp_i) < R_i$$

so we are done. □

The results obtained in Proposition 5 simply states that the rational web service agents in discouraged to act maliciously and analyzing the environment, the obtained payoff is higher once the agent acts truthfully. This does not represent an equilibrium since agents could freely adopt any strategies. However, we strengthen the choice of truthful action and lead agents towards truthful environment.

Theorem 1. *q^{t_n} is increasing with respect to Imp_i and decreasing with respect to Pn_i*

Proof. From Proposition 5, we obtain the lower bound of Cg's accuracy q^{t_n}: $\frac{Imp_i}{Pn_i+2Imp_i}$ ($q^{t_n} > \frac{Imp_i}{Pn_i+2Imp_i}$). Let B_i denote this lower bound. We have:

$$\frac{\partial B_i}{\partial Imp_i} = \frac{Pn_i}{(Pn_i + 2Imp_i)^2}$$

As $Pn_i \geq 0$, $\frac{\partial B_i}{\partial Imp_i} \geq 0$, which means B_i is increasing with respect to Imp_i. Consequently, q^{t_n} is also increasing with respect to Imp_i.
On the other hand, we have:

$$\frac{\partial B_i}{\partial Pn_i} = \frac{-Imp_i}{(Pn_i + 2Imp_i)^2} \leq 0$$

B_i is then decreasing with respect to Pn_i. Consequently, q^{t_n} is also decreasing with respect to Pn_i. □

This theorem is important and very intuitive as it tells us if the improvement in the web service's reputation is very high, then the controller should be very cautious as probably the improvement is a result of some malicious actions. On the other hand, if the agent is less cautious, then this should be balanced by making the penalty high, so that web services will be discouraged to act maliciously. This theorem is inline with the result found in [21] according to which "buying agents will not be harmed infinitely by dishonest selling agents and therefore will not incur infinite loss, if they are cautious in setting their penalty factor".

Theorem 2. *In n-shot repeated game, if $Pn > \frac{1-2q^{t_n}}{q^{t_n}} Imp_i$, acting normal and being ignored is both Nash and Pareto-Optimal.*

Proof. From Proposition 4, we know that ignoring normal acts in all the shots is Pareto-Optimal. On the other hand, from Proposition 5, we deduce that i would

Table 2: Implemented environment details

Agent Type	Number	Agent Subtype	Percentage	Tendency
Controller	1	-----	-----	-----
Web service	100	Fixed	Truthful 20%	0%
			Malicious 20%	100%
		Random	30%	50%
		Observation	30%	$f(p_1,p_2,...,p_n)$
Consumer	1000	Fixed	Truthful 20%	0%
			Malicious 20%	100%
		Random	30%	50%
		Observation	30%	$g(p_1,p_2,...,p_n)$

have less cumulative payoff if it fakes given that it is aware of the assigned penalty Pn_i and Cg's accuracy. Therefore, the dominant strategy for i would be acting N. If i plays N as its dominant strategy, the best response from Cg would be I in all shots (see Table 1). Therefore, if the condition $Pn > \frac{1-2q^{tn}}{q^{tn}} Imp_i$ holds, then playing N and I is Nash, where $N^{\mu_1} \dots N^{\mu_n}$ and $I^{\mu_1} \dots I^{\mu_n}$ are dominant strategies for i and Cg, which completes the proof. □

This theorem shows that if web services are aware of the penalties and the controller's accuracy, then the system will achieve a secure and healthy state.

6. Simulation and Experimental Results

We developed a simulator in a java-based platform hosting different agents having broad range of characteristics and capabilities. Three types of agents are implemented: controller agent, web service agents, and consumer agents. During the simulation runs, web services and consumers might leave or join the network if they wish so. Table 2 provides detailed information regarding the implemented environment. We categorized the consumer and web service agents into three classes with respect to their acting strategies through simulation runs: (1) acting strategies using fixed opinions; (2) acting strategies using random movements; and (3) acting strategies using environment observations. Acting using fixed opinions means agent are completely (100%) truthful (0% tendency to act maliciously) or completely malicious (100% tendency to act maliciously). Acting using random movements means agents randomly decide to act truthfully or maliciously and can

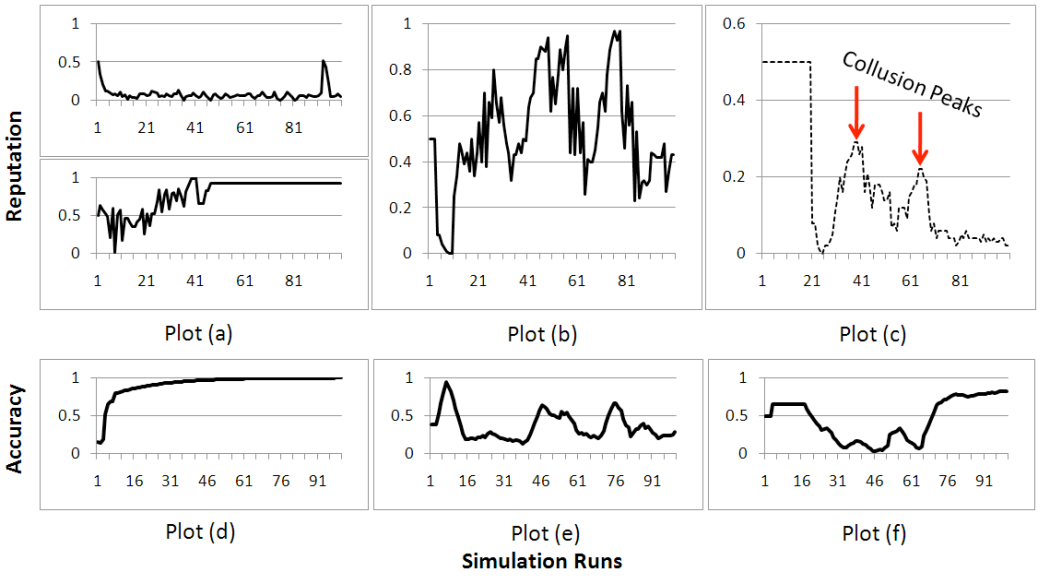

Figure 2: Overall reputation and accuracy assessment regarding different types of web services

change their decisions continuously. This type of agents, which represents 30% of the population with 50% tendency to act maliciously, makes the environment more realistic with presence of noise. Finally, acting through observations means agents are strategic and change their behaviors based on their observations of Cg's performance and their tendency to act maliciously is function of previous and current observations $p_1, \ldots, p_n$. The objective of this simulation is to analyze the outcome and performance of these agent types in different scenarios.

The first group of agents follow their predefined strategies regardless of the environment changes. The agents following this strategy fall into two groups of malicious and truthful. Figure 2 plot (a) illustrates two graphs reflecting the accumulated reputation of two typical web services (truthful and malicious) over the simulation runs. The truthful web service (lower graph) gradually maintains its actual reputation value, which converges to its publicly announced quality of service. This is the normal case in the implemented environment as the active web service collects the feedback with respect to the offered quality of service and thus, the accumulated reputation would reflect the actual quality value. The malicious web service (upper graph) eventually loses its accumulated reputation because based on its fixed strategy, it will continuously be involved in collusion scenarios. The controller agent recognizes the collusion made by web services following fixed malicious strategies. As consequence, the reputation dramatically decreases

at a certain time. Figure 2 plot (d) illustrates the overall reputation mechanism efficiency (i.e. Cg's accuracy) with respect to all the actions made by the web services and consumers and the reactions maintained by the controller agent. As shown by the graph, the controller agent acts accurately. In fact, recognizing the malicious actions maintained through fixed strategies is easy to learn for the controller. Therefore, the accuracy obtained by the controller agent is relatively high. We would carry on illustrating the efficiency graph in the rest of this section in order to compare the impacts on the reputation mechanism imposed by diverse parameters in the environment.

Figure 2 plot (b) represents the same results according to the observed reputation of a typical agent following random behavior as acting strategy. As represented in Table 2, agents of this type are developed with 50% chance of acting maliciously. Observed in different simulations, the controller agent is capable of recognizing these agents from time to time and penalizes them as it keeps the information regarding the past detections and web services with history of being detected are investigated more carefully. Hence, the web services which do not consider the controller's existence in their acting strategies would fail to accumulate a stable reputation value. In this figure, plot (e) illustrates the corresponding reputation mechanism efficiency with respect to the maintained actions. In fact, the unpredictable behavior of this type of web services confuses the controller agent because the agent that has maintained some collusion attempts, might act truthfully at some periods of time, where the controller agent has got very suspicious about the agent (because of a number of detected collusion attempts). The unpredictable and random behavior of this agent would generate a number of false detections for the controller agent, which brings about an oscillating efficiency.

Figure 2 plot (c) illustrates the results regarding a typical web service that considers the environment parameters (the controller agent's accuracy) in its acting strategy. The behavior of this type of agent is more dynamic compared to the previously discussed agents. In this plot, the considered web service maintains collusion attempt twice, which in both, the controller agent recognizes the attempt. Overall, controlling this type of agents is easy, but takes some time for the controller to completely learn from their behaviors and as illustrated by plot (f), the corresponding reputation mechanism efficiency increases once the behavior is being learnt, which reflects the controller agent's overall capability to manage the detections. In the rest of this section. we analyze the reputation assessment and reputation alteration in no collusion, collusion, and collusion-resistant environments. The exposed graphs are upon observed data from different experiments to avoid unpredicted randomization effects.

6.1. Reputation Assessment with No Collusion

We ran the simulation in a safe environment within which, web services act truthfully and the accumulated feedback reflect the actual reputation of the web services. The rational behind this experiment is to emphasize the fact that based on truthful actions, the accumulated reputation of a web service would approach its actual quality of service.

Figure 3 illustrates different curves obtained from separated simulation runs regarding only one typical web service i holding a quality Q_i. As shown by the figure, the overall reputation of this web service approaches its actual quality of service QoS_i over different experiments. This fact is analyzed via the reputation assessment procedure that is formalized in Equation 5 in Section 3. The reputation value regarding web service i is computed by aggregating web service's quality Q_i with the web service's market share M_i. In the simulations, M_i follows a normal distribution $\mathcal{N}(Q_i, 0.2)$.

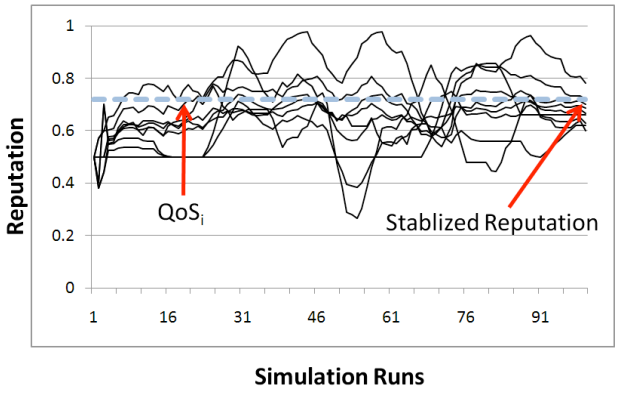

Figure 3: Reputation assessment with no collusion

According to a truthful web service i, Q_i percent of services are satisfactorily offered and thus, web service i expects positive feedback for "Q_i" percent of total posted feedback. However, there is another parameter that comes to play, which is the probability of posting unbiased feedback k from a consumer agent upon reception of a service ($Pr(unb(k))$). Therefore, the probability of receiving positive feedback ($Pr(k \in \mathcal{P}_i)$) for web service i is computed in Equation 20.

$$Pr(k \in \mathcal{P}_i) = Q_i \times Pr(unb(k)) \tag{20}$$

The value $Pr(unb(k))$ would be different in experiments according to the reaction of the consumers. This value is out of control and is completely based on

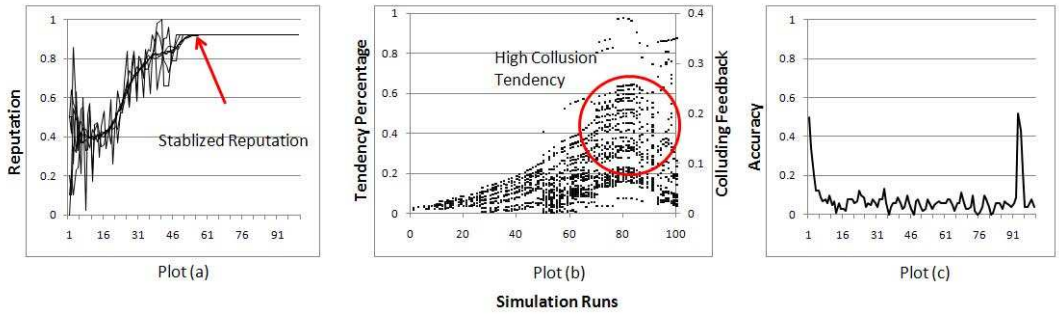

Figure 4: Reputation assessment through collusion

the distribution that the consumer uses to produce accurate feedback regarding the received service. In Figure 3, different curves are shown reflecting a number of experiments in which, the consumers use dynamic probabilities of providing unbiased feedback. However, overall in all of the graphs, the total reputation of the web service approaches its general quality of service (QoS_i) value. This means that, in a honest environment in which agents do not perform collusion, one's quality of service overall reflects its accumulated reputation. The reputation mechanism efficiency in this case is pretty high and very similar to the one shown in Figure 2 plot (d). The controller agent can easily manage the system control (as long as there is no collusion and web services act truthfully) and quickly recognize that the active users do not attempt to collude.

6.2. Reputation Assessment Through Collusion

In this simulation, we investigate the collusion impacts on the malicious agents' reputation values in a scenario where the controller agent imposes no penalty during simulation runs. Figure 4 plot (a) illustrates one typical malicious web service's reputation value extracted from different experiments. As depicted by the curves, the malicious web service performs collusion in all of them. This is due to the fact that the web service at the earlier collusion experiments recognizes that the controller agent is dormant and therefore, there would be no penalty applied after a performed collusion. This results in a dramatic increase of the probability of colluding.

Figure 4 plot (b) represents the collusion tendency of the whole network involving all the web services that are capable of acting maliciously, which represents 80% of the population ($20\% + 30\% + 30\%$, see Table 2). The X-axis of this plot denotes the elapse time over the simulation runs. The left Y-axis denotes the percentage of colluding web services (obtained from whole active web services in the network). This value is increasing over time, which expresses the increasing

tendency of the web services to act maliciously. The right Y-axis denotes the number of colluding feedback, which reflects the amount of increase in faked positive feedback set in the collusion agreement between the colluding web service and consumer. The dots in plot (b) show the extent to which the colluding web service maintains faked feedback in the collusion process. It is observed that overall, the amount of faked feedback is increasing over time when the malicious action is widespread in the environment. In such a chaos system, the performance of reputation mechanism (controller agent's accuracy) decreases dramatically as shown in Figure 4 plot (c)).

6.3. One-shot Game and Penalty Impact on Reputation Assessment

In this part, we expose the results obtained after one-shot game between the web service and consumer agents. Figure 5 plots (a), (b), and (c) represent respectively the reputation graphs obtained after a series of experiments. We study this result on three different types of web services (acting upon fixed opinions, acting randomly, and environment observers). In plots (a) and (b) typical agents follow strategies within which the controller agent's action is not considered. However, plot (c) shows agents following a strategy which considers controller agent's action. All these agents adopt malicious actions and get penalized via the controller agent, which confirms the theoretical result discussed in Proposition 1 that represents the Nash equilibrium. As shown in plots (d), (e), and (f), the controller agent expresses accurate collusion detection system and thus, the efficiency graph is increasing over time. However, the social situation depicted by the Nash is not well-accepted since the collusion is maintained regardless of the controller's accuracy.

6.4. Repeated Game and Penalty Impact on Reputation Assessment

To simulate the repeated game case, we ran the simulation with agents capable of analyzing the history of interactions in order to adopt the most appropriate strategy. The web services, which belong to categories of fixed and random opinions are not considered in these experiments as they carry on the same behavior shown in Figure 5 plots (a) and (b). The repeated game and history analysis only affect the agents, which consider the environment characteristics. To this end, we run many experiments considering these agents with tendency to maintain malicious actions over the time. Figure 6 shows different simulations running different web services capable of observing the environment characteristics and analyzing the history of previous interactions with the controller agent. In these simulations, the controller agent adopts different detection and penalty settings, which imposes

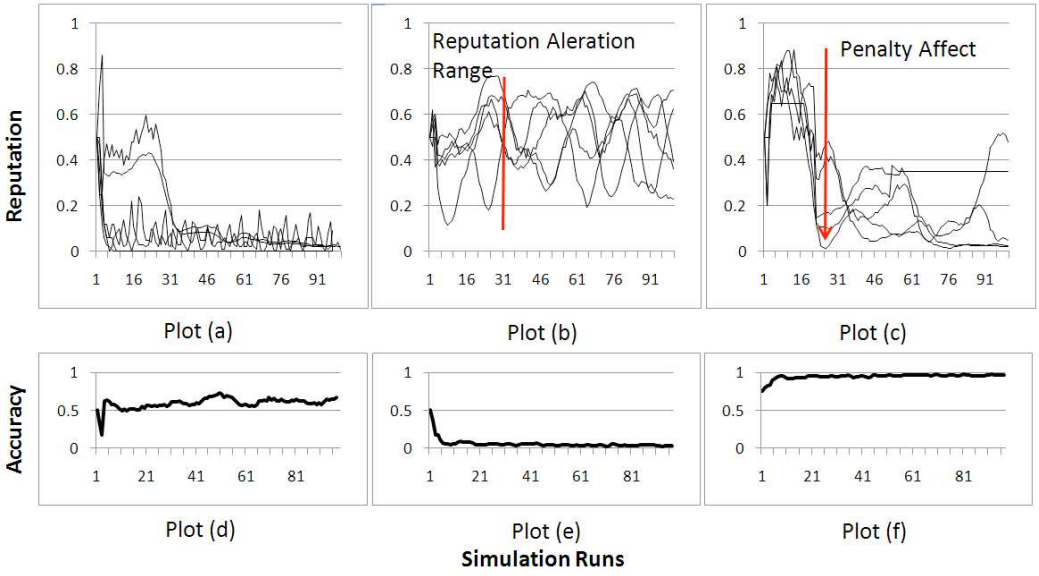

Figure 5: Reputation assessment and penalty impact in one-shot game

some impacts on the behavior and convergence of web services to a truthful reputation mechanism. In these experiments, the involved web services are all capable of collusion attempts. However, as shown in all the graphs, the collusion attempt is being detected and the corresponding penalty is applied, which results in decreasing the reputation value. As mentioned in Theorems 1 and 2, the assigned penalty exceeding the specified threshold brings about truthful actions made by malicious web services and affects controller agent's accuracy to some certain extend. Obviously for the sake of true detections, the controller agent cannot increase the assigned penalty with no limit.

The graphs shown in plots (a), (b), and (c) are representative of reputation management regarding different penalty settings that the controller agent imposes in a repeated game to a set of malicious web services. We capture the reputation of web services in different settings to observe over all influence of controller agent's settings to web services' attitudes. Figure 6 plot (d) shows overall reputation management efficiency reflected by the accuracy of the controller agent in detecting malicious actions. The analysis of the reputation management efficiency shows that obtaining high efficiency does not necessarily make web service adopt the truthful strategy as shown in Figure 2 Plot (d). However, it is crucial to obtain this efficiency where web services also tend to act truthfully. The results in plots (a), (b), and (c) show that reputation values are affected by the collusion and penalties

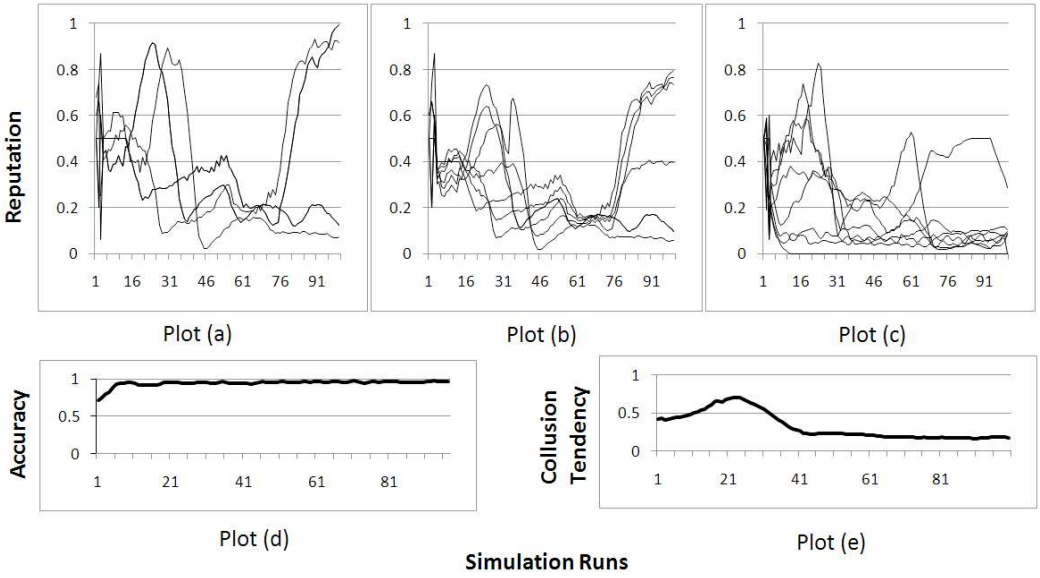

Figure 6: Reputation assessment and penalty impact in repeated game

assigned by the controller agent, but web services also increase their reputation towards their actual quality of service while the run number is increasing. Figure 6 plot (e) shows the overall tendency of malicious web services to attempt collusion. Over simulation runs, the tendency of these agents is decreasing, which reflects the trustful action as Nash equilibrium.

7. Related Work

Reputation is measured in open systems using different methodologies [2, 9]. In the literature, the reputation of web services have been intensively stressed [12] [19]. In [20], the authors have developed a framework aiming to select web services based on the reputation policies expressed by the users. The framework allows the users to select a web service matching their needs and expectations. In [16], authors have proposed a model to compute the reputation of a web service according to the personal evaluation of the previous users. In [17], a reputation model for web services called RATEWeb is proposed. In RATEWeb, individual web services share experiences of service providers with their peers through feedback ratings, which are aggregated to derive the overall service provider reputation. These proposals have the common characteristic of measuring the reputation of web services by combining data collected from users or other peers. To this

end, the credibility of the user or the peer that provides the data is important. In [15], authors have designed a sound mechanism to address the credibility of the collected data from users. In [18], a multi-agent framework based on an ontology for QoS has been designed. The users' ratings according to the different qualities are used to compute the reputation of the web service. In [10, 11], service-level agreements are discussed in order to set the penalties over the lack of QoS for the web services. In [5], a layered reputation assessment system is proposed mainly addressing the issue of anonymity. In this work, the focus is on the layered policies that are applied to measure the reputation of different types of agents, specially the new comers. Although, the proposed work is interesting in terms of anonymous reputation assessment, the layered structure does not optimally organize a community-based environment that gathers web services and users, and also the computational expenses seem to be relatively high.

In all the aforementioned frameworks, the service selection is based on the data that could not be reliable. The main issue we addressed in this paper, and which makes it different from the existing proposals is that web services are selfish agents and utility maximizers. Thus, if those agents are not provided with an incentive to act truthfully, they can violate the system to maliciously increase their reputation level. Analyzing the relationship between the payoffs and systems efficiency is another issue that has not been addressed in related proposals.

In [6], a trust-based game-theoretic model for web services collaboration is introduced. The focus of this paper is to determine the web service to perform a specific task. In each round of the game, a web service bids with a cost for performing the task and the task owner combines this bid with the web service trust value to compute the overall cost called trust-based cost. The winner is the web service that has the minimum trust-based cost. This work is close to our proposal as both use game-theoretic analysis and stress incentives for truth telling, but they are different as unlike our proposal, collusion is not considered in [6].

8. Conclusion

The contribution of this paper is the theoretical analysis and simulation over the reputation-based infrastructure that hosts agent-based web services as providers, users as consumers, and controller agent as reputation manager of the system. In the deployed infrastructure, web services can act maliciously to increase self reputation. Meanwhile, controller agent investigates user feedback and penalizes malicious web services. Controller agent may fail to accurately function, which is an incentive for some web services to act maliciously. The discussion is formed

in terms of a game that is analyzed in one-shot and then repeated cases. This analysis is concluded by denoting the best social state in which selfish services are discouraged to act maliciously and increase self reputation. The analysis is accompanied by empirical results that highlight reputation system's parameters. In experimental results, malicious services are observed and their characteristics are measured over time. In general, the Pareto-Optimality is observed to be a stable state for both web services and the controller agent.

Our plan for future work is to advance the game theoretic analysis such that web services that risk the malicious act deploy a learning algorithm that enables them to measure their winning chance. To this end, a continuous game can be extended, so that both players update their selected policies. Similarly, we need to discuss more about the different false detection cases that distract the reputation management.

References

[1] D. Banerjee and S. Sen. Reaching pareto-optimality in prisoners dilemma using conditional joint action learning. Autonomous Agents and Multi-Agent Systems, 15(1):91-108, 2007.

[2] V. Botêlhoa, F. Enembrecka, B. Ávilaa, H. de Azevedob, E. Scalabrina. Using asymmetric keys in a certified trust model for multiagent systems. Expert Systems with Applications, 38(2):1233-1240, 2011.

[3] Y. Chen, Y. Liu, and C. Zhou. Web service success factors from users behavioral perspective. Proc. of the 10th Int. Conf. on Computer Supported Cooperative Work in Design III, LNCS 4402, pp. 540-548, 2006.

[4] Z.P. Fana, W.L. Suoa, B. Fengc, and Y. Liu. Trust estimation in a virtual team: A decision support method. Expert Systems with Applications, 38(8):10240-10251, 2011.

[5] E. Fourquet, K. Larson, and W. Cowan. A reputation mechanism for layered communities. ACM SIGecom Exchanges, 6(1):11-22, 2006.

[6] H. Yahyaoui. A trust-based game theoretcal model for web services collaboration. Knowledge-Based Systems, 27:162-169, 2012.

[7] T.D. Huynh, N.R. Jennings, and N.R. Shadbolt. An integrated trust and reputation model for open multi-agent systems. Journal of Autonomous Agents and Multi-Agent Systems, 13(2):119-154, 2006.

[8] M. Jacyno, S. Bullock, M. Luck, T.R. Payne. Emergent service provisioning and demand estimation through self-organizing agent communities. 8'th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS), pp. 481-488, 2009.

[9] A. Josang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. Decision Support Systems, 43(2):618-644, 2007.

[10] R. Jurca, B. Faltings, and W. Binder. Reliable QoS monitoring based on client feedback. Proc. of the 16'th Int. World Wide Web Conf. (WWW), pp. 1003-1011, 2007.

[11] R. Jurca and B. Faltings. Reputation-based service level agreements for Web services. Proc. of the Int. Conf. on Service Oriented Computing (ICSOC 2005), Lecture Notes in CS, Volume 3826, pp. 396-409, 2005.

[12] S. Kalepu, S. Krishnaswamy, S. W. Loke. A QoS metric for selecting Web services and providers. Proc. of the 4'th Int. Conf. on Web Information Systems Engineering Workshops, pp. 131-139, 2003.

[13] B. Khosravifar and J. Bentahar, M. Gomrokchi, and R. Alam. CRM: An Efficient Trust and Reputation Model for Agent Computing. Knowledge-Based Systems, Elsevier. DOI: 10.1016/j.knosys.2011.01.004, 2011.

[14] B. Khosravifar, J. Bentahar, A.Moazin, and P. Thiran. On the reputation of agent-based web services. Proc. of the 24'th AAAI Conf. on Artificial Intelligence (AAAI), pp. 1352-1357, 2010.

[15] B. Khosravifar and J. Bentahar and A. Moazin and P. Thiran. Analyzing communities of web services using incentives. International Journal of Web Services Research, 7(3):30-51, 2010.

[16] Z. Malik and A. Bouguettaya. Evaluating rater credibility for reputation assessment of web services. 8'th Int. Conf. on Web Information Systems Engineering (WISE), pp. 38-49, 2007.

[17] Z. Malik and A. Bouguettaya. RATEWeb: reputation assessment for trust establishement among web services. Very Large Data Bases, 18(4):885-911, 2009.

[18] E. M. Maximilien. Multiagent system for dynamic web services selection. The 1'st Workshop on Service-Oriented Computing and Agent-based Engineering, pp. 25-29, 2005.

[19] S. Paradesi, P. Doshi, S. Swaika. Integrating behavioral trust in web service compositions. 7'th Int. Conf. on Web Services (ICWS), pp. 453-460, 2009.

[20] E. Shakshuki, L. Zhonghai, and G. Jing. An agent-based approach to security service. International Journal of Network and Computer Applications, 28(3):183-208, 2005.

[21] T. Tran. Protecting buying agents in e-marketplaces by direct experience trust modelling. Knowledge and Information Systems, 22(1):65-100, 2010.

[22] A. Yassine, A.A. Shirehjini, S. Shirmohammadi, and T. Tran. Knowledge-empowered agent information system for privacy payoff in ecommerce. Knowledge and Information Systems, DOI: 10.1007/s10115-011-0415-3, 2011.

[23] J. Zhang, R. Cohen. An incentive mechanism for eliciting fair ratings of sellers in e-marketplaces. In Proc. of the 6'th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS), pp. 108, 2007.

[24] J. Zhang, R. Cohen. Design of a mechanism for promoting honesty in e-marketplaces. Proc. of the 22'nd Conf. on Artificial Intelligence (AAAI), pp. 1495-1500, 2007.