

Considering Quality of a Service in an Intentional Approach

Assia Ait Ali Slimane, Manuele Kirsch Pinheiro, Carine Souveyet

▶ **To cite this version:**

Assia Ait Ali Slimane, Manuele Kirsch Pinheiro, Carine Souveyet. Considering Quality of a Service in an Intentional Approach. Stephan Reiff-Marganiec and Marcel Tilly. Handbook of Research on Service-Oriented Systems and Non-Functional Properties: Future Directions, IGI Global, pp.334-351, 2012. hal-00673524

HAL Id: hal-00673524

https://paris1.hal.science/hal-00673524

Submitted on 23 Feb 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Handbook of Research on Service-Oriented Systems and Non-Functional Properties:

Future Directions

Stephan Reiff-Marganiec
University of Leicester, UK

Marcel Tilly
European Microsoft Innovation Center, Germany

Information Science
REFERENCE

Managing Director: Lindsay Johnston
Senior Editorial Director: Heather Probst
Book Production Manager: Sean Woznicki
Development Manager: Joel Gamon
Development Editor: Michael Killian
Acquisitions Editor: Erika Carter
Typesetters: Lisandro Gonzalez, Mackenzie Snader
Print Coordinator: Jamie Snavely
Cover Design: Nick Newcomer

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Copyright © 2012 by IGI Global. All rights reserved. No part of this publication may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher. Product or company names used in this set are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI Global of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

Handbook of research on service-oriented systems and non-functional properties : future directions / Stephan Reiff-Marganiec and Marcel Tilly, editors.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-61350-432-1 (hardcover) -- ISBN 978-1-61350-433-8 (ebook) -- ISBN 978-1-61350-434-5 (print & perpetual access) 1. Service-oriented architecture (Computer science)--Handbooks, manuals, etc. 2. Business--Data processing--Handbooks, manuals, etc. I. Reiff-Marganiec, Stephan. II. Tilly, Marcel, 1971-
TK5105.5828.H36 2012
004.6’54--dc23

2011031924

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Chapter 15
Considering Quality of a Service in an Intentional Approach

Assia Ait-Ali-Slimane
Université Paris1 Panthéon Sorbonne, France

Manuele Kirsch-Pinheiro
Université Paris1 Panthéon Sorbonne, France

Carine Souveyet
Université Paris1 Panthéon Sorbonne, France

ABSTRACT

The success of service-based applications is based on service technologies such as Web services. Nevertheless, the benefits of the Service-Oriented Architecture (SOA) remain mainly at the software level, since business people are often unable to fully exploit its benefits due to their unfamiliarity with such software level technology. The intentional Service-Oriented Architecture (iSOA) suggests a move from the function-driven SOA to intention-driven SOA in order to provide service description understandable by business practitioners. However, such transposition from business to implementation level should also consider Quality of Service (QoS) aspects. In this paper, we propose modeling the Quality of intentional Service (QoiS) by introducing the quality goals and their qualitative and quantitative evaluation. We also propose populating the intentional service registry of the iSOA architecture with the QoiS description.

CONSIDERING QUALITY OF A SERVICE IN AN INTENTIONAL APPROACH

Service-Oriented Computing (SOC) is the computing paradigm that utilizes services as fundamental elements for developing software applications (Papazoglou et al., 2008). SOC relies on the Service-Oriented Architecture (SOA) (Alonso et al., 2004) that is a way of reorganizing a portfolio of legacy applications into services that are self-describing computational elements, which are platform independent, accessible through standard interfaces and can be assembled in complex compositions based on standard messaging protocols. Service based applications are considered as support for Business-to-Consumer (B2C) interactions and Business-to-Business (B2B) collaborations.

DOI: 10.4018/978-1-61350-432-1.ch015

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Services, usually referred as e-services, provide well-defined functionalities that allow users and applications to meet their functional requirements (Casati & Shan, 2001).

SOC is a way of designing a software system that is *function-driven* and it remains at the software level. Consequently, although business people are completely familiar with the notion of service, they are totally unable to fully exploit its benefits, since they are not familiar to such software level technology. SOC technology, such as WSDL (W3C, 2007) and OWL-S (W3C, 2004) is understandable by software professionals, but far to be easily comprehensible by business practitioners. Business practitioners use to reason in terms of business goals.

The *intentional Service-Oriented Architecture approach* (iSOA) (Rolland et al., 2009) suggests a progress from the function-driven SOC to intention-driven SOC in order to provide a service description understandable by business practitioners. The function-driven SOC focuses on a functional view of services, whereas the intention-driven SOC spells out the purpose, the intention behind a service. The main goal of iSOA is to fill the gap between high level business services, referred to *intentional service*, and low level software services. The iSOA approach proposes a higher abstraction level that allows business practitioners to publish, search and compose, in terms of goals and strategies, services that can be executed in the SOA level.

On both level (SOA and iSOA), the service selection remains an important challenge, especially, when a set of services fulfills the same functionality. Among these services, one will be eventually invoked by user, generally depending on a combined QoS evaluation. The QoS can be defined as a set of non functional properties related to software service such as performance, security, accuracy and fault tolerance mechanisms (O'Sullivan et al., 2002; W3C, 2003). On the business level, Fedosseev (2003/2004) describes QoS of business process as a set of qualitative and quantitative characteristics needed to meet the initial requirements of this process. On both levels, QoS stands for non-functional properties that the service provider can ensure and that are (or can be) demanded by the service user.

The QoS plays then an important role in the software service for different reasons (O'Sullivan et al., 2002; Aiello & Giorgini, 2004): (1) a service provider may offer the same functionality with differentiated QoS (for example different prices) and must therefore publish the different qualities for this same functionality; (2) a service requester may decide for a particular service based on its QoS properties; and (3) a service may depend on other services and it needs to be aware of the QoS of the collaborating services. Therefore, QoS becomes a main concern for providers and customers of a service.

Several works in the literature (Zeng et al., 2003; Herssens et al., 2008; Aiello & Giorgini, 2004; Penserini & Mylopoulos, 2005; Ma et al., 2009) propose QoS models. Certain authors (Zeng et al., 2003; Herssens et al., 2008) consider the QoS as a collection of metrics related to non functional properties of services. Zeng et al. (2003) proposes a QoS model to describe concepts of QoS such as execution price and execution duration, whereas Herssens et al. (2008) recommend using the Unified Modeling Language (UML) to enable QoS modeling. Requirement engineering community (Aiello & Giorgini, 2004; Penserini & Mylopoulos, 2005; Ma et al., 2009) propose to reason about non functional requirements, based on the qualitative framework (Mylopoulos et al., 1992), by considering the QoS as soft goal that a service can satisficed.

The expression of the QoS through the different metrics, as defined by (Zeng et al., 2003; Herssens et al., 2008), focuses on the implementation mechanisms of QoS, their computation and their monitoring. Other works such as (Aiello & Giorgini, 2004; Penserini & Mylopoulos, 2005; Ma et al., 2009) focus on the business process. However, an important gap remains between

these approaches: business consideration on QoS remains often disconnected from their implementation counterpart.

We advocate that metrics used on the software level should be raised up the intentional level, in order to be understandable for business practitioners. In our point of view, QoS handling on iSOA should respect and follow an intention driven approach that makes QoS of software service more accessible to business practitioners (Ait-Ali-Slimane & Souveyet, april, 2008).

In this chapter, we propose to extend the original iSOA proposal (Kaabi et al., 2004; Rolland et al., 2009) to deal with the quality dimension. We propose: (1) a model to describe the *Quality of intentional Service* (QoiS), named ISM-q (Intentional Service Model extended to quality dimension*)* that considers QoiS as the set of the quality goals that an intentional service contribute satisficing; (2) a method for evaluating the QoiS associated with an atomic and aggregate service; (3) a XML description of ISM-q that allows publishing intentional service QoiS into the intentional service registry; and (4) the use of XQuery language to retrieve the services that meet users' requirements as considering QoiS. Our approach considers two main characteristics. First, we consider that the intentional service achieves a goal and can satisfices several quality goals (based on (Aiello & Giorgini, 2004; Penserini & Mylopoulos, 2005)). Second, we consider that the intentional service QoiS can be the high level abstraction of the software service QoS.

The chapter is organized as follows. Section 2 presents the related work. In Section 3, an overview of the ISM model is presented. In Section 4, we describe the ISM-q and the associated evaluation methods. The Section 5 illustrates the XML description of ISM-q that is published in the registry. Section 6 considers a query language for retrieving an intentional service considering its QoiS. Finally, Section 7 outlines conclusion and future works.

BACKGROUND

Maximilien and Singh (2004) identify selection of services as the step that enable differentiating the services whose provide similar functionality, by considering their QoS, such as performance, security and accuracy (Mani & Nagarajian, 2002; O'Sullivan et al., 2002; W3C, 2003).

The selection process supports several QoS models (Zeng et al., 2003; Herssens et al., 2008; Maximilien & Singh, 2004; Dobson et al., 2005; Toma et al., 2006; Ma et al., 2009) that enable providers and users to share the same concepts in the perspective of facilitating selection algorithm execution. Certain authors (Zeng et al., 2003; Herssens et al., 2008) consider the QoS as a collection of metrics related to non functional properties of services. Zeng et al. (2003) propose a QoS model to describe concepts of QoS such as execution price and execution duration, whereas Herssens et al. (2008) recommend using the Unified Modeling Language (UML) to enable QoS modeling. Another authors (Maximilien & Singh, 2004; Dobson et al., 2005; Toma et al., 2006) propose an ontology to describe the quality concepts. Practically, Toma et al. (2006) proposes extending the Web Service Modeling Ontology (WSMO) with QoS support by modeling QoS as quality attributes.

Requirement engineering community (Aiello & Giorgini, 2004; Penserini & Mylopoulos, 2005; Ma et al., 2009) propose to reason about non functional requirements, based on the qualitative framework (Mylopoulos et al., 1992), by considering the QoS as soft goal that a service can satisficed. Soft goals cannot be satisfied in a clear cut manner by opposition to hard goals for which this is possible. Soft goals are said satisficed within acceptable limits, related to a satisficing degree, by opposition to hard goals which can be said satisfied. Satisficing degree can be "very satisficed", "satisficed", "neutral", ''not Satisficed" and "not at all satisficed", noted respectively by the symbols "++", "+","?", "-", and "--".

The expression of the QoS through the different metrics and attributes, as defined by Zeng et al., 2003; Herssens et al., 2008; Maximilien & Singh, 2004; Dobson et al., 2005; Toma et al., 2006) focuses on the implementation level of QoS. It describes what a software service can provide as quality. However, it does not specify how the latter can meet the users' non functional requirements. Instead, works such as (Aiello & Giorgini, 2004; Penserini & Mylopoulos, 2005; Ma et al., 2009) focus on business process and non functional requirements of users. Nevertheless, they do not consider the software service QoS.

Moreover, the evaluation of QoS must be global in order to evaluate the QoS of composite services. Zeng et al. (2003) propose a global approach by introducing a set of aggregation function for the computation of the QoS of a composite service. Aiello and Giorgini (2004) recommend axioms that make the propagation of full and partial satisficing, whereas (Maximilien & Singh, 2004; Penserini & Mylopoulos, 2005; Toma et al., 2006; Herssens et al., 2008) do not consider the case of a global QoS.

The *intentional Service-Oriented Architecture* approach (iSOA) (Rolland et al., 2009) provides a service description understandable by users. The *Intentional Service Model* (ISM) enables describing services for service publishing, retrieval and composition in a goal driver manner. The intentional service focuses on the goal it allows to achieve rather than on the functionality it performs. Besides, atomic intentional services are operationalized by a software services enabling fill the gap between the users' functional requirements and the software service.

Currently, ISM does not consider how to represent QoS eventually associated with an intentional service, neither with the corresponding operational service. We propose to extend the ISM to deal with the quality dimension in order to profit of the benefit of the iSOA approach while considering QoS issues, as defined in (Ait-Ali-Slimane & Souveyet, April, 2008).

INTENTIONAL SERVICES: THE ISM MODEL

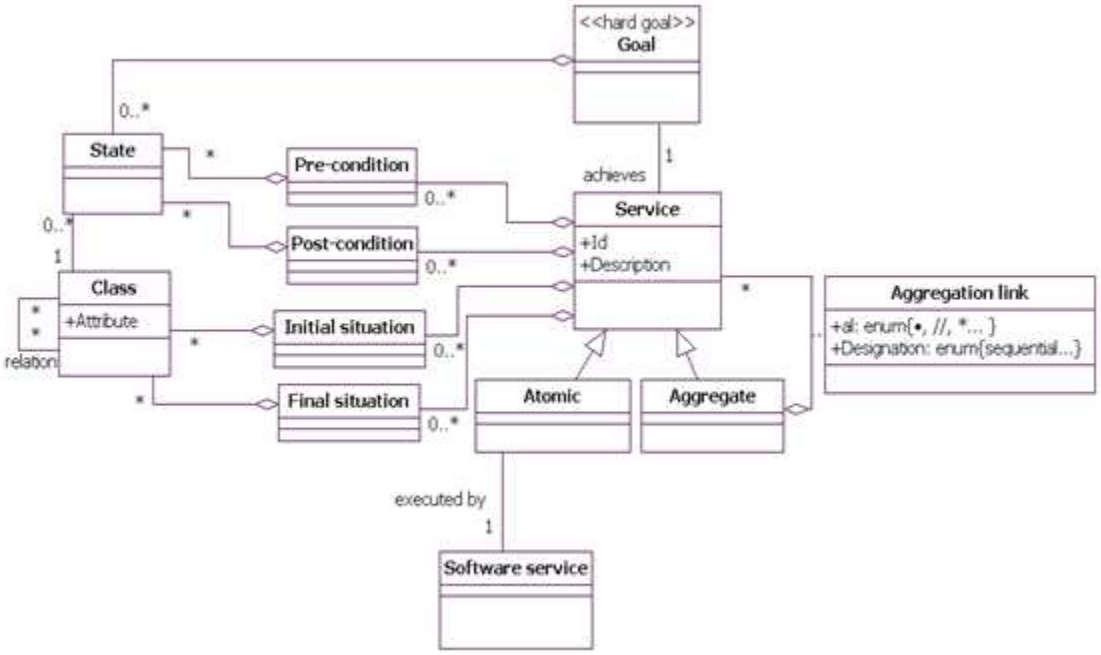
In (Kaabi et al., 2004; Rolland et al., 2009) the authors introduce the concept of *intentional service*. An intentional service is defined as a service captured at the business level, in business comprehensible terms and described in a goal perspective. The intentional service focuses on the goal it allows to achieve rather than on the functionality it performs. Intentional services fill the gaps between the user functional requirements and the software services.

The intentional service model (ISM) is traditionally composed of three parts (Kaabi et al., 2004; Rolland et al., 2009), namely the *service interface*, the *service behavior* and the *service composition*, whose elements are represented in Figure 1.

The service interface describes the visible parts of a service that permits the fulfillment of a *goal,* given *an initial situation* and terminating in a *final situation.* Thus, *Make Room Booking* is a goal to make a reservation for rooms in a hotel. The achievement of this goal leaves the system in the state *Booking made.* If *Accept Payment* is the goal of a service, then the initial situation refers to the *booking* and *customer* classes, whereas the final situation comprises also the *payment* class. Similarly to SOA, in which service users search services based on their functional interface, iSOA approach considers that business agents find an intentional service by the goal it allows to fulfill. The goal related to an intentional service plays then the central role in the service retrieval, since the user (the business agent) will look for a service based on the goal it fulfills, a goal that should match his functional needs. In order to underline this point, we note an intentional service using its goal. For instance, the intentional service allowing to *Make Room Booking* is noted $S_{Make\ Room\ Booking}$.

The service behavior is specified through its *pre* and *post conditions* that are the initial and

Figure 1. Elements composing the intentional service model (ISM)



final sets of states characterizing the initial and the final situation, respectively. For instance, in the *Accept Payment* service example, <*booking.state='OK'* ∧ *customer.status='registered'*> and <*booking.state='paid'*∧*payment.status = 'done'* > are the pre and post-conditions, respectively.

Similarly to software services, intentional services can be composed. The service composition part on the ISM model indicates this possibility through two different kinds of services, namely the *aggregate* service and the *atomic* one. The former is composed of other services whereas the latter is not. *Atomic services* have goals that are *operationalized by software service* (see Figure 1). In contrast *aggregate services* have high-level goals that need to be decomposed in lower level services till atomic intentional services are found. Aggregation of services can involve *variants*, *i.e.* services which are alternative to the others or composites (see (Rolland et al., 2009) for more detail).

In the *iSOA* approach, services are defined in a goal-driven manner focusing on the '*whys'* of the functionality provided by the underlying software service. In other words, these compositions are represented by a set of aggregation links (•, //,⊗, ν,* and ∪) (see Figure 1) indicating order or variability on service composition. Composition itself is goal-driven which are grounded in *XOR* (related to ⊗, and ∪), *OR* (related to ν), *AND* (related to •, // and *) relationships among intentional services. However, the intentional service proposed by (Kaabi et al., 2004; Rolland et al., 2009) deals with the functional aspects of the service that are related to user functional requirements. Therefore, the notion of quality of service is not considered by the intentional service. We introduce in (Ait-Ali-Slimane et al., 2009) the notion of *quality goals* which enables capturing the user non functional requirements. This notion is used here in order to introduce a *quality dimension* on iSOA approach.

PROPOSAL OVERVIEW

We extend ISM model by considering that an intentional service should also be described by a quality dimension. This quality dimension is introduced in the ISM-q model through the QoiS (Quality of intentional Service) element, which introduces the quality properties of the intentional service. We consider that quality aspects can be captured by quality goals (Ait-Ali-Slimane et al., 2009).

The adoption of quality goals is not new. Other authors (Yu et al., 1995; Mylopoulos et al., 1992) have considered this notion. Mylopoulos et al. (1992) propose to handle quality goals as soft goals in a qualitative framework. The qualitative framework (Mylopoulos et al., 1992) is a goal-oriented approach for addressing non functional requirements such as security, performance and accuracy. We adopt the concepts of the qualitative framework, including the decomposition process, the contribution links and the evaluation process, in order to represent the non functional requirements and the QoiS as quality goals.

In order to make sure that quality goals are used uniformly by service providers and business users, we propose to combine quality goals in a quality referential. Although quality referential is out of the scope of this particular paper, we would like to underline the significance of building a quality referential in order to enable improving the quality consensus between users and service providers, corresponding to a specific domain.

The quality goals satisfaction is based on the concept of satisficing (Simon, 1981), which means a quality goal can be satisficed within acceptable limits (see (Mylopoulos et al., 1992; Chung et al., 2000) for more detail). The satisficing values can be qualitative one (Chung et al., 2000) or a quantitative one (Keller et al., 1990; Chung & Subramanian, 2001). The quality referential adopts a dual evaluation comprising a qualitative one and a quantitative one. The qualitative evaluation corresponds to the satisficing degree. The expressions "*very satisficed"*, "*satisficed*", "*neutral*", "*not Satisficed*" and "*not at all satisficed*", noted respectively by the symbols "++", "+","?", "-", and "--" correspond to the satificing degrees of the quality goal. We selected these notations for their understandability and readability for business users. However, as this satisficing scale is subjective we complement this evaluation by introducing the concept of *metric*. Each quality goal can be evaluated by one or more metrics which facilitate the quantification of the quality goal satisficing. Each metric has a set of *reference values* (corresponding to the quality of software services) that are associated with the qualitative scale (represented by satisficing degree). For instance, a quality goal C*onfidentiality* can be associated with different metrics, such as *Fraud Rate* and *Security Level*. The *Fraud Rate* metric can be associated with a qualitative scale in which the satisficing degree *"++"* is obtained when the *Fraud Rate is lower than 0,02%,* whereas the satisficing degree *"-"* is obtained when the *Fraud Rate is greater than 1%.* The use of such metrics allows service providers to connect high level satisficing degree with corresponding operational measures performed by software services.

The quality referential is used by both sides: service providers and users. The former to describe the quality of theirs services, and the latter to express their non functional requirements.

Besides, intentional services can be atomic or aggregate. Aggregate services represent composition and variability on service definition. Thus, the quality (QoiS) of an atomic service is said *simple* and the quality of an aggregate service is said *global*: the simple QoiS reflects the contribution of an atomic service to the quality goals associated with it. It is calculated as a set of quality goals (q_j) that a service contributes satisficing and the related satisficing degree (d_j). The global QoiS results of a compilation of QoiS related to the constituent services of an aggregate service.

Hence, it is important to calculate the quality of the intentional service as a whole. The global

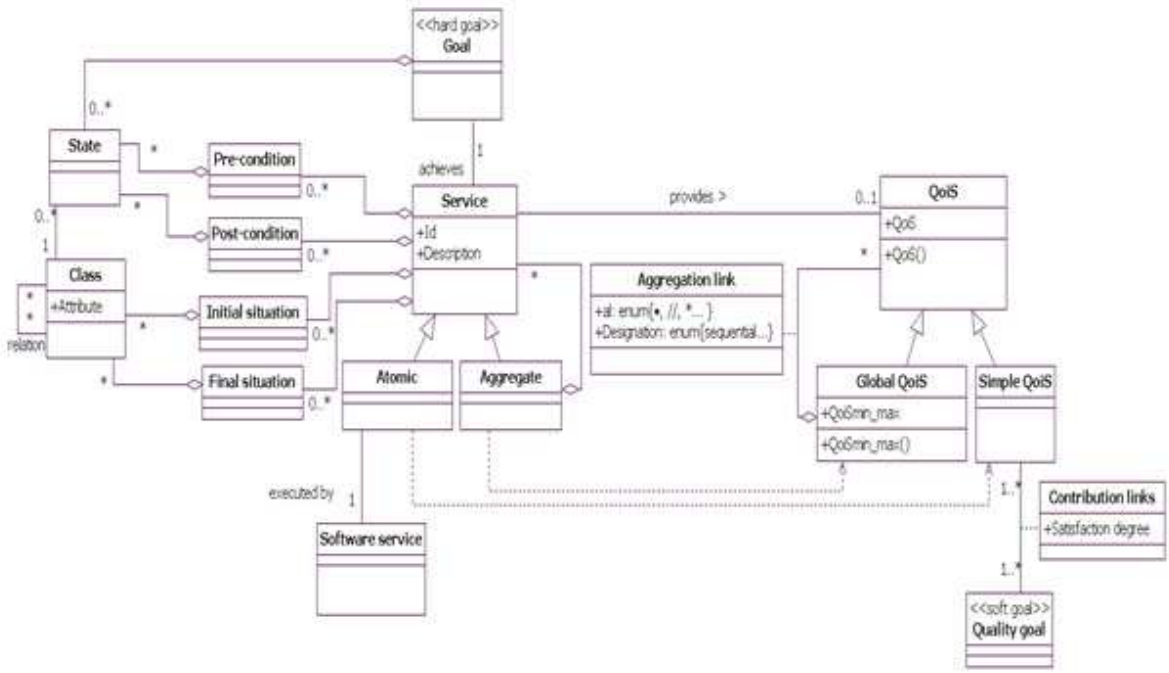
QoiS is based on the different kind of aggregation links binding aggregate and components services. In order to make use of the QoiS of an aggregate service in the selection algorithms, we propose calculating the $QoiS_{min_max}$ corresponding to the set of quality goals (q_j) that an aggregate service contributes satisficing and the related satisficing degree (d_j). The calculation of $QoiS_{min_max}$ depends on the aggregation link forming the global QoiS, namely composite quality or variant quality.

Every intentional service must be available in the intentional service registry. This enables retrieval of atomic and aggregate services and their adaptation to users' needs. Retrieval is based on goal matching that is, given a goal *G*, the registry searches a service that can satisfy this goal. In order to publish intentional service on an intentional registry, we adopt XML view of ISM-q. Such view represents all service dimensions: interface, behavior, composition and QoiS. Besides, we consider in this paper the possibility of using XQuery expressions in order to query registry, taking advantage of the quality description of these services. Traditionally, iSOA approach considers that business users submit their requests through a user interface. In order to consider QoiS issues, such user interface should allow business users to express their preferences considering quality goals. In such preferences, business users can indicate what quality goals they expect from intentional services and the corresponding satisficing degree they consider as minimal. We call this user's quality context.

DEFINING THE QUALITY OF INTENTIONAL SERVICES (QoiS)

We define the quality of an intentional service (QoiS) as a set of quality goals that a service contributes satisficing and the associated satisficing degree. In this section, we present the ISM extended to QoiS (ISM-q) which models the QoiS aspects of different kinds of intentional services. We also present the evaluation and the calculation methods of these.

Figure 2. Intentional service model extended to QoiS (ISM-q)



QoiS Description

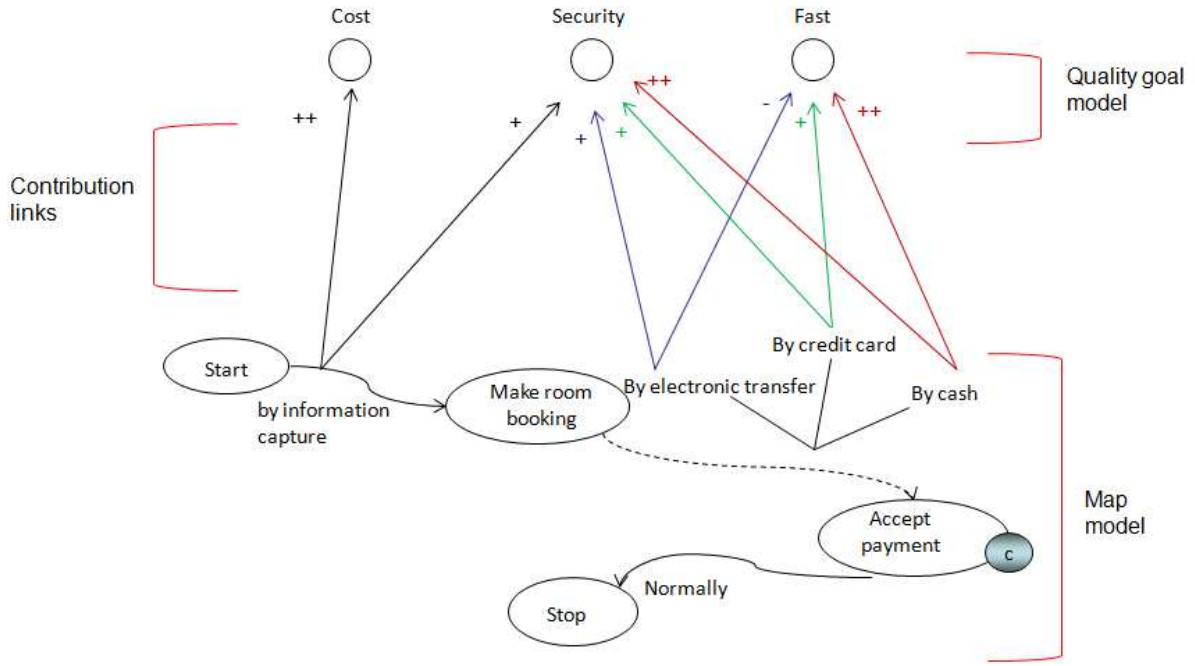
In addition to service interface, service behaviour and service composition (see Figure 1), we consider that an intentional service should also be described by a quality dimension. The QoiS element shown in Figure 2 introduces this dimension, with the quality properties of the intentional service. This quality dimension introduces the quality goals and their qualitative evaluation on ISM-q (Ait-Ali-Slimane et al., 2009). We consider the *goal* (the goal element shown in Figure 2) associated with the service as a *functional hard goal*, and the *quality goal* (the quality goal element shown in Figure 2) as a *non functional soft goal*, capturing non-functional requirements that the service can contribute satisficing (Ait-Ali-Slimane & Souveyet, may, 2008). The goal satisfaction can be established through verification technique, whereas the quality goal satisfaction is subjective and cannot be established in clear cut manner (Mylopoulos et al., 1992; Jureta et al., 2006).

We propose using two models-oriented goal for identifying intentional services and their QoiS: a Map model (Rolland & Prakash, 2000) and a quality goal model. The Figure 3 presents an overview of both models.

The Map model (Rolland & Prakash, 2000) is used for modeling functional requirements as a map of goals (a hard goal). A map is a labeled directed graph with goals as nodes and strategies as edges. The map is capable of representing many strategies that can be used for achieving a goal. This map is used as an architectural style for determining business services and their composition (Rolland et al., 2009), by defining an intentional service as a section of the map (Kaabi et al., 2004). A Section is a triplet <*Source goal, Target goal, Strategy*> that represents a way to achieve the target goal from the source goal following the strategy.

The quality goal model represents criteria that services should meet. We propose that all service providers consult the quality referential to deter-

Figure 3. Map model, quality goal model and contribution links



mine quality goals that theirs services can satisfice, enabling truthful descriptions in terms of QoiS.

Thus, the satisficing of those quality goals (soft goals) is encoded in the map. Their satisficing reflect the partial contribution of a service (map) towards (or against) a particular quality goal (Ait-Ali-Slimane & Souveyet, may, 2008). *Contributions links* are used to represent the links among the Map and the quality goal models, as showed by the "Contributions link" element in the Figure 2. The semantic of these links are based on the satisficing degree attribute in the Figure 2.

For example, Figure 3 describes the variant intentional service $\boldsymbol{S}_{Accept\ payment}$. The identification of intentional services from the map model is detailed in (Kaabi et al., 2004). This service is composed of $\boldsymbol{S}_{Accept\ payment\ by\ transfer}$, $\boldsymbol{S}_{Accept\ payment\ by\ credit\ card}$ and $\boldsymbol{S}_{Accept\ payment\ by\ cash}$. Among these variant services, the $\boldsymbol{S}_{Accept\ payment\ by\ transfer}$ service contributes to make satisficing (+) a *Security* quality goal and contributes to not satisficing (-) a *Fast* quality goal.

The expression of a quality goal satisficing allows the service provider to indicate the quality level that their services can guarantee. Such degrees can then be matched with users' quality context. The integration of the quality dimension in the intentional service description aims at a user-centered selection of services. Although service selection is out of the scope of this paper, we define the intentional selection of services as the step that allows selecting services that satisfice user' quality context.

As shown in Figure 2, the quality (QoiS) of an atomic service is said *simple* and the quality of an aggregate service is said *global*: the simple QoiS reflects the contribution of the atomic service to the quality goals associated with it. The global QoiS results of a compilation of QoiS related to the constituent services of an aggregate service. Hence, it is important to calculate the quality of the intentional service as a whole.We detail this aspect in the following.

QoiS Evaluation

The QoiS of an intentional service is defined as a set of quality goals that a service contributes satisficing and the associated satisficing degree. As mentioned in the Figure 2, the service may define several QoiS. The QoiS can be a *simple QoiS* related to the *atomic service* or a *global QoiS* associated with the *aggregate service*. For instance, the service $\mathbf{S}_{\text{Make Room Booking}}$ defines several QoiS that depend on the different variant services.

Simple QoiS

As shown in Figure 2, the QoiS of an atomic service (S_{at}) is said simple as it reflects the contribution of this service to satisficing quality goals. The Simple QoiS is calculated as a set of quality goals (q_j) that a service contributes satisficing and the related satisficing degree (d_j). We adopt the following structure, inspired from (Rohleder et al., 2009), to represent the simple QoiS:

$$\text{QoiS}(S_{at}) = \mathbf{S}_{at}.\{<q_j,d_j>\} \quad (1)$$

For example, the QoiS of the atomic services $\boldsymbol{S}_{Accept\ Payment\ by\ electronic\ transfer}$ is expressed as follows: $QoiS(\boldsymbol{S}_{AcceptPaymentbyelectronictransfer})= \boldsymbol{S}_{Accept\ Payment\ by\ electronic\ transfer}.\{<Security, +>,<Fast, ->\}$. This QoiS defines that the service $\boldsymbol{S}_{Accept\ Payment\ by\ electronic\ transfer}$ contributes to make "*satisficing (+)*" the *Security* quality goal and "*not satisficing (-)*" the *Fast* quality goal.

Global QoiS

In the *ISOA* approach, QoiS is defined in a goal-driven manner focusing on the '*whys'* of the quality values (non functional properties) provided by the software service. Such relationships are represented by the aggregation links (•, //,⊗, v,* and ∪), indicating order or variability on QoiS composition. Moreover, QoiS evaluation is itself grounded in *XOR*, *OR*, *AND* relationships among

QoiS of intentional services that compose an aggregate service (service components).

As shown in Figure 2, the QoiS of an aggregate service is said global as it results of a compilation of QoiS related to its components services. In other terms, the QoiS of an aggregate service ($\boldsymbol{S}$) is calculated in a recursive manner, depending on the QoiS of components services.

The global QoiS calculation depends on the different kinds (composite and variant) of the aggregate service. As represented by the *Aggregation Link* element in the Figure 2, we propose to extend the semantic of the aggregation link to define the different kind of global QoiS. Given an aggregation link ($al: al \in \{\bullet, //, \otimes, \nu, *, \cup\}$) and a set of components services ($\boldsymbol{S}_n$), we adopt a following generic structure, inspired from (Rohleder et al., 2009), to represent the global QoiS of the aggregate service ($\boldsymbol{S}_{ag}$):

$$\text{QoiS}(\mathbf{S}_{ag}) = al\ (\text{QoiS}(\mathbf{S}_1), \text{QoiS}(\mathbf{S}_2) \ldots \text{QoiS}(\mathbf{S}_n)) \quad (2)$$

We apply this structure to represent the different kinds of global QoiS related to the aggregate service. We detail this aspect in turn.

A *composite quality* is related to the composite service. It reflects a composition of the QoiS of its component services, corresponding to an AND relationship between the QoiS of the component services. The composition is denoted "•", "//" or "*" symbol, depending on the kind of the composite service, respectively sequential, parallel or iterative. It is noted as:

$$\text{QoiS}(\bullet\ (\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_n)) = \bullet\ (\text{QoiS}(\mathbf{S}_1), \text{QoiS}(\mathbf{S}_2) \ldots \text{QoiS}(\mathbf{S}_n))$$

For example, The QoiS of the composite service $\boldsymbol{S}_{Make\ Confirmed\ Booking} = \bullet(\boldsymbol{S}_{Make\ Room\ Booking}, \boldsymbol{S}_{Accept\ Payment})$ is defined as the composition of the QoiS of both $\boldsymbol{S}_{Make\ Room\ Booking}$ and $\boldsymbol{S}_{Accept\ Payment}$. It is noted as:

$$\text{QoiS}(\mathbf{S}_{\text{Make Confirmed Booking}}) = \bullet(\text{QoiS}(\mathbf{S}_{\text{Make Room Booking}}), \text{QoiS}(\mathbf{S}_{\text{Accept Payment}}))$$

Introduction of *variability* in quality modelling is justified by the need to introduce flexibility in quality goal satisficing and adaptability in intentional service execution. There are three types of *variant quality*, namely *alternative, choice* and *multi-path.*

An *alternative* variation corresponds to an XOR relationship between the QoiS of the alternative components, *i.e.* each QoiS of alternative component can be satisficed in exclusively way. The symbol "⊗" is used to denote the alternative variation and it is noted as:

$$\text{QoiS}(\otimes\ (\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_n)) = \otimes\ (\text{QoiS}(\mathbf{S}_1), \text{QoiS}(\mathbf{S}_2) \ldots \text{QoiS}(\mathbf{S}_n))$$

For example, The QoiS of the alternative service $\boldsymbol{S}_{Accept\ Payment} = \otimes\ (\boldsymbol{S}_{Accept\ Payment\ by\ electronic\ transfer}, \boldsymbol{S}_{Accept\ Payment\ by\ credit\ card}, \boldsymbol{S}_{Accept\ Payment\ by\ cash})$ is defined as the XOR relationship between the QoiS of both $\boldsymbol{S}_{Accept\ Payment\ by\ electronic\ transfer}$, $\boldsymbol{S}_{Accept\ Payment\ by\ credit\ card}$ and $\boldsymbol{S}_{Accept\ Payment\ by\ cash}$. It is noted as:

$$\text{QoiS}(\mathbf{S}_{\text{Accept Payment}}) = \otimes\ (\text{QoiS}(\mathbf{S}_{\text{Accept Payment by electronic transfer}}), \text{QoiS}(\mathbf{S}_{\text{Accept Payment by credit card}}), \text{QoiS}(\mathbf{S}_{\text{Accept Payment by cash}}))$$

A *choice* variation corresponds to an OR relationship between the QoiS of the alternative components. The difference between alternative and choice lies on the fact that the former implies exclusion of variants QoiS whereas the latter authorizes the selection of several of the choice when the variant QoiS is satisficed. The symbol "**v**" is used to denote the choice variation and it is noted as:

$$\text{QoiS}(\nu\ (\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_n)) = \nu\ (\text{QoiS}(\mathbf{S}_1), \text{QoiS}(\mathbf{S}_2) \ldots \text{QoiS}(\mathbf{S}_n))$$

For example, The QoiS of the alternative service

$\boldsymbol{S}_{Investigate\ Candidate\ Booking} = \boldsymbol{v}(\boldsymbol{S}_{Investigate\ Candidate\ Booking\ on\ the\ Internet}, \boldsymbol{S}_{Investigate\ Candidate\ Booking\ by\ visiting\ a\ travel\ agent})$

is defined as the OR relationship between the QoiS of both $\boldsymbol{S}_{Investigate\ Candidate\ Booking\ on\ the\ Internet}$ and $\boldsymbol{S}_{Investigate\ Candidate\ Booking\ by\ visiting\ a\ travel\ agent}$. It is noted as:

$$QoiS(\mathbf{S}_{\text{Investigate Candidate Booking}}) = \mathbf{v}\ (QoiS(\mathbf{S}_{\text{Candidate Booking on the Internet}}), QoiS(\mathbf{S}_{\text{Investigate Candidate Booking by visiting a travel agent}}))$$

Finally a *multi-path* variation occurs when several QoiS compositions of an intentional service allow achieving the same intentional service and satisficing different quality goals. The multi-path is denoted "∪". It is noted as:

$$QoiS(\cup\ (\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_n)) = \cup\ (QoiS(\mathbf{S}_1), QoiS(\mathbf{S}_2) \ldots QoiS(\mathbf{S}_n))$$

For example, let us define the QoiS of the multi-path service $\boldsymbol{S}_{Make\ Confirmed\ Booking} = \cup(\bullet(\boldsymbol{S}_{Make\ Room\ Booking}, \boldsymbol{S}_{Accept\ Payment}), \boldsymbol{S}_{Get\ a\ Rewarded\ Booking})$. Thus, there are two paths to providing the intentional service *Make a Confirmed Booking:* one by achieving the sequence of intentional services *Make a Booking* and Accept *payment*, and another by achieving the service *Get a Rewarded Booking*. Therefore the QoiS of $\boldsymbol{S}_{Make\ Confirmed\ Booking}$ should consider the QoiS of the first path $QoiS(\bullet\ (\boldsymbol{S}_{Make\ Room\ Booking}, \boldsymbol{S}_{Accept\ Payment}))$ and the QoiS of the second path $QoiS(\boldsymbol{S}_{Get\ a\ Rewarded\ Booking})$. It is noted as:

$$QoiS(\mathbf{S}_{\text{Make Confirmed Booking}}) = \cup\ (QoiS(\bullet\ (\mathbf{S}_{\text{Make Room Booking}}, \mathbf{S}_{\text{Accept Payment}}), QoiS(\mathbf{S}_{\text{Get a Rewarded Booking}})))$$

Previously, we present the definition of QoiS of an intentional service: a simple QoiS is related to the atomic service, whereas the global QoiS is related to the aggregate service. However the global QoiS specification remains not sufficient because it cannot be used by the selection algorithms that enable choosing among various intentional services (that achieve the same functional goal) those that meet user' quality context. For that, we propose in the next section to calculate the $QoiS_{min_max}$ offered by the aggregate intentional service.

$QoiS_{min_max}$ Calculation

We propose calculating the $QoiS_{min_max}$ of an aggregate service in order to specify the set of quality goals that it contributes satisficing and the related satisficing degrees. The calculation of $QoiS_{min_max}$ depends on the aggregation link forming the global QoiS, namely composite quality ("•", "//", "*") or variant quality ("⊗", "v", "∪"). Given an aggregate service (S_{ag}), we adopt a following generic structure to represent the $QoiS_{min_max}$:

$$QoiS_{min_max}\ (\mathbf{S}_{ag}) = \{<Q,D>\} \quad (3)$$

The $QoiS_{min_max}$ of the aggregate service $\mathbf{S}_{ag}$ corresponds to the set of quality goals Q and the related satisficing degrees D that a service S_{ag} contributes satisficing. The calculation of the $QoiS_{min_max}$ is based on the propagation rules defined in the requirement engineering community (Chung et al., 2000). As its name indicates, the minimal and the maximal degrees are calculated (or propagated) in the case of a composite and a variant service, respectively.

In the case of a *composite quality* (for instance using the "•" operator), the rule R 1 is proposed. The notation of $QoiS_{min_max}$ is based on the equation *(3)*. The $QoiS_{min_max}$ is expressed as: (i) the union (∪) of quality goals (q_j) satisficed by the component services of the composite service *(S)*; and (ii) the minimum satisficing degree guaranteed by the component services per q_j. In the case of a composite quality (related to a composite service), the $QoiS_{min_max}$ is calculated as follows:

$$Q=\bigcup_{j=1}^{m} q_j \quad \mathbf{R}\ 1$$

$$D=\min_{1\le i\le n;1\le j\le m} d_{ij} \,/\, q_j$$

vice $\mathbf{S}_{\text{Make Confirmed Booking}}$. The QoiS of this service id specified as $\text{QoiS}(\mathbf{S}_{\text{Make Confirmed Booking}})= \bullet (QoiS(\mathbf{S}_{\text{Make Room Booking}}), QoiS(\mathbf{S}_{\text{Accept Payment}}))$. The $\text{QoiS}_{\text{min_max}}(\mathbf{S}_{\text{Make Confirmed Booking}})$ is computed as follows: (i) the set of quality goals the $\mathbf{S}_{\text{Make Confirmed Booking}}$ contributes satisficing, namely *Fast*, *Security* and *Cost*; and (ii) the minimum satisficing degree per quality goal: for instance $\mathbf{S}_{\text{Make Room Booking}}$ contribute to make "*satisficing (+)*" the *Security* quality, whereas $\mathbf{S}_{\text{Accept Payment}}$ contribute to make it "*not satisficing (-)*". Thus a minimum degree of *Security* quality is "*not satisficing (-)*". As a result, the $\text{QoiS}_{\text{min_max}}(\mathbf{S}_{\text{Make Confirmed Booking}})$ = {<*Fast,*+>,<*Security*,->, < *Cost,*++> }.

However, in the case of a *variant quality* (for instance using the "∪" operator), the rule R 2 is proposed. The notation of $\text{QoiS}_{\text{min_max}}$ is also based on the equation *(3)*. The $\text{QoiS}_{\text{min_max}}$ is also expressed as: (i) the union (∪) of quality goals (q_j) satisficed by the component services of the variant service *(S)*; and (ii) the maximum satisficing degree guaranteed by the component services per q_j. In the case of a variant quality (related to a variant service), the $\text{QoiS}_{\text{min_max}}$ is calculated as:

$$Q=\bigcup_{j=1}^{m} q_j \quad \mathbf{R}\ 2$$

$$D=\max_{1\le i\le n;1\le j\le m} d_{ij} \,/\, q_j$$

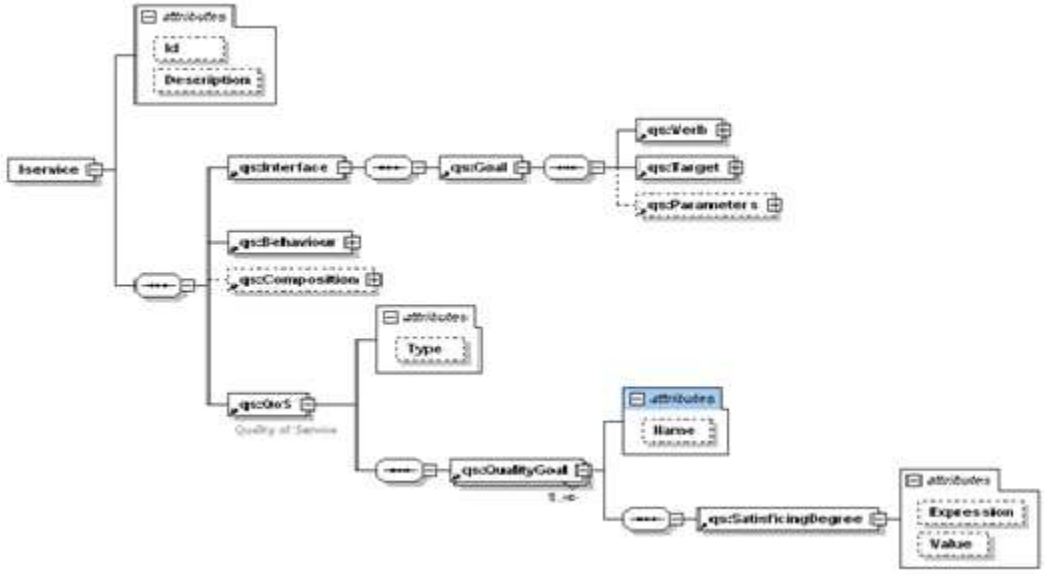
For instance, let us consider the $\text{QoiS}_{\text{min_max}}$ of the variant service $\boldsymbol{S}_{Accept\ Payment}$. The QoiS of this service id specified as $QoiS(\boldsymbol{S}_{Accept\ Payment})= \otimes(QoiS(\boldsymbol{S}_{Accept\ Payment\ by\ electronic\ transfer}), QoiS(\boldsymbol{S}_{Accept\ Payment\ by\ credit\ card}), QoiS(\boldsymbol{S}_{Accept\ Payment\ by\ cash}))$. The $\text{QoiS}_{\text{min_max}}(\boldsymbol{S}_{Accept\ Payment})$ is computed as follows: (i) the set of quality goals the $\boldsymbol{S}_{Accept\ Payment}$ contributes satisficing, namely *Fast* and *Security*; and (ii) the maximum satisficing degree per quality goal: for instance $\boldsymbol{S}_{Accept\ Payment\ by\ electronic\ transfer}$ contribute to make "*not satisficing (-)*" the *Security* quality, whereas $\boldsymbol{S}_{Accept\ Payment\ by\ credit\ card}$ contribute to make it "*satisficing (+)*". Thus a maximum degree of *Security* quality is "*satisficing (+)*". As a result, the $\text{QoiS}_{\text{min_max}}(\boldsymbol{S}_{Accept\ Payment})$ = {<*Fast,*+>,<*Security*,+>}. More explicitly, this latter indicates that $\boldsymbol{S}_{Accept\ Payment}$ service provides the $\text{QoiS}_{\text{min_max}}$ corresponding to "*satisficing (+)"* the *Fast* quality and "*satisficing (+)*" the *Security* one. This $\text{QoiS}_{\text{min_max}}$ is depending of the chosen variant, namely $S_{Accept\ Payment\ by\ electronic\ transfer}$, $\text{S}_{\text{Accept Payment}by\ credit\ card}$ or $S_{Accept\ Payment\ by\ cash}$.

The previous demonstrate the importance of calculating the QoiS related to the intentional service in order to differentiate the various services that achieve the same functional goal. The simple QoiS and the $\text{QoiS}_{\text{min_max}}$ are both used to select among services those that meet users' quality context. Next section discusses the description of ISM-q that is published in the intentional service registry.

PUBLISHING THE QOIS

Every intentional service must be available in the intentional service registry. This enables retrieval of atomic and aggregate services and their adaptation to users' needs. Retrieval is based on goal matching; that is, given a goal G, the registry searches a service that can satisfy this goal. We consider, in the following, the issue of making available the QoiS descriptions in the registry. In order to publish intentional service on an intentional registry, we adopt XML view of *ISM-q*. Such view represents all service dimensions: interface, behavior, composition and QoiS. Figure 4 illustrates the XML Schema used to specify intentional service description in XML. It is worth noting that in this paper, we

Figure 4. XML schema of intentional service description



focus only QoiS dimension. Details concerning interface, behavior and composition can be found in (Rolland et al., 2009).

As shown in the Figure 4, the intentional service description considers the QoiS dimension of the intentional service additionally to the interface, the behavior and the composition ones. Such description allows service selection based on the QoiS provided by the service, in addition to goal it satisfies. In particular, the service provider can publish, for an atomic service, the simple QoiS associated with this service (Figure 4 element *qs:QoS* attribute *Type=SimpleQoS*). For an aggregate service, the service provider can publish the $QoiS_{min_max}$ guaranteed by this service (Figure 4 element *qs:QoS* attribute *Type=GlobalMinQoS*). Each QoiS is composed of a set of quality goals (element *qs:QualityGoal* in Figure 4) satisficed by the intentional service and their corresponding satisficing degrees (element *qs:SatisficingDegree* in Figure 4).

The XML file presented in the Figure 5 specifies the QoiS of the atomic service $\boldsymbol{S}_{Accept\ Payment\ by\ electronic\ transfer}$, which achieves the goal *Accept payment* and contributes satisficing the quality goals *Security* ("+") and *Fast ("-")*. The XML file shown in the Figure 6 describes the $QoiS_{min_max}$ associated with the aggregate service $\mathbf{S}_{Make\ Confirmed\ Booking} = \bullet(\mathbf{S}_{Make\ Room\ Booking}, \mathbf{S}_{Accept\ Payment})$ that achieves the goal *Make Confirmed Booking* and contributes satisficing the quality goals *Security* ("-"), *Cost* ("++") and *Fast ("+")*. As one can observe on Figure 5 and Figure 6, the satisficing degrees proposed for each quality goal are described using a dual representation. Actually, in addition to the notation proposed on ISM-q, which indicates the satisficing degree using understandable expression for business agents, XML service description also includes a numeric value (attribute *value* in Figure 4). Such numeric values ($-2 \leq value \leq +2,\ value \in \aleph$) translates the qualitative satisficing degree in such a way that queries involving such degrees can be easily executed. For instance, service $\boldsymbol{S}_{Accept\ Payment}$ provides the quality goal *Security* with a satisficing degree of "+", which is translated to the value "*+1*" (Figure 5 line 15).

Figure 5. Example of simple QoiS

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <qs:Iservice xsi:schemaLocation="qs.xsd QoS_Iservice_QoS.xsd"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:qs="qs.xsd"
3   Id="http://www.booking.com/APet" Description="SAccept Payment by electronic transfer
4   ">
5   <qs:Interface>
6           <qs:Goal>
7             <qs:Verb Verb="Accept"/>
8             <qs:Target Target="Payment"/>
9             <qs:Parameters> Parameters="by electronic transfer" </qs:Parameters>
10          </qs:Goal>
11  </qs:Interface>
12  <qs:Behaviour>. . .             </qs:Behaviour>
13  <qs:QoS Type="SimpleQoS">
14      <qs:QualityGoal Name="Security">
15          <qs:SatisficingDegree Expression="Satisficed (+)" Value="+1"/>
16      </qs:QualityGoal>
17      <qs:QualityGoal Name="Fast">
18          <qs:SatisficingDegree Expression="Not Satisficed (-)" Value="-1"/>
19      </qs:QualityGoal>
    </qs:QoS>
    </qs:Iservice>
```

Figure 6. Example of $QoiS_{min_max}$

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <qs:Iservice xsi:schemaLocation="qs.xsd QoS_Iservice_QoSmin.xsd"
…   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:qs="qs.xsd"
3   Id="http://www.booking.com/MCB"  Description="SMake Confirmed Booking">
4   <qs:Interface>
5           <qs:Goal>
6                   <qs:Verb Verb="Make"/>
7                   <qs:Target Target=" Confirmed Booking"/>
8                   <qs:Parameters>. . .             </qs:Parameters>
9           </qs:Goal>
10  </qs:Interface>
11  <qs:Behaviour>. . .             </qs:Behaviour>
12   <qs:QoS Type="GlobalMinQoS">
13      <qs:QualityGoal Goal="Security">
14        <qs:SatisficingDegree Expression="Not Satisficed (-)" Value="-1"/>
15      </qs:QualityGoal>
16      <qs:QualityGoal Goal="Cost">
17        <qs:SatisficingDegree Expression="Very Satisficed (++)" Value="+2"/>
18      </qs:QualityGoal>
19      <qs:QualityGoal Goal="Fast">
20        <qs:SatisficingDegree Expression="Satisficed (+)" Value="+1"/>
21      </qs:QualityGoal>
22   </qs:QoS>
23  </qs:Iservice>
```

Figure 7. Example of query for an intentional service expressed in XQuery

```
for $x in //Iservice
let $g := $x/Interface/Goal
let $q := $x/QoS/QualityGoal
where $g/Verb/@Verb = "Make" and
    $g/Target/@Target = "Confirmed Booking" and
    ( ($q/@Goal = "Security" and $q/SatisficingDegree/@Value >= -1)
     or
      ($q/@Goal = "Cost" and $q/SatisficingDegree/@Value >=1 ) )
  return $x
```

Moreover, the Figure 6 describes the $QoiS_{min_max}$ of the aggregate service $\mathbf{S}_{Make\ Confirmed\ Booking}$, including three different quality goals: *Security* (Figure 6 line 13), *Cost* (Figure 6 line 16) and *Fast* (Figure 6 line 19). For each quality goal, the satisficing degree is indicated. For example, the service described in Figure 6 supplies a quality goal *Security* with a satisficing degree "*Not satisficed (-)*" (Figure 6 line 14).

QUERYING INTENTIONAL SERVICES REGISTRY

Although query evaluation and service selection are out of the scope of this particular paper, we would like to underline the possibility of using XQuery expressions in order to query intentional service registry, taking advantage of the quality description of these services.

Traditionally, iSOA approach considers that business agents submit their requests through a user interface. In order to consider QoiS issues, such user interface should allow business agents to express their preferences considering quality goals. Through the user's quality context, business agents can indicate what quality goals they expect from intentional services and the corresponding satisficing degree they consider as minimal. The user's quality context will be used by the intentional service registry. We propose using a Multi-Criteria Decision-Making (MCDM) methods in order to select from several services those that meet the user's quality context. The MCDM methods can deal with conflicting QoiS such as security and performance.

Based on the user's quality context, the user interface can form XQuery expression representing functional requirement formulated by the business agent as well as quality goals this agent demands. For instance, the Figure 7 presents a query for intentional service achieving the goal *Make confirmed booking* and providing *Security* and *Fast* quality goals with a minimal satisfaction degree "Not satisficed (-)" and "Satisficed (+)" respectively.

By considering XQuery for querying intentional registry, we focus on the advantages of using a standard query language. W3C XQuery standard is largely accepted by SOA community and offers a simple way of handling XML data (Innocenti, 2007), as the XML description proposed for intentional services.

CONCLUSION

In the last few years, the notion of service became a transversal notion that talks to technical software community as well as to business IT community. However, vocabulary used by these communities is not the same. For software community, services refer mainly to Web services and related technologies, with their technical constraints and implications. For business IT community, services based applications are considered as a support to improve cooperation between companies. In this context, the notion of intentional service is used to

improve the service semantic by decreasing the gap between the users' requirements, considered by business IT community, and the software service, handled by software community. However, until now, ISM did not consider the quality dimension, but only functional requirements associated with intentional services. This chapter proposed an extension to the ISM specification by introducing the quality dimension.

Our contribution is varied. Firstly, we described the ISM extended to QoiS (ISM-q) and a whole evaluation methods enabling evaluating the global QoiS. Secondly, we recommend an XML description of the ISM-q in order to publish QoiS descriptions in the intentional service registry. Finally, we suggest the XQuery language to retrieve services considering their quality dimension.

As future work, we develop the selection algorithm by adopting the Technique for Order Preference by Similarity to Ideal Solution (Hwang & Yoon, 1981; Zavadskas, 1986) to select from several intentional services according to the users' quality context. Since the aggregate service supports variability in achieving the functional goal of user, we are investigating as future work how to configure this service according to the users' quality context. A prototype that supports intentional service selection and the configuration of an aggregate service is under development.

REFERENCES

W3C. (2003). *QoS for Web services: Requirements and possible approaches*. Retrieved from http: // www.w3c.or.kr/ kr-office/ TR/ 2003/ws-qos/

W3C. (2004). *OWL-S: Semantic markup for Web services*. Retrieved from http: //www.w3.org/ Submission/OWL-S/

W3C. (2007). *Web service description language (WSDL) version 2.0 part 1: Core language*. Retrieved from http: //www.w3.org/ TR/wsdl12/

Aiello, M., & Giorgini, P. (2004). Applying the Tropos methodology for analysing Web services requirements and reasoning about qualities of services. *Journal for the Informatics Professional*, *5*(4).

Ait-Ali-Slimane, A., Kirsch-Pinheiro, M., & Souveyet, C. (2009). *Goal reasoning for quality elicitation in the ISOA approach*. International Conference on Research Challenges in Information Science (RCIS), IEEE, Maroc.

Ait-Ali-Slimane, A., & Souveyet, C. (2008, May). *Une Approche Méthodologique Orientée But pour Eliciter les Qualités de Service comme des Aspects*. INORSIDE. France.

Ait-Ali-Slimane, A., & Souveyet, C. (2008, April). *A goal driven approach to deal with quality of service as potential aspects*. In Inter. Workshop on Advanced Information Systems for Enterprises (IWAISE'08), Algeria.

Alonso, G., Casati, F., Kuno, H., & Machiraju, V. (2004). *Web services: Concepts, architecture, and applications*. Springer Verlag.

Casati, F., & Shan, M. (2001). Dynamic and adaptive composition of e-services. *The 12th International Conference on Advanced Information Systems Engineering (CAiSE 00)*. *Journal for Information Systems*, *26*(3), 143–163. doi:10.1016/ S0306-4379(01)00014-X

Chung, L., Nixon, B., Yu, E., & Mylopoulos, J. (2000). *Non-functional requirements in software engineering*. Kluwer Academic Publishers.

Chung, L., & Subramanian, N. (2001). Process-oriented metrics for software architecture adaptability. *In Proceedings of Fifth IEEE International Symposium on Requirements Engineering* (pp.310–311).

Dobson, G., Lock, R., & Sommerville, I. (2005). *QoSOnt: A QoS ontology for service-centric systems*. In Software Engineering and Advanced Applications. 31st EUROMICRO Conference.

Fedosseev, P. (2003/2004). *Composition of Web services and QoS aspects*. Seminar: Data Communication and Distributed Systems. Retrieved from http: //www.nets.rwthaachen. de/ content/ teaching/ seminars/sub/ 2003_2004_ws_docs/ WebServices.pdf

Herssens, C., Jureta, I. J., & Faulkner, S. (2008). Capturing and using QoS relationships to improve service selection. *In Proceeding of CAISE*, France.

Hwang, C.-L., & Yoon, K. (1981). *Multiple attribute decision making: Methods and applications*. Berlin, Germany: Springer.

Innocenti, C. (2007). SOA and the importance of XQuery. *The SOA Magazine, 10.* Retrieved from http: // www.soamag.com/I10/ 0907-3.pdf

Jureta, I. J., Faulkner, S., & Schobbens, P.-Y. (2006). A more expressive soft goal conceptualization for quality requirements analysis. *In Proceedings of 25th International Conference on Conceptual Modeling* (pp. 281-295).

Kaabi, R. S., Souveyet, C., & Rolland, C. (2004). *Eliciting service composition in a goal driven manner*. Int. Conf. on Service Oriented Computing (ICSOC), New York.

Keller, S. E., Kahn, L. G., & Panara, R. B. (1990). Specifying software quality requirements with metrics. In Thayer, R. H., & Dorfman, M. (Eds.), *System and software requirements engineering* (pp. 145–163). Washington, DC: IEEE Computer Society Press.

Ma, W., Liu, L., Xie, H., Zhang, H., & Yin, J. (2009). Preference model driven services selection. In *21st International Conference on Advanced Information Systems Engineering* (CAISE'09), (pp. 216-230).

Mani, A., & Nagarajian, A. (2002). *Web services: Understanding quality of service for Web services*. IBM Developer Works. Retrieved from http:// www-106.ibm.com/ developerworks/ library/ ws-quality.html

Maximilien, E. M., & Singh, M. P. (2004). A framework and ontology for dynamic Web services selection. *IEEE Internet Computing*, *8*(5), 84–93. doi:10.1109/MIC.2004.27

Mylopoulos, J., Chung, L., & Nixon, B. (1992). Representing and using non-functional requirements: A Process-oriented approach. *IEEE Transactions on Software Engineering*, *18*(6), 483–497. doi:10.1109/32.142871

O'Sullivan, J., Edmond, D., & Hofstede, A. T. (2002). What's in a service? Towards accurate description of non-functional service properties. *Distributed and Parallel Databases Journal - Special Issue on E-Services, 12*.

Papazoglou, M.-P., Traverso, P., Dustdar, S., & Leymann, F. (2008). Service-oriented computing: A research roadmap. *Int. J. Cooperative Inf. Syst.*, *17*(2), 223–255. doi:10.1142/ S0218843008001816

Penserini, L., & Mylopoulos, J. (2005). *Design matters for Semantic Web Services*. ITC-IRST, Technical report: T05-04-03.

Rohleder, C., Marhold, C., Salinesi, C., & Doerr, J. (2009). *Quality data model and quality control in the product lifecycle management*. International Conference on Product Lifecycle Management (ICPLM), Bath, Royaume Uni.

Rolland, C., Kirsch-Pinheiro, M., & Souveyet, C. (2009). *An intentional approach to service engineering. Transactions on Service Computing (IEEE-TSC), Special issue on REFS (Requirements Engineering for Services)*. IEEE.

Rolland, C., & Prakash, N. (2000). Bridging the gap between organizational needs and ERP functionality. *Requirement Engineering Journal*, *5*(3), 180–193. doi:10.1007/PL00010350

Simon, H. A. (Ed.). (1981). *The science of the artificial* (2nd ed.). Cambridge, MA: The MIT Process.

Toma, I., Foxvog, D., & Jaeger, M. C. (2006). Modeling QoS characteristics in WSMO. In *Proceeding of the 1st Workshop on Middleware for Service Oriented Computing* (MW4SOC 2006).

Yu, E., Du Bois, P., Dubois, E., & Mylopoulos, J. (1995). From organization models to system requirements- A cooperating agents approach. In *Proc. 3rd Int. Conf. on Cooperative Information Systems (CoopIS-95)*, Vienna, Austria, (pp. 194-204).

Zavadskas, E. K. (1986). The method of ranking of construction-technological alternatives on the basis of the distance from the ideal solution. *In New construction technology for buildings and structures* (pp. 52-57). (in Russian)

Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., & Sheng, Q. Z. (2003).Quality driven web services composition. In *Proceedings of the International WWW*.

KEY TERMS AND DEFINITIONS

Context Models: Informational models representing context information in a well-defined structure.

Context: Any information that can be used to characterize the situation of an entity (a person, place, or object considered as relevant to the interaction between a user and an application).

Context-Aware Services: Services of which description is enriched with context information related to non-functional requirements, describing the service execution environment or its adaptation capabilities.

Context-Aware Systems: Systems that are able to adapt their operations to the current context, aiming at increasing usability and effectiveness by taking into account environmental context.

Service Selection: The process allowing the identification of all services, among the available ones, that match functional and non-functional requirements.

Services: Independent entities, with well-defined interfaces, that can be invoked in a standard way, without requiring the client to have knowledge about how the service actually performs its tasks.

Similarity Measures: Measures used to compare the degree of similarity (or dissimilarity) between two concepts on a domain.