On the Reputation of Agent-based Web Services

Babak Khosravifar[1], **Jamal Bentahar**[1], **Ahmad Moazin**[1], **and Philippe Thiran**[2]
[1]Concordia University, Canada, [2]University of Namur, Belgium
$b\_khosr@encs.concordia.ca, bentahar@ciise.concordia.ca, a\_moazi@encs.concordia.ca, pthiran@fundp.ac.be$

Abstract

Maintaining a sound reputation mechanism requires a robust control and investigation. In this paper, we propose a game-theoretic analysis of a reputation mechanism that objectively maintains accurate reputation evaluation of selfish agent-based web services. In this framework, web services are ranked using their reputation as a result of provided feedback reflecting consumers' satisfaction about the offered services. However, selfish web services may alter their public reputation level by managing to get fake feedback. In this paper, game-theoretic analysis investigates the payoffs of different situations and elaborates on the facts that discourage web services to act maliciously.

Introduction

Web services are deployed to maintain continuous interactions between applications. Abstracting web services using agents will benefit them from flexible and intelligent interactions that agents are able to manage (Jacyno et al. 2009). However, an important issue in agent-based environments is reputation, which is a significant factor that regulates the process of service selection. During recent years, there have been extensive work addressing the reputation in multi-agent and service environments (Kalepu, Krishnaswamy, and Loke 2003), (Malik and Bouguettaya 2007), (Maximilien 2005), (Jurca, Faltings, and Binder 2007; Jurca and Faltings 2005). Many of the proposed models are based on data collected from different sources that are supposed reliable. However, this might not be the case in many concrete situations. 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.

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. We aim to advance the state of the art by analyzing the system's parameters using game theory. We investigate the incentives to cheat that malicious web services can have and the incentives to act truthfully while being aware of the possible penalties assigned by a special agent called *the controller agent*. In fact, we theoretically and empirically analyze the obtained payoffs according to the agent's followed strategy. We conclude with incentives for web services to act truthfully and identify the state that is socially acceptable for all the system components.

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

Preliminaries

Service Consumers are intelligent agents that continuously seek for the services provided by some other agents. Service consumers could be encouraged by some web services to temporarily support them by reporting false feedback. This issue will be detailed later in this paper.

Feedback File is used in the proposed system to gather the submitted feedback from the service consumers. Consumers' report is the aggregation of all feedback, which reflects the total quality of a given web service.

Controller agent Cg is the assigned agent that takes the feedback file under surveillance. Cg is able to remove the cheated feedback that support particular web services. In general, Cg might fail to accurately detect the fake feedback or similarly might recognize truthful feedback as fake. Therefore, malicious web services will most likely consider this failure chance in their decisions of cheating.

Reputation Mechanism

This is a mechanism that enables the service consumers to evaluate the credibility of the web services they want to invoke. In this system, Cg updates its surveillance algorithm and web services learn 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 (Chen, Liu, and Zhou 2006): quality (Q_i), and market share (M_i). Besides the aforementioned parameters, there is also a significance level parameter (α) that is proposed by the controller agent to measure the confidence this agent has on the received feedback. Such a parameter measures the confidence that outcomes are not reached by random chance and they depend on the data size and noise associated to the measurement (Sackett 2001). After explaining

each factor, we will formalize the reputation of a typical web service as aggregation of these factors.

Reputation Parameters

Quality Q_i is used to measure the mean rate that is given to the 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 its launching in the web. Thus, the acceptance factor would be simply computed as $|\mathcal{P}_i|/|\mathcal{T}_i|$.

In such a trivial way, this calculation is not highly effective 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 by (Huynh, Jennings, and Shadbolt 2004): $ln^{-1}\alpha\Delta t_k$, where Δt_k is the time difference between the current time and feedback k submission time. Therefore, at each moment k, the feedback is referred to as $ln^{-1}\alpha\Delta t_k$. The quality factor Q_i of web service i is then measured in Equation 1.

$$Q_i=\frac{\int_{k\in\mathcal{P}_i} ln^{-1}\alpha\Delta t_k dt_k}{\int_{k\in\mathcal{T}_i} ln^{-1}\alpha\Delta t_k dt_k}=\frac{\int_{k\in\mathcal{P}_i}\frac{1}{\alpha}e^{\Delta t_k}dt_k}{\int_{k\in\mathcal{T}_i}\frac{1}{\alpha}e^{\Delta t_k}dt_k}=\frac{\frac{1}{\alpha\Delta t_k}e^{\Delta t_k}|_{k\in\mathcal{P}_i}}{\frac{1}{\alpha\Delta t_k}e^{\Delta t_k}|_{k\in\mathcal{T}_i}} \quad (1)$$

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 of the web service in the sense that the high service load together with high provided quality bring higher number of consumers (as a successful web service). Equation 2 defines the market share for the web service i. 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.

$$M_i=\frac{\int_{k\in\mathcal{T}_i} ln^{-1}\alpha\Delta t_k dt_k}{\int_{k\in\mathcal{G}} ln^{-1}\alpha\Delta t_k dt_k}=\frac{\frac{1}{\alpha\Delta t_k}e^{\Delta t_k}|_{k\in\mathcal{T}_i}}{\frac{1}{\alpha\Delta t_k}e^{\Delta t_k}|_{k\in\mathcal{G}}}=\frac{1}{\frac{1}{\alpha\Delta t_k}e^{\Delta t_k}|_{k\in\mathcal{G}-\mathcal{T}_i}} \quad (2)$$

Reputation Assessment

Taking the aforementioned parameters into account, we propose the estimated total reputation for the web service i that is crucial for its selection process and its 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 $r_i=\beta_1 Q_i+\beta_2 M_i$. In our reputation mechanism, Cg is dedicated to manage the sound reputation assessment. On top of the rates that a web service i receives from collecting the consumers' feedback file, the rate that is given by Cg for this service (C_i) affects its total reputation. If C_i is so low, that means the web service has a bad-reputed history that might cause users to avoid i. If the rate is relatively high, the consumers rely more on what they have evaluated from the files. Equation 3 gives then the formula of computing the total reputation R_i.

$$R_i=\gamma_1 r_i+\gamma_2 C_i \text{ such that: } \begin{cases} \gamma_2-\gamma_1= & r_i-C_i \\ \gamma_1+\gamma_2= & 1 \end{cases} \quad (3)$$

Reputation updates

Malicious Actions

In an open environment populated with selfish agents, web services might desire to increase self reputation to the level that they have not been ranked to. The increased reputation level would bring more requests under the assumption that the increasing process may not be recognized. To this end, malicious web services can manage to collude with some consumers to provide some continuous positive feedback supporting them. Such consumers can be encouraged to do that by promising them some privileges, such as low pricing, outstanding quality of service, etc. These web services can also manage to get virtual fake users sending positive feedback. For a typical web service i, this normally takes place by increasing Q_i and M_i. However, the big risk is the rate submitted by the controller agent (C_i) in the sense that if the malicious act is detected, C_i would be fairly small reflecting bad history of i. To this end, i should consider the risk of getting detected and penalized by Cg. Aiming to increase its reputation level, a malicious web service's challenges are: 1) the decision of acting maliciously; 2) when to act maliciously; 3) who to collude with; and 4) how many fake feedback to provide. To be focussed, in this paper we only consider the genuine and 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.

Detecting Malicious Actions

The controller agent, adversely aims to seize malicious acts and maintain a sound reputation system. Failing to detect malicious acts leads to false alarm, which is composed of two cases: the case of penalizing a good web service by mistake (false positive), and the case of ignoring a malicious web service by mistake (false negative). Cg analyzes the applied penalty to minimize malicious acts in the network. Therefore, Cg always looks for an optimum penalty value, which minimizes malicious acts and maximizes self-performance level. Cg is then required to be equipped with a mechanism to analyze the interactions of the web services with the consumers. Inequation 4 gives a detection criterion that this agent uses to capture suspected behavior of the web services. In this inequality, R_i is the current reputation level of the web service i and $R_{i,\alpha}$ is the general assessment of the same web service reputation without considering the $1-\alpha$ percent of the recent feedback. This inequality considers the fact that α is the percentage that Cg is confident in its performed actions. The threshold ν is application-dependant

and set by the controller depending on the tolerated risk level. A small value would be chosen if the application domain is critical, so only a minimum risk can be taken.

$$|\frac{R_i - R_{i,\alpha}}{R_{i,\alpha}}| > \nu \quad (4)$$

Suspecting Phase

The controller agent initiates a secret suspecting phase about the web service i when Inequation 4 is satisfied. In this stage, the behavior of the 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 (ΔQ_i), Cg might penalize it for its faked feedback. If Cg is sure about the i's malicious action, the suggested C_i would be decreased to αC_i. 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 a wrong decision. In general there are four cases: (1) the web service acts maliciously and accordingly gets penalized by Cg; (2) the web service acts maliciously, but gets ignored by Cg; (3) the web services acts truthfully, but gets penalized by Cg; and (4) the web service acts truthfully and Cg considers its action normal. Cases (1) and (4) represent the fair situations. However, cases (2) as false negative and (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 the fact of changing ones reputation level by which social opinions could be influenced. Conversely, the reputation is updated once Cg applies some penalties to the detected malicious acts. In general, the feedback file is subject to be modified by some unauthorized agents or an authorized controller agent. The interaction between a selfish web service and the controller agent can be modeled as a repeated game over time. The game consists of actions (made by the web service) and reactions (made by Cg). Here we consider the aforementioned four cases and obtain the corresponding payoff of each case. 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 (R_i) might be different because of false positives or negatives.

Malicious Act Ignored (false negative). This is the case where the 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 Q_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 5 gives the corresponding values for the current reputation level R_i and the actual (fair) reputation value R'_i.

$$R_i = R_{i,\alpha} + Imp_i; \quad R'_i = R_{i,\alpha} - Pn_i \quad (5)$$

$$R_i - R'_i = Imp_i + Pn_i = \omega \quad (6)$$

The difference between the actual (fair) and current reputation values reflect the payoff that we can use in our game-theoretic analysis (Equation 6). We use this difference to be able to compare two possible scenarios and consider their distance in 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 (false positive). This is the case where the web service i acts normal, but Cg decides to penalize it. In this case, i would lose its actual reputation value as shown in Equation 7. Equation 8 shows the negative obtained payoff, which reflects the fact that the web service i loses ω. This basically affects Cg as well in the sense that the correct decision is not made, so there is a negative effect applied to its accuracy level.

$$R_i = R_{i,\alpha} - Pn_i; \quad R'_i = R_{i,\alpha} + Imp_i \quad (7)$$

$$R_i - R'_i = -\omega \quad (8)$$

Truthful Act Ignored. This is the fair case where i acts normal and Cg refuses to penalize. In this case, the current reputation is the same as the actual reputation ($R_i = R'_i$). Thus, the payoff assigned to i is zero ($\omega = 0$).

Malicious Act Penalized. 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 discussed here, we also need to discuss about the obtained payoff for the controller agent. 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 detections are correct, Cg obtains the positive payoff ($+\pi$). 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, which brings the concept of learning the detection process, penalizing, and also the significance level α set by Cg. 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.

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 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 in acting normal, 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. Assume 1 denotes acting normal in the game, and 0 denotes acting maliciously. Also let $Pr[1|i]$ be the probability that web service i acts normal. In general, Cg's expected value for normal action from a typical web service i that is likely to be a good type is $P[S_G]Pr[1|S_G]+P[S_B]Pr[1|S_B]$ when $Pr[1|S_G]$ and $Pr[1|S_B]$ are calculate by Cg using the feedback and significance level parameter α.

As mentioned before, the chosen strategies of web service i is to fake (F) or act truthfully (i.e. normal) (N). This chosen strategy imposes the behavior that the web service exhibits and thus, the controller agent observes after the action is performed. Therefore, Cg has an observation O_i on web service i's performed actions. Cg also chooses between two strategies: penalizing (P) and ignoring (I) the event. The payoff range of i is $[-\omega, +\omega]$ and the payoff range of Cg is $[-\pi, +\pi]$. In the rest of this section, we start by analyzing the one-shot game, then extend it to the repeated one.

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

Proof. *By acting fake by the service i, the controller agent Cg would have a best utility if penalizing strategy is chosen rather than ignoring ($+\pi$). On the other hand, if Cg chooses to penalize, i would not change its chosen strategy since in both cases i will lose $-\omega$. Adversely, the normal act by i would lead 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.*

In one-shot game, players only consider the present information and they rationally tend to stay in fake-penalized state. This unique Nash is *socially* 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 estimates 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 (Banerjee and Sen 2007), which is the best situation for both players. We call this situation *Pareto-Optimal Socially Superior*. However, our work is different from (Banerjee and Sen 2007) as our focus is not on learning strategies, but on the incentives that make intelligent Web services act honestly.

Definition 1 ***Pareto-Optimality.*** *A situation in a game is said to be Pareto-Optimal once there is no other situation, in which one player gains better payoff and other players do not lose their current payoff.*

In the following, we extend the one-shot game to the repeated game over periods that we typically denote by $[t_0, t_2]$. Therefore, following different strategies in time intervals will generate the corresponding payoffs to the players. At time $t_1 \in [t_0, t_2]$, Cg would decide whether to continue or stop investigating. To this end, e_0 is referred to as the case of no effort in doing investigation (i.e. ignoring all actions). Otherwise, the best effort is made by Cg doing investigation and therefore, at time t_2, Cg decides about a proper strategy. Obviously, if Cg chooses e_0 and i plays fake, Cg would lose right away. We split the game time to two time intervals of $[t_0, t_1]$ and $[t_1, t_2]$ and strategy of acting in each interval needs to be decided. We apply a weight to each time interval regarding its importance, which reflects the payoff portion. Consider μ as the payoff coefficient for the acts done in $[t_0, t_1]$ and $1 - \mu$ as 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, the web service i as player 1 has two information sets. The first set contains decisions over fake F and act normal N given that i is in the first decision time spot (weighted by μ). The second set contains decisions over fake and act normal given that i is in the second decision time spot (weighted by $1 - \mu$). Since i is the game starter, and the controller agent Cg initially decides whether to stop or continue the game, we consider two continuous actions that reflect our game the best. Therefore, web service'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 $\sum_{i=1}^{n} \mu_i = 1$. On the other hand, 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. The payoff function will be denoted $\chi(O_i, O_i(M-1))$, which represents the assigned payoff to i when it selects the strategy $O_i \in \{F, N\}$ at current moment and $O_i(M-1)$ represents its $M-1$ previous chosen strategies.

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 importance of the time spot (μ or $1 - \mu$). 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 $+\mu\pi$ that is added to the negative payoff regarding its wrong detection in the second time spot $-(1-\mu)\pi$. The crucial key to survive in the environment for both players is the fact of considering 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.*

Table 1: Two-shot game of the 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	$\omega,-\pi$	$\mu\omega,-\mu\pi$	$(1-\mu)\omega,-(1-\mu)\pi$	0,0
$P^{\mu}P^{1-\mu}$	$-\omega,\pi$	$-\omega,(2\mu-1)\pi$	$-\omega,(1-2\mu)\pi$	$-\omega,-\pi$
$P^{\mu}I^{1-\mu}$	$(1-2\mu)\omega,(2\mu-1)\pi$	$-\mu\omega,\pi$	$(1-2\mu)\omega,-\pi$	$-\mu\omega,(1-2\mu)\pi$
$I^{\mu}P^{1-\mu}$	$(2\mu-1)\omega,(1-2\mu)\pi$	$(2\mu-1)\omega,-\pi$	$-(1-\mu)\omega,\pi$	$-(1-\mu)\omega,(2\mu-1)\pi$
$I^{\mu}I^{1-\mu}$	$\omega,-\pi$	$\mu\omega,(1-2\mu)\pi$	$(1-\mu)\omega,(2\mu-1)\pi$	$0,\pi$

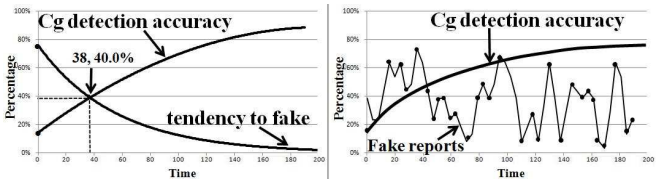

Figure 1: Controller agent's accuracy measurements vs. simulation RUNs.

Uniqueness. *We prove that the 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 maximum received payoff is zero and thus, it can be increased by changing the status. In 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 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 ommited because of space limit.

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 .*

We would like to elaborate more on the fact that web service being aware of Cg's assigned penalty, tend to act more normal than maliciously. In Figure 1, we illustrate two graphs that are obtained from our simulations of behaviors of two distinguished sets of web services. In the left graph, the set of web services are active in the environment while they are equipped with the algorithm that considers Cg's penalties over time. Such consideration affects their total tendency to choose to fake feedback. As it is clear, over 175 runs, there is below 10% of them that still consider to fake. Over time, Cg's accuracy level is increased in the sense that it maintains a good control over the environment. In contrast, in the right graph we collect a set of active agents that do not consider Cg's assigned penalties in past and act independently. Therefore, we observe haphazard behavior of them in malicious efforts. Such a chaotic environment also imposes a negative effect on Cg's detection process so that its significance level is not increased over time. However, Cg recognizes the malicious acts, which increases its accuracy, even though it is less than the case in the left graph.

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 are done in the past, given that actions are observed by Cg is: $Pr[n|O_i] = Pr[n|1]Pr[1|O_i] + Pr[n|0]Pr[0|O_i]$. Considering $\theta \in \{0, 1\}$ we have the binomial distribution for $Pr[n|\theta]$ given in Equation 9. Consequently, we compute the probabilities $Pr[\theta|O_i]$ and $Pr[O_i]$ from Bayes'law in Equations 10 and 11.

$$Pr[n|\theta] = c(M-1,n)Pr[1|\theta]^n(1-Pr[1|\theta])^{M-1-n} \quad (9)$$

$$Pr[\theta|O_i] = \frac{Pr[O_i|\theta]Pr[\theta]}{Pr[O_i]} \quad (10)$$

$$Pr[O_i] = Pr[O_i|1]Pr[1] + Pr[O_i|0]Pr[0] \quad (11)$$

We use this probability to measure the accumulative payoff for i in the sense that n actions were normal. Let $V(O_i) = \sum_{n=0}^{M-1} Pr[n|O_i]\chi(O_i, n)$ be this payoff using strategies recorded in O_i. As rational agent, i would select the next strategy in the sense that $V(O_i) > V(0_i(M-1))$. Therefore, the following inequality would be used as a constraint to select a strategy that would maximize i's obtained payoff. Moreover, the expected value for the obtained value could be measured in Equation 13.

$$\sum_{n=0}^{M-1} Pr[n|O_i]\chi(O_i, n) > \sum_{n=0}^{M-1} Pr[n|O_i]\chi(1-O_i, n) \quad (12)$$

$$E[V(O_i)] = Pr[1]\sum_{n=0}^{M-1} Pr[n|1]\chi(1,n) + Pr[0]\sum_{n=0}^{M-1} Pr[n|0]\chi(0,n) \quad (13)$$

Let q be the probability of correct recognition by Cg that impacts the strategy that i adopts in the repeated game ($Pr[n|1] = 1-q$ and $Pr[n|0] = q$). Therefore, in the repeated game, these probabilities of Cg are labelled as $q^{t_0}, \ldots, q^{t_n}$, which reflects the evolution of Cg's accuracy over time. Indeed, Cg's accuracy has impact on expected obtained value that i estimates given the penalty and improvement it makes. The controller agent Cg applies then this penalty that discourages i to act maliciously.

Proposition 5 *If $Pn > \frac{1-q^{t_n}}{q^{t_n}} Imp$, then the web service i receives less reputation value if it acts fake.*

Proof. *The details of this proof is skipped for the space limit. However, expanding the reputation formula would clarify that.*

From proposition 5, we obtain the lower bound of Cg's accuracy: ($q^{t_n} \geq \frac{1}{Imp+Pn} = \frac{1}{\omega}$). The obtained relation highlights the dependency of the received payoff (ω) to the recognition probability of Cg. Therefore, in the repeated game, if q increases (as a result of more experience and learning over time), ω would decrease, which reflects two facts: 1) i's lower tendency to fake; and 2) Cg's higher probability of penalizing since faking history is taken into account. Therefore, over time i tends to act truthfully.

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

Proof. *According to Proposition 4, ignoring normal intervals (or zero effort in normal intervals) is Pareto-Optimal. On the other hand, deduced from proposition 5, i would have less reputation if it fakes given that it is aware of the assigned penalty 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 intervals. This state is a healthy state that both players would be stable. This would be Nash once the condition expressed in Proposition 5 holds. In this case, $N^{\mu_1} \ldots N^{\mu_n}$ and $I^{\mu_1} \ldots I^{\mu_n}$ are dominant strategies for i and Cg.*

Discussion

Reputation is measured in open systems using different methodologies (Yao and Vassileva 2007). In the literature, the reputation of web services have been intensively stressed (Kalepu, Krishnaswamy, and Loke 2003). (Malik and Bouguettaya 2007) have proposed a model to compute the reputation of a web service according to the personal evaluation of the previous users. These proposals have the common characteristic of measuring the reputation of web services by combining data collected from users. To this end, the credibility of the user that provides the data is important. However, unlike our work, these proposals ignore the malicious acts from the users and their behaviors are not analyzed. In (Khosravifar et al. 2010), authors have designed a sound mechanism to address the credibility of the collected data from users. (Maximilien 2005) has designed a multi-agent framework based on an ontology for QoS. The users' ratings according to the different qualities are used to compute the reputation of web services. In (Jurca, Faltings, and Binder 2007; Jurca and Faltings 2005), service-level agreements are discussed in order to set the penalties over the lack of QoS for the web services. In all the aforementioned frameworks, the service selection is based on the data that is supposed reliable. These proposals do not consider the case where web services are selfish agents, and if not provided with an incentive to act truthfully, they can violate the system to maliciously increase their reputation level.

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 in the system. In the deployed infrastructure, web services can act maliciously to increase self reputation. Meanwhile, the controller agent investigates user feedback and penalizes malicious web services. The 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 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 the experimental results, malicious services are observed and their characteristics are measured over time. In general, the best Pareto-Optimal is observed to be a stable state for both the web services and 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.

Acknowledgments. Jamal Bentahar is supported by NSERC (Canada), NATEQ and FQRSC (Quebec). Philippe Thiran is partially supported by Banque Nationale de Belgique.

References

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

Chen, Y.; Liu, Y.; and Zhou, C. 2006. Web service success factors from users behavioral perspective. In *Proc. of the 10th Int. Conf. on Computer Supported Cooperative Work in Design*, volume 4402 of *LNCS*, 540–548.

Huynh, T. D.; Jennings, N. R.; and Shadbolt, N. R. 2004. An integrated trust and reputation model for open multi-agent systems. *Autonomous Agents and Multi-Agent Systems* 13(2):119–154.

Jacyno, M.; Bullock, S.; Luck, M.; and Payne, T. 2009. Emergent service provisioning and demand estimation through self-organizing agent communities. In *Proc. of the 8'th Int. Conf. on Autonomous Agents and Multiagent Systems*, 481–488.

Jurca, R., and Faltings, B. 2005. Reputation-based service level agreements for web services. In *Proc. of Service Oriented Computing (ICSOC)*, volume 3826 of *LNCS*, 396–409.

Jurca, R.; Faltings, B.; and Binder, W. 2007. Reliable QoS monitoring based on client feedback. In *Proc. of the 16'th Int. World Wide Web Conf.*, 1003–1011.

Kalepu, S.; Krishnaswamy, S.; and Loke, S. W. 2003. Verity: A QoS metric for selecting web services and providers. In *Proc. 4'th Int. Conf. on Web Information Systems Eng. Workshops*, 131–139.

Khosravifar, B.; Bentahar, J.; Moazin, A.; and Thiran, P. 2010. Analyzing communities of web services using incentives. *International Journal of Web Services Research* 7(3):(in press).

Malik, Z., and Bouguettaya, A. 2007. Evaluating rater credibility for reputation assessment of web services. In *Proc. of 8'th Int. Conf. on Web Inf. Sys. Engineering (WISE)*, 38–49.

Maximilien, E. M. 2005. Multiagent system for dynamic web services selection. In *The 1'st Workshop on Service-Oriented Computing and Agent-based Eng.*, 25–29.

Sackett, L. 2001. Why randomized controlled trials fail but needn't: 2. Failure to employ physiological statistics, or the only formula a clinician-trialist is ever likely to need (or understand!). *CMAJ* 165(9):1226–1237.

Yao, W., and Vassileva, J. 2007. A review on trust and reputation for web service selection. In *1'st Int. Workshop on Trust and Reputation Management in Massively Dis. Comp. Sys.*, 22–29.