QoS and Reputation-aware Service Selection

Noura Limam
University of Paris VI
Paris, France
Email: noura.limam@lip6.fr

Raouf Boutaba
University of Waterloo
Waterloo, ON, Canada
Email: rboutaba@uwaterloo.ca

***Abstract*—The advent of service-oriented architectures has created a unique opportunity for business providers and consumers to establish more versatile and flexible interactions across the Internet by means of a new generation of services that are discoverable, composable, configurable, and reusable. In order to support such services all along their life cycle, underlying service-oriented infrastructures have to provide various functionalities, including service discovery. In large scale environments, like the Internet, a discovery process may result in a very large number of matching services. Service quality, cost and reputation are substantial aspects for differentiating between similar services. In order to help users select the most appropriate service, an automated service selection algorithm is proposed. The devised algorithm helps to accurately predict service suitability to quality requirements while taking into account the reputation parameter.**

I. Introduction

The significant growth and globalization of the IT service market have created a strong incentive for more flexible IT solutions in the support of more productive and more competitive businesses. In this perspective, the Service-Oriented Architecture (SOA) has been designed to leverage IT to deliver the level of agility that is necessary for business solutions to compete. Organizations pushing their way to the global economy are offered to gain in agility and value through the consumption of interoperable, loosely-coupled and reusable services in business composition. Leveraging existing services is expected in fact to reduce substantially integration cost, maintenance effort and time to market.

Service reuse is the key challenge of SOA. Reusability is certainly profitable to business agility, but most importantly essential to the success of SOA infrastructures. In this perspective, a number of enabling functions are designed to support and facilitate service reuse. Service discovery, for instance, is a critical function through which service opportunities are made available to organizations for completing new or on-going businesses. Service discovery is indeed necessary to building and composing new businesses, but it is also essential to their management. In fact, composite processes are inherently vulnerable to failures, including software, machine or communication failures, due to the distributed and highly volatile nature of their service components. Failures may result in the unavailability of a service component and hence the failure of a business sub-process that may prevent the successful execution of the whole business process. Given that alternatives to each service component may be offered by different providers, a replacement component could be discovered and substituted for the failed one.

In large scale environments, the response to a discovery request may however result in a very large number of matching services, equivalent functionality-wise. A dynamic service selection mechanism is hence required to find at design time the most appropriate service to complete the business process, and at run-time the best alternative to a failed component. In a highly competitive business environment like the Internet, quality and cost are substantial aspects for differentiating between similar services. However, in order to ensure that the delivered quality will meet the offer made by the provider, the requirements of the business and the expectations of the consumers, it is essential that the reputation of the service is taken into consideration.

We investigate in this work the automation of service selection. We devise a service selection algorithm to accurately predict service suitability to user's quality and cost constraints while taking into account service reputation. A computational model is also provided to measure the trustworthiness and credibility of service offerings. The proposed model is based on the measurement of service utility and relies on the expectancy-disconfirmation theory from market science.

The remainder of the paper is organized as follows. Section II defines the terminology used throughout the paper. Section III discusses the motivations and the contributions of the work presented in this paper. In Section IV, we present our automated service selection algorithm. Section V provides few hints for the automation of the rating process and the design of a reputation system. Section VI presents simulation results and Section VII discusses related works. Finally, Section VIII concludes this paper.

II. Terminology

- *Quality of service (QoS)*: We define QoS as a measure of the fulfilment of the service agreed upon. QoS is a broad concept that encompasses multiple nonfunctional properties, or dimensions, some of which can be service-specific and others more general such as availability, responsiveness and reliability.
- *Trust*: Trust is commonly defined as a belief of confidence or a feeling of certainty that one person has in another person or thing that he/she is interacting with. In our context, similarly to [1] we define trust as the probability

978-1-4244-2066-7/08/$25.00 ©2008 IEEE

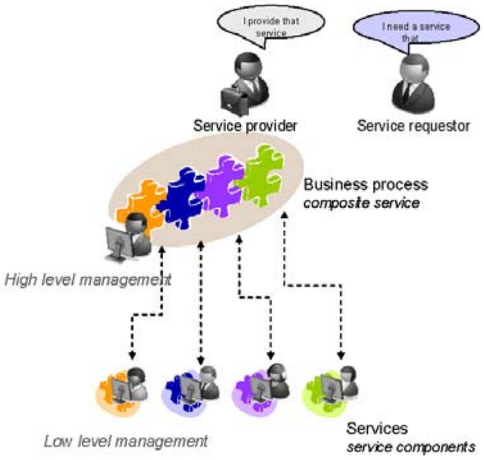

Fig. 1. Business integration and management

by which a user expects that a service performs a given action on which its welfare depends.
- *Feedback or rating*: After the completion of a service transaction, the user is expected to rate the service by providing a feedback. The feedback measures user's experience with the service; it represents the opinion of the user on the fulfillment of the service considering the agreement between the user and the service provider.
- *Reputation*: The reputation of a service is a collective measure of the opinion of a community of users regarding their experience with the service. It is computed as an aggregation of users' feedbacks and reflects the reliability, trustworthiness and credibility of the service and its provider.

III. Motivations and contributions

To fully leverage SOA principles, business processes are built on any opportunity of reusing existing services within the organization or across its boundaries (see Figure 1). The composed business is brought into the market with a quality offer that the provider commits to deliver to its customers. The distributed and cross-boundary nature of service components raises however a critical management issue. More generally, the management of composite services is structured in two levels (see Figure 1). Higher level management is supported by the provider of the composite service and includes the tasks required to manage the fulfillment of the overall process, in particular the execution of the workflow. The lower level involves the management of service components by their respective providers, with respect to partnership agreements. Considering that the composite service provider has a very limited control over the execution quality of service components, maintaining the "promised" quality level is very challenging. In addition, recovering a component failure requires generally applying changes to the current composite service which may conflict with the current quality offer and/or agreement.

The autonomous and loosely-coupled nature of service components facilitates the recovery task. In some cases, the component may be replaced at run-time by an equivalent one. More generally, the complexity of recovering a component failure is variable and different actions are required accordingly. The failure of a service and its recovery may result in a revenue shortfall for the business provider. To minimize this risk, the following strategy should be implemented:

- *Assurance of the delivered service quality*: QoS assurance is required at composition-time and more precisely at the discovery and selection phase. It is hence important that the selection procedure results in the service component that will respect the quality agreement.
- *Recovery time optimization*: recovery implies the search for an alternate component to replace the failed one and hence involves a discovery and selection process. Service selection is commonly conducted by the users. This task is energy and time-consuming with the increasing number of candidate services.

Service discovery and selection are critical to the management and maintenance of composite services. For the sake of time effectiveness it is important to automate the selection process. Commonly, service components are selected according to the providers' quality offers. Thus, the compliance of the actual execution quality with the offer is not determined until the service is rendered. Reputation systems have been introduced to help predicting the trustworthiness of service providers. As a matter of fact, it is possible to measure the confidence in the claimed quality offer and predict the conformance of this latter to the future service execution quality, provided reputation reports are credible.

The management of composite services creates thus an incentive for an automated quality and reputation aware service selection mechanism. In the following we present a service selection mechanism that we believe will help in enforcing SOA principles and increasing business agility.

IV. Automated selection process

As opposed to service discovery which has been subject to intensive research works, few studies have tackled the selection automation issue and fewer have considered service reputation in decision support. Service selection is a multi-criteria multi-choice decision making process which resolution commonly involves a trade-off between quality and cost. As explained earlier, there is no guarantee on service quality at selection time, however reputation can help predicting the likeliness of a quality offer to be met. As a matter of fact, selection can translate into a three-criteria decision making problem involving reputation, quality and cost. This problem can be simplified into a single-criterion problem provided quality reputation and cost are aggregated into a single selection metric.

The resolution of the selection problem entails three steps: *match-making*, *evaluation* and *ranking*.

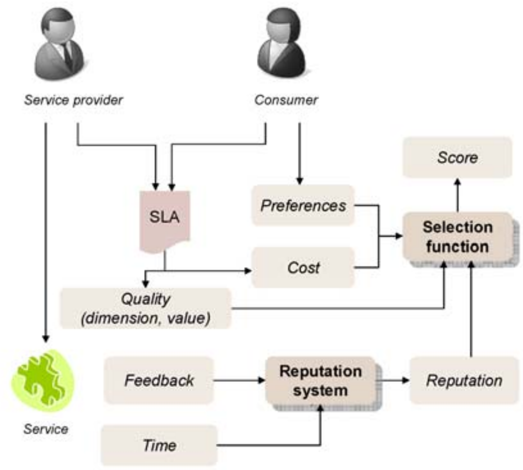

Fig. 2. Automated selection process

A. Match-making step

This step consists in comparing service offers against user requirements. All the offers which do not meet user requirements, expressed commonly in terms of quality and cost constraints, are ignored. Quality offers are commonly defined as a vector of (Q_i, q_i) pairs where Q_i refers to a quality dimension, and q_i refers to the corresponding value. Quality dimensions may have different characteristics, such as different definition domains and different evaluation rules. For instance, service availability is defined on $[0,1]$ and obeys to the "more-is-better" evaluation rule, whereas service response time is defined on $[0,+\infty[$ and as opposed to the availability dimension, obeys to the "less-is-better" evaluation rule. The evaluation of a quality offer, is challenging considering that it requires the knowledge of such properties. In this perspective, it is essential to provide formal specification of quality dimensions. Figure 2 suggests that quality dimensions are formally specified in the SLA.

For each discovered service s, let qos be the offered quality vector, R its reputation and $cost$ its cost. At design-time requirements may be expressed as end-to-end, i.e. business-level, constraints. At management time however, and more precisely when a component must be replaced, service selection is performed with component-level constraints. Let $qos_r = (q_1)_r, (q_2)_r, ..., (q_N)_r$ be the vector of quality constraints.

We denote by QOS^+ the subset of *more-is-better*-like QoS parameters, and QOS^- the subset *less-is-better*-like parameters.

Let S be the set of preselected services,

$$s \in S \quad \text{if} \quad \forall i = 1..N \begin{cases} \text{if} \quad Q_i \in QOS^-, & (q_i)_r \leq q_i(s) \\ \text{if} \quad Q_i \in QOS^+, & (q_i)_r \geq q_i(s) \end{cases}$$

Similarly restrictions on cost may apply; unaffordable services are also ignored.

B. Evaluation step

During this phase, services are evaluated with respect to two aspects: the offer and the likeliness that consumer's expectations will be met. At this stage, all the eligible services offer a quality level that is equal to or higher than requested and come at affordable costs. We will thus evaluate service offers in terms of the *gain* in quality and cost that is proposed. As outlined earlier, R is the measure of service's likeliness to meet consumer's quality constraints. Let Q and C be the evaluation metrics of gains in quality and cost respectively. We hereby define R, Q and C as scalar values comprised between 0 and 1.

Considering that the quality offer is defined as a vector, we first evaluate the gain in each quality dimension. For each quality dimension Q_i, we define two parameters $(q_i)_{max}$ and $(q_i)_{min}$ as follows:

$$\begin{cases} (q_i)_{max} = \begin{cases} \max_{s \in S} q_i(s) & \text{if} \quad Q_i \in QOS^+ \\ (q_i)_r & \text{if} \quad Q_i \in QOS^- \end{cases} \\ (q_i)_{min} = \begin{cases} \min_{s \in S} q_i(s) & \text{if} \quad Q_i \in QOS^- \\ (q_i)_r & \text{if} \quad Q_i \in QOS^+ \end{cases} \end{cases}$$

The scaling function $Scal_i$ is defined on $dom(Q_i)$ and takes values in $[0,1]$. $Scal_i$ is increasing for $Q_i \in QOS^+$ and decreasing for $Q_i \in QOS^-$.

$$Scal_i(q_i) = \begin{cases} \frac{q_i - (q_i)_{min}}{(q_i)_{max} - (q_i)_{min}} & \text{if} \quad Q_i \in QOS^+ \\ \frac{(q_i)_{max} - q_i}{(q_i)_{max} - (q_i)_{min}} & \text{if} \quad Q_i \in QOS^- \\ 1 & \text{if} \quad (q_i)_{max} - (q_i)_{min} = 0 \end{cases}$$

We can easily demonstrate that for all $i = 1..N$, $Scal_i((q_i)_r) = 0$.

We now derive the scalar metric Q from the vector $(Scal_i(q_i))$. We denote by $W = (w_1), (w_2), ..., (w_N)$ consumer's quality preferences, where $0 \leq w_i \leq 1$ and $\sum_{k=1}^{N} w_i = 1$. $(Scal_i(q_i))_{i=1..N}$ represent the coordinate of the candidate service s in the N-dimensional Euclidean space where the origin represents the coordinate $Scal_i((q_i)_r)$ of, say, service s_0. We compute Q as the weighted Euclidean distance $Q = ||s - s_0||$ as follows:

$$Q = \sqrt{\sum_{i=1}^{N} w_i q_i^2(s)}$$

Q also represents the weighted root-mean-square of $(q_i)_{i=1..N}$. We finally compute C the same way we have scaled a quality parameter in QOS^-.

C. Ranking step

In the following, we aggregate R, Q and C into $Score(s)$, the ultimate selection metric, and select the service with the highest $Score(s)$. In [2], a weighted mean-like aggregation function on all quality parameters including cost and reputation is used, where weights are user-defined constants. In our opinion, reputation should not be considered as a quality parameter; we view reputation as a moderator between service quality and quality guarantee. Moreover, a weighted mean-like score function implies that the score is evenly sensitive to variations in reputation, respectively quality and cost, independently of reputation, respectively quality and cost, values. Although we can approve that the score is evenly sensitive to variations in Q or C, we believe that it should be less sensitive to variations in "low" reputation values than to variations in higher values. In fact, we observe that under a reputation ceil, consumer's "faith" in the service is lost which means that the service is no longer believed to deliver the expected quality.

According to the above observations, the $Score(s)$ should increase linearly with Q and C and exponentially with R; this would emphasize the insensitivity of $Score(s)$ to variations in low R values. We also consider that $Score(s)$ takes values in $[0, 1]$. We denote by $Score_R(s)$, $Score_Q(s)$ and $Score_C(s)$ the partial derivatives of $Score(s)$ with respect to R, Q and C respectively. We obtain:

$$\begin{cases} Score_R(s) = \lambda \; Score(s) \\ Score_Q(s) = \alpha \\ Score_C(s) = \beta \\ \lambda \quad > \quad 0 \quad and \quad \alpha, \beta \quad \in \quad [0, 1] \end{cases} \tag{1}$$

We also have:

$$Score(s) = 1 \quad \text{if} \quad R, Q, C = 1 \tag{2}$$

$$Score(s) = 0 \quad \text{if} \quad R, Q, C = 0 \tag{3}$$

We integrate the partial derivatives of $Score(s)$ from Equation 1, we obtain:

$$Score(s) = e^{\lambda(R-\gamma)} + \alpha \; Q + \beta \; C + \psi \tag{4}$$

With $\gamma > 1$ and $\psi = -e^{-\lambda\gamma}$ (from Equation 3). α and β weight the impact of the quality and the cost attributes, respectively, on the score function. As shown in Figure 3, λ and γ control the impact of reputation on the score function. More precisely, λ controls the growth rate and γ the function's range. We observe that the higher is λ, the more convex is the function, while the closer is γ to 1, the larger is the function's range. A high λ value and a γ value close to 1 are more compliant with the desired characteristics of the score function. Moreover, we obtain a very desirable simplification of $Score(s)$ by setting γ to 1:

$$Score(s) = e^{\lambda(R-1)} + e^{-\lambda}(\alpha e^{\lambda} \; Q + \beta e^{\lambda} \; C - 1)$$

From Equation 3, we obtain:

$$\alpha e^{\lambda} + \beta e^{\lambda} = 1$$

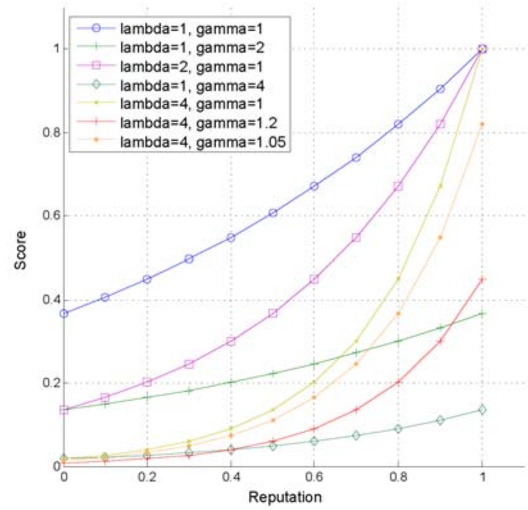

Fig. 3. Impact of λ and γ on the $Score$ function

In the remainder of the paper the relative weights ω_c and ω_q will denote βe^{λ} and αe^{λ} respectively.

We denote by R_0 the ceil under which a reputation value is considered unsatisfactory, i.e. the ceil under which $Score(s)$ is much less sensitive to variations in R. R_0 should be the point at which the score function's growth rate becomes faster, i.e. graphically speaking, the point at which the tangent angle gets sharper. By choosing R_0, and the tangent angle $\theta \in [0, \frac{\pi}{4}]$ that we consider sharp enough, we can derive λ. In fact:

$$\theta = \arctan\left(Score_R(R_0)\right)$$

which leads to λ verifying the following equation:

$$\lambda \exp\left(\lambda(R_0 - 1)\right) - \tan(\theta) = 0$$

It is worth noting that the choice of R_0 should depend on the service nature. In fact, subscribers to expensive services have higher expectation with respect to service quality; R_0 should be higher for costly services than for cheap ones.

V. Reputation system

Commonly the reputation of a service is computed on the basis of feedbacks left by the consumers of that service. Feedbacks may be subjective or malicious, thus, the value of reputation reports can not be granted unless the objectivity and trustworthiness of consumers is ensured. Generally, it is harder to maintain a per-consumer reputation system compared to a per-service reputation system mainly because services are less versatile, more traceable and come in smaller number. Moreover, it is harder to manage user identities especially for malicious users who are likely to change theirs quite often (e.g. sybil attacks [3]).

The automation of the rating process aims at freeing users from the burdening rating task while ensuring the objectivity of feedbacks. In order to automate the service rating process

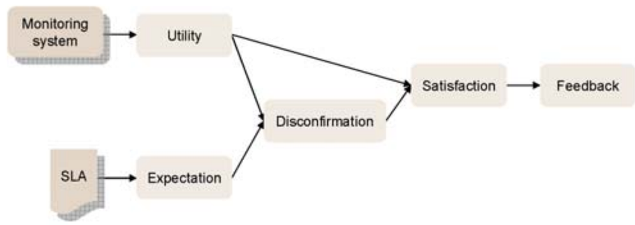

Fig. 4. Feedback forecasting model

we elaborate a feedback forecasting model based on customer satisfaction. This model relies on the measurement of service utility in such a way that any quality monitoring system can be enhanced with a rating function.

For the sake of simplicity, we assume that monitoring is achieved by a trusted system that produces credible QoS reports. According to [4] quality monitoring can either be supported by companies that manage service directories and are eager to control their services' quality, or by third party companies hired to achieve QoS monitoring tasks for them. The trustworthiness of the monitoring system when enhanced with a ranking function ensures the credibility of feedbacks as well.

A. Feedback vs. Satisfaction

We base our model (see figure 4) on the expectancy- disconfirmation theory. This theory relates expectation, perceived quality and quality disconfirmation, i.e. the difference between the expected and perceived quality, to human being customer satisfaction. From this theory and from studies in the field of psychology and marketing, user satisfaction forecasting models, like [5], have been derived. Inspired from the latter work, we assimilate in our model customer expectation to the contracted service quality level (agreed QoS) and the perceived quality to the measured one. A negative disconfirmation between the agreed QoS and the measured one will lead to the dissatisfaction of the user and hence to a negative feedback; the feedback reflects user's satisfaction with the service.

In our model, we relate quality disconfirmation to the feedback that the user will leave after experiencing the service. In a service chain, where a user is typically a piece of software, the rating process must be achieved without human intervention. An automatic conversion of quality disconfirmation into feedback is hence necessary. As opposed to the user satisfaction model in [5] where user's subjectiveness is taken into account in the prediction of its satisfaction with the service, our model must reflect objectively quality disconfirmation in the feedback.

B. Feedback computation

We denote by $U_{QoS}(s)$ the utility function of service s. We represent service quality as a vector of N dimensions, where N represents the number of quality parameters $QoS_{dim} = Q_1, Q_2, ..., Q_N$. We consider service availability as a quality parameter, and, for the sake of objectivity, we do not consider the customer care aspect.

In [6], the utility function is defined as follows:

$$U_{QoS}(s) = \prod_{Q_i \in QoS_{dim}} F_{Q_i}^{c_{Q_i}}$$

where for each QoS parameter Q_i in QoS_{dim}, F_{Q_i} is a function that gives the utility associated to the parameter Q_i, and the weight $c_{Q_i} \in [0, 1]$ reflects how much the user cares about the quality parameter Q_i. c_{Q_i} being user specific, we consider in our model $c_{Q_i} = 1$ for each parameter Q_i for the sake of objectivity. We define the function F_{Q_i} as the probability for a measured value of Q_i to meet the quality requirement.

In [5], the satisfaction $CSAT_i(s)$ of a customer i with service s is defined as follows:

$$CSAT(s) = f1(U_{QoS}(s)) + f2(U_{QoS}(s) - U^e_{QoS}(s)) \quad (5)$$

Where $U_{QoS}(s)$ is the measured utility, $U^e_{QoS}(s)$ the expected utility, $f1$ the perception function which maps the perceived utility to user's "baseline" satisfaction, and $f2$ the disconfirmation function which reflects the subjectivity of user's evaluation given its expectation as reference point.

According to [5], customer expectation evolves over time based on experienced disconfirmation; positive disconfirmation increases future expectation while negative disconfirmation has the opposite effect. For the sake of objectiveness, we consider $U^e_{QoS}(s)$ as constant; the customer expects the agreement to be respected, hence $U^e_{QoS}(s) = 1$.

By mapping $FEEDBACK(s)$ to $CSAT(s)$ in the equation 5, we obtain:

$$FEEDBACK(s) = f(U_{QoS}(s))$$

f is an increasing function defined in $[0, 1]$ and is bounded between $f(0) = 0$ and $f(1) = 1$. According to [5], *as the utility increases, the customer becomes less sensitive to changes in utility*. We believe that users are not much more sensitive to changes for very low utility values than they are for high utility values. In our model, we consider that f varies slightly for utility values nearby 0 as well as for utility values nearby 1. Hence the concavity of f changes at a particular utility value, say U_0 in $[0, 1]$. Like in [5], and with respect to the above requirements, we adopt a polynomial rate of change for the perception function f as follows:

$$f''(x) = \mu(U_0 - x) \quad \text{with} \quad \mu \geq 0$$

As detailed in [7], f can be defined as follows:

$$f(x) = -\frac{\mu}{6}x^3 + \frac{\mu U_0}{2}x^2 + (1 + \mu(\frac{1}{6} - \frac{U_0}{2}))x \quad (6)$$

As shown in figure 5, U_0 and μ define the shape and concavity of the perception function f respectively, the concavity of the perception function is the highest (i.e. the most pronounced), for values of μ verifying:

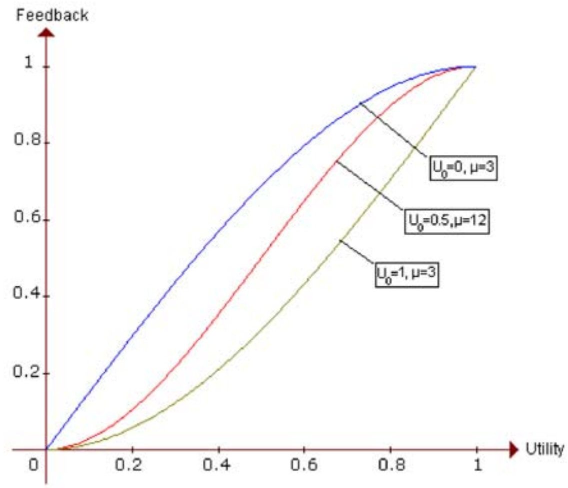

Fig. 5. The perception function

$$\begin{cases} \mu = \frac{6}{2-3U_0} & for \quad U_0 \in [0, \frac{1}{2}] \\ \mu = \frac{6}{3U_0-1} & for \quad U_0 \in [\frac{1}{2}, 1] \end{cases} \tag{7}$$

We will try in the following paragraph to establish a relationship between μ, U_0 and the cost (price) of the service.

In our model, we would like to penalize expensive services that deliver low utility levels and reward cheap services with high utility levels. Hence, in one hand, for a particular service cost, we provide higher feedback for high-utility services, and on the other hand, for a particular utility value, we provide higher feedback for low-cost services. In addition, we believe that cost has an impact on user quality expectation; the higher the cost of the service, the more the costumer is sensitive to quality disconfirmation. In fact the costumer gets easily dissatisfied with the contracted service, when its cost is relatively high, as soon as the utility drops below the expected value. Since the feedback reflects costumer satisfaction/dissatisfaction, the concavity of the feedback curve is consequently mostly positive for high cost values. On the other hand, the feedback curve's concavity is mostly negative for low cost values. In other words, when the cost increases, U_0 tends to 1 and when cost decreases, U_0 tends to 0.

Let c denote the cost of the service and υ the function that relates service cost to the value U_0. υ is a positive increasing function that takes value in $[0,1]$. Let c_{min} be the lowest available service cost and c_{max} the highest. In the following, and for the sake of simplicity, we define υ as follows:

$$\upsilon : \begin{cases} Dom(c) \longrightarrow [0,1] \\ c \longmapsto \upsilon(c) = \begin{cases} \frac{1}{2} & \text{if} \quad c_{max} - c_{min} = 0 \\ \frac{c - c_{min}}{c_{max} - c_{min}} & \text{else} \end{cases} \end{cases}$$

This way, $U_0 = 1$ for $c = c_{max}$, and $U_0 = 0$ for $c = c_{min}$.

C. Reputation computation model

By definition, reputation helps predicting the reliability and credibility of the service and its provider at time t, time at which the agreement is being conducted. Let R_s be the reputation function of service s,

$$R_s : \begin{cases} Timestamp \longrightarrow [0,1] \\ t \longmapsto R_s(t) \end{cases}$$

We compute R_s based on users' past feedbacks. These feedbacks reflect service's past behavior and may give an indication on its future behavior; feedbacks may be randomly distributed when service's behavior is not deterministic, they may follow a trend, e.g. increasing feedbacks may reflect an improvement in service quality, they may even be cyclic when there is a periodicity in service's behavior, e.g. quality may decrease in rush hours which leads to low feedbacks at those times.

When the feedback series does not show any trend, it is very hard to predict service's future behavior. However, when feedbacks exhibit a trend, this latter should be taken into account by the reputation function as it helps predicting future behavior. Smoothing-based forecasting techniques, like *Moving Average*, *Weighted Moving Average*, and *Exponential Smoothing* [8] can be used to predict near future behavior from past behaviors. Other more specific techniques like the Holt's Linear Exponential and the Holt-Winters' Forecasting are more suitable for long-term forecasting over data showing a trend and periodicity respectively.

VI. Simulation and Evaluation

We hereby consider a set of six instances of service S with six different quality and cost offers. We generate for each service a utility function, where utility stands for the measure of the conformance of the delivered service quality to the agreement. The utility of each service instance changes over time; it either increases, decreases or fluctuates around a utility value. Changes in utility series are generated randomly every M time units. Utility at time t represents the utility that a user has experienced before leaving a feedback at time t. We use our rating model [7] to derive feedbacks from utility values. We use the *simple exponential smoothing* (SES) forecasting technique to compute service reputation.

We assume that: (1) a user initiates a request for S each time unit, (2) all service instances satisfy user's request and quality and cost constraints, and (3) a feedback is left for every service instance at the end of each time unit.

At each time unit we hence look into each service's score; the selected service is the one that has the highest score. When selecting service S_i at time t, the user experiences a utility value that corresponds to the utility of S_i at time $t+1$. At $t+1$ the user leaves the exact feedback that we previously derived from the utility of S_i at time $t+1$, and starts another selection process.

Table I lists our simulation parameters.

Parameter	Definition	Value
C_{S_1}	Cost of S_1	90
C_{S_2}	Cost of S_2	70
C_{S_3}	Cost of S_3	60
C_{S_4}	Cost of S_4	40
C_{S_5}	Cost of S_5	30
C_{S_6}	Cost of S_6	10
Q_{S_1}	Quality of S_1	512
Q_{S_2}	Quality of S_2	512
Q_{S_3}	Quality of S_3	256
Q_{S_4}	Quality of S_4	256
Q_{S_5}	Quality of S_5	128
Q_{S_6}	Quality of S_6	128
α	SES smoothing parameter	0.8
λ	Score function convexity parameter	2.6
w_q	Relative quality weight	0.4
w_c	Relative cost weight	0.6

TABLE I
SIMULATION PARAMETERS

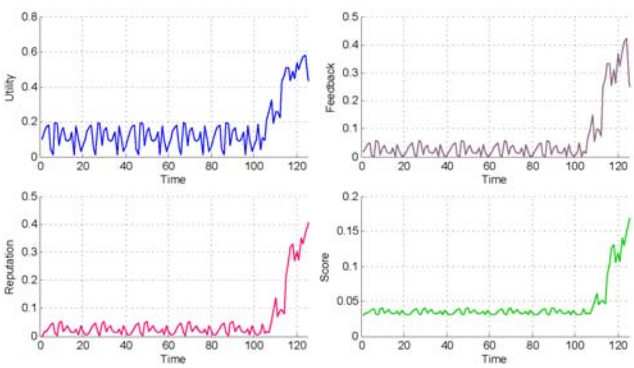

Fig. 6. Utility, feedback, reputation and score evolution of S1

Figure 6 shows the utility feedback, reputation and score functions of S_1. It shows how closely feedbacks, reputations and scores follow variations in utility values.

Initially, at $t = 0$, there is no feedback in the reputation system. The user selects the service instance that maximizes the weighted quality and cost parameters; here S_6. However, by choosing S_6, the user perceives a low service utility compared to what she/he would have experienced with S_1 or S_3. As a matter of fact, the user leaves a low feedback which shows user's dissatisfaction with the service. As soon as the first feedbacks are introduced into the reputation system, the user ends up making the best choices most of the time as shown in Figure 7. These experiments have also shown that the user ends up selecting the service instance that delivers the highest utility almost 83% of the time and that the user would not have been more satisfied with a service instance other than with her/his choice more than 87% of the time.

Simulation results show that the overall system evolves well over time. Hence, our system has succeeded in capturing

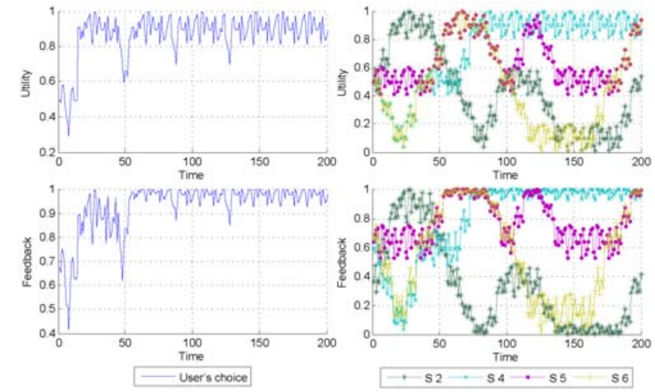

Fig. 7. User's choice vs. service offers: experienced utility and resulting feedback

service behavior and providing best possible choices.

VII. RELATED WORKS

Service selection is a wide research topic that emerged lately with the advent of SOA. Few research works have addressed so far this topic, each from a different perspective, including QoS and trust assessment, reputation management, and QoS and trust ontologies.

Works like [2], [9] and [10] have focused on QoS awareness and more precisely on service composition with end-to-end QoS constraints. [2] considers two models for service selection. The first model promotes the selection of those services that present the optimal service-level quality offers. We have adapted this latter model to our service-level quality-constrained selection problem. As opposed to this first model which does not however provide an optimal solution for the overall composite service but rather locally optimal solutions, the second model promotes service selection with global planning. This latter selection model uses the Integer Linear Programming (ILP) method to find optimal solutions. Considering the high complexity of the ILP method, [9] proposes heuristic algorithms to find near-optimal solutions in polynomial time. Similarly, [10] provides a Mixed Integer Linear Programming (MILP)-based formulation of the selection problem and considers a greedy heuristic to find near-optimal solutions.

The above works consider service selection as a multi-criteria constraint satisfaction problem but ignore the fact that services may not deliver the exact promised quality. Although the reputation factor is considered in [2], it is defined as a quality parameter and used as any quality parameter in the ranking function.

Other works like [11] and [12] promote the sharing of reports on the experienced service quality among consumers to help them predict the trustworthiness of services at selection time. [11] mainly considers the assessment of the credibility of quality reports. Few trusted parties are assumed to provide credible reports on the conformance of the delivered quality to the quality offer. These latter reports are used to evaluate the credibility of other reports. False reports are then detected and ignored in the selection process. The future conformance of the delivered quality to the offer is predicted using a linear regression method on past QoS conformance reports, each weighted by its evaluated credibility. Similarly, [12] addresses the issue of predicting the capability of a service to deliver the level of quality that would meet user's requirements. In this perspective, a Bayesian network-based QoS assessment model is devised along with a fuzzy logic-based reasoning approach for inferring service capability under various combinations of users QoS requirements. More precisely, the performance of the service is tracked and recorded while the service is being executed and the compliance between the users QoS requirements and the services delivered QoS is computed. Based on these compliance values, a fuzzy reasoning approach is introduced to infer the services overall capability and the corresponding Bayesian network is updated with the assessment outcome.

False quality reports may indeed jeopardize the selection process. The report evaluation system in [11] raises however a critical issue. In fact, user reports can not be objectively and fairly compared to reports made by trusted monitoring agents unless both monitoring results are achieved under the same circumstances. If trusted monitoring agents can be deployed within the same context as users then user reports become useless. We propose a solution that have similarities with the above works in that we do promote the sharing of consumers feedbacks on the experienced QoS where feedbacks are generated by an automated rating engine.

[13], [14] and [15] promote the use of trust and reputation to rate candidate services in the selection process. [13] uses a multidimensional trust model to evaluate service and service provider's properties like credibility, quality and reliability. A trustworthy vector describes users' experience with those service aspects. The trustworthiness of the service, referred to as the confidence in the service is estimated using the Hypothesis Evaluation theory. [14] considers reputation-based selection function for atomic and composite services. The paper introduces the concept of execution context: service is evaluated with regards to a specific execution context (application domain or user type). Reputation is a weighted mean of time-decaying feedbacks within the desired context. A time-decaying factor, called "forgetting" factor is also used in [15] along with a bayesian estimation model to compute the reputation of a service on the basis of past users' testimonials. In our work however, we consider that the reputation system should take into account the evolution trend in users' feedbacks.

VIII. CONCLUSION

We have presented in this paper an automated QoS- and reputation-based framework for service selection. We have designed a selection algorithm to help users choose the most appropriate service among equivalent services functionality-wise. This algorithm predicts the suitability of a given service to user's quality requirements and the conformance of the delivered quality to the initial offer.

We have evaluated our selection algorithm through simulations. The results showed that the system succeeded in capturing service behaviors and in providing users with best available choices.

REFERENCES

[1] C. B. Audun Josang, Roslan Ismail, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, 2007.

[2] L. Zeng, B. Benatallah, A. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "Qos-aware middleware for web services composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, 2004.

[3] J. R. Douceur, "The sybil attack." in *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, 2002.

[4] Y. Wang and J. Vassileva, "A review on trust and reputation for web service selection," in *Proceedings of the 27th International Conference on Distributed Computing Systems Workshops (ICDCSW '07)*, 2007.

[5] J. Xiao and R. Boutaba, "Assessing network service profitability: Modeling from market science perspective," *ACM/IEEE Transactions on Networking, To appear*.

[6] V. Poladian, J. P. Sousa, D. Garlan, and M. Shaw, "Dynamic configuration of resource-aware services," in *Proceedings of the 26th International Conference on Software Engineering (ICSE '04)*, 2004.

[7] N. Limam, "Discovery and selection in internet-scale service infrastructures," Ph.D. dissertation, Universite Pierre et Marie Curri. Paris VI, 2007.

[8] P. Brockwell and R. Davis, *Introduction to Time Series and Forecasting*, ser. Springer Texts in Statistics. New York: Springer-Verlang, 2002.

[9] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for web services selection with end-to-end qos constraints," *ACM Transactions on the Web (TWEB)*, vol. 1, no. 1, 2007.

[10] J. Anselmi, D. Ardagna, and P. Cremonesi, "A qos-based selection approach of autonomic grid services," in *Proceedings of the Workshop on Service-oriented computing performance: aspects, issues, and approaches (SOCP '07)*, 2007.

[11] L.-H. Vu, M. Hauswirth, and K. Aberer, "Qos-based service selection and ranking with trust and reputation management," in *Proceedings of the 13th Conference on Cooperative Information Systems (CoopIS '05)*, 2005.

[12] G. Wu, J. Wei, X. Qiao, and L. Li, "A bayesian network based qos assessment model for web services," in *Proceedings of the IEEE International Conference on Services Computing (SCC '07)*, 2007.

[13] S. J. H. Yang, J. S. F. Hsieh, B. C. W. Lan, and J.-Y. Chung, "Composition and evaluation of trustworthy web services," in *Proceedings of the IEEE International Workshop on Business services networks (BSN '05)*, 2005.

[14] A. S. Ali, S. Majithia, O. F. Rana, and D. W. Walker, "Reputation-based semantic service discovery," 2004.

[15] R. Wishart, R. Robinson, J. Indulska, and A. Josang, "Superstringrep: reputation-enhanced service discovery," in *Proceedings of the Twenty-eighth Australasian conference on Computer Science (ACSC '05)*, 2005.