A UTILITY-BASED REPUTATION MODEL FOR SERVICE-ORIENTED COMPUTING*

Gheorghe Cosmin Silaghi
STFC Rutherford Appleton Laboratory and *University of Coimbra, DEI*
Chilton, Didcot, OX11 0QX, UK *Polo II, 3030-290, Coimbra, Portugal*
g.c.silaghi@rl.ac.uk gsilaghi@dei.uc.pt

Alvaro E. Arenas
STFC Rutherford Appleton Laboratory
Chilton, Didcot, OX11 0QX, UK
a.e.arenas@rl.ac.uk

Luis Moura Silva
University of Coimbra, DEI
Polo II, 3030-290, Coimbra, Portugal
luis@dei.uc.pt

Abstract Reputation systems have emerged as a method for fostering trust amongst strangers in electronic transactions. In this paper we propose a utility-based model for reputation tailored for service-oriented computing. In contrast to most other reputation models that require direct feedback from users, our model build the reputation from information provided by monitoring systems, making it suitable for service-oriented settings such as Grids. Usefulness of the model is described by showing how the efficiency of resource brokering in Grids can be improved by using a reputation-based scheduler scheme.

Keywords: Reputation; Service-Oriented Computing; Utility Computing; Grids.

*This work is funded by the European Commission under the IST FP6 projects CoreGRID –project No.004265– and GridTrust –project No. 033827–.

1. Introduction

Reputation systems have emerged as a method for fostering trust amongst strangers in electronic transactions. A reputation system gathers, distributes, and aggregates feedbacks about participants' behaviour. The feedback is usually provided as a-posteriori operation requiring human intervention. One major drawback is that users will try to misrepresent the obtained quality of services in order to make more profit, or to lie or provide misleading ratings in order to achieve some specific goals [5].

This way of building reputation is useful in semi-automatic contexts, such as electronic marketplaces where users rate sellers, but it becomes a limitation in fully automatic contexts. For instance, Grid computing focuses on the development of a distributed service environment that integrates a variety of resources with various quality of service capabilities in order to support scientific and business problem solving environments. In a typical Grid scenario, middleware services like brokers select resources by obtaining basic information about them from directory services; the type of information usually includes functional and quality of service properties of the resource. The selection process can be enriched by providing reputation information about resources. However, in several Grid applications, resource usage is transparent to the end user, making difficult to obtain a qualification.

This paper proposes a reputation model that overcomes above limitations by using some basic assumptions considered valid in service-oriented architectures, and Grids in particular, such as the existence of trustworthy monitoring systems. A monitoring system provides information about services and the actual service delivery. Instead of asking the user to provide the feedback after having a transaction, we use monitoring information as a substitute to build the direct reputation.

Our reputation model is based on ideas of utility computing. User feedback is represented as a utility function which reflects the satisfaction a user perceives from consuming a service. The user provides such a utility function *before* committing to use a service either by building the utility function herself or by selecting the function from a library of utility functions. The user's utility function will be then applied on the monitoring information in order to calculate the reputation of the service.

Section 2 describes our proposed reputation model. Next, in section 3 we show how the metrics of the model work by experimenting with various types of service delivery, and describe a potential application of our reputation model for improving resource brokering in Grids. Then, section 4 compares related work. Finally, we conclude the paper in section 5 by summarising main results and highlighting future work.

2. A Utility-Based Reputation Model

This section describes a utility-based reputation model that aims at registering the reputation of service providers based on the satisfaction of users.

DEFINITION 1 ***(Services, Issues and Expectations)***
Let $X = \{x_1, x_2, \ldots, x_n\}$ denote the set of services, with x ranging on X. Let SP denote the set of services providers, with b ranging on SP, and function $S : SP \rightarrow \mathscr{P}(X)$ denoting the services provided by a service provider, where $\mathscr{P}$ represents the power set operator. Let SC denote the set of users (service consumers) of the system, with c ranging on SC.

Each service has associated issues of interest, denoted by set I, which users are interested in monitoring; variable i ranges on I. Function IS represents the set of issues of interest for a service: $IS : X \rightarrow \mathscr{P}(I)$. Function $\mathscr{O}^c : X \times SP \times I \rightarrow \mathscr{R}$ denotes the expectation of user c on the services he uses, where $\mathscr{R}$ denotes the real numbers. Notation $v_{x,i}^{b,c}$ represent the expectation of user c on issue i of service x supplied by provider b.

For instance, in classical service-oriented architectures, a potential issue of interest could be the *quality of service*. In this case, the user expectation would be the *service level agreement*, the formal negotiated agreement between a user and his service providers.

Based on his expectation, a user can develop a utility function which reflects the satisfaction he perceives from consuming the service.

DEFINITION 2 ***(Utility Function)***
Let $U_{x,i}^{c,b}(v)$ denote the utility that user c gets by obtaining the actual value $v \in \mathscr{R}$ on issue i from service x of provider b. Utilities will be normalized and scaled to $[0,1]$, getting the user a utility of 1 if provider b actually supplies with the expected value $v_{x,i}^{b,c}$ for issue i from service x. If the provider supplies a better quality, the user gets the utility of 1. Therefore, we have $U_{x,i}^{c,b} : \mathscr{R} \rightarrow [0,1]$.

If the service has a direct valuation scale (i.e. bigger supplied value, better the satisfaction the user gets), equation (1) can be an example of a utility function.

$$U_{x,i}^{c,b}(v) = \begin{cases} 1 & v \geq v_{x,i}^{b,c} \\ \frac{v}{v_{x,i}^{b,c}} & v < v_{x,i}^{b,c} \end{cases} \tag{1}$$

We assume that the IT infrastructure provides a trustable monitoring service that delivers regularly events indicating the current value of the issues of interest for those services in execution. Events are captured by the reputation engine in order to generate the reputation values at different levels - issue, service or service provider.

DEFINITION 3 ***(Events)***
An event $e = (\ (c, b, x, i),\ t,\ v\ )$ indicates that at time t the issue of interest i for service x provided by b for user c has value v. The set of events E consists of triples $(\ (SC \times SP \times X \times I) \times \mathscr{N} \times \mathscr{R}\ )$, where $\mathscr{N}$ and $\mathscr{R}$ stand for the natural and real numbers respectively.

For each event reported by the monitoring service, having the utility function of a consumer c, one can compute the instant utility $U^{c,b}_{x,i}(v)$, indicating the actual satisfaction the consumer is getting at that moment. This allows us to calculate the reputation of a service provider.

DEFINITION 4 ***(Reputation Function for Issues of Interest)***
The reputation of a service provider b in relation to issue i of service x at time t can be defined as follows:

$$R^b_{x,i}(t) = \frac{\sum_{c \in SC} \sum_{((c,b,x,i),t_e,v_e) \in E \wedge t_e \leq t} \varphi(t, t_e) * U^{c,b}_{x,i}(v_e)}{\sum_{c \in SC} \sum_{((c,b,x,i),t_e,v_e) \in E \wedge t_e \leq t} \varphi(t, t_e)} \tag{2}$$

where $\varphi(t, t_e)$ is a time discount function which puts more importance on events closer to present.

It is worth noticing that the reputation measure incorporates information supplied by various users who consumed the service in the past. Reputation equation (2) is inspired by the aggregation presented in [9].

As in [9], we developed the reputation deviation to provide a fitness measure for the reputation value. The reputation deviation shows how much the reputation varies in time. In contrast to [9], in our model, the lower the reputation deviation, the better the confidence one can put on the reputation value $R^b_{x,i}$.

DEFINITION 5 ***(Reputation Deviation for Issues of Interest)***
The reputation deviation of a service provider b in relation to issue i of service x at time t can be defined as follows:

$$DR^b_{x,i}(t) = \frac{\sum_{c \in SC} \sum_{((c,b,x,i),t_e,v_e) \in E \wedge t_e \leq t} \varphi(t, t_e) * \left| U^{c,b}_{x,i}(v_e) - R^b_{x,i}(t) \right|}{\sum_{c \in SC} \sum_{((c,b,x,i),t_e,v_e) \in E \wedge t_e \leq t} \varphi(t, t_e)} \tag{3}$$

Based on above definitions, we can derive the reputation of a service as the aggregation of the reputations on the issues of interest of such service.

DEFINITION 6 ***(Reputation and Reputation Deviation for a Service)***
The reputation and reputation deviation for a service provider b in relation to

service x at time t can be defined as follows:

$$R_x^b(t) = \frac{\sum_{i \in IS(x)} R_{x,i}^b(t)}{\#IS(x)} \qquad DR_x^b(t) = \frac{\sum_{i \in IS(x)} \left| R_{x,i}^b(t) - R_x^b(t) \right|}{\#IS(x)} \tag{4}$$

where $\#$ corresponds to the cardinality of a set.

Likewise, the reputation of a service provider can be defined as the aggregation of the reputation of the services it provides.

DEFINITION 7 ***(Reputation and Reputation Deviation for a Service Provider)*** *The reputation and reputation deviation for a provider b in relation with all services it delivers at time t can be defined as follows:*

$$R^b(t) = \frac{\sum_{x \in S(b)} R_x^b(t)}{\#S(b)} \qquad DR^b(t) = \frac{\sum_{x \in S(b)} \left| R_x^b(t) - R^b(t) \right|}{\#S(b)} \tag{5}$$

In our model we require a user to deliver the utility function for their tasks. Can the user indeed formulate the utility function for his tasks? In [2], it is acknowledged that finding the utility function of all grid actors is a difficult task. In economic market approaches to the grid ([2, 1]), the emphasis is put on producers and consumers which take decisions according with their internal utility functions and the outcome of these decisions is the pricing assessment for the grid services. Therefore, one can learn the internal utility functions of the grid agents by observing how they price the grid services. As our goal is to keep our reputation model simple, we do not intend to enter the scope of accountability in grids. Therefore, if the user is not able to define his utility functions, the reputation manager will simply ask the user to characterize how important is the realization of the expected values for a given task. Based on this response (which can be a discrete one, i.e. very important, don't know, not so important), the reputation manager can assign a utility function selected from a template library.

3. Evaluation of the Reputation Model

Some simulations were performed in order to validate the intended properties of our reputation model.

3.1 Initial Experiments

Initially, we consider the simplistic approach in which a resource provider supplies one service –storage– and the service has one issue of interest –storage capacity–. The user expectation for storage capacity is 100Gb, and uses the same utility function (equation (1)) for all his tasks during the time of the experiment.

Further, we assume that the provider delivers the service according with some pre-established patterns. The reputation function for a issue of interest (equation 2) requires a time discount function; following [3], the time discount function is defined as $\varphi(t, t_j) = e^{-\frac{t-t_e}{\lambda}}$. We took a time frame of 1000 time units (tu) for our experiments, using $\lambda = 200$. The reputation engine is set up to compute the reputation values according with the formulas presented in section 2 at every 20 time units.

The provider supplies a storage capacity uniformly distributed around the expected value of 100Gb. We took a variation band of 20Gb, from 85-105. Figure 1 shows how the reputation measure reflects this pattern of delivery for the service and the issue. We can notice that after a short learning time frame, the reputation stabilizes itself around some value. Next, we generated a harder drop down in the delivery of the capacity issue at a provider site. Between time units 200 and 300 the provider drops the supplied capacity with 20Gb. Figure 2 shows the results. We can notice that the reputation value immediately drops after the the fall in delivery and it does not recover at the initially existing level even at the end of the time frame.

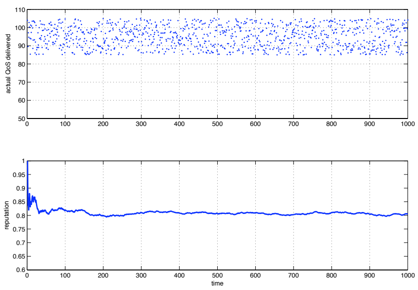

Figure 1. Reputation when the issue is delivered normally distributed around the expected value

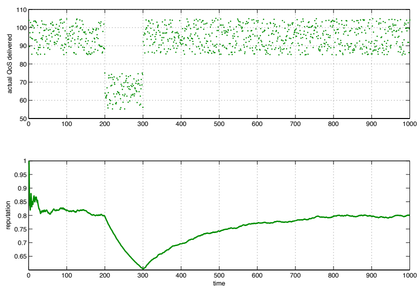

Figure 2. Reputation when there is a decay in delivery

The next experiment emphasizes the usage of reputation deviation. We consider two providers: first provider supplying uniformly distributed values in a variation band (line dotted with circles in Figure 3), and second provider supplying normally distributed values with the same mean and a deviation being half of the bandwidth selected for the first provider (line dotted with squares in Figure 3). Let the variation band to be between 85 and 105. Analyzing the upper diagram of figure 3, we can notice that reputation itself is not enough to characterize which provider is more reputable. However, the reputation deviation, as expected, is smaller (i.e. better) for the second provider, which makes the difference between the providers. Based on the reputation deviation, one

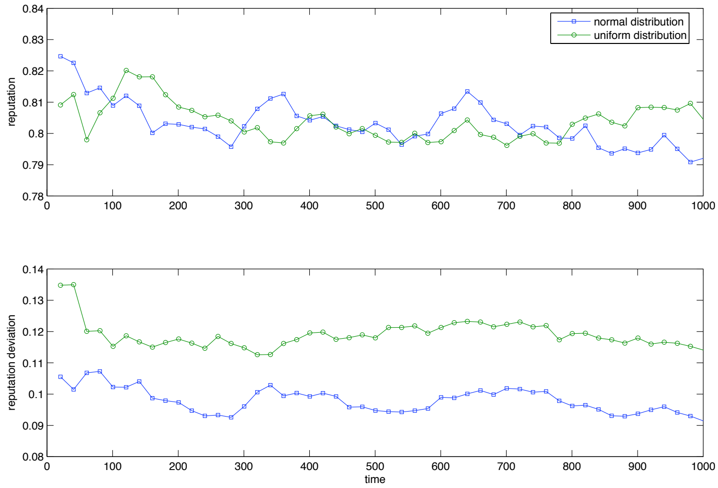

Figure 3. Reputation and reputation deviation in the case of different QoS deliveries

can decide which reputation value is more reliable and distinguish between the above patterns of delivery.

3.2 Enhancing Resource Brokering with Reputation

Resource brokering is defined as the task of selecting and allocating the most appropriate resource for a given job. Resource brokering is a wide research area in grids and is not the scope of this paper to enter the full details of this problem. Instead, we want only to point out that using reputation can be beneficial for the effectiveness of the brokering process, compared with simplistic approaches that use no intelligent technique to tackle this problem.

We performed our experiments using the SimGrid simulator [8], on which we implemented the following reputation-based scheduling scheme: when scheduling a job to a resource, the resource broker considers all available nodes that fulfill all service requirements for that job; it then schedules the job to the most reputable node. We implemented also a monitoring service that observes the quality of the service delivery during the execution and reports events to a reputation manager, following our reputation model.

In our experiments, we compared a simple brokering algorithm like round-robin scheduling with our reputation-based scheme. Round-robin scheduling has also been used as a base comparison in [2]. In the round-robin approach, the broker immediately schedules a task that arrives in its queue to the next grid node that comes in the round-robin scheme. To measure the difficulty of the brokering problem, load factor parameter was used [2]: load factor is light if in a certain period of time the number of jobs submitted is small, and the length of the jobs are short; otherwise the system load is heavy. We use the same parameter for the x dimension of Figures 4 and 5. For our experiment, we allow 20% of the nodes to produce random values uniformly distributed in a variation band between $85 - 105\%$ of the expected service value. We use the time discount function defined previously, with λ tuned to 200. In order to keep low the memory of events related with an issue of a service provider, we discard all events for which $\varphi(t, t_e) < 0.01$

In the experiment we record two parameters: the *total completion time* for the entire batch of jobs and the *total welfare* produced in the system by counting all utilities acquired by the users for the submitted jobs. Figure 4 depicts the total completion time for different loads of the system. We can notice that with reputation-based scheduling, the total completion time is better with around 25%. This 25% gain in completion time can be very significant in the case of high load factors.

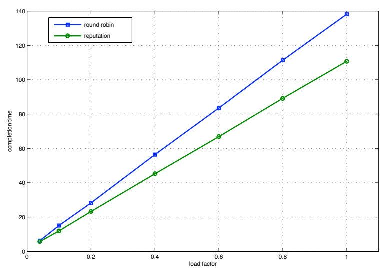

Figure 4. Completion time comparison

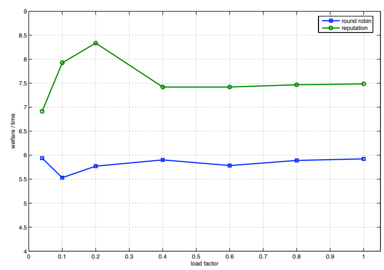

Figure 5. Total welfare produced per unit of time comparison

Figure 5 compares the total welfare produced on the time unit. In the case of the reputation-based scheduling, the efficiency of the system, measured as the amount of welfare produced on a time unit, is higher. Regardless the load of the system, the user feels more satisfied from consuming the services in the utility-based scheduling. A good outcome is the fact that the amount of welfare produced per unit of time stabilizes for high load of the system. Again, about 25% gain of welfare is recorded. For both cases, only 20% of nodes in the grid violated the expectations with only maximum 15%, leading to an average of only 3% violation of the expectations for the overall grid.

4. Related Work

There are some previous models of reputation for Grids system. The GridEigen-Trust model [7]integrates trust management as part of the QoS management system. They consider both direct and indirect trust, acquired at the level of grid entities and contexts (i.e. service delivery), after transaction execution. PathTrust[6]proposed a reputation system for member selection in the formation phase of a virtual organization. When inviting members to join a VO, the initiator selects only those members whose reputation is above a certain threshold and probabilistically selects a member to be in the VO, as we did for selecting node where to schedule some task. The reputation is built by aggregating positive and negative feedback the user submits after transaction execution. These two approaches have the limitation on depending on the direct feedback from users.

Reputation can be built based on the internal beliefs of the agents, as in the subjective logic of [4]. In multi-agent research, techniques based on aggregating over several sources of trust have been considered [3, 9]. REGRET model [9]aggregates over the individual, social and ontological dimension in order to obtain a reputation to be used on an electronic marketplace. They inspired us with the aggregation metric for the reputation and with the reputation deviation. But their model is based on impressions collected after the transaction, which limits the applicability of the model to the grid. We distinguish from them overcoming this limitation by asking the utility function of the user before the transaction and by the usage of the trusted monitoring information. More, in our approach, the reputation is aggregated considering the view of several users.

5. Conclusion

In this paper we have presented a reputation model based on ideas of utility computing, tailored for service-oriented contexts. For each service, the user defines issues of interest and expected values on such issues. The satisfaction of the user on a service is measured by a utility function. Reputation of a service issue is then built by comparing its expected value with the actual value, delivered by a monitoring system. Reputation of a service is built as the aggregation of the reputations of its issues. Likewise, reputation of a service provider is built as the aggregation of the reputation of all services it delivers. This robust and scalable way of calculating reputation does not depend on direct feedback collected after the transaction. We have shown the usefulness of the model in improving the efficacy of resource brokering in Grids when using a reputation-based scheduling scheme.

References

[1] R. Buyya, D. Abramson, and J. Giddy. A case for economy grid architecture for service oriented grid computing. In *IPDPS '01: Proceedings of the 10th Heterogeneous Computing Workshop - HCW 2001 (Workshop 1)*, page 20083.1, IEEE Computer Society, 2001.

[2] Li Chunlin and Li Layuan. Multi economic agent interaction for optimizing the aggregate utility of grid users in computational grid. *Applied Intelligence*, 25(2):147–158, 2006.

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

[4] A. Jøsang and S.J. Knapskog. A metric for trusted systems. In *Proceedings of the 21st National Information Systems Security Conference, (NIST-NCSC 1998)*, 1998.

[5] A. Jøsang, R. Ismail, and C. Boyd A Survey of Trust and Reputation Systems for Online Service Provision *Decision Support Systems*, 43(2):618–644, 2007.

[6] F. Kerschbaum, J. Haller, Y. Karabulut, and P. Robinson. Pathtrust: A trust-based reputation service for virtual organization formation. In *iTrust2006: Proceedings of the 4th International Conference on Trust Management*, volume 3986 of *Lecture Notes in Computer Science*, pages 193–205. Springer, 2006.

[7] G. von Laszewski, B.E. Alunkal, and I. Veljkovic. Towards reputable grids. *Scalable Computing: Practice and Experience*, 6(3):95–106, 2005.

[8] A. Legrand, L. Marchal, and H. Casanova. Scheduling distributed applications: the SimGrid simulation framework. In *CCGRID '03: Proceedings of the 3st International Symposium on Cluster Computing and the Grid*, page 138, IEEE Computer Society, 2003.

[9] J. Sabater and C. Sierra. Regret: a reputation model for gregarious societies. In *4th Workshop on Deception, Fraud and Trust in Agent Societies*. ACM Press, 2001.