2013 IEEE 10th International Conference on Ubiquitous Intelligence & Computing and 2013 IEEE 10th International Conference on Autonomic & Trusted Computing

Adaptable Web Service Registry for Publishing Profile Annotation Description

Chiraz El Hog
MIRACL, ISIMS, Cité El Ons
Route de Tunis Km 10
Sakiet Ezziet 3021, Sfax, Tunisia
Email: elhog.chiraz@gmail.com

Raoudha Ben Djemaa
MIRACL, ISIMS, Cité El Ons
Route de Tunis Km 10
Sakiet Ezziet 3021, Sfax, Tunisia
Email: Raoudha.Bendjemaa@isimsf.rnu.tn

Ikram Amous
MIRACL, ISIMS, Cité El Ons
Route de Tunis Km 10
Sakiet Ezziet 3021, Sfax, Tunisia
Email: Ikram.Amous@isecs.rnu.tn

***Abstract*—Web services are described via an XML-based language called Web Service Description Language (WSDL). They are then discovered through registries and repositories built upon a standard called Universal Description, Discovery and Integration (UDDI). Due to this registries worldwide businesses and services can be listed and dynamically located on the Internet. Web Service users have heterogeneous platforms and widely different profiles and needs, thus a Web service should be adaptable to the multiple users profiles. In our previous works we introduced a modeling language for adaptive Web services AWS-UML as well as a description language called AWS-WSDL. Nevertheless, the registry should be able to publish adaptable services and offers discovery functionalities upon context criteria. In this paper, we introduce a new registry that provides publishing adaptable Web services (AWS). Our registry allows to publish and to find Web services not only through the functional criterion but also through users contexts and profiles.**

***Keywords*-Adaptable Web Service; UDDI; WSDL;**

I. Introduction

Web services are rapidly gaining acceptance as a fundamental technology in Web fields. They are becoming the cutting edge of communication between various devices over the Web. Dealing with Web services concentrates not only on interoperability but also recognizes three fundamental actions: description using the standard WSDL [11] , discovery through UDDI [10]registries and invocation using SOAP [12] messages.

Those current technologies are limited to functional point of view, they do not deal with issues related to adaptable Web services. In fact, UDDI is a database that allows service requesters to find a Web service that fulfills their functional need regardless their working environment and context constraints. Moreover, WSDL does not support non functional properties mainly user profile characteristics in the service description.

In order to be more efficient and up to date with the spread of new ubiquitous sensors and hand held devices, Web service technology should be easily adapted to the specific user profile. Hence, worldwide users have different contexts i.e. multiple used devises (laptop, Smartphone, PDA ...), different connections and locations, and have also various preferences (in term of content and presentation). A Web service should be empowered to be self-adaptive to the user profile that means it should be able to adjust its behavior according to the user's rights, preferences, context and interest without any explicit intervention.

Nevertheless, most of the existing Web services are not originally developed to be adaptable. Essentially, tools and mechanisms are required to describe and publish adaptable Web service. In fact, current web service standards WSDL, UDDI and SOAP are limited to functional view and do not support profile properties. A number of efforts have been made to deal with non-functional characteristics. MVWSDL (Multiview WSDL) [4] gives an enrichment of the description file by including the user's access rights to the service functionality. [5] presented extensions of WSDL and UDDI with version information. Eventhough previous solutions take into account non-functional properties of a Web service, they are limited to QoS and versioning. Moreover, most of researches focus only on the service description level, they did not deal with the publishing issue of the extended descriptions.

In the following, we propose a UDDI extension that supports adaptable Web service (AWS) based on profile description annotations. We propose a new registry called Adaptable Web Service Registry (AWSR) that publishes adaptable web services and allows users to select services according to their profiles. Our current work is built on our earlier research work around the Adaptable Web Service Description Language (AWS-WSDL) [2] and [3]. Due to AWS-WSDL, we are able to describe the functional as well as non functional properties related to the user's profile in the same description file.

The remainder of this paper is structured as follows. Section II gives an overview of the AWS-WSDL language. Section III is interested on the registry extension. A comparison of Web service publishing on the standard UDDI registry and publishing in the extended registry AWSR is carried out through a case study on section IV. Section V discusses some related works. Finally, section VI summarizes the main results and gives some directions for ongoing and future work.

978-1-4799-2481-3/13 $31.00 © 2013 IEEE
DOI 10.1109/UIC-ATC.2013.53

II. BACKGROUND

The standard Web Service architecture is based on the interactions between three components: service provider, service registry (the broker) for storing service descriptions, and service requestor (the consumer). These interactions based on Web service standards do not support the adaptation notion related to the Web Service requester profile. Therefore, current standards must evolve in order to be able to consider supported profiles together with service functionality. This evolution have to be suitably managed in the whole steps of a Web service life cycle mainly the description and publication ones. To overcame the standard architecture limitation, we proposed in [3] an Adaptable Web Service Archirecture (AWSA) enhanced with suitable extentions. We concentrate in this section in giving a brief overview of the AWS-WSDL description language and in showing the relationship between WSDL and UDDI standards.

A. Overview of AWS-WSDL

The Web service description is provided by the standard WSDL. The description contains essentially a list of features offered by the service and links invocation. This is a contract between the provider and the service consumer. Service provider and service consumer can communicate with the help of the standard WSDL structured as shown in figure 1. However, this standard is insufficient to introduce the adaptation criteria supported by a web service. In addition, it does not differentiate access rights to service features. It describes only functional aspects. Operations and request are treated in a similar way for all user profiles. This marks the limits of standard WSDL. In order to fulfill provider and consumer expectation, we suggest additional information in the description file making possible to deal with context and preferences.

We therefore proposed in [3] an extension of the standard WSDL named *AWS-WSDL* and based on the OMG specification of the MDA architecture. We gathered in the same document; the description of functional elements (operations, messages, types ...) and non-functional ones (profile information). Figure 2 illustrates the AWS-WSDL document structure.

Thus, we extend the *portType* element by adding an attribute *userCategory* referring to the user profile having the ability to use operations described in the current portType. The user profile is also described by *Profile* element which provides detailed description of the profile (context and preferences) supported by the current web service. Besides, as the *definition* element in WSDL specifies the multiple *namespaces* used throughout the document, we have added a new namespace http://localhost:8080/AWS-WSDL which contains the definition of *Profile*. We are extending the WSDL 1.1 version.

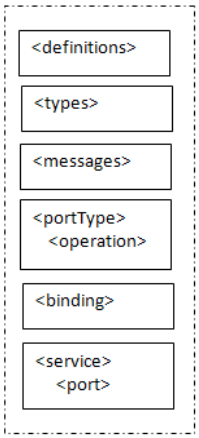

Figure 1. WSDL structure

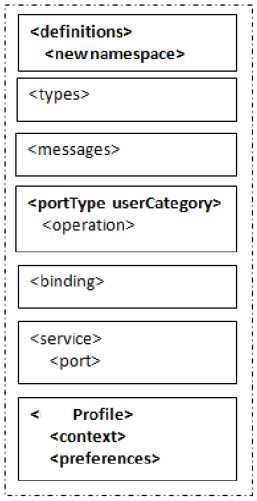

Figure 2. AWS-WSDL structure

B. Relationship between WSDL and UDDI

Universal Description Discovery and Integration (UDDI) provides a method for publishing and finding service descriptions. The UDDI data entities provide support for defining both business and service information. The Web Services Description Language (WSDL) allows to define and describe the details regarding communication with a Web service. It consists of the service interface definition and the service implementation definition.

The WSDL service interface definition is published in a UDDI registry as a *tModel*. Some of the tModel elements (name and overviewURL) are constructed using the information that is copied from the WSDL service interface definition. The WSDL service implementation definition is published in UDDI registry as a *businessService* with all relevant information copied into the *businessService*.

Currently, UDDI and WSDL allow to define and publish the means for communicating with the service, and allow a consumer to discover and use the desired service. As in our case, we proposed a service profile and specify it within the AWS-WSDL file; an UDDI extension is required to enable the current registry to publish and discover services based on the proposed profile information.

In the next section, we aim to define the UDDI extension named *Adaptable Web Service Registry* (*AWSRegistry*)by adding a data entity to save the profile which is associated with the service. When the service is searched, a match will be calculated between the users' profile and supported profile stored with the service.

III. AWSR: ADAPTABLE WEB SERVICE REGISTRY

The most known Web service registry is UDDI which is an XML-based standard that was proposed for allowing providers to publish their web services, so that they can

be located afterwards by consumers. UDDI provides a service query API to locate appropriate Web services. It also defines a service publication API for service providers, who can use it to advertise their services. A UDDI registry is structured into three layers: *BusinessEntity* or white pages, *BusinessService* and *BindingTemplate* or green pages. These layers allow searching a UDDI registry according to the three data types [10]. Unfortunately, Web service discovery through UDDI-based registries is insufficient to carry out consumer non functional requirements. This necessity is increasing every day with the emergence of hand held devices and various user preferences. Hence, provider and consumer wants to have a registry providing a fairly precise publish and search capabilities to render the use of user aware Web services more efficient. To overcome these needs, we propose a new Web service registry offering the ability to save in addition of features information, the adaptation criteria supported by each service. Our register is called Adaptable Web Service Registry (AWSR) and it supports the publication of the AWS-WSDL description presented in section II. Next we describe how we can include profile information in UDDI with the aim of discovering the services and supported profile informations easily.

A. AWSR data structure

UDDI information model is composed of data structures expressed in XML. They are persistently stored in UDDI registries. To allow the publication of user-aware Web service described according to the AWS-WSDL description language, an extension of the UDDI data structure is required. We propose adding an element named **AdaptationCriteria** that references the item *Profile* in the description file AWS-WSDL.

Figure 3 aggregates the adaptation criteria and its relationships to the standard UDDI data structure.

The AWSR data structure elements are described below:

- businessEntity: top-level structure within UDDI that contains descriptive information about a provider organization, such as contact and classification. Each businessEntity may provide various adaptableServices;
- **adaptableService**: represents a logical Web service that may have multiple implementations. It includes businessService and adaptation criteria;
- businessService: represents a logical Web service that may have multiple implementations. It includes descriptive information about a Web service, such as name and classification;
- bindingTemplate: represents a service implementation (Web service) and provides the information needed to bind with the service. Each bindingTemplate contains information such as access point and transport protocol;
- tModel (Technical Model): represents unique concepts in UDDI, such as namespaces and category systems. Examples include tModels based on WSDL and other documents that specify service interfaces;
- **adaptationCriteria**: represents the user profile element describing the context of the service.

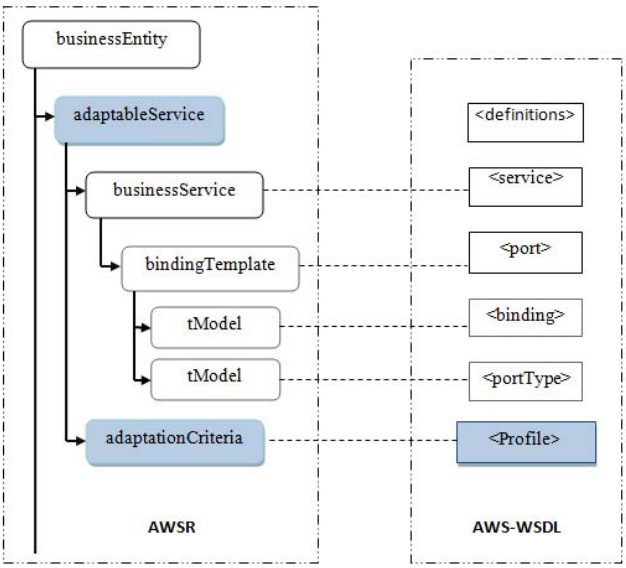

Figure 3. Data structure of the AWSR

To be able to save AWSR model elements we need to extract corresponding values, to analyze it and to save it in a suitable entity.

B. Publishing Adaptable Web Service

To be able to save functional and profile information in the AWSR, we propose to add a table in the UDDI database named **service profile**. When a provider requests the publication of his service at the AWSR, an analyzer module retrieves the AWS-WSDL service description and performs an analysis of the different elements of the file and then the extraction of the values provided for each element. Then features and access points description will be routed to the **ServiceFeatures** database and those corresponding to the adaptation criteria will be routed to the **ServiceProfile** database. To implement the publishing module of the AWSR, we propose a multi-agent system consisting of two main agents: The *provider agent* and the *publication agent* shown on figure 4. This system is developed on Java and is based on the juddiV3 java implemented API for the UDDI register. Therefore, we use the apache Tomcat as a Web server and the MySQL as a database management system. Agents are developed using Jade (Java Agent Development Framework).

Our multi-agent system offers a Java user interface allowing the provider to specify enterprise and service information and to specify the URL of the description file of its service. Then, the *provider agent* acts as an interface between the

Figure 4. Publication multi agent system

system and the Web service provider. It allows the retrieval of the AWS-WSDL description file throw its URI. The *publication agent* has to analyze the AWS-WSDL file. It allows the extraction of the functional description contained in items (types, portype, message, binding, service) and stores it in the database *ServiceFeatures*. Then it extracts the profile description contained in the element *Profile* and saves it in the database *ServiceProfile*. The provider has a simple interface allowing to provide enterprise name and description, service name and description, to choose the publication server and finally to choose the corresponding web service profile annotation description file.

Figure 5 illustrates an excerpt of the *AWS-WSDL* document describing the room booking service. The user's profile is incorporated in this document using the newly added elements, *UserCategory* and *Profile* to the WSDL schema. Here in this example, we only give the detailed description of the extended elements.

```
<portType name="specifierPrefs" categorieUtilisateur="xsi:profileTouriste" vue="utilisateurHumain">
    <operation name="specifierPrefs">
        <input message="specifierPrefsRequest"/>
        <output message="specifierPrefsResponse"/>
    </operation>
</portType>
<binding name="bindingName" type="y:typeName"/>
<service name="serviceName">
    <port name="SimplePort" binding="wsdlns:SimpleBinding">
        <soap:address location="http://www.elmouradi.tn"/>
    </port>
</service>
<profile name="profileTouriste" type="utilisateurHumain">
    <droitdacces designation="interaction" estAutorise="true"/>
    <preferences>
        <prefsaffichage value="FR" name="langage" />
        <prefscontenu value="YES" name="displayText" />
        <prefscontenu value="NO" name="displayVideo" />
        <prefscontenu value="NO" name="displayImage" />
    </preferences>
    <context>
        <localisation latitude="34" longitude="9"/>
        <dispositif>
            <materiel type="desktop" tailleecran="1366" memoire="4048528"/>
            <logiciel SE="windows 7" />
            <connexion type="3G"/>
        </dispositif>
    </context>
</profile>
```

Figure 5. Excerpt of the room booking description file

Next step is to publish the AWS-WSDL description of the service in AWSR. The definition document which is associated with the supported profile is chosen by the provider via the publishing interface illustrated in figure 6. The provider have to specify the enterprise information, the service information, the web server and the description file.

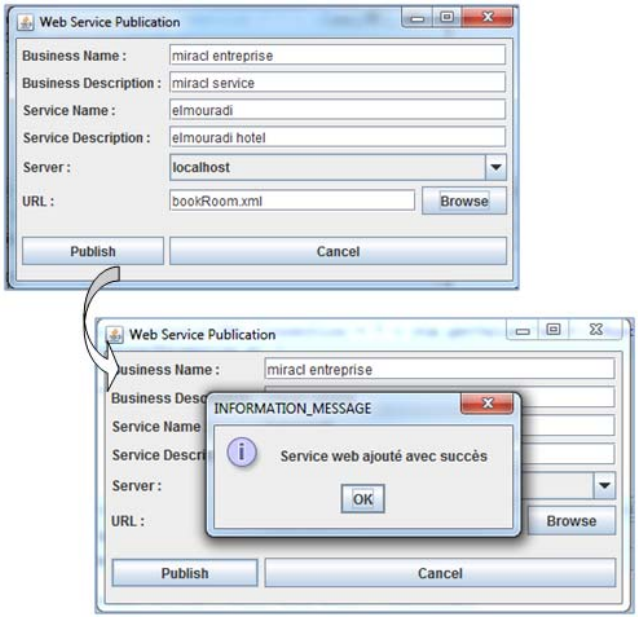

Figure 6. Publishing booking room web service in the AWSR

After getting service information through the publishing interface, a parser will get the aws-wsdl description provided and extract features description to be saved in the *ServiceFeatures* database and the profile to be saved in the *ServiceProfile* database.

C. Finding Adaptable Web Service

The AWSR contains also a query analyzer module used in the selection and discovery phase. This module communicates with the database **ServiceFeatures** and **ServiceProfile** in order to select the suitable Web service meeting functional and non functional queries.

When looking for a service, the query analyzer offered by the AWSR allows the customer to:

- Extract the needs in terms of functionality through keywords.
- Extract the needs in terms of adaptation by specifying preferences and by detection of his context hrought sensors and contextual preferences afforded by the user.

The selecting module (cf figure 7) is implemented using a multi agent system; a *Discovery agent* and a *Selecting agent*.

- *Discovery agent:* This agent retrieves the request from keywords and preferences specified by the user and context information (device, connection and location)

detected using a context detection system E-WDM (Extended WildCat Detection Module). Preferences and context captured will be stored in an XML file (user-Context.xml). Then, the agent enables the discovery of web services. Keywords will be adressed to the Service features and context captured will used to discover service with corresponding profile. Results will be transfered to the selecting agent.

- *Selecting agent:* This agent selects the suitable web service that satisfies the user's request.

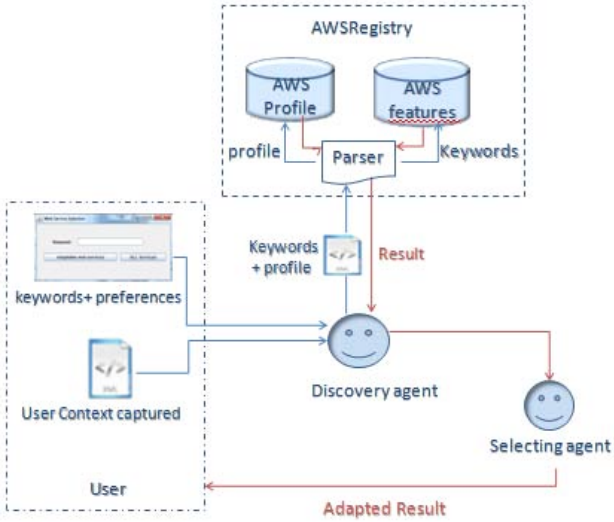

Figure 7. Adaptable Web service selecting module

IV. AWSR Experimentation

In this section, we provide experiment results when publishing Web services in the AWSRegistry. We use the *JUDDIv3* API which provides a Java based open source implementation of the UDDI register. To save web services description, the *JUDDI* could communicate with any relational database. In our work we have chosen the MySQL database. We have extend the *JUDDI* API to be able to extract and save profile characteristics from the AWS-WSDL document. We also extend the JUDDI database with a table named *db-profile* in order to save the service profile's information. Our experiments are organized into two sides: The first experiment determined the performance of the standard UDDI by measuring the speed of the publish function dealing with WSDL document. The second experiment determined the performance of the AWSR by measuring the speed of the publish function dealing with AWS-WSDL document. Both measures are illustrates in table I.

From the table bellow, it was noted that the publishing time are very close. Figure 8 demonstrates using an histogram a comparison between time taken to publish WSDL description and the one taken to publish AWS-WSDL description.

Table I
Execution time to publish Web service into AWSRegistry

Services	Name	wsdl time	aws-wsdl time
Service 1	BankValidate	472	480
Service 2	BookRoom	352	572
Service 3	Calculatrice	435	683
Service 4	Currency	520	641
Service 5	GlobalWeather	483	511
Service 6	Latex	353	468
Service 7	Stockquote	417	499
Service 8	Weather	724	737

Figure 8. Execution time required to publish web services to the AWSRegistry

V. Related Work

In this section, we take a look at some research works interested in the extension of the description language WSDL and the UDDI registry. We provide an overview of these works.

El Asri et al. [4] propose a model driven approach for the modeling of user-aware web services on the basis of the multiview component concept. The multiview component is a class modeling entity that allows the capture of the various needs of service clients by separating their functional concerns. This work takes into account access rights of the expected user. Despite that, the user preferences, device capacity, network characteristic, localization ... are not taken into account.

The work presented in [1] covered all kind of non-functional attributes needed to express web service's capabilities. It focused on providing a model extension of WSDL thought two-level descriptive model: *Class Attribute* (CA) is an overall description of services features and *Class Description* (CD) shows the details of a special CA.

PA-WSDL [7] is a WSDL extension aiming to integrate payment information in the web service description. This work proposed adding new a *PaymentDescriprion* element to the standard WSDL file. This element describes the payment

model, the price, the payment protocol ... Then a brief overview is given describing how to include payment information in UDDI with the aim of discovering the services and their payment information easily.

[5] presented extensions to WSDL and UDDI to support run-time and development-time versioning information of web service. These work aimed to build a version-aware service model based on the WSDL service model. Also, we present a design of a versionaware service directory by extending the current UDDI service registry with the UDDI tModel structure to store the meta-data of the version-aware services.

Sellami et al. [8] are interested in Web services discovery in a distributed registry environment. They proposed a semantic model to describe Web services registries (WSRD). This semantic description is functionality driven and is benefit to discover the appropriate service that best fits requester needs. A WSRD results from the aggregation of the different Web services functional descriptions advertised by a registry. It gives an idea of the functionalities offered by the Web services of a registry.

In [6], a novel Web services discovery model based on the metaphor of social networks was developed. Presented solution shows how to assist Web services in building, using, and maintaining their respective social networks. Then, they discussed the use of social networks to discover Web services.

While there are solutions on listing non-functional aspects of services, the focus is mainly on QoS factors. The interest is mostly on showing how quickly a service can generate a response and so forth. Other researches have been interested in versioning of Web services. However, there are a major lack in considering adaptation notion when describing or providing Web services. In this paper, we have been interested in covering this lack.

VI. Conclusion

The expansion of the Web and the wide spread of the Web service as a basic technology of universal interoperability through Internet have led to an increasing need of user-aware adaptation. The same service should be able to adjust its offered functionality according to the user specific need regarding his profile (context and preferences). Thus, Web service adaptation is an urgent and challenging issue. In the literature, multiple adaptation solutions have been proposed but they are all restricted to services description and selection steps.We proposed in this paper an adaptable Web service registry called AWSR. It supports saving and searching Web services according to adaptation criteria together with functional needs. We detailed our registry data structure and implementation. We also tested the performance of our registry AWSR and comparing its publishing robustness with the standard UDDI registry. Our next step is to evaluate the search result returned by the AWSR registry in response to a functional and adaptation user's request.

References

[1] C. Dai and Z. Wang, *"A flexible extension of WSDL to describe non-functional attributes,"* Proc. 2nd IEEE International Conference on e-Business and Information System Security (EBISS 10), May 2010, pp.1-4, doi:10.1109/EBISS.2010.5473641.

[2] C. ElHog, , R. Ben Djemaa and I. Amous, *"AWS-WSDL: A WSDL Extension to Support Adaptive Web Service,"* Proc. International Conference on Information Integration and Web-based Applications Services (iiWAS11), Dec 2011, pp. 477–480, doi: 10.1145/2095536.2095634

[3] C. ElHog, , R. Ben Djemaa and I. Amous, *"Profile Annotation for Adaptable Web Service Description,"* Proc. ACM Symposium on Applied Computing (SAC 12), March 2012, pp. 1935–1940, doi:10.1145/2245276.2232096

[4] B. El Asri, A. Kenzi, M. Nassar, A. Kriouile and A. Barrahmoune,*"Multiview Components for User-Aware Web Services,"* Proc 11th International Conference Enterprise Information Systems (ICEIS 09), May 2009, pp. 196–207, doi:10.1007/978-3-642-01347-817

[5] M.B Juric, A. Sasa, B. Brumen and I .Rozman, *"WSDL and UDDI extensions for version support in web services,"* Journal of Systems and Software, vol 82, Sep 2009, pp. 1326-1343, doi:10.1016/j.jss.2009.03.001

[6] Z. Maamar, L.K. Wives, Y. Badr, S. Elnaffar, K. Boukadi and N. Faci, *"LinkedWS: A novel Web services discovery model based on the Metaphor of "social networks","* Simulation Modelling Practice and Theory, vol 19, 2011, pp.121-132, doi:10.1016/j.simpat.2010.06.018

[7] A. Ruiz-Martinez, O.C Reverte and A.F Gomez-Skarmeta, *"Payment Annotations for Web Service Description Language (PA-WSDL),"* Computer and Information Science, vol 131, 2008, pp. 65-76, doi:10.1007/978-3-540-79187-46

[8] M. Sellami, O. Bouchaala, W. Gaaloul, S. Tata, *"WSRD: A Web Services Registry Description,"* Proc 10th Annual International Conference on New Technologies of Distributed Systems (NOTERE), June 2010, pp.89-96, doi:10.1109/NOTERE.2010.5536781

[9] B. Soukkarieh, *"L'adaptation au contexte dans un systme d'information web = Plateforme CA-WIS (context adaptation of web information system),"* Revue des sciences et technologies de l'information, vol 14, 2009, pp.91-116, doi:10.3166/isi.14.1.91−116

[10] UDDI Spec Technical Committee Draft, $http : //uddi.org/pubs/uddi_v3.htm$

[11] $http : //www.w3.org/TR/wsdl$

[12] $http : //www.w3.org/TR/soap/$