# A Game Theoretic Approach for Analyzing the Efficiency of Web Services in Collaborative Networks

Babak Khosravifar, Mahsa Alishahi, Jamal Bentahar
*Faculty of Engineering and Computer Science*
*Concordia University*
*Montreal, Canada*
*b_khosr, m_ali, bentahar@encs.concordia.ca*

Philippe Thiran
*Department of Computer Science*
*University of Namur*
*Namur, Belgium*
*pthiran@fundp.ac.be*

***Abstract*—Because web services are loosely-coupled business applications, they are called to cooperate in distributed computing for the sake of efficiency. In this paper we propose a model formalizing web services efficiency considering different related parameters and a game-theoretical framework analyzing the web services strategies allowing them to maximize this efficiency. Many theoretical results are proved and confirmed through extensive simulations.**

***Keywords*-Web Service; Reputation; Agent;**

## I. Introduction

During recent years, web services have obtained a great attention as they represent distributed cooperation in IT networks. Using XML artifacts, Web services are capable of interactions and compositions that enhance productivity in enterprize networks. Because web services are loosely-coupled business applications, they are called to cooperate in order to increase efficiency, the ultimate goal in business settings. Efficiency in service computing can be defined in terms of the number of requests a service can receive and handle, which is function of different parameters such as *service reputation*, *market share* and *capacity*. To address the efficiency issue, there have been efforts attempting to model and analyze collaboration among web services [5], [10], [11]. Moreover, many frameworks have been proposed towards gathering functionally similar web services and representing the group as a community that manages task allocation [7], [4]. In this context of communities, we distinguish web services collaboration from web services composition. By collaboration, we mean that the community aggregates web services capable of interacting with one another to manage allocated tasks, for example by allowing a web service to replace another that is incapable of executing a task. By composition, we mean the extension initiated by a web service to finalize a specific task. In all the proposed frameworks into collaboration, the objective is to increase efficiency in distributed computing. However, in such frameworks, strategies web services can follow to achieve this goal are very limited to aggregation and more sophisticated strategies have not been thoroughly analyzed and investigated yet. Such sophisticated strategies can help communities and single web services achieve higher efficiency.

The aim of this paper is to investigate strategies as rational behaviors that web services and communities can adopt to increase efficiency. We present a game-theoretical model in which web services either act alone or cooperate with other web services within a community. Each entity (single web service or community of web services) manages its reputation, market share, capacity, and efficiency parameters. There are also service users that select services based on their reputation. A game is defined between typical single web service and the representative of a typical community (called master web service). Each entity seeks maximum efficiency following strategies of *joining*/*leaving* a community, *accepting*/*refusing* a request to join a community, and *inviting* to join a community. In different scenarios, we investigate the situation that maximizes players' efficiencies. Overall contributions of the paper are threefold: (1) we provide a distributed network of web services and users where the task allocation problem is regulated by a mechanism taking reputation, market share, and efficiency into account; (2) we propose a game-theoretic analysis investigating the stabilized situation within which, entities achieve high efficiency; and (3) we identify thresholds allowing the master web service to identify the optimal number of web services associated to the community. We also provide experiments that show and uphold the impact of our game-theoretic analysis on the behavior of rational web services.

## II. Preliminaries

In this section, we introduce the preliminary concepts that we use in the rest of this paper. More details about those concepts are discussed in [5].

**Web Service** is a rational entity providing a service that seeks to maximize the individual income by increasing service requests. In general, a web service has limited capabilities (in terms of available resources and quality of service regarding the allocated task), which forces collaboration with other web services in the surrounding environment. In this paper, we refer to a typical web service as $w_i$

978-0-7695-4462-5/11 $26.00 © 2011 IEEE
DOI 10.1109/SCC.2011.76

**Community of Web Services** is a set of functionally similar web services aggregated to represent a group where members can collaborate. The community is represented by the master web service, which uses a task allocation mechanism to select a web service when a request is received. As argued in [6], communities allow increasing web services' performance compared to the case where these web services act alone. We refer to a typical community by $c_j$.

**Service User** is the entity that continuously seeks for web services (either those acting alone or within a community) to request given services. Users initiate service requests and accordingly rate the web services using a feedback system.

**Feedback** is a rating posted by a user to express his satisfaction about the provided service. Feedback are accumulated in the feedback file to compute and analyze the web service's reputation.

**Reputation** is a value between 0 and 1 computed from the feedback file. Web services, as rational entities, aim at increasing this value to maximize their income. We compute the reputation value of a single web service $w_i$ ($R_{w_i}$) and a community of web services $c_j$ ($R_{c_j}$ with $n$ web services) in Equation 1. In this Equation, $PF_{w_i}$ denotes the number of positive feedback posted for the web service $w_i$ and $TF_{w_i}$ represents the total submitted feedback to this web service. These parameters could be evaluated during a fixed period of time such as a day or a week.

$$R_{w_i} = \frac{PF_{w_i}}{TF_{w_i}} \qquad R_{c_j} = \frac{\sum_{w_i \in c_j} R_{w_i}}{n} \tag{1}$$

**Market Share** is a value between 0 and 1 denoting the portion of the total requests directed to the web service (or community) to all initiated requests. The value is computed in Equation 2. In this Equation, $Req_{w_i}$ denotes the requests sent to to the web service $w_i$ and $TReq$ represents the total requests filed.

$$M_{w_i} = \frac{Req_{w_i}}{TReq} \qquad M_{c_j} = \sum_{w_i \in c_j} M_{w_i} \tag{2}$$

**Capacity** is a parameter denoted by $Cp_{w_i}$ that the web service $w_i$ holds as an individual attribute. This value is fixed and only reflects the service total ability in handling simultaneous requests. For the community, the total capacity is simply the sum of its members' capacities (see Equation 3).

$$Cp_{c_j} = \sum_{w_i \in c_j} Cp_{w_i} \tag{3}$$

## III. The Model

In this section, we formalize the attributes of rational services. In general, all rational entities, including users and web services, tend to maximize their efficiencies. To make the paper focussed, we only consider the perspective of web services. Thus, we propose a heuristic (see Equation 4) for computing the efficiency $E_x$ as a function $f$ of $R_x, M_x$ and $Cp_x$ where $x \in \{w_i, c_j\}$

$$E_x = f(R_x, M_x, Cp_x) \tag{4}$$

The function $f$ should satisfy the following properties.

***Property** 1:* $f$ is continuous.

This property says that at each moment the efficiency of a web service or a community can be evaluated with respect to the current attributes.

***Property** 2:* $f$ is strictly increasing in $R_X$ and $M_X$.

This property says that the efficiency of the service increases if it holds high reputation and market share in the system. Consequently, services and communities will have incentive to do better to get their overall efficiency increased.

***Property** 3:* $f$ is monotonically decreasing in $M_X - Cp_X$.

This property says that the efficiency of a service or a community decreases if it fails to make a good balance between its capacity and the requests it should handle. Consequently, services and communities will have incentive to analyze their capacities and manage to have acceptable market share. The idea is that the more service provider entity succeeds in making balance between its capacity and market share, the higher the efficiency would be.

Equation 5 gives a possible definition of $f$.

$$f = \frac{R_x \times M_x}{|M_x - Cp_x| + 1} \tag{5}$$

***Theorem** 1:* The function $f$ satisfies Properties 1, 2 and 3.

*Proof:* Satisfaction of Property 1 is straightforward as all the parameters are defined at each moment in time, so the function is continuous. Property 2 can be proved by computing the partial derivatives $\frac{\partial f}{\partial R_x}$ and $\frac{\partial f}{\partial M_x}$, which are clearly positive. Property 3 can be proved by considering $|M_x - Cp_x|$ as a variable, say $v$ and compute the partial derivative $\frac{\partial f}{\partial v}$, which is manifestly positive, so we are done. ■

The other attribute that categorizes services is the risk factor $S_X$. This factor is denoted as how flexible the service is in loosing its efficiency. For example, if the risk factor associated to $w_i$ is $\%20$ ($S_{w_i} = 0.20$), then the web service $w_i$ would consider any situation in its strategy analysis where estimated efficiency is more than $\%80$ of its current efficiency. $\overline{E_{w_i}}$ is defined as the estimated efficiency of the web service $w_i$ after taking any strategy for updating its status ($\overline{E_{c_j}}$ would corresponds to the community $c_j$). To this end, the web service $w_i$ would discard all the strategies (and choices of updating the current status) that yield to an estimated efficiency less than $(1 - S_{w_i})E_{w_i}$.

The reason behind using the provider risk factor is the fact that web services or communities need to be flexible in choosing strategies. For the rest of this section, we discuss two different cases where the web service is outside and inside the community. In each case, we analyze the best strategies that culminate in maximum efficiency level for both the web service and community.

### A. Web Service Out of Community

In this scenario, the single web service $w_i$ is facing the community $c_j$ with different strategies that would end in either the single web service $w_i$ joins the community $c_j$ or not. This action could be initiated or ceased by the web service or community representative. Doing so, there are four different cases: (a) $w_i$ attempts to join $c_j$ and the attempt is accepted; (b) $w_i$ attempts to join, but $c_j$ refuses the join request; (c) $c_j$ invites the web service $w_i$ but $w_i$ refuses the invitation; and (d) there is neither invitation from $c_j$ nor join request from $w_i$. From the outcome perspective, the cases of "$w_i$ attempts to join and $c_j$ accepts" and "$c_j$ invites and $w_i$ accepts" are similar. However, refusal from any party would lead to different estimating efficiencies and this is why we consider them as two separated cases. In the following, we compute the estimated efficiency of each entity with respect to the taken action.

**Case (a)** The web service $w_i$ that takes the risk of join ($S_{w_i}$) would update its reputation, market share and capacity parameters respectively in Equations 6 and 7, where $n$ denotes the current cardinality of the community set.

$$\overline{R_{w_i}} = \frac{n \times R_{c_j} + R_{w_i}}{n+1} \tag{6}$$

$$\overline{Cp_{w_i}} = Cp_{w_i} \qquad \overline{M_{w_i}} = \frac{M_{c_j} + M_{w_i}}{n+1} \tag{7}$$

In his case, our assumptions are as follows: (1) the reputation of a web service would be updated to the average of the community reputation. To this end, each registered web service in the community holds its individual reputation, but broadcasts the public reputation of the community; and (2) we consider the capacity as a fixed attribute. Therefore, the capacity of the web service stays unchanged, but the community accumulates the joined web service's capacity. When it comes to the market share, the community simply accumulates the market share of the new web service. However, the joined web service is going to obtain a share of total market share from the community. The corresponding attribute updates regarding the community $c_j$ are formulated in Equations 8 and 9.

$$\overline{R_{c_j}} = \overline{R_{w_i}} \tag{8}$$

$$\overline{C_{c_j}} = C_{c_j} + C_{w_i} \qquad \overline{M_{c_j}} = M_{c_j} + M_{w_i} \tag{9}$$

In this case, both entities consider the estimated parameters and compute their new efficiency values (see Equation 5). The case would take place when the following inequalities hold:

$$\overline{E_{w_i}} \geq (1 - S_{w_i})E_{w_i} \qquad \overline{E_{c_j}} \geq (1 - S_{c_j})E_{c_j}$$

**Case (b)** In this case, $w_i$ requests joining, but the community does not accept the request. The difference between the cases (a) and (b) is that in case (a) the join takes place, which brings actual updated efficiency for both entities. However, in case (b) the join does not take place, which keeps the analysis at the estimation level. The corresponding estimated efficiencies are characterized by the following inequalities:

$$\overline{E_{w_i}} \ngeq (1 - S_{w_i})E_{w_i} \qquad \overline{E_{c_j}} \geq (1 - S_{c_j})E_{c_j}$$

**Case (c)** This case is similar to the case (b), except the fact that the refusal is caused by the web service. The corresponding estimated efficiencies are characterized by the following inequalities:

$$\overline{E_{w_i}} \geq (1 - S_{w_i})E_{w_i} \qquad \overline{E_{c_j}} \ngeq (1 - S_{c_j})E_{c_j}$$

**Case (d)** In this case, both entities are not encouraged to attempt joining and therefore, the join does not take place. In this case, we have:

$$\overline{E_{w_i}} \ngeq (1 - S_{w_i})E_{w_i} \qquad \overline{E_{c_j}} \ngeq (1 - S_{c_j})E_{c_j}$$

### B. The Game Set up for Single Web Service

Table I
PAYOFF REGARDING 2 PLAYERS WHEN WEB SERVICE IS OUTSIDE THE COMMUNITY.

| Web service (Player 1) \ Community (Player 2) | Accept/Invite Join | Refuse/Not Invite Join |
|---|---|---|
| Join/Accept Invitation | $J_{w_i,c_j}, A_{w_i,c_j}$ | $JR^{c_j}_{w_i}, JR^{w_i}_{c_j}$ |
| Stay/Refuse Invitation | $SI^{c_j}_{w_i}, SI^{w_i}_{c_j}$ | 0 , 0 |

$$J_{w_i,c_j} = E'_{w_i} - E_{w_i}$$
$$A_{w_i,c_j} = E'_{c_j} - E_{c_j}$$
$$JR^{c_j}_{w_i} = E_{w_i} - \overline{E_{w_i}}$$
$$JR^{w_i}_{c_j} = E_{c_j} - \overline{E_{c_j}}$$
$$SI^{c_j}_{w_i} = E_{w_i} - \overline{E_{w_i}}$$
$$SI^{w_i}_{c_j} = E_{c_j} - \overline{E_{c_j}}$$

Upon the discussed cases, we develop a game-theoretic model consisting of the web service $w_i$ as player 1 and community $c_j$ as player 2. The player 1 follows the strategy profile of (join/not join) when is initiating the game (i.e. play first), and follows the strategy profile of (accept join/refuse join) when is reacting to the opponent's move (i.e. play second). Since for our analysis it is only important whether the join takes place or not, the order of playing does not matter when calculating payoffs (represented in terms of efficiency). Table I shows the assigned payoffs for both players in different cases. As shown in the table, the values of $J_{w_i,c_j}$ and $A_{w_i,c_j}$ are the generalized form of "join/accept" or "invite/accept join" cases. These values are actual differences in efficiency values after the join ($E'_{w_i}$ and $E'_{c_j}$). The obtained payoffs could be either positive or negative. The negative payoff denotes the wrong decision the entity regrets. The payoffs obtained in the other cases are all upon estimations.

The developed game is only a one-stage game between a typical web service and a typical community. The game could be set up between any other two entities and is repeated over time when entities are active in the network. Moreover, rational entities consider the information obtained in one game in their further strategy analysis. We formalize the results we obtain from the set up game between these entities in the following.

***Proposition** 1:* In one-stage game, there is no pure strategy Nash equilibrium.

*Proof:* In the set up one-stage game, the payoff of web services regarding accepted join request ($J_{w_i,c_j}$) could be either more or less than that of refusing the invitation (as it refers to the actual efficiency evaluation). This is also the case for the master of the community. Consequently, there is no dominant strategy for any player. Therefore, no pure strategy Nash equilibrium can be found. ■

As a consequence of this proposition, there is no stable situation rational entities can try to achieve by playing the game. Both players should then consider the risk parameter in their strategy selections. To this end, we define web service and community's mixed strategy probabilities respectively as $w_i(S_{w_i}, 1-S_{w_i})$ and $c_j(S_{c_j}, 1-S_{c_j})$. Thus, we compute web services expected payoff $\alpha_{w_i}$ of join (or accept to join) versus the mixed strategy profile of the community in Equation 10. Equation 11 computes the related value regarding the refusal of join.

$$\alpha_{w_i}(join, c_j(S_{c_j}, 1-S_{c_j})) = S_{c_j}(J_{w_i,c_j}) + (1-S_{c_j})(JR^{c_j}_{w_i}) \tag{10}$$

$$\alpha_{w_i}(stay, c_j(S_{c_j}, 1-S_{c_j})) = S_{c_j}(SI^{c_j}_{w_i}) + (1-S_{c_j})(0) \tag{11}$$

The web service aims at maximizing its payoff. Therefore, for all adopted strategies, we need to consider the best response (to the other player) and discard the others. For instance, if the web service obtains a higher expected payoff with the joining strategy, it would change its probability profile to $(1, 0)$, so the join would be the dominant strategy.

Since each player in each stage game chooses between only two strategies, and since any of these strategies could be the best response in a particular situation, we analyze the case where the expected payoffs are equal. By so doing, we can compute a threshold ($\mu_{w_i}$), which is used to identify which strategy is dominant. The threshold $\mu_{w_i}$ is used by the master to control the expected payoff of the web service in the sense that the web service adopts the master's desirable strategy as dominant. Thus, the master would pay the least possible cost to obtain its desirable control on the web services. This eventually would lead to the control mechanism of the master web service over cardinality of the community set. The threshold is computed in Equation 12.

$$\alpha_{w_i}(join, c_j(S_{c_j}, 1-S_{c_j})) = \alpha_{w_i}(stay, c_j(S_{c_j}, 1-S_{c_j}))$$
$$\Rightarrow S_{c_j}(J_{w_i,c_j}) + (1-S_{c_j})(JR^{c_j}_{w_i}) = S_{c_j}(SI^{c_j}_{w_i})$$
$$\Rightarrow S_{c_j}(E'_{w_i} - E_{w_i}) + (1-S_{c_j})(E_{w_i} - \overline{E_{w_i}}) = S_{c_j}(E_{w_i} - \overline{E_{w_i}})$$
$$\Rightarrow \mu_{w_i} = \frac{(S_{w_i})(E'_{w_i}) + E_{w_i} - 3(S_{w_i})(E_{w_i})}{1 - 2S_{w_i}} \tag{12}$$

The threshold $\mu_{w_i}$ obtained in Equation 12 is in terms of the estimated efficiency $\overline{E_{w_i}}$, which could be changed by the master $c_j$. So if the expected efficiency of join is computed to be more than $\mu_{w_i}$, the web service $w_i$ would adopt the join or accept the invitation to join strategy. We have then the following result.

***Proposition** 2:* In mixed strategy one-stage game, there is a threshold $\mu_{w_i}$ such that if $\overline{E_{w_i}} > \mu_{w_i}$, joining the community would be the goal of the web service. Otherwise, the web service $w_i$ would not join the community.

***Corollary** 1:* If the master web service considers the expected efficiency value computed by the web service and provides (broadcasts) a reputation value that let $\overline{E_{w_i}}$ exceeds $\mu_{w_i}$, the master can control adopting strategy of the web service.

### C. Web Service in the Community

In the previous sections, we analyzed the case where the web service $w_i$ was acting alone outside the community $c_j$. We also set up a game and analyzed the payoffs regarding different adopting strategies. In this part, we analyze the same system where the web service $w_i$ is already acting in collaboration with other web services inside the community $c_j$. In this case, the web service chooses its actions from strategy profile of "leave/accept to leave" or "stay/refuse to leave" (we assume that any action that ends up in changing the status of the web service is being made upon agreements between the web service and the master of the community). The community $c_j$ also refers to the strategy profile of "accept of leave/fire" or "refuse the leave/not fire". Doing so, there are four different cases: (a) $w_i$ attempts to leave the community $c_j$ and the attempt is accepted; (b) $w_i$ attempts to leave, but $c_j$ refuses the leaving request; (c) $c_j$ encourages (fires) the web service $w_i$, but $w_i$ refuses the invitation; and (d) there is neither firing from $c_j$ nor leaving request from $w_i$. Similar to the case where the web service was outside the community, we analyze the cases with their parameter updates.

**Case (a)** The web service $w_i$ that takes the risk of leave ($S_{w_i}$) would update his reputation, market share and capacity parameters respectively in Equation 13

$$\overline{R_{w_i}} = R"_{w_i} \qquad \overline{Cp_{w_i}} = Cp_{w_i} \qquad \overline{M_{w_i}} = M"_{w_i} \quad (13)$$

In his case, our assumptions are as follows: (1) the reputation of a web service would be back to its previous individual reputation ($R"_{w_i}$). To this end, each registered web service in the community holds its individual reputation when joining a community. However, the community recalculates its average reputation; (2) we consider the capacity as a fixed attribute. Therefore, the capacity of the web service stays unchanged but the community reduces the left web service's capacity. A similar analysis can be obtained for the market share where $M"_{w_i}$ is the previous value. The corresponding attribute updates regarding the community $c_j$ are formulated in Equation 14.

$$\overline{R_{c_j}} = \frac{n(R_{c_j}) - R_{w_j}}{n-1} \quad \overline{C_{c_j}} = C_{c_j} - C_{w_i} \quad \overline{M_{c_j}} = M_{c_j} - M_{w_i} \tag{14}$$

In this case, both entities consider the estimated parameters and compute their new efficiency values (see Equation 5). The case would take place when the following inequalities hold:

$$\overline{E_{w_i}} \geq (1 - S_{w_i})E_{w_i} \qquad \overline{E_{c_j}} \geq (1 - S_{c_j})E_{c_j}$$

**Case (b)** In this case, $w_i$ attempts to leave, but the community does not accept the leaving request. The difference between the cases (a) and (b) is the same as explained in the previous section. We have then the following inequalities:

$$\overline{E_{w_i}} \ngeq (1 - S_{w_i})E_{w_i} \qquad \overline{E_{c_j}} \geq (1 - S_{c_j})E_{c_j}$$

**Case (c)** This case is similar to the case (b) except the fact that the refusal is caused by the web service:

$$\overline{E_{w_i}} \geq (1 - S_{w_i})E_{w_i} \qquad \overline{E_{c_j}} \ngeq (1 - S_{c_j})E_{c_j}$$

**Case (d)** In this case, both entities are not encouraged to attempt leaving:

$$\overline{E_{w_i}} \ngeq (1 - S_{w_i})E_{w_i} \qquad \overline{E_{c_j}} \ngeq (1 - S_{c_j})E_{c_j}$$

### D. The Game Set up for the Joined Web Service

In this section, we also develop the game-theoretic analysis consisting of the web service $w_i$ as player 1 and community $c_j$ as player 2. The player 1 follows the strategy profile of (leave/not leave) when is the initiator and follows the strategy profile of (acceptance fire/refuse fire) otherwise. Table II shows the assigned payoffs for both players in different cases. As shown in the table, the values of $L_{w_i,c_j}$ and $F_{w_i,c_j}$ are the generalized form of "leave/accept" or "fire/accept join" cases. These values are actual differences in efficiency values after the join ($E'_{w_i}$ and $E'_{c_j}$). The obtained payoffs could be either positive or negative. We formalize the results we obtain from the set up game between these entities in the following.

Table II
PAYOFF REGARDING 2 PLAYERS WHEN WEB SERVICE IS INSIDE THE COMMUNITY.

| Web service (Player 1) \ Community (Player 2) | Accept Leave/Fire | Not Fire/ Refuse Leave |
|---|---|---|
| Leave/Accept Leaving | $L_{w_i,c_j}, F_{w_i,c_j}$ | $IR^{c_j}_{w_i}, IR^{w_i}_{c_j}$ |
| Stay/Refuse Leaving | $SF^{c_j}_{w_i}, SF^{w_i}_{c_j}$ | 0 , 0 |

$$L_{w_i,c_j} = E'_{w_i} - E_{w_i}$$
$$F_{w_i,c_j} = E'_{c_j} - E_{c_j}$$
$$IR^{c_j}_{w_i} = E_{w_i} - \overline{E_{w_i}}$$
$$IR^{w_i}_{c_j} = E_{c_j} - \overline{E_{c_j}}$$
$$SF^{c_j}_{w_i} = E_{w_i} - \overline{E_{w_i}}$$
$$SF^{w_i}_{c_j} = E_{c_j} - \overline{E_{c_j}}$$

***Proposition** 3:* In one-stage game, there is no pure strategy Nash equilibrium.

*Proof:*
The proof is similar to the one given for Proposition 1. ■

Referring to the obtained payoffs shown in Table II, we would have the same best response analysis that we did in the case for the single web service. To this, the obtained threshold $\mu_{w_i}$ is set the same.

***Proposition** 4:* In mixed strategy one-stage game, there is a threshold $\mu_{w_i}$ such that if $\overline{E_{w_i}} > \mu_{w_i}$, leaving the community would be the goal of the web service that is already member of the community. Otherwise, the web service $w_i$ would not leave the community.

***Corollary** 2:* If the master web service considers the expected efficiency value computed by the web service and provides a market share value that let $\overline{E_{w_i}}$ exceeds $\mu_{w_i}$, the master can control the strategy of the web service.

The market share offered to the web service by the community could cause dissatisfaction of the joined web services. Thus, this would generate a low $E_{w_i}$ value, which would cause the web service to leave considering its previous individual efficiency value.

## IV. EMPIRICAL ANALYSIS

We used a realistic multi-agent simulator in a java-based platform and developed many agents with broad range of characteristics and capabilities. In the multi-agent based environment, we exposed dynamism in agents' actions and therefore, we could obtain results that are based on the performed realistic experiments. In the implemented environments, there are three types of agents: (a) user agents; (b) web service agents; and (c) master web service agents that represent communities. We do not emphasize the user agents for the sake of simplicity. However, in general, they look for best possible web service (either from a single or a community of web services). During simulation runs, web services and users might leave or join the network. Table III

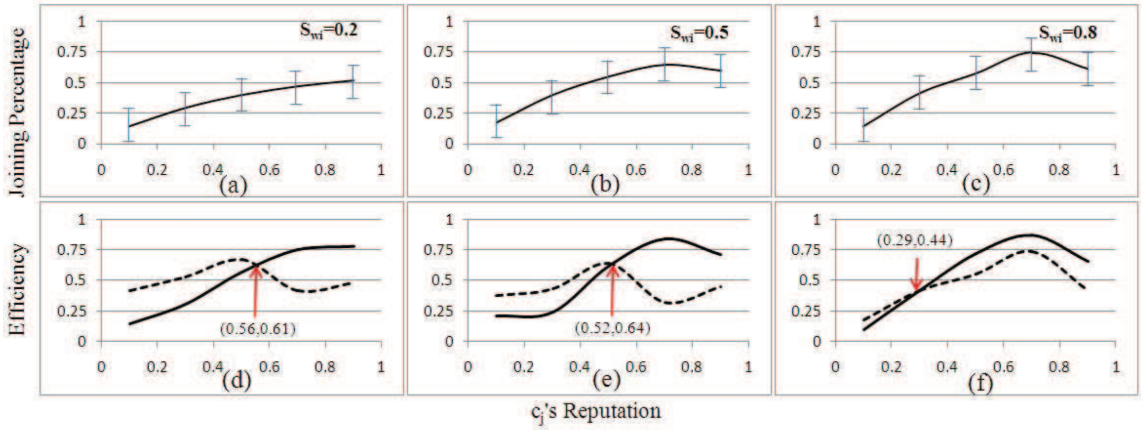


Figure 1. Efficiency of three categorized web services on joining a community.

provides the details regarding the implemented environment. We categorize the web services and masters based on the risk they take in adopting strategies. There are three classes of services that obtain different payoffs during the games.

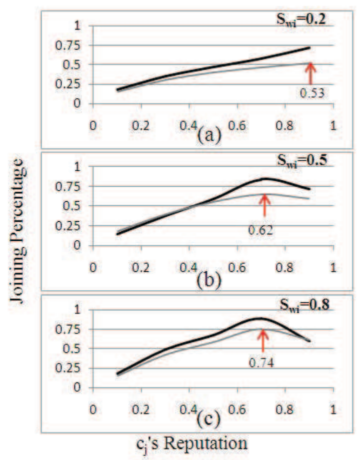


Figure 2. Efficiency of three categorized web services on joining a community while threshold being investigated.

Table III
ENVIRONMENT CHARACTERISTICS.

| Entity Type | Number | Risk Level | Environment Percentage |
|---|---|---|---|
| User | 1000 | --- | 86 |
| Web service | 50<br>50<br>50 | 0.2<br>0.5<br>0.8 | 4<br>4<br>4 |
| Community | 5<br>5<br>5 | 0.2<br>0.5<br>0.8 | 0.4<br>0.4<br>0.4 |

In this section, we investigate the characteristics of the single web services that act alone outside the community. During the simulation runs, we set up a number of one-stage games analyzing the strategies that web services take in different situations. We repeat the same process using three different classes of the web services according to their risk attribute. Figure III-C illustrates 6 plots categorizing three different types of single web services that are involved in the one-stage game regarding joining the community. In plots (a), (b), and (c) the x-axis denotes community's public reputation that is broadcasted by the player 2 ($c_j$) in the game. The y-axis denotes the percentage of the web services that considered to join the community. In this experiment, the community is willing to accept joining web services since its market share is not balanced with its limited capacity ($M_{c_j} > C_{c_j}$). As it is shown in plots, there are different joining percentages regarding the situation that either encourages or discourages most of the web services. In Figure III-C, plots (d), (e), and (f) illustrate the average efficiency comparison between the case where the web service was acting alone (the dotted curve) and the case where the web service joined the community (the bold curve). The updates in efficiencies clarify the extent to which the joining strategy is chosen wisely. In this experiment, the community adopts its strategies according to its individual efficiency analysis regardless of the threshold that could lead the web services to join.

We also lunch the experiment with the community representative that is capable of analyzing the threshold that would enhance the control of the master web service over the adopting strategies of the single web services willing to join and obtain higher efficiency. Figure IV plots the same group of web services (categorized in plots (a), (b), and (c)) facing a community whose master web service analyzes the threshold that could encourage the web services to join. As shown in this Figure, $c_j$ is more successful in games

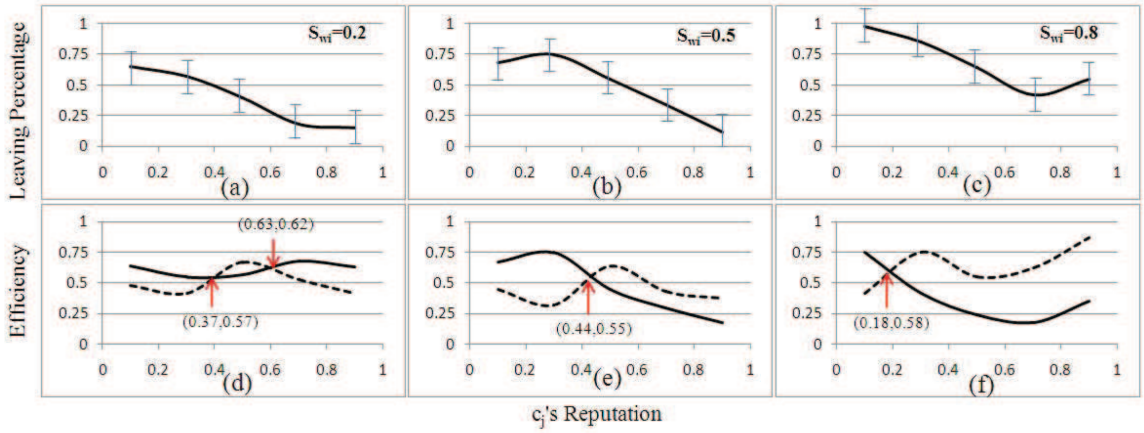


Figure 3. Efficiency of three categorized web services on leaving a community.

with players that hold relatively high risk attribute. In lower risky web services, the community is more successful in absorbing the web services by advertising higher reputation. This fact is promising according to web services' desire to increase self efficiency. However, the community facilitates the joining process and meanwhile, obtains the control on the strategies that the web services adopt. Thus, the master web service acts better compared to the case where the master web service considers self parameters in games.

We carry on the experiments with analysis on the efficiency updates regarding the joined web services that are involved in one-stage game facing the community representative. Figure IV illustrates 6 plots categorizing three types of web services according to their risk attribute class. These plots illustrate the percentage of leaving the community together with their corresponding efficiency update. As it is clear in plots (a), (b), and (c), the web services with lower risk levels act more or less according to their satisfaction of joining the community. Therefore, the percentage of leaving is decreased by increasing the reputation of the community. Note that the public reputation of the community cannot be faked in this case as long as the web service is already member of the community. The experiment shows the web services with higher risk level could adopt leaving strategy with weaker reasoning mechanism. Consequently, we observe a more chaotic behavior of the joined web service with higher risk level acting in a community with relatively low reputation value. This chaotic percentage is regulated while the reputation of the community is increased. In this case, the web services consider to refuse the leave.

Figure IV illustrates the leaving percentage of the web services in the same experiment but facing a community that manages to recognize the threshold $\mu_{w_i}$. In these plots we observe a better handling of the web services, which reflects community's success in controlling the adopting strategies of the web services.

In Figure IV, we compare the total efficiency of different communities categorized based on their efficiencies ($S_{c_j} = 0.2$, $0.5$, and $0.8$). In these plots, the bold curves represent the efficiency of the community when the threshold $\mu_{w_i}$ is taken into account and the dotted ones represent the community when the threshold is not taken into account. As shown in the plots, the efficiency of communities are enhanced when they consider the computed threshold.

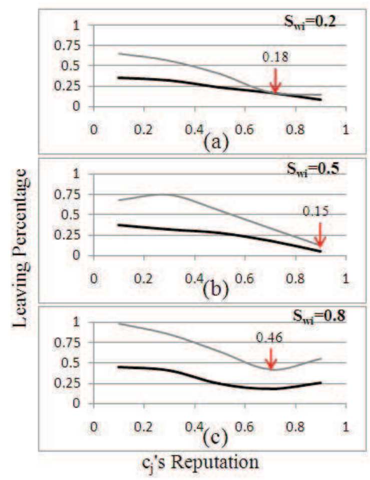


Figure 4. Efficiency of three categorized web services on leaving a community while threshold being investigated.

## V. Related Work

In many frameworks proposed in the literature, service selection and task allocation are regulated based on the reputation parameter [10], [11]. In [1], the proposed framework regulates the service selection based on the trust policies expressed by the service users. In [9], authors propose ontology for quality of service. Users compute the web services' reputation using ratings. The frameworks proposed in [3], [8] address effective reputation mechanism for web services. All these models address the reputation in

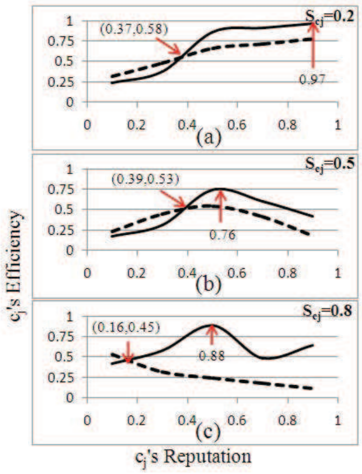


Figure 5. Efficiency of three categorized communities of web services.

environments where Web services function alone. In such models, web service efficiency is not discussed in details and in general, balancing the market share with the capacity is not considered as an issue for web service besides its reputation.

There have been few work addressing the communities of web service. The objective is to facilitate and improve the process of web service selection and effectively regulate he process of request and task allocation [2]. In [4], authors propose a reputation-based architecture for communities and investigate the collusion scenarios that might falsely increase communities' reputation in the network. In [5], the authors mainly address the overall assessed reputation that is used as a main reason for service selection. The authors do not consider efficiency as a parameter that impacts service selection in future. In general, the recent aforementioned proposals motivate the existence of communities rather than single functional web services, but fail to systematically provide potential benefits and technically compare different scenarios that increase service providers' efficiency.

## VI. Conclusion

The contribution of this paper is the proposition of a game-theoretic based model to analyze the best efficiency characteristics for the active services in open networks. The proposed framework measures the efficiency of the web services considering a number of involved factors. The proposed game measures the threshold that lead to a control of strategies adopted by the single web service.

Our model has the advantage of being simple and taking into account four important factors: (1) rational services seek better efficiency in the environment; (2) in service computing the collaboration concept is well defined if the maximum efficiency is posed as the main goal; (3) rational web services might meet higher performance either by joining a community (for the sake of collaboration) or acting alone (for managing the task alone); and (4) the community is capable of managing the number of involving web services. The resulting model shows that the efficiency of the community is increasing once the game-theoretic analysis is considered to impose parameters to control the cardinality of the community set. As future work, we plan to consider the user role in the game to obtain more accurate results when users act rationally. Moreover, we would like to achieve a collusion resistant efficiency mechanism.

## References


[1] A.S. Ali, S.A. Ludwig, and O.F. Rana. A cognitive trust-based approach for Web service discovery and selection. In Proc. of the 3rd European Conference on Web Services, pp. 38-40, 2005.

[2] M. Jacyno, S. Bullock, M. Luck, T.R. Payne. Emergent Service Provisioning and Demand Estimation through Self-Organizing Agent Communities. 8th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 481-488, 2009.

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

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

[5] B. Khosravifar, J. Bentahar, P. Thiran, A.Moazin, and A. Guiot. An approach to incentive-based reputation for communities of Web services. In Proc. of IEEE 7th International Conference on Web Services, pp. 303-310, 2009.

[6] B. Khosravifar, J. Bentahar, A. Moazin, Z. Maamar, and P. Thiran: Analyzing communities vs. single agent-based web services: trust perspectives. In Proc. of the IEEE international Conference on Services Computing, pp. 194-201, 2010.

[7] Z. Maamar, S. Subramanian, P. Thiran, D. Benslimane, J. Bentahar. An approach to engineer communities of web services: Concepts, architecture, operation, and deployment. In International Journal of E-Business Research 5(4): 1-21, 2009.

[8] E.M. Maximilien and M.P. Singh. Conceptual model of Web service reputation. SIGMOD Record 31(4):36-41, 2002.

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

[10] S. Rosario, A. Benveniste, S. Haar, and C. Jard. Probabilistic QoS and soft contracts for transaction based web services. IEEE International Conference on Web Services, pp. 126-133, 2007.

[11] M. Ruth and T. Shengru. Concurrency issues in automating RTS for Web services. IEEE International Conference on Web Services, pp. 1142-1143, ICWS 2007.