

# Managing QoS Acceptability for Service Selection : A Probabilistic Description Logics Based Approach

Salima Benbernou<sup>1</sup>, Allel Hadjali <sup>2</sup>, Naouel Karam<sup>3</sup>, Mourad Ouziri<sup>1</sup>

<sup>1</sup>Université Paris Descartes, Sorbonnes Paris Cité, France  
{mourad.ouziri,salima.benbernou}@parisdescartes.fr

<sup>2</sup> LIAS/ENSMA, Poitiers, France  
allel.hadjali@ensma.fr ,

<sup>3</sup>Department of Mathematics and Computer Science, Freie Universität Berlin,  
Germany

naouel.karam@fu-berlin.de

**Abstract.** Quality of Service (QoS) guarantees are usually regulated in a Service Level Agreement (SLA) between provider and consumer of services. Such guarantees are often violated, it may however be the case where the available services do not match exactly all required QoS, leading the system to grind to a halt. It would be better to look for an approximation for acceptable QoS and avoid the complete stopping of the running services. This paper aims at making dynamic QoS acceptability easy for service selection. The proposed model is based on an extension of existing probabilistic description logics reacting to QoS variations. The contributions made are twofold (1) a query description language to express the required QoS by means of a probabilistic description logic (2) a reasoning algorithm for decision making about the acceptability of QoS w.r.t the probabilistic description.

## 1 Introduction

In a Service Oriented Architecture, service providers need to characterize their services defining both the offered functionalities and the offered quality. Then, the quality of service (QoS) properties can be critical elements for achieving the business goals of service providers. Establishing QoS contracts, described in a SLA, that can be monitored at runtime, is therefore of paramount importance. The SLA must stipulate the values of QoS attributes that a service provider is expected to deliver to a client.

Moreover, a successful execution of a service composition implies continuous monitoring of QoS at runtime. However, at this step, variations could occur in the QoS and most likely induce violations of the agreement. For that, the service composition should support a dynamic QoS-driven adaptation. A static adaptation is not adequate due to variations of the QoS. Existing approaches deal with static QoS, they do not address the attribute variations issue. A *formal and declarative approach* to achieve a dynamic adaptation is then highly desired.

In this paper, we propose a probabilistic description logic based approach to describe the QoS attributes and handle their variations through the use of linguistic concepts. The proposed approach helps to select services in order to build approximate compositions which may not satisfy all the requirements, hence avoiding to completely stop running the processes. It provides the basis allowing the representation and reasoning about the introduced symbolic concepts, where the variation of QoS is modeled in a probabilistic way, hence managing the QoS acceptability. Our main contributions are summarized in the following:

1. We introduce a query description language to express the required QoS using both linguistic concepts and probabilistic representations. To this end, we extend existing probabilistic description logics to handle QoS variations.
2. We develop a suitable reasoning algorithm for computing the QoS acceptability of the selected service w.r.t the probabilistic description. The algorithm achieves an approximate reasoning using inference rules involving probabilistic statements.

The remainder of the paper is structured as follows. In section 2, we provide first, a discussion about existing works dealing with QoS and, second a review of the approaches related to probabilistic description logics. Section 3 describes our proposal to manage the QoS acceptability. In section 4, we present the syntax and semantics of the proposed language. Section 5 details the reasoning algorithm and section 6 concludes the paper.

## 2 Related Work

Most of existing approaches focused on the contract definition and on mechanisms for contract enactment. [1] describes a matchmaking algorithm for ranking functionally equivalent services. In [8], a fuzzy service adaptation approach that leverages the degrees of QoS satisfaction, is discussed. [11] proposes a soft constraint-based framework to seamlessly express QoS properties reflecting both customer preferences and penalties applied to unfitting situations. The work in [2] discusses a constraint based approach for quality assurance in a choreography system. In [10], a new trend called service skyline computation for handling multiple quality criteria including user preferences, is proposed.

On the other hand, semantic technologies have been applied in recent works for reasoning about QoS of different systems during service composition. [7] proposes a meta-model for non functional property descriptions targeted to support the selection of Web services. The contribution in [3] focuses on the analysis of the requirements for a semantically rich QoS-based Web Service Description Model and an accurate, effective QoS-based WS Discovery (WSDi) process. [9] presents an overview of prominent research works related to developing QoS ontologies.

It is worthy to note that the above approaches do not handle the variations of QoS at runtime. An effective service composition management requires a support for the acceptance of QoS values, and should also allow for a flexible

acceptability for autonomous corrective actions. We provide here a probabilistic description logic based approach allowing self-healing in reaction to QoS variations by expressing their semantics.

In the literature, a number of works on probabilistic description logics exists [6, 5, 4]. In cite[4], the authors proposed a probabilistic extension of the expressive description logics P-SHIF(D) and P-SHOIN (D) that encompasses both statistical and subjective features, and also addresses the non-monotonic aspects of probabilistic knowledge using a semantics based on Lehmanns lexicographic entailment. In the cited approaches, the terminological and assertional knowledge is extended with conditional constraints and referred to as probabilistic knowledge. Those constraints express probabilities relating concepts or individual assertions. Our approach describes probabilities in a higher level of abstraction which allows to define probabilistic concepts that can be used in complex descriptions. We provide a reasoning algorithm dealing with such concepts.

### 3 Dynamic QoS Acceptability Framework

As depicted in Figure 1, our work can be integrated into a service composition system as a real-time QoS acceptability process. The inputs are the QoS required by the service selection process and the events captured by the monitoring process. The output is a set of acceptable services regarding their effective QoS. Events dealing with real-time execution of services are captured by the mon-

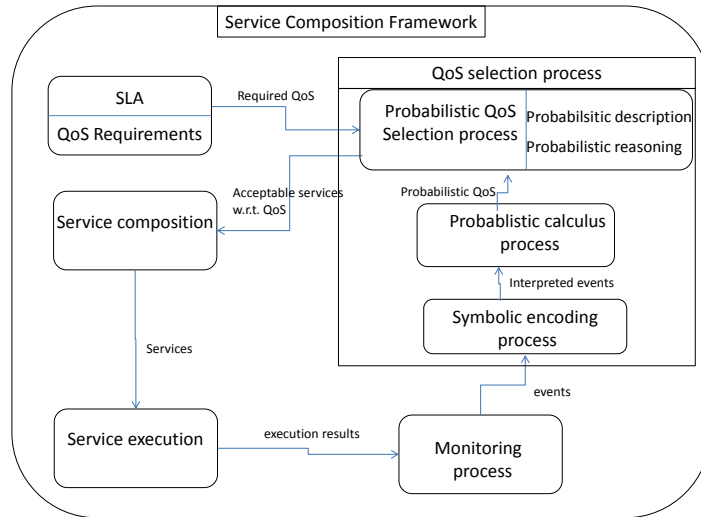


Fig. 1. QoS variations framework for service selection

itoring process. From these events, we extract real-time measures of the QoS provided by services. To efficiently handle such measures for the purpose of decision making, they are expressed thanks to symbolic concepts while taking into account their variations in a probabilistic way. We illustrate this idea through the following example.

Let us consider two services  $S_1$  and  $S_2$  involved in a web services composition. Assume that the monitoring process has captured the events given in Table 1.

Event	Execution time	Service ID	Response time (TR)	RAM Consuming (RC)
1	10:15:45	$S_1$	12ms	3MB
2	10:55:05	$S_1$	3ms	11MB
3	21:30:00	$S_1$	2ms	4MB
4	10:15:45	$S_1$	4ms	13MB
5	21:35:55	$S_2$	1ms	80MB
6	23:10:09	$S_2$	2ms	75MB

**Table 1.** Example of events captured by the monitoring process

### 3.1 Symbolic encoding of the events

Each quantitative measure captured by the monitoring process is interpreted by symbolic value using business rules. Business rules may be provided by the application domain experts in a Service Level Agreement (SLA). In what follow are examples of business rules:

- Rule 1 (for the interpretation of Time Response measures): A response time is said to be good (GoodTR) if it is less than 3ms, medium (MediumTR) if it is between 4ms and 6ms and bad (BadTR) otherwise.
- Rule 2 (for the interpretation of RAM consuming measures): A RAM consuming is said to be good (GoodRC) if it is less than 5Mb, bad (BadRC) otherwise.

Given the above two rules, the measures of Time Response and RAM are shown in Table 1 can be expressed in the following qualitative/symbolic descriptions:

- Event 1:  $S_1$  : BadTR,  $S_1$  : GoodRC.
- Event 2:  $S_1$  : GoodTR,  $S_1$  : BadRC.
- Event 3:  $S_1$  : GoodTR,  $S_1$  : GoodRC.
- Event 4:  $S_1$  : MiddleTR,  $S_1$  : BadRC.
- Event 5:  $S_2$  : GoodTR,  $S_2$  : BadRC.
- Event 6:  $S_2$  : GoodTR,  $S_2$  : BadRC.

### 3.2 Probabilistic QoS

The real-time QoS of a given service is not static but varies over different executions of the service. The QoS of each service has to be represented by a unique description in terms of a probability distribution <sup>1</sup>. Hence, the multiple values about a symbolic QoS are described using the Bayesian probability defined as follows:

$$\begin{aligned} P(\text{SymbQoS}|S_i) &= \frac{P(\text{SymbQoS} \cap S_i)}{P(S_i)} = \frac{|\text{SymbQoS} \cap S_i|}{|S_i|} \\ &= \frac{\text{number of appearances of SymbQoS in the events of } S_i}{\text{Number of executions of } S_i} \end{aligned}$$

where

$\text{SymbQoS} \in \{\text{BadTR}, \text{GoodTR}, \text{MiddleTR}, \text{BadRC}, \text{GoodRC}, \text{MiddleRC}\}$ .

Back to the example introduced in section 3.1, the services  $S_1$  and  $S_2$  have been executed four times and twice, respectively. Hence, the probability values associated with their QoS can be computed as follows:

- $S_1$  received once BadTR. So,  $S_1$  is BadTR with a probability of 0.25.
- $S_1$  received three times GoodTR. So,  $S_1$  is GoodTR with a probability of 0.75.
- $S_1$  received twice GoodRC. So,  $S_1$  is GoodRC with a probability of 0.5.
- $S_1$  received twice BadRC. So,  $S_1$  is BadRC with a probability of 0.5.
- $S_2$  received twice GoodTR. So,  $S_2$  is GoodTR with a probability of 1.
- $S_2$  received twice BadRC. So,  $S_2$  is BadRC with a probability of 1.

### 3.3 Probabilistic QoS selection process

Given a required QoS, a traditional selection process decides whether a known service is acceptable or not. In our case, the selection process operates in an uncertain (probabilistic) environment since each QoS is described thanks to a probability distribution.

To this end, we propose a formal model based on probabilistic DLs to process acceptability of QoS. In this model, the required and provided QoS are described using a formal description language. A reasoning algorithm to decide the acceptability, is proposed as well.

As example for a required QoS, let's consider the following query: Retrieve the services that should provide a Good Response Time with a probability greater than 0.7 and Bad RAM Consuming with a probability less than 0.55. Intuitively, Only service  $S_1$  is acceptable with respect to this required QoS, the Service  $S_2$  is not acceptable because it does not provide Bad RAM consuming with probability less than 0.55.

<sup>1</sup> The interpretation of probability considered here is the so-called "the a-priori interpretation" which is the oldest and simplest one. The probability of an event is the number of favorable cases, where this event occurs, divided by the total numbers of possibles cases. So, w.r.t the same service and the same attribute, the required axiom that probabilities add up to 1 holds

In what follows, we present the proposed formal model for QoS acceptability decision making.

## 4 Probabilistic DL for Handling QoS Acceptability

The model relies on a probabilistic extension of DL defined with the following components:

- An expression language, including a set of symbols that are used to express knowledge about QoS.
- A semantic of symbols of the expression language, based on an interpretation domain and an interpretation function.
- A reasoning algorithm based on a set of inference rules to make a decision about the QoS acceptability.

### Syntax of the description language

The proposed syntax is summarized in Table 2 (with  $\circ \in \{<, >, \leq, \geq\}$ ). The introduced description language allows expressing two types of information related to QoS:

1. The knowledge about dynamic QoS, and
2. The information related to the required QoS.

Syntax	Interpretation
$Q_{\circ p}$	Refers to services that provide the QoS $Q$ with a probability $\circ p$ ( $Q_{\circ 1}$ is noted $Q$ )
$Q1_{\circ p} \sqcap Q2_{\circ q}$	Refers to services that provide the QoS $Q1$ and $Q2$ with probability, respectively, $\circ p$ and $\circ q$
$Q1_{\circ p} \sqcup Q2_{\circ q}$	Refers to services that provide the QoS $Q1$ or $Q2$ with probability, $\circ p$ and $\circ q$ , respectively
$\neg Q_{\circ p}$	Refers to services that do not have the QoS $Q$ with a probability $\circ p$
$Q1 \sqsubseteq Q2$	Means: Any service that provides the QoS $Q1$ , then it provides automatically the QoS $Q2$ .

**Table 2.** Syntax of the proposed description language

For instance, the dynamic QoS of the example provided in the previous section can be written as follows:

$$BadTR_{0.25}(S_1), GoodTR_{0.75}(S_1), GoodRC_{0.5}(S_1), BadRC_{0.5}(S_1),$$

$$GoodTR_1(S_2), BadRC_1(S_2).$$

Similarly, the description of the required QoS can also be expressed using this language. For example, the following formulas are two different descriptions of possible required QoS:

$$GoodTR_{\geq 0.7} \sqcap GoodRC_{\geq 0.4}$$

$$\neg BadTR_{\geq 0.1} \sqcap GoodRC_{\geq 0.9}$$

In the first, we require a good time response with a minimum rate of 70% and good RAM Consuming with the minimum rate of 40%. While in the second one, we require services that provide good RAM Consuming with probability greater or equal than 90% and, at the same time, do not provide bad time response with probability greater than 10%.

### Formal semantics of the description language

The formal semantics of terms and constructors of the description language is defined by the pair  $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$  where:

- $\Delta^{\mathcal{I}}$  is an interpretation domain. It is composed of individuals that represent the services managed in the BPM (Business Process Management) system.
- $\cdot^{\mathcal{I}}$  is an interpretation function that assigns terms  $Q$  of the language to individuals of  $\Delta^{\mathcal{I}}$  as shown in Table 4 (where  $\bar{op}$  stands for the complementary of  $op$ , for instance  $\bar{\geq}p = \leq p$ ):

Syntax	Formal interpretation
$Q_{op}$	$Q_{op}^{\mathcal{I}} = \{s \in \Delta^{\mathcal{I}} \mid Q^{\mathcal{I}}(s) \circ p\}$
$Q1_{op} \sqcap Q2_{oq}$	$Q1_{op}^{\mathcal{I}} \sqcap Q2_{oq}^{\mathcal{I}} = \{s \in \Delta^{\mathcal{I}} \mid Q1^{\mathcal{I}}(s) \circ p \wedge Q2^{\mathcal{I}}(s) \circ q\}$
$Q1_{op} \sqcup Q2_{oq}$	$Q1_{op}^{\mathcal{I}} \sqcup Q2_{oq}^{\mathcal{I}} = \{s \in \Delta^{\mathcal{I}} \mid Q1^{\mathcal{I}}(s) \circ p \vee Q2^{\mathcal{I}}(s) \circ q\}$
$\neg Q_{op}$	$(\neg Q)_{op}^{\mathcal{I}} = Q_{\bar{op}}^{\mathcal{I}} = \{s \in \Delta^{\mathcal{I}} \mid Q^{\mathcal{I}}(s) \bar{\circ} p\}$
$Q1 \sqsubseteq Q2$	$\forall s \in \Delta^{\mathcal{I}}, Q1^{\mathcal{I}}(s) \Rightarrow Q2^{\mathcal{I}}(s)$

**Table 3.** Formal semantics of the proposed probabilistic DL

For instance, the probabilistic concept  $Q_{\geq 0.5}$  is interpreted as the set of individuals with a probability degree greater than 0.5. Interpretation of complex and composed descriptions is also allowed as stated in Table 3.

## 5 Reasoning Algorithm for the Dynamic QoS Acceptability Decision

Based on the syntax and semantics of the description language presented in the previous section, we develop a reasoning algorithm that decides whether a provided dynamic QoS is acceptable with respect to a required one.

The decision problem is formulated as follows:

”Is the dynamic QoS provided by the service  $S_i$  acceptable with respect to a given required QoS  $RQ$  ?”

This acceptability decision problem can be formalized by the following logical deduction:

$$\langle (TBox, ABox) \rangle \models RQ(S_i)$$

where  $(TBox, ABox)$  is the knowledge base with two components:

- A TBox: containing knowledges about the considered domain application. For example, it may contain the following axioms:

$$\{GoodTR_1 \sqsubseteq BadTR_0, BadRC_1 \sqsubseteq GoodRC_0\}$$

- An ABox: containing knowledges about dynamic QoS provided by services. For example, the ABox may contain the previous QoS of services  $S_1$  and  $S_2$ :

$$\{BadTR_{0.25}(S_1), GoodTR_{0.75}(S_1), GoodRC_{0.5}(S_1), BadRC_{0.5}(S_1), GoodTR_1(S_2), BadRC_1(S_2)\}$$

The logical deduction is achieved by checking the inconsistency of the following knowledge base:

$$\langle TBox, ABox \cup \{\neg RQ(S_i)\} \rangle$$

The inference machinery calls on the propagation rules introduced in the following sub-section.

### 5.1 Algorithm for QoS acceptability

A sample of the propagation rules that constitute the foundations of our probabilistic DL is depicted in Figure 2. Those rules are defined with respect to the formal semantics given in section 4. For sake of clarity, the *R<sub>reduction</sub>* rules are given for  $\leq$  and  $\geq$ , one can easily deduce the corresponding rules for  $<$  and  $>$ . The algorithm checks the consistency of the knowledge base:

$$\langle TBox, ABox \cup \neg RQ(S_i) \rangle$$

by applying the propagation rules until termination. Each application of a propagation rule generates a new inference system  $IS_i$ . The algorithm terminates if:

1. There exists a clash in the current inference system. In this case, the knowledge base is not consistent which means that the checked QoS is acceptable.
2. No more rule can be applied to generate a new inference system. In this case, the knowledge base is consistent, which means that the checked QoS is not acceptable.

An inference system  $IS_i$  contains clash if:

$$IS_i = \{Q_{<p}(s), Q_{>p'}(s), p \leq p'\}, \text{ or}$$

$$IS_i = \{Q_{\leq p}(s), Q_{\geq p'}(s), p < p'\}, \text{ or}$$

$$IS_i = \{Q_{<p}(s), Q_{\geq p'}(s), p < p'\}, \text{ or}$$

$$IS_i = \{Q_{\leq p}(s), Q_{>p'}(s), p < p'\}, \text{ or}$$

$$IS_i = \{Q_{=p}(s), Q_{=p'}(s), p \neq p'\}$$

The algorithm is sound because for any satisfiable concept of the proposed description logic, the application of propagation rules given in Fig. 2 terminates on at least one state free of clash. By the same way, starting from an inconsistent concept, all terminal states of the algorithm after applying propagation rules contain clash.



<p>Rule <math>R_1</math> :</p> $IS_i = \{Q(s)\} \longrightarrow IS_{i+1} = IS_i \cup \{Q_{\geq 1}(s)\}$ <p>Rule <math>R_0</math> :</p> $IS_i = \{\neg Q(s)\} \longrightarrow IS_{i+1} = IS_i \cup \{Q_{\leq 0}(s)\}$ <p>Rule <math>R_{\neg}</math> :</p> $IS_i = \{\neg Q_{op}(s)\} \longrightarrow S_{i+1} = IS_i \cup \{Q_{\bar{o}p}(s)\}$ (Note that: $\bar{<} = \geq$ , $\bar{\leq} = >$ , $\bar{>} = \leq$ , $\bar{=} = <$ ) <p>Rule <math>R_{\cap}</math> :</p> $IS_i = \{(Q1_{op} \cap Q2_{oq})(s)\} \longrightarrow IS_{i+1} = IS_i \cup \{Q1_{op}(s), Q2_{oq}(s)\}$ <p>Rule <math>R_{\sqcup}</math> :</p> $IS_i = \{(Q1_{op} \sqcup Q2_{oq})(s)\} \longrightarrow IS'_{i+1} = IS_i \cup \{Q1_{op}(s)\}, IS''_{i+1} = IS_i \cup \{Q2_{oq}(s)\}$ <p>Rule <math>R_{\sqsubseteq}</math> :</p> $IS_i = \{(Q1_{op} \sqsubseteq Q2_{oq}), Q1_{op}(s)\} \longrightarrow IS_{i+1} = IS_i \cup \{Q2_{oq}(s)\}$ <p>Rule <math>R'_{\sqsubseteq}</math> :</p> $IS_i = \{(Q1_{\geq p} \sqsubseteq Q2_{oq}), Q1_{=r}(s), p \leq r\} \longrightarrow IS_{i+1} = IS_i \cup \{Q2_{oq}(s)\}$ <p>Rule <math>R_{reduction}</math> :</p> $IS_i = \{Q_{\geq p}(s), Q_{\leq p}(s)\} \longrightarrow IS_{i+1} = IS_i - \{Q_{\geq p}(s), Q_{\leq p}(s)\} + \{Q_{=p}(s)\}$ $IS_i = \{Q_{\geq p}(s), Q_{\geq q}(s), p \geq q\} \longrightarrow IS_{i+1} = IS_i - \{Q_{\geq q}(s)\}$ $IS_i = \{Q_{\geq p}(s), Q_{=q}(s), q \geq p\} \longrightarrow IS_{i+1} = IS_i - \{Q_{\geq p}(s)\}$ $IS_i = \{Q_{\leq p}(s), Q_{\leq q}(s), p \leq q\} \longrightarrow IS_{i+1} = IS_i - \{Q_{\leq q}(s)\}$ $IS_i = \{Q_{\leq p}(s), Q_{=q}(s), p \geq q\} \longrightarrow IS_{i+1} = IS_i - \{Q_{\leq p}(s)\}$
--

**Fig. 2.** A sample of the propagation rules

## 5.2 An illustrative example

We illustrate the use of the proposed algorithm to decide whether the dynamic QoS of services  $S_1$  and  $S_2$  is acceptable with respect to the following required QoS:

$$RQ = (GoodRC_{\geq 0.4} \cap \neg BadTR_{\geq 0.8}) \sqcup GoodRC_{\geq 0.9}.$$

**Acceptance of the QoS of  $S_2$**  The QoS of the service  $S_2$  is acceptable w.r.t  $RQ$  if:

$$\langle TBox_{exple}, ABox_{exple} \rangle \models RQ(S_2).$$

That is,  $\langle TBox_{exple}, ABox_{exple} \cup \{\neg RQ(S_2)\} \rangle$  is inconsistent

where,

$$TBox_{exple} = \{BadRC_1 \sqsubseteq GoodRC_0\}$$

$$ABox_{exple} = \{GoodTR_1(S_2), BadRC_1(S_2)\}$$

The reasoning algorithm generates inference systems  $IS_i$  by applying the propagation rules as follows:

$$\begin{aligned}
IS_0 &= \{TBox_{exple}, ABox_{exple}, \neg RQ(S_2)\} \\
&= \{BadRC_1 \sqsubseteq GoodRC_0, GoodTR_1(S_2), BadRC_1(S_2), \neg[(GoodRC_{\geq 0.4} \sqcap \\
&\neg BadTR_{\geq 0.8}) \sqcup GoodRC_{\geq 0.9}](S_2)\} \\
IS_1 &= IS_0 \cup \{GoodRC_0(S_2), [\neg(GoodRC_{\geq 0.4} \sqcap \neg BadTR_{\geq 0.8}) \sqcap \neg GoodRC_{\geq 0.9}](S_2)\} \\
&\text{(generated by } R_{\sqsubseteq} \text{ and pushing the negation)} \\
IS_2 &= IS_1 \cup \{\neg(GoodRC_{\geq 0.4} \sqcap \neg BadTR_{\geq 0.8})(S_2), \\
&\neg GoodRC_{\geq 0.9}(S_2)\} \text{ (generated by } R_{\sqcap}) \\
IS_3 &= IS_2 \cup \{(GoodRC_{< 0.4} \sqcup BadTR_{\geq 0.8})(S_2), GoodRC_{< 0.9}(S_2)\} \text{ (generated} \\
&\text{by } R_{\neg}) \\
IS'_4 &= IS_2 \cup \{(GoodRC_{< 0.4}(S_2), GoodRC_{< 0.9}(S_2)\} \text{ or } IS''_4 = IS_2 \cup \{(BadTR_{\geq 0.8})(S_2), \\
&GoodRC_{< 0.9}(S_2)\} \text{ ( } IS'_4, IS''_4 \text{ generated by } R_{\sqcup}) \\
IS'_5 &= IS'_4 \cup \{(GoodRC_{< 0.4}(S_2)\} \text{ (generated by } R_{reduction} \text{ on } IS'_4) \\
IS'_6 &= IS'_5 \cup \{(GoodRC_0(S_2)\} \text{ (generated by } R_{reduction} \text{ with } GoodRC_0(S_2)) \\
&\text{The algorithm stops at the inference system } IS'_6 \text{ because no more rule can} \\
&\text{be applied.}
\end{aligned}$$

As there is no clash in  $IS'_6$ , the system is not inconsistent (without performing the inference system  $IS'_4$ ). That is, it exists an interpretation such that:

$$\langle TBox_{exple}, ABox_{exple} \rangle \models \neg RQ(S_2)$$

We can deduce that the service  $S_2$  is not acceptable w.r.t. the required QoS  $RQ$ .

**Acceptance of the QoS of  $S_1$**  The QoS of the service  $S_1$  is acceptable w.r.t  $RQ$  if:

$$\langle TBox_{exple}, ABox_{exple} \rangle \models RQ(S_1).$$

That is,

$$\langle TBox_{exple}, ABox_{exple} \cup \{\neg RQ(S_1)\} \rangle \text{ is inconsistent}$$

where,

$$TBox_{exple} = \{\Phi\}$$

$$ABox_{exple} = \{BadTR_{0.25}(S_1), GoodTR_{0.75}(S_1), GoodRC_{0.5}(S_1), BadRC_{0.5}(S_1)\}$$

The reasoning algorithm generates inference systems  $IS_i$  by applying the propagation rules as follows:

$$\begin{aligned}
IS_0 &= \{TBox_{exple}, ABox_{exple}, \neg RQ(S_1)\} \\
&= \{BadTR_{0.25}(S_1), GoodTR_{0.75}(S_1), GoodRC_{0.5}(S_1), BadRC_{0.5}(S_1), \\
&\neg[(GoodRC_{\geq 0.4} \sqcap \neg BadTR_{\geq 0.8}) \sqcup GoodRC_{\geq 0.9}](S_1)\} \\
IS_1 &= IS_0 \cup \{GoodRC_0(S_1), [\neg(GoodRC_{\geq 0.4} \sqcap \neg BadTR_{\geq 0.8}) \sqcap \neg GoodRC_{\geq 0.9}](S_1)\} \\
&\text{(generated by pushing the negation)} \\
IS_2 &= IS_1 \cup \{\neg(GoodRC_{\geq 0.4} \sqcap \neg BadTR_{\geq 0.8})(S_1), \neg GoodRC_{\geq 0.9}(S_1)\} \text{ (gen-} \\
&\text{erated by } R_{\sqcap}) \\
IS_3 &= IS_2 \cup \{(GoodRC_{< 0.4} \sqcup BadTR_{\geq 0.8})(S_1), GoodRC_{< 0.9}(S_1)\} \text{ (generated} \\
&\text{by } R_{\neg}) \\
IS'_4 &= IS_3 \cup \{GoodRC_{< 0.4}(S_1), GoodRC_{< 0.9}(S_1)\} \text{ or } IS''_4 = IS_3 \cup \{BadTR_{\geq 0.8}(S_1), \\
&GoodRC_{< 0.9}(S_1)\} \text{ ( } IS'_4, IS''_4 \text{ are generated by } R_{\sqcup}) \\
IS'_4 &\text{ contains the clash } \{BadTR_{\geq 0.8}(S_1), BadTR_{0.25}(S_1)\} \\
IS'_5 &= IS'_4 \cup \{(GoodRC_{< 0.4}(S_1)\} \text{ (generated by } R_{reduction} \text{ on } IS'_4) \\
IS'_5 &\text{ contains the clash : } GoodRC_{< 0.4}(S_1), GoodRC_{0.5}(S_1)
\end{aligned}$$

The algorithm stops because all the inference systems contain a clash. Therefore, the system is inconsistent.

This means that the service  $S_1$  is acceptable w.r.t. the required QoS  $RQ$ .

## 6 Conclusion and Future Work

In this paper, we proposed a high level support for describing the semantic QoS variations in order to react to the service behavior changes. It provides an appropriate adaptation and avoid the process to grind to a halt. The key element of our model is the probabilistic extension of DL proposed. This extension allows to express the QoS requirements and performs an efficient reasoning mechanism to let the system self healing. Our variant of description logic has a minimal set of logical constructors. It contains only concept terms and two constructors which are conjunction and disjunction. As future work, we plan to extend the proposed approach by providing more constructors.

## References

1. Marco Comuzzi and Barbara Pernici. A framework for qos-based web service contracting. *TWEB*, 3(3), 2009.
2. Dragan Ivanovic, Manuel Carro, and Manuel V. Hermenegildo. A constraint-based approach to quality assurance in service choreographies. In *ICSOC*, pages 252–267, 2012.
3. Kyriakos Kritikos and Dimitris Plexousakis. Requirements for qos-based web service description and discovery. *IEEE T. Services Computing*, 2(4):320–337, 2009.
4. Thomas Lukasiewicz. Expressive probabilistic description logics. *Artif. Intell.*, 172(6-7):852–883, 2008.
5. Thomas Lukasiewicz and Umberto Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *Web Semant.*, 6(4):291–308, November 2008.
6. Carsten Lutz and Lutz Schröder. Probabilistic description logics for subjective uncertainty. In Fangzhen Lin, Ulrike Sattler, and Miroslaw Truszczynski, editors, *Principles of Knowledge Representation and Reasoning (KR 2010)*, pages 393–403. AAAI Press; Menlo Park, CA, 2010.
7. Flavio De Paoli, Matteo Palmonari, Marco Comerio, and Andrea Maurino. A meta-model for non-functional property descriptions of web services. In *ICWS*, pages 393–400, 2008.
8. Barbara Pernici and Seyed Hossein Siadat. A fuzzy service adaptation based on qos satisfaction. In *CAiSE*, pages 48–61, 2011.
9. Vuong Xuan Tran and Hidekazu Tsuji. A survey and analysis on semantics in qos for web services. In *Proceedings of the 2009 International Conference on Advanced Information Networking and Applications*, pages 379–385, Washington, DC, USA, 2009. IEEE Computer Society.
10. Qi Yu and Athman Bouguettaya. Multi-attribute optimization in service selection. *World Wide Web*, 15(1):1–31, 2012.
11. Mohamed Anis Zemni, Salima Benbernou, and Manuel Carro. A soft constraint-based approach to qos-aware service selection. In *ICSOC*, pages 596–602, 2010.