



HAL
open science

Selector: A tool for dynamic service selection and management

Adel Alti, Abdallah Boukerram, Philippe Roose

► **To cite this version:**

Adel Alti, Abdallah Boukerram, Philippe Roose. Selector: A tool for dynamic service selection and management. Journal of Computing, 2012, 4 (4), pp.1-10. hal-00689782

HAL Id: hal-00689782

<https://hal.science/hal-00689782v1>

Submitted on 20 Apr 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.

Selector: A tool for dynamic service selection and management

ALTI Adel, BOUKERRAM Abdallah, ROOSE Philippe

Abstract—This paper presents the tool SELETOR which has been developed for services selection. This paper proposes also a new web management system that allows for considering quality and context-awareness while discovering and selecting web service. The main idea of the proposed web service oriented architecture consists in the ontology Context-aware Quality Semantic Web Service called (CxQWS). At the first step, services are defined as a set of semantic metadata, reflecting service requirements and QoS parameters. At the second step, services with a semantic contextual metadata are elaborated. Such a procedure ensures that the selection decisions should be based on the semantic quality representation of the created services. The tourism services in a mobile environment have a critical role in creating tourist satisfaction. They are neither a uniform group, nor able to give consistently high service quality. Indeed they have significantly different platforms and a variety of heterogeneous service providers which make the management of service qualities complex. This complexity is clearly noticeable when a given service is provided by two or more providers, within the same area of the mobile clients; a decision should be made to select the most appropriate service based on contextual description with the best QoS.

Index Terms—context-aware, selection of services, quality of service, context matching



1 INTRODUCTION

RECENTLY the *World Tourism Organization* emphasized that the future of the tourism industry depends on the ability to instill a culture of mobile communication and quality in tourism services [1, 11]. The tourism services in a mobile environment have a critical role in creating tourist satisfaction; they are neither a uniform group, nor able to give consistently high service quality. Indeed they have significantly different networks environments and a variety of heterogeneous service providers which make the management of service qualities complex. This complexity is clearly noticeable when a given service is provided by two or more providers, within the same area of the mobile clients; a decision should be made to select the most appropriate service based on contextual description with the best quality of service. The identical services have common semantics characteristics (name, city, category), within different providers, versions, QoS and executions contexts.

The heterogeneity of components regarding embedded sensors, CPU power, communication mechanisms (*GPRS, WIFI, Bluetooth, ZigBee, etc.*), speed of transmission as well as the media variety (*sound, video, text and image*) requires taking into account adaptation to a semantic level in order to avoid the ad hoc solutions which are not reusable and/or generalized. Diversity of languages, protocols, and hardware platforms lead to major incompatibility issues. Moreover, multimedia services adaption guided by user and community requirements and preferences are not easy to be performed in this context. Communication

protocols, security, and QoS are other non-functional concerns which present new design challenges.

The application of context-awareness has been demonstrated in a large number of tourism applications [3, 8]. Context aware computing plays an important role in tourism applications to better understand user context and adapt service with the continuous evolution of its context. The web service is an ideal fit in the tourism services, because it allows the service easy integration over heterogeneous systems.

The semantic web service ontology (OWL-S) offer sophisticated capabilities including automated discovery, composition, invocation and monitoring [6]. But OWL-S ontology does not address how to consider non-functional concerns, i.e. how to represent them. Furthermore, OWL-S doesn't really take into account the contextual information in web services quality selection for very limited mobile device like PDA, Smartphone, etc.

In this paper, we present the tool SELECTOR which has been developed for service selection. We present also the ConteXt-aware Quality of Semantic Web Service Ontology (CxQWS) includes support for web service quality control and resources requirements changes. The explicit separation of functional and non-functional concerns is the main concern of the CxOWS aiming to well capturing resources-awareness and controlling quality at semantic level.

The *contribution* of this work is deals with a semantic level which help improving services qualities provided to tourists in a discovery and interaction mobile environment. The goal of the proposed ontology is to provide semantic level facilities that allow the management of quality parameters of heterogeneous services. It provides also a dynamic adaptation of selection services based on context metadata parameters (*user, environment, device,*

-
- Adel Alti is with the Computer Science Department, University of Sétif, Sétif, Algeria, CO 19000.
 - Abdallah Boukerram is with the Computer Science Department, University of Sétif, Sétif, Algeria, CO 19000.
 - Philippe Roose LUIPPA, IUT of Bayonne, France, CO 64600

and service provider context) for the better flexibility, adaptation and customization of mobile client software.

The key idea of the proposed web service oriented architecture consists in the Ontology ConteXt-aware Quality of Semantic Web Service (CxQWS). At the first step, services are defined as set of semantic metadata, which reflect service requirements and QoS parameters. At the second step, services with a semantic contextual metadata are elaborated. Such a procedure ensures that the selection decisions should be based on the semantic quality representation of the created services.

The rest of this paper is organized as follows: Section 2 presents our motivations and other related works. Section 3 details CxQWS ontology, while section 4 presents an overview of the proposed framework. Section 5 presents application scenarios of our ontology as well as implementation issues. Finally, Section 6 concludes and presents some perspectives.

2 MOTIVATIONS AND RELATED WORKS

2.1 Management of Context-aware Web Services Qualities

There is a diversity of contextual information and a variety of quality attributes, we find several ontological languages such as WSMO [19], DAML-S [20], ConteXtML [21], CxBR [22], CWSC4EC [7], CA-Discovery [24] and CxG [23]. These languages provide the means for defining QoS and context in specific application domains such as pervasive and mobile computing. However, neither various communication protocols, not heterogeneous services are discussed. Our approach addresses the limitations of existing approaches, suggesting the mechanisms to integrate new tourism quality attributes of mobile-based application, to provide a dynamic adaptation of selection services based on context metadata parameters (user, environment, device, and service provider context) and the management of the heterogeneity of service needs, of mobile devices capacities and their various communication protocols (*GPRS, WIFI, Bluetooth, etc.*) as well as the media variety (*sound, video, text and image*).

A semantic-based architecture for intelligent destination management system is proposed in [5]. In this architecture, a semantic web ontology is used to model the tourism destinations, user profiles, and the user contexts. The proposed semantic web service ontology (OWL-S) is extended with context information taken into account. This architecture is context-aware but suffered from a semantic-based quality evaluation for achieving an efficient framework.

Boukadi and al. [7] introduced a new context categorization suitable for the enterprise collaboration environment and proposed a context-aware web service ontology for describing and selecting web services enhances dynamicity and agility of collaboration. The proposed approach takes into account only contextual information necessary for web service collaboration, and not considers QoS parameters and their semantics for web services discovery and selection.

Nicoliciu-Georgescu and al [16].exploited ontologies

and ontology-based rules data for improving query response times in data warehouses groups by modifying cache memory allocations. This work ensures a better data warehouse management at a lower cost and enables autonomy by using autonomic loop control. However, this work concentrated only on service performance as indicator of the system quality and not considers other QoS parameters like security and dynamic adaptation.

Wang and al. [17] present a QoS model for service selection. This work is based on QoS ontology, which introduces a number of specific quality attributes and their respective quality measurements used for web service quality evaluation and selection. The authors have presented an interesting approach for annotating service description with QoS data. However, neither various communication protocols, not heterogeneous services are discussed. Therefore, it does not provide support for dealing with the dynamic service selection in order to adapt to highly varying context conditions. Moreover, proposed intelligent mechanisms (e.g. intelligent software agents) and techniques can be exploited in the practice of a new tourism quality management policy [13].

Currently, the semantic web service ontology (OWL-S) does not include a semantic description of contextual information. Thus, in its current state, it does not enable web business quality selection in pervasives computing environments. Moreover, proposed intelligent mechanisms (e.g. intelligent software agents) and techniques can be exploited in the practice of a new QoS policy [13].

Recently, Kerses and al. [4] proposed a model that makes an automatic selection of best service provider that is based on mixed context and QoS ontology for a given set of parameters of QoS. It approach provides interesting benefits for QoS enhancement, allows a well interoperability between heterogeneous services, but disagree in the automatic service selection using multi-dimensional QoS and did not address any issue related to service integration in the limited portable devices.

MMSA (*MultiMedia Software Architecture*) [26] is an approach based on multimedia software architectures. It proposes a generic solution to the problem of incompatibility of services, in term of media type to another, or of a media format to another format. This work treats syntactic interoperability between the heterogeneous services and did not take into account the semantics of services and that of the location, category, QoS and context constraint and providers.

More recently, Lamolle and al. [9] introduced an ontology framework for automatic networked multimedia systems deployment and configuration. Based on well-known standards, this framework integrates multimedia, distribution and platform concepts. This work offering excellent quality service management and deployment but disagree in the complex semantic context matching.

The methods and contributions of this paper are different from above related works. Firstly, the paper exploits services selection of mobile devices for maximizing user QoS under resource constraint in mobile applications. Secondly, the paper proposes an intelligent business quality management system, which is based on the

Context-aware Quality of Semantic Web Service Ontology (CxQWS). The success of this approach provides its members intelligent access to business quality relatively inexpensively. This approach provides improved performance, a dynamic adaptation of selection services based on context metadata parameters (user, environment, device, and service provider context) for the better flexibility. Above two contributions don't appear in other related works.

2.2 Uses and Reasons of Ontology Importance

To investigate the importance of ontology from the technical view, for four reasons:

1. *Homogenization*: describing different contextual tourism information and different qualities tourism service using the same ontology confers to the ontology a role of homogenization for at least the service quality and the context information that it describes [12, 15].
2. *Interoperability*: ontologies provide a good way to treat semantic interoperability between the heterogeneous representations of web services and thus QoS parameters and context parameters of tourism services by sharing common representation and semantics [13, 14].
3. *Expressiveness*: most ontology languages such as RDF and OWL-S with high expressiveness are efficient and perform much better for expressing semantics of various portable devices properties (PDA, smart phone, iPods and Palms) and various QoS parameters.
4. *Easy service discovery and selection*: ontologies make the discovery and the selection of web business quality services easier, faster and automatic. The main mechanism for service discovery is *service registry* and *semantics* can be used for the discovery of web business quality service registries and for the management of QoS parameters and context parameters. This is for a better service choice among several equivalent functionalities [13].

2.3 Objectives and Motivations

Currently, the web service was described with XML standards languages such as semantic web service ontology (OWL-S) that facilitates the web service discovery and selection. However, contextual information and mobility awareness is not considered by OWL-S. This results to basic drawbacks:

- The poor service quality results from employing syntactic and semantic description without semantic-based context technologies and semantic-based mobility information,
- Difficulty in meeting service quality such as reliability and efficiency,
- The quality management applications for tourism business depend on their hardware and software platforms.

These disadvantages can be tided, if we extend OWL-S for matching user requirement with tourism destination specifications at the semantic level, with context informa-

tion and QoS parameters both takes into account. Thus an efficient mechanism is provided making the web service of mobile application/system easy to discover and associate with the semantic information contained in the ontology. Our proposed ontology introduces semantics related to the web business quality and also semantics related to the context-aware service which is defined as set of semantic metadata (see Figure 1). Due to heterogeneous services as well as its various communication protocols (GSM, 3G, Bluetooth, etc.) can be specified more easily using a semantic representation to better support resource-awareness. The automated processing of web business qualities in mobile-based applications requires that the user should be able to locate a business quality through its meaning independent of which service context is evolved. In our case, we refer by context-aware QoS modelling the semantic web service ontology that will be used for matching user quality requirements with tourism destination specifications at the semantic level, with context information of heterogeneous resources taken into account. Since, there is a diversity of quality attributes with a variety of contextual information, we find several contextual languages such as WSMO[19], CxS[18] (*Contextual Schema*), DAML-S [20], ContextML [21], CxBR [22] (*Context-Based Reasoning*), CWSC4EC [7] (*Context Web Service and Community Enterprise Collaboration*), CA-Discovery (*Context-awareness in Context Discovery*) [24], and CxG [23] (*Contextual Graphs*). These languages provide the means for defining QoS and context in specific application domains such as pervasive and mobile computing. However, neither various communication protocols, not heterogeneous services are discussed. Moreover, mobility introduces higher degree of heterogeneity and dynamicity than traditional distributed systems. The semantic quality service discovery and selection becomes essential in order to adapt to highly varying context conditions.

3 CxQWS: CONTEXT-AWARE QUALITY WEB SERVICE ONTOLOGY

The user should be able to locate a service quality through its meaning, independent of what the service quality is called, which language is used, which service context is evolved, and who the service provider is. Not only the QoS of different services need to be compatible, but also the devices performing the service execution need to respect certain properties and constraints. The resource properties and constraints nature make service qualities management and sharing an even more difficult.

To exploit the context-aware service quality to their full potential, we need ontologies to describe their semantics. Any web service can be based on our proposed ontology Context-aware Quality Web Service Ontology (CxQWS). Our ontology aiming to well capturing resources-awareness and controlling quality of services at semantic level, to well share their devices properties and constraints, to provide a semantic service selection facilities, to adapt and personalize a service to a device's capabilities and to user preferences.

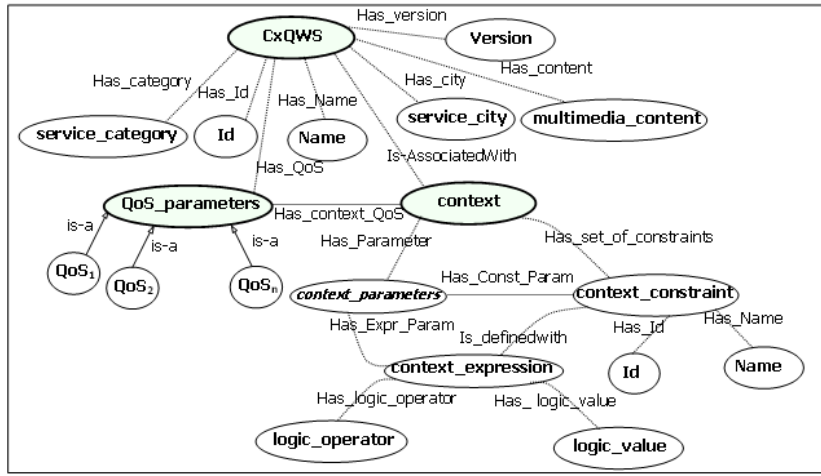


Fig. 1. The structure of CxQWS meta-ontology.

Based on the previously mentioned works, the quality of any mobile-based application can be represented on our ontology CxQWS, which is defined as set of metadata parameters. These QoS parameters are: 1) - customer care and attention, 2) - professional level of staff, 3) - presentation of food and drink, 4) - quality of food and drink, 5) - safety and security, 6) - level of service, 7) - speed and efficiency of service, 8) - quality of facilities, 9) - variety of facilities, 10) - tourist information, 11) - accessories in bedrooms, 12) - decoration, 13) - availability of breakfast, dinner and room services, 14) - continuity of service, 15) - durability of service. Figure 1 presents the relevant classes of our meta-ontology. It allows the services providers to publish metadata about their services within their context constraints in the UDDI registry. Providers would instantiate the classes from the ontology and publish the resulting individuals as OWL files on their web sites. The CxQWS define common concepts as well as relationships among those concepts to represent QoS parameters and context constraints. CxQWS distinguishes the following concepts:

- service has a name, a version and unique ID.
- *service_category* = {hotels, hostels, motels, restaurants, cafés, camping, bungalows}.
- *service_city*: the city which the service facilities are located.
- *QoS_parameters*: the list of the QoS parameters. This class includes the above fifty basic subclasses. QoS parameters concern not only the services but also the device resources where the execution is taking place.
- *multimedia_content* = {text, images, graphic, sound, video}.
- *context* = {service provider, user, device, environment}
- *context_parameters*: Each context category has specific context parameters. For example, the device-related context is a collection of parameters (memory size, CPU power, bandwidth, battery lifecycle, etc). Some context parameters may use semantically related terms, e.g. CPU power, CPU speed.
- *context_constraint*: consists of simple or composite

context expression. For example a context constraint can be *bandwidth=10000* or *memory_size='less'*.

- *context_expression*: denotes an expression which consists of context parameter, logic operator and logic value.

The quality services selection must take the capabilities and limitations of a mobile device into account as well as any user requirements or preferences with respect to the service. Context is any information can be collected from service (name, version, resources needs), from device (resources capacities), from environment (time, location) and from user (preferences). CxQWS uses these informations to select a service quality in a dynamic environment, by confronting a required context (user, service) for a provided context (mobile node). Figure 2 shows the basic subclasses of the class *context* of CxQWS ontology.

- *context_environment*: The context environment representing available information characterizing network connections, time, date and location constraints. Network connections connect hardware components (PDAs, smart phone) having a limited bandwidth. Due to heterogeneous services as well as its various communication protocols (GSM, 3G, Bluetooth, etc.) can be specified more easily using a semantic representation to better support resource-awareness. The location, the date and the time represent respectively the geographical location within the exactly period of time that the service opportunity should be accomplished.

Context_environment: {Time, Location, Date, Network}

- *Network* = {type, protocol, bandwidth}
- *type* = {wired | wireless}
- *protocol* = {GSM | 3G | WLAN | Bluetooth}
- *bandwidth* = {128K | 256K | 512K | 256K | 256K | 1M}
- *context_device*: The hardware components are mobile devices like PDAs or smart phone, are constrained in their resources (memory size, CPU

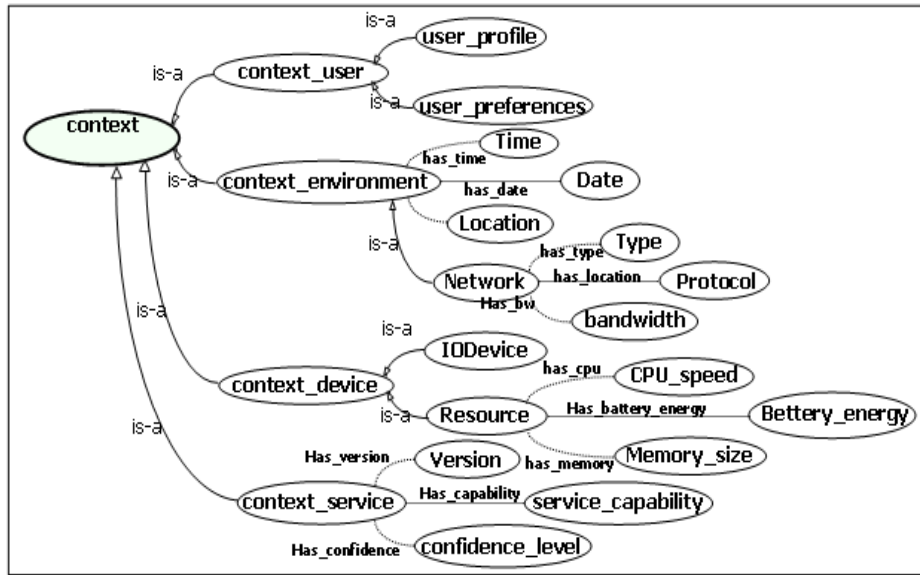


Fig. 2 Context types ontology.

speed, battery energy, etc) and act as execution environment for web service. A resource-awareness about current usage of processing power, memory, etc. is a prerequisite to guarantee a minimum QoS.

- *context_service*: Service capability describes the operational features of a web service. Service capability includes semantic attributes like pre-conditions, effects and post-condition. The service confidence level is very interesting in the case of having two web services providing the same functionalities. They may have varying values for their performance quality attributes.

4 OVERVIEW OF THE FRAMEWORK

The proposal framework is a generic, dynamic and interpretable system regarding the management of web tourist service qualities. The framework facilitates semantic discovery and interoperability of web services that manage and deliver parts of tourist service qualities (for future reservation). It offers two operations. The first is the find web services qualities. This takes as input the user query, then returns list of candidate web services from the web service registry. The second is to allow providers to publish metadata about their services dynamically and to define their own specializations of the default classes based on the ontology CxQWS as illustrated in Figure 1.

Our framework for the context-aware quality web service discovery, as shown in Figure 3, is a three layers architecture consisting of top-level layer dedicated to the semantic query layer. The middle layer consists in three basic agents which are the Quality Semantic Service Discovery Agent (QSDA), the Quality Semantic Context Evaluator Agent (QCEA) and the Quality Semantic Service Sensor Agent (QSSA). The basic layer is a group of web services which the sequel consumes the local web

services exposed by every web services provider offering services qualities and contexts descriptions. The discovery and selection process is a 3-step process:

Step.1 The mobile client has no previous knowledge about the discovery services. In addition, when a given service is provided by two or more providers within the same area of the mobile clients; a decision should be made to select the most appropriate service based on contextual description with a better quality of service. Users submit queries through the mobile client application interface. As soon as a user describes the service qualities queries, the QSDA initiate the web service discovery mechanism based on the functional constraints.

Step.2 The QSDA responsible to communicate with the corresponding web services in order to initiate the query and finally to present personalized results. This is made thanks to semantic-based matching discovery which the QSDA could exploit the categorization of the ontology hierarchy to find suitable matches. The QSDA captures

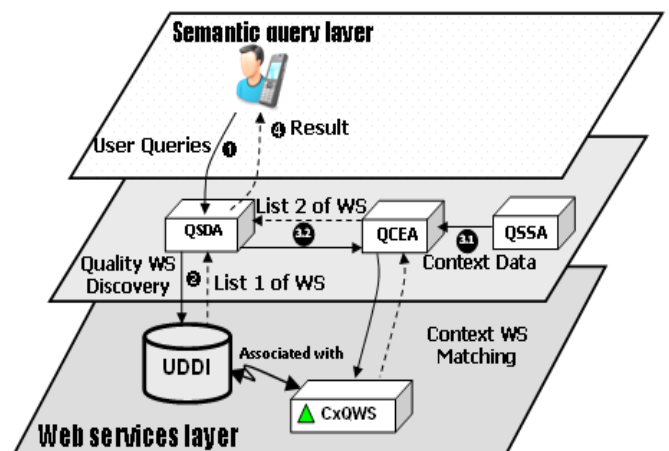


Fig. 3 Discovery framework architecture.

user's goals from user's queries which express a user's intention that s/he expects to get the results from the framework. Each goal represents functionalities and non-functionalities that a particular business domain offers. The information in goal is semantically described in our ontology by the service properties (*service_category*, *service_city*, *QoS parameters*, *multimedia_content*). Each goal can be expressed as an AND-OR of simple expression. The QSDA identify web services which matches functional requirements and QoS parameters of user's queries. A list of matched web services is passed to the QCEA for further selections.

Step.3 Depending on the resources availability on the mobile device, on the environment context and on the service provider context, the function of QCEA is to select a better service from the primary list identified in the previous step within the equivalent functionalities.

Step.3.1 The QSSA is validated to provide collections of context metadata (*context metadata*) based on the ontology as illustrated in Figure 1. In case of changes in the client context (i.e. less available memory, less processor power) or in the environment context (i.e. less bandwidth), the QSSA will be notified by a set of contexts metadata in order to dynamically skip the next selected service of the primary list of services offering the same service functionalities which requires less available resources.

Step.3.2. This step consists on matching context metadata of each service of the primary list of Step 1 with the device context parameters from the QSSA. The service selection are based on inference techniques and semantic interpretation to (1) detect context semantic matching between two services contexts based on their semantic representation (*context type*, *context parameters* and *logical operators*), (2) evaluate the resulting matching and (3) automatically add a service in the list of selected web services. In [10], this comparison uses a simple syntactic comparison of web services policies but our top level ontology resolves the semantic and syntactic heterogeneities that resulted from the different representation languages and service providers. This step returns a list of web services which provide a better service quality to a specific mobile client. Table 1 details the algorithm for the dynamic service selection described above.

5 APPLICATION SCENARIOS

An adventurous tourist is planning a backpacking trip and looking for new places to explore. His Smart phone with integrated GPS has become indispensable for finding nearby activities that match his interests and is handy for finding cheap accommodations. To facilitate quality service discovery, CxQWS ontology on the Smart phone provides a semantic markup of the user profile, the Smart phone context, and the link to the CxQWS ontology. Tourists want to find service qualities, in which their quality's profiles are expressed by a certain media types (e.g. video, image, audio, text). For example, tourists are interested for service quality profiles that are represented by image and graphics, while viewing impaired visitors

TABLE 1
ALGORITHM FOR DYNMAIC SERVICE SELECTION

Dynamic Service Selection Algorithm
Input Goal = { $g_1, g_2, g_3, \dots, g_n$ } // a set of user's goals
Service = { s_1, s_2, \dots, s_n } // a set of services
Output
MatchedService_List1 $\leftarrow \phi$ // a set of services that meet the user's goal
MatchedService_List2 $\leftarrow \phi$ // a set of services that meet the user's goal and sensor context constraints
Begin
// match user's goal with services
for each s in Service {
 for each g in Goal {
 if match_service_category(g, s) **AND** match_service_city(g, s) **AND**
 AND match_multimedia_content(g, s) match_service_QoS_parameters (g, s) **then**
 { MatchedService_List1.add(s); }
 }
} // match context constraints of Quality Semantic Service Sensor Agent (QSSA) with selected service constraints
if MatchedService_List1.size != 0 {
 Context sensor_cx = sensor_getContext (); // return a set of contexts constraints
 for each s in MatchedService_List1 { // return context description of service
 Context s_cx = service_getContext (service ID, UDDI);
 for each cx in sensor_cx {
 if match_context_type(cx, s_cx) **AND** match_context_constraint(cx, s_cx) **then**
 { MatchedService_List2.add(s); }
 }
 }
} }
return MatchedService_List2;
End;

are interested for service quality which represented by audio. The above requirement can be satisfied by using the semantic metadata *multimedia_content*.

First scenario: A first simple scenario, tourists having specific requirements want to find service qualities with combined criteria. For example, visitors want to find restaurants map, located in Setif, with high speed and efficiency of service, as well as statutory obligations. This query can be answered by using the following semantic metadata: *service_category*, *service_city*, *speed_and_efficiency_of_service* and *statuary_obligations_of_service*. Our selection tool (**Selector Tool**) allows tourists to specify the type of service, its location and different quality of services parameters; the type of context is its properties (e.g. throughput and memory size). This query can be submitted as follows:

(service_category="Restaurant") **AND**
(multimedia_content="Image") **AND** (service_city="Setif")
AND (speed_and_efficiency_of_service = "high") **AND**
(statuary_obligations_of_service = "True")

Second scenario: A second scenario, tourists want to find services all services on the Smart phone, that semantically matches the Smart phone context (i.e. the less available memory) and the environment context (i.e. the less bandwidth). The context constraint can be:

(context_type="Client") **AND** (memory_size="low") **AND**
(context_type="Environment") **AND** (bandwidth="128K")

Third scenario: when he asked for the restaurant map, the Quality Semantic Service Discovery Agent returned two services: a *black and white map* and a *coloured map*. To limit the search results when browsing the restaurant map on the Smart phone, only the first one is returned

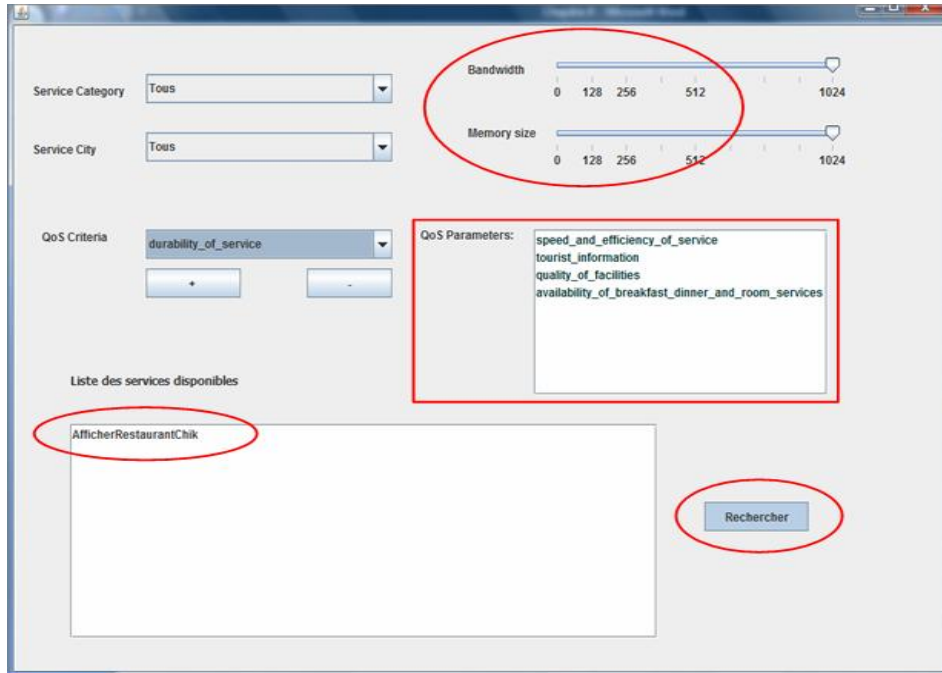


Fig. 4. Research results for the third scenario.

that semantically matches the Smart phone context (i.e. the less available memory) and the environment context (i.e. the less bandwidth). The context constraint can be:

```
(service_category = "Restaurant") AND
(multimedia_content = "Image") AND (service_city = "Setif")
(speed_and_efficiency_of_service = "high") AND (statu-
ary_obligations_of_service = "True") AND
(context_type = "Client") AND (memory_size = "low") AND
(context_type = "Environment") AND (bandwidth = "128K")
```

5.1 Implementation of CxQWS

We have utilized Protégé [2] to build the CxQWS ontology using RDF/S (Resource Description Framework) language in which each statement consists of a triplet (subject, predicate and object). In particular we have used the RDF schema to describe our ontology concepts and its relationships. We briefly illustrate selected features of implementation of our ontology through snippets of the RDF code. Fig.5 illustrates our ontology.

5.2 Intelligent Software Agents in the Context-aware Quality Semantic Service

Soon, providers of web services will advertise their services and contexts constraints on the semantic web, so that intelligent software agents can find and adapt them dynamically. For, these agents (see Figure 3), the semantic web infrastructure will be based on the ontology as illustrated in Figure 1. This ontology would be published as OWL file. The ontology CxQWS would allow providers to instantiate the classes from the ontology and publish the resulting individuals as OWL files on their web services. Then, a semantic web service specialized in enter-

TABLE 2
CxQWS ONTOLOGY USING RDF

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://univ-
setif.dz/schemas/context_aware_quality_service_ontology">
  <rdfs:Class rdf:ID="QoS_parameters"/>
  <rdfs:Property rdf:ID="Service_id"/>
  <rdfs:Property rdf:ID="Service_category"/>
  <rdfs:Property rdf:ID="Service_city"/>
  ...
  <rdfs:Property rdf:ID="Has_QoS">
    <rdfs:range rdf:resource="#QoS_parameters"/>
    <rdfs:domain rdf:resource="#Service"/>
  </rdfs:Property >
  <rdfs:Property rdf:ID="Is_Associatedwith">
    <rdfs:range rdf:resource="#Context"/>
    <rdfs:domain rdf:resource="#Service"/>
  </rdfs:Property >
  <rdfs:Class rdf:ID="QoS_parameters"/>
  <rdfs:Property rdf:ID="Service_city"/>
  <rdfs:Property rdf:ID="QoS_1"/>
  <rdfs:Property rdf:ID="QoS_2"/>
  ...
  <rdfs:domain rdf:resource="#QoS_values"/>
  <rdfs:domain rdf:resource="#xsd:string"/>
  <rdfs:Class rdf:ID="Context"/>
  ...
  <rdfs:Class rdf:ID="ressource">
    <rdfs:subClassOf rdf:resource="#Context_Device"/>
    <rdfs:Property rdf:ID="Memory_size"/>
    <rdfs:Property rdf:ID="CPU_speed"/>
    <rdfs:Property rdf:ID="Battery_energy"/>
```

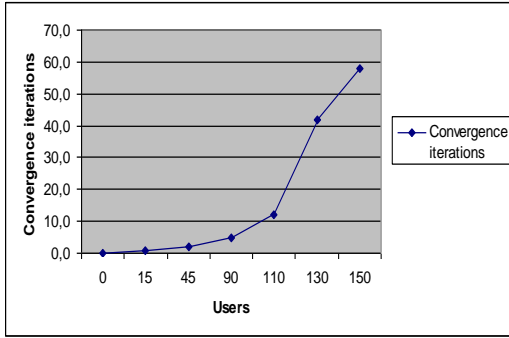



Fig. 5. Convergence iterations under various device arrival rate.

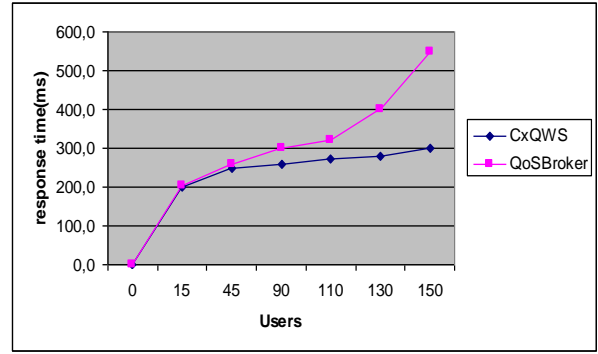


Fig. 6. response time under various device mobiles

tainment context could send out a crawler agent to collect the available activities.

If a user then asks for a specific type of *restaurant* for mobile limited device (low bandwidth, less memory, etc.), the software agent could exploit the categorization of the ontology hierarchy to find suitable matches, and call auxiliary web services.

Providers of web services cannot only publish their metadata dynamically, but can also define their own specializations of the default classes. For example, an ontology module could define *chic_restaurant* as subclass of *restaurant*, and put semantic restrictions on these classes to describe its characteristics. Then, if a software agent searches for restaurant facilities it would also find the instances of the subtype, and also learn that *chic restaurants* are more expensive than usual restaurant, because they involve the ability for guests to sing etc. When a *chic restaurant* service delivers a high density color image or video sequence, a notification was sent to the QSSA telling that there is not enough available memory to continue storing such an image. In this case, the QCEA will skip to the next available service offering the same image with requires less memory (due to different internal representation technique). Moreover, proposed intelligent software agents (i.e. QSDA, QCEA and QSSA agents) can be exploited in the practice of a new tourism quality management policy [13].

5.3 Experiments Framework

In this section, we evaluate the performance of dynamic mobile selection algorithms (SSA) for querying semantic web services into context-aware applications using our Java VM simulator. The experiments are conducted to compare our CxQWS selection algorithms (SCA) for querying of semantic web services into context-aware applications with QoS Broker algorithm [25], which is quality and distributed based service selection model for mobile-based application. The reason for choosing reference [25] as the comparison is that both our work and reference [25] provide a various selection policies based selection algorithm for mobile environment.

We simulate a tourist case study (second scenario) for two approaches using our Java VM simulator and we have varied the user (and respectively, the mobile device) from 1 to 150. The users use the system as modeled by a Poisson-process with an arrival rate of 10 per hour. There are

two communication protocols between the server and the mobile client, either a GPS link, which has a rather a low bandwidth, 3G link, which has a higher bandwidth. The simulation took not more than approximately 10 seconds on a 3.5Ghz Pentium PC. In the simulator, different agents are used QSDA, QCEA, and QSSA. The framework dynamic service selection starts a listening thread that listens to the requests from users. It receives the requests of users and puts them into the queue. While the queue is not empty, the QSDA starts the dynamic service selection algorithm to find the right match.

5.4 Analysis and Discussion

The first experiment is to test convergence of mobile services selection algorithms (SSA) which includes quality service semantic level algorithm and resources-awareness semantic level algorithm. In Fig.4, when user arrival number is low ($n=15$), the convergence iteration of SSA is one. When user arrival rate increases to 45, the convergence iterations of SSA increase to two. In task intensive case ($n = 90$), there are more request waiting for a match than the complete responses. When user arrival rate increases quickly, the match of the services will increase, and some users will not afford requests services.

The second experiment is to test performance of our CxQWS with QoS Broker. Figure 5 shows the evaluation results, meaning that our approach turns out to be the best. When user arrival rate is small, there are enough resources (e.g. network bandwidth) for mobile users to select efficient service, and obtain better response times. Compared with QoS Broker, the response times of SCA increase slowly than QoS Broker when the user arrival rate increases. This result is practically significant as well related to the low adaptation effort e.g. takes into account all the heterogeneous services as its various communication protocols (GSM, 3G, Bluetooth, etc.) as consequences of self-selection for environment evolution (e.g. network bandwidth) guided by the adaptation policies.

6 CONCLUSION AND FUTURE WORKS

The paper proposed CxQWS ontology being an extension to the web service, allows for considering quality and resource-awareness while discovering and selecting web service. It allows web services to share their media resources using semantic representation to the end-user.

Using quality and context parameters for describing and selecting web services enhances dynamicity and transparency to mobile applications. The paper proposed also an intelligent context-aware quality service management framework. This framework allows web services to inter-operate both with internal and external applications using the semantic quality and context matching techniques. The idea of presented framework consists of two steps. At the first step, services are defined as set of semantic metadata, which reflect service requirements and QoS parameters. At the second step, services with a semantic contextual metadata are elaborated. Such a procedure ensures that the selection decisions should be based on the semantic quality representation of the created services. We presented application scenarios of our ontology. As a result, Therefore, CxQWS provides a great flexibility and enables intelligent adaptation and customization of mobile client software. In this paper we have presented tool for dynamic service selection and management (Selector) in order to explore service space at a semantic level. The most innovative characteristic of the tool is that it profits from the potential of ontological space (i.e. CxQWS) to achieve semantic quality and context-awareness while discovering and selecting web service. For the near future, our tool will be evaluated on large-size systems with industrial partners and we plan to include refinements on the ontology proposed for the representation of the context-aware services qualities.

ACKNOWLEDGMENT

We would like to thank our colleagues and students for testing SELECTOR tool and specially Hamza Reffad and Roumili. The authors appreciate the in-depth comments given by the anonymous reviewers to improve this work.

REFERENCES

- [1] A. Pashtan, R. Blattler, A. Heusser, P. Scheuermann, "CATIS: A Context-Aware Tourist Information System," *Proceedings of the 4th International Workshop of Mobile Computing, Rostock*, June 2003.
- [2] The Protégé Ontology Editor and Knowledge Acquisition System, <http://protege.stanford.edu/>
- [3] Abowd, G., "Cyberguide: a Mobile Context-Aware Tour Guide", *Wireless Networks*, pp. 421-433, 1997.
- [4] Keskes, N., Lehierche, A., and Rahmoun, A., "Web Services Selection Based on Context Ontology and Quality of Services," *International Arab Journal of e-Technology*, Vol. 1, No.3, January 2010.
- [5] Kanellopoulos, D., Kotsiantis, S., " A semantic-based Architecture for intelligent Destination Management Systems," *International Journal of Soft Computing*, Vol. 2, No. 1, pp. 61-68, 2007.
- [6] McIlraith, S., Martin, D., "Bringing semantics to web services," *IEEE intelligent Systems*, Vol.18, n No. 1, pp. 90-93, 2003.
- [7] Boukadi, K., C. Ghedira, S. Chaari, L. Vincent, E. Bataineh, "CWSC4EC: How to Employ Context, Webn Service, and Community in Enterprise Collaboration," *Proceedings of 8th International Conference on New Technologies in Distributed Systems (NOTERE'08)*, Vol.1, pp.22 – 33, Lyon, France, 2008.
- [8] Pashtan, A., Heusser, A., and Sheuermuem, P., "Personnel Service Areas For Mobile Web Applications", *IEEE Internet Computing*, Vol. 8, No. 6, pp. 34 – 39, 2004.
- [9] Lamolle, M., Gomez, J., Exposito, E., "MODA : une architecture multimédia dirigée par les ontologies pour des systèmes multimédia en réseau," *Proceedings of 4ème Conférence francophone sur les Architectures Logicielles (CAL'2010)*, pp. 137–151, Pau, France, 2010.
- [10] Verma, K., Akkiraju, R., and Goodwin, R., "Semantic Matching of Web Services Policies," *In 2nd International Workshop on Semantic and Dynamic Web Process (SDWP'2005)*, 2005
- [11] WTO Quality Support Committee, "Quality in Tourism, A Conceptual Framework," May 2003, Varadero, Cuba.
- [12] Mizoguchi, R., "Ontology Engineering Environments," *Handbook of Ontologies*, Berlin, Springer, pp.275 – 298, 2004.
- [13] Handler, J., "Agents and the Semantic Web," *IEEE Intelligent Systems*, Vol. 16, No. 2, pp. 30 – 37, 2007.
- [14] Mao, M., Peng, Y., Spring, M., "An adaptive ontology mapping approach with neural network based constraint satisfaction," *Journal of Web Semantics*, Vol. 8. N°.1, pp. 14-25, 2010.
- [15] J.A. Gargentilla, A. Gómèz-Pérez, "A survey on ontology-based applications: e-commerce and knowledge management will deserve special attention," *OntoWeb Deliverable 1.6.*, 2004.
- [16] Nicoliciu-Georgescu, V., Benatier, V., Lehn R., Briand H. ,"An Ontology-Based Autonomic System for Improving Data Warehouse Performances," *Proceedings of 13th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES 2009)*, pp. 262–269, 2009.
- [17] Wang, X., Vivtar, T., Kerrigan, M., Toma, I., "A QoS-aware Selection Model for Semantic Web Services," *Proceedings of 4th International Conference on Service Oriented Computing (ICSOC'06)*, Lecture Notes in Computer Science Springer 4294, pp. 390–401, 2006.
- [18] Turner, R.M., "Context-mediated representation of intelligent agents," *Human-Computer studies*, Vol. 48, pp. 307–330, 1998.
- [19] WSMO., "Web Service Modeling Ontology.," http://www.wsmo.org/TR/d2/v1.2/20050413/D2v1-2_20050414.pdf.
- [20] Horrocks, I., "CDAML+OIL : a Description Logic for the Semantic Web," *IEEE Data Engineering Bulletin*, Vol. 25, pp. 4–9, March 2002.
- [21] Rayn, N., "ConteXtML : Exchanging Contextual Information between a Mobile Client and the FieldNote Server.," <http://www.cs.kent.ac.uk/projects/mobicomp/fnc/ConteXtML.html>.
- [22] Gonzlaez, A.J., and Ahlers, R., "Context-based representation of intelligent behavior in training simulations," *International Transactions of the Society for Computer Simulation*, pp. 153-166, 1999.
- [23] Brezillon, P., "Context-based modeling of operators Practices bu Contextual Graphs," *Proceedings of 14th Mini Euro Conference in Humman Contered Processes*, 2003.
- [24] Suraci, V., Silvano, M., Massimiliano, T., "Context-awareness in Content Discovery architectures," *STreaming Day 2007*.
- [25] Badidi, E., Esmahi, L., Serhani, M.A., A Qureying Model for Service Selection of Multi-classes QoS-aware Web Services, pp. 17-24, 2004.
- [26] Derdour, M., Ghoualmi Zine, N., Roose, P., Dalmau, M., Altı, A., "UML-Profile for Multimedia Software Architectures", *International Journal of Multimedia Intelligence and Security (IJMIS)* Inderscience Publishers, Vol. 1, No. 3, pp. 209-231, ISSN : 2042-3462 2010.

Adel Alti obtained the Master degree from the University of Setif (UFAS), Algeria, in 1998. He is holding a Ph.D. degree in software engineering from UFAS university of Sétif, Algeria, 2011. Right now he is an associate professor at University of Sétif. He is a member of the research group LRSD. His area of interests includes automated software engineering, mapping multimedia concepts into UML, semantic integration of architectural description into MDA platforms, context-aware quality software architectures and automated service management, Context and QoS. During his work he has published number of publications concerning these subjects.

Abdallah Boukerram is an associate professor at university of Setif (UFAS), Algeria. He obtained his PhD degree in computer science from university of L. Pasteur, Strasbourg, France, 1991. He is interesting in Software Parallel Architecture, and Grid Computing. He is the President of the scientific comity of Computer Science Department (UFAS), where he is supervising many Master and PhD students. He is also the head of a research group in the Computer Science Department.

Roose Philippe is an associate professor at the LIUPPA/UPPA – FRANCE. He obtained his PhD degree in computer science from university of Bayonne, France, 2001. He head of the MOJITO and AEXIUM projects. His research interests are software architecture and platforms, pervasif and ubiquitous computing, mobility, software components services, context and QoS, multi-parts profiles. He is the co-author of three books on software component technologies.