I.J.Modern Education and Computer Science, **2013, 8, 27-42**
Published Online October 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2013.08.04

Context-Aware Web Service Discovery Based on A Quantitative Similarity Measure

Bouyakoub Fayçal M'hamed
University of Sciences and Technology Houari Boumediene, Algiers, Algeria
Email: fbouyakoub@usthb.dz

Belkhir Abdelkader
University of Sciences and Technology Houari Boumediene, Algiers, Algeria
Email: kaderbelkhir@hotmail.com

***Abstract*—**This paper presents a new context-based solution for Web services discovery. The service description includes an enriched representation, in order to make more efficient the discovery and selection stages.

Our approach gives, services publishing and searching, another dimension. Services context-based selection uses a new quantitative similarity measure to calculate the correspondence degree between the client and the services contexts in order to provide users with appropriate services according to their contexts.

Index Terms—Web services, Services discovery, Similarity measure, User's/service's contexts.

I. Introduction

Web services are software components that provide functionalities accessible via standardized web protocols. Based on XML (*eXtensible Markup Language*), web services are independent from platforms and operating systems. This characteristic involves their adoption by various commercial and industrial organizations offering their services across the Web, and therefore increases the number of offered services.

Web services are based on standards technologies [1] like: WSDL (*Web Services Description Language*), UDDI (*Universal Discovery, Description and Integration*) and SOAP (*Simple Object Access Protocol*).

WSDL is used to describe Web services in terms of parameters, communication protocols [2] etc. UDDI is a Web services descriptions directory; it is used for discovering services [3]. Finally, SOAP is a transmission protocol between the user and the service provider [4].

Web services discovery is based on a syntactic search of the WSDL descriptions using the UDDI register.

However, with the diversity of users and the conditions under which they access Web services, other parameters must be considered during the discovery, such as the client terminal (PDA, laptop, etc.), the client preferences, his location, etc… All these parameters form a particular context of use called the *user profile*.

Also, publishing methods available in the UDDI do not involve a formal model to describe services' contexts, therefore services discovery could not be achieved efficiently without taking into account their contexts.

Indeed, when a user sends his/her request to services discovery system, he/she would like to have as a result services adapted to his/her preferences and to his/her device characteristics (screen size,…).

Thus, it becomes necessary to propose a model which provides relevant and adapted results according to the user context.

Since the user context can vary during a session, the system must be able to adapt it in order to select services according to the new context.

However, the functional parameters of a Web service, contained in the WSDL description, are not sufficient to implement discovery mechanisms. In order to fill this lack (representation of Web services), we propose to extend Web services descriptions in order to take into account the necessary concepts in the discovery process. Thus, Web services descriptions include the context of the user, via a general structure allowing the use of a common vocabulary to represent the two entities: the user and the service contexts.

Finally, the context is used during the discovery stage by selecting services corresponding as well as

possible to the user request. This discovery is based on the use of a similarity measure, which estimates the correspondence degree between the user context and the service context.

This paper is organized as follows: section 2 is an overview of the different approaches proposed for services discovery. For each approach various solutions are presented and a comparison is done at the end of this section. Section 3 is dedicated to similarity measures concept. In the next section we present our Web service discovery framework based on contexts: First, we start by presenting the objectives of the framework and then the user/service contexts components and finally the framework architecture, composed of a set of agents. Section 5 presents the quantitative similarity measure used in the discovery process and the next section evaluate the efficiency of the proposed measure. Finally, section 7 concludes this paper.

II. Web Services Discovery

The success of Web services allowed the adoption of this technology by various services providers, which induced an increase number of services, making their discovery a complicated task.

Various discovery mechanisms were proposed in the literature. In [5], authors define the discovery process as being "*the act of localizing a description, manageable by a machine, of an unknown Web service before describing certain functional criteria*".

Currently, services descriptions are published in registers (like UDDI) conceived, specially, for this function. The aim of these registers is to facilitate the discovery of services published by various commercial organizations.

Generally, Web services discovery is based on terms such as *request*, *keyword*, *matching* or *mapping*. Several definitions of these terms were proposed in the literature:

- **Request:** Generally a request is addressed by a user to a search engine, during a session with one or more keywords.
- **Keyword:** A keyword is a succession of characters not containing blanks and appearing in the field reserved for this purpose in the request. Operators (characters used to make complex expressions) are not considered as key words.
- **Matching:** The matching is a mechanism which aims to find semantic similarities, and possibly structural similarities, between two information: the required information (necessary) and the provided information by the system (published) [6], using a matching algorithm. The matching algorithm compares all the announcements (descriptions of published services) with the request, to provide, as a result, services which are close to the request.
- **Mapping:** The mapping consists of finding semantic relations which allow the transition from an entity to another. That is why we must look for correspondences to establish transformations between objects having comparable nature but not having the same form. Consequently, the mapping uses the matching results to carry out the possible transformations of the objects.

Different approaches were proposed, in the literature, to support web services discovery [7][8][9][10]. Initially, Web services discovery was mainly based on a syntactic search (syntactic correspondence between the request key words and services descriptions). But with the development of new technologies like semantic Web, these techniques became primarily semantic (semantic degree of similarity between the terms of the request and semantic descriptions of the services).

In general, discovery approaches can be classified in three categories:

- Approaches based on syntactic descriptions of Web services;
- Approaches based on semantic Web;
- And context-based approaches.

For each approach, we have two architectures: centralized and distributed architectures. In a centralized architecture, services descriptions are saved in the same register and in distributed architecture services descriptions are saved in different registers creating a cloud.

II.1) Syntactic Representation Based Approaches

The principle of syntactic-based approaches is simple: The client sends a request composed of key words; these words will be compared with services' descriptions. In spite of its simplicity and its facility of implementation, this approach presents some limitations. Indeed, syntactic search does not make it possible, always, to have good results. Moreover, a software agent cannot examine textual descriptions intended for human use.

Others solutions were proposed for distributed architecture, as [11]. The main idea is to connect an arbitrary number of nodes (cloud or UDDI federation) to form in a virtual UDDI register, and each node contains a part of services' descriptions. When a user sends a request to one of the nodes, the node transmits the request to its neighbours, and so on for the nodes receiving this request. The results, obtained in each node, are then sent to the source node.

The AASDU system (*Agent Approach for Service Discovery and Utilization*), proposed in [12], is an example of distributed systems. AASDU is composed of:

1. A Graphical User Interface (GUI);
2. A Query Analyser Agent (QAA);
3. A reference system of agents' expertise fields, which reference agent according to their expertise;
4. The services module which allows services providers to publish descriptions, to start a negotiation to select services and finally to invoke a selected service.

II.2) Semantic Web Based Approaches

In the second discovery class (semantic approaches) authors focused on the semantic description of services. This development is increasingly significant since it seems to be able to approach certain insufficiencies of syntactic approaches.

For centralized architectures, we have:

- **The OWL-S** (*Web Ontology Language*) **approach** [13]**:** Among the ontologies proposed for the description of services we have the DAML-S ontology (*DARPA Agent Markup Language for Services*). This ontology is based on the DAML language ontology. DAML-S describes a service using three profiles:
  1. **ServiceProfile:** defines the service;
  2. **ServiceModel:** defines the service operations;
  3. **ServiceGrounding:** defines how to reach the service.
- **The IRS-II platform** [14]**:** The main components of this architecture are:
  1. The ***IRS-II Server*** contains services' descriptions.
  2. The ***IRS-II Publisher*** has two functions. Firstly, it allows linking services to their respective semantic descriptions. Secondly, it automatically generates a program which wraps the Java code of the service, in order to invoke it.
  3. The ***IRS-II Client*** invokes a service with a request.

For distributed architectures we have:

- The **PSWSD** Architecture (*P2P-based Semantic Web Discovery Service*) [15] is a service discovery architecture in a P2P network. In this architecture, providers publish services' descriptions in various distributed registers. A subscriber looking for a service can question any register of the network. When the register receives the request, it will direct it towards the register(s) which can satisfy this request. This information is sent to the *matchmaker module* which selects services descriptions having a semantic correspondence with the user request.

- The **Speed-R** system [16] aims to connect all private UDDI registers (each service provider has its own UDDI register) via a P2P network. In order to have semantics in services descriptions, authors associate to each register specific ontologies. Semantics are brought to services descriptions by making a mapping between services specifications and concepts of ontologies.

II.3) Context Based Approaches

Dey [17] defines a context as being "*all information being able to be used to characterize the situation of an entity, where an entity is a person, a place, or an object who can be relevant for the interaction between the user and the application, including the user and the application themselves*" [18].

With this definition, we can say that each entity (user and Web service) have its own context. The service context can group the service localization (geographical restriction), the service cost, the service category, quality of service parameters, etc. The user context can be formed of his localization, his preferences, etc.

Several context-based discovery solutions were proposed, among these solutions we cite:

a) The UDDI+ approach

The principal idea of this approach is to make extensions on UDDI register in order to take into account context information during the service discovery. The new UDDI server is called *UDDI+* [19]. The objectives of UDDI+ are:

- To allow a Web service semantic discovery;
- To provide users with services according to their contexts (localization, etc.).

UDDI+ is composed of four principal components:

1. The Proxy which allows the publication of syntactic and semantic descriptions of services. When receiving a publication or an update request, the Proxy transmits it to the UDDI server which assigns a single identifier called UUID (*Universal Unique Identifier*). The Proxy checks if the description message contains a T-Model describing

the service in DAML-S (*DARPA Agent Markup Language-Service*). If the TModel exists, the semantic description is saved in DAML-S repertory with the UUID identifier, if not the description is saved in UDDI. The provider can specify the period of validity of the published information. This information is used by the *planning module* (scheduler) responsible for the deletion of services descriptions in UDDI and in the DAML-S repertory when the validity date is expired.

2. The second component is a simple search interface relating to UDDI. For an advanced search, authors proposed an interface named *Inquiry+* allowing the user to introduce a description of the desired service as well as context information.

3. The *Matcher* uses ontologies to discover services corresponding (semantically) to the user request and filters services which are not appropriate to the user context.

4. The last component is an ontologies database.

b) SOAP

The approach proposed in [20] integrates the context in the SOAP (*Simple Object Access Protocol*) communication protocol. The goal of this approach is to seek and select services according to a context integrated in SOAP. This context takes mainly into account the characteristics of the user device and his localization. The selection, based on services contexts, carried out by the *Context Manager*, is realized in four stages:

1. The SOAP request is pre-treated by the *Context Manager*, in order to extract the transmitted context. This process makes it possible to find, at the next stage, a service corresponding to the user needs.

2. The *Context Manager* invokes a service according to the defined context.

3. The answer of the invoked service is transmitted to the *Context Manager* to check if it corresponds well to the context.

4. Finally, the result of the treated service is sent to the *SOAP Request Processing* in order to be translated in this language and sent to the client.

c) CASD architecture

CASD (*Context Aware Discovery Service*) [20] contains a semantic discovery module which determines the services that have a semantic relationship with the user request using specific ontologies. When the user formulates his/her request (*Qusr*), another request (*Qctx*) is attached to the first one. *Qctx* contains information about the user context, such as the device type and the user profile. This information is recovered, respectively, by the two databases: the user profiles database and terminal type database.

Other information composing the user context like his localization is attached to *Qctx*.

Qusr allows finding services which have a semantic relationship with the user search topics while *Qctx* filters services which have a contextual correspondence with the user.

d) CB-Sec architecture

CB-Sec is an architecture for the discovery and the composition of services based on contexts [21]. This architecture is composed of four layers:

1. The physical layer: this layer represents the resources belonging to the environment. The resources describe all the entities which can be implied during the execution of an application such as software components…

2. The context layer: this layer recovers and processes the contextual data. It is composed of two modules: the *collection of contextual data* module and a database for contexts data. The *collection of contextual data* module is based on a multi-agents system where each agent tries to recover and to treat a certain type of contextual data (user context, terminal characteristics, etc.) and to save these data in the database. Contextual data is acquired by sensors, or by agents.

3. The services layer: This layer is responsible for the discovery, the composition and the execution of Web services.

4. The user application layer: Consist of a set of interfaces allowing various users to specify their profiles and preferences.

II.4) Synthesis

Initially, Web services descriptions were made only at a syntactic level, services discovery was based on techniques from information retrieval. However, with the development of semantic Web technologies and in order to automate the discovery task, new approaches based on services semantic descriptions were proposed.

In general, discovery approaches depend on the representation level (semantic or syntactic) of services descriptions. Moreover, the approaches vary according to the adopted architecture (centralized or distributed).

We can notice that generally multi-agents approaches are used in distributed architectures. That is well justified since this technology is well adapted to the distributed nature of the problem (Table 1).

It is also noticed that the semantic approaches suggested, such as OWL-S [13], Speed-R [16] and

PSWSD [15], are based on the same technique which consists on calculating the semantic correspondence level between the services functional parameters and those mentioned in the user request.

Table 1 Synthesis of services discovery systems

	AASDU	OWL-S	Speed-R	IRS-II	IRS-III	PSWSD
Services description	**WSDL**	**OWL ontologies**	**WSDL & ontologies**	**Ontologies**	**WSMO ontologies**	**WSMO ontologies + WSDL**
Architecture	**Distributed**	**Centralized**	**Distributed**	**Centralized**	**Centralized**	**Distributed**
Discovery techniques	**TFIDF**	**Matching**	**Key words**	**SM**	**SM**	**Matching**
Adopted technology	**Multi-agents**	**SOAP**	**Multi-agents**	**SOAP**	**SOAP**	**Multi-agents**

The difference between these approaches is the ontology language used. In PSWSD [15] a WSMO ontology is adopted while in [13] an OWL-S ontology based on OWL is used.

Discovery mechanisms must take into account services context in order to propose to users only services which meet, as well as possible, their needs. In other terms, we must choose services whose context is adequate with the user one. Table 2 represents the user's context components considered by the suggested approaches.

Table 2 Users' context components supported by the presented approach

	UDDI+	SOAP	CB-Sec	CASD
User Localization	+	+	+	+
User terminal	+	-	+	+
User profile	+	-	-	+

There are three approaches to create user context: manual approach, automatic and semi-automatic approach. In the manual approach, the user introduces his context parameters, via an interface dedicated for this purpose. The UDDI+ [19] is based on a manual approach.

In the two other approaches, the context is built automatically via software agents. However, for the semi-automatic approach, a part of the context is built manually, such as for example in CASD [20] and CB-Sec [21].

Web services context includes several parameters like the service category, QoS parameters, localization, etc. However, the majority of suggested approaches focus only on two parameters: localization and the device type. Moreover, few works took into account QoS parameters during the discovery.

In the other hand, one of the big problems of Web search systems is the definition of a correspondence function between the representation of the proposed service and the user request. This function must model the relevance of the search result to the user [22].

The search relevance is a complex concept. Closely related to user judgment, the relevance is paradoxically evaluated by technologies because the capacity to perceive similarities and analogies is one of the most fundamental aspects of human knowledge [23]. Consequently, to be able to offer, to users, services corresponding to their requirements, a search solution must be based on a relevance model. This model will permit to calculate, for each request, the relevance of its information. Those that have the best relevance score will then be presented to users in a descending order.

In the majority of the cases [24], the correspondence between what is offered and what is required is evaluated using a similarity measure to obtain useful information about their compatibilities. Calculating the similarity was considered as a subject of research strongly recommended in the fields of semantic Web and artificial intelligence [24].

Similarity Measures

The similarity is defined by the degree of resemblance between two objects. Indeed, any system having for goal to analyse or organize automatically a whole of data or knowledge must use, in a form or another, a similarity operator to establish the resemblances or the relations existing between the used data.

In general, a similarity measure is defined in a universe U which can be modelled using a quadruplet: (Ld, Ls, T, SF) [25]:

Ld the representation language used to describe the data;

Ls the representation language of the similarities;

T a set of knowledge about the studied universe;

SF the similarity binary function, such as:

$$SF: Ld * Ld \rightarrow Ls$$

A similarity measure is a function which satisfies the following properties:

$$\forall x, y \in Ld: SF(x, y) \geq 0 \quad (1)$$

$$\forall x, y \in Ld: SF(x, x) = SF(y, y) \geq SF(x, y) \quad (2)$$

$$\forall x, y \in Ld: SF(x, y) = SF(y, x) \quad (3)$$

In the same way, a dissimilarity measure is defined as a function which checks the following properties:

$$\forall x, y \in Ld: DF(x, y) \geq 0 \quad (4)$$

$$\forall x \in Ld: DF(x, x) = 0 \quad (5)$$

$$\forall x, y \in Ld: DF(x, y) = DF(y, x) \quad (6)$$

It is also possible to transform a similarity measure SF to a dissimilarity measure DF by using the following relation:

$$\forall x, y \in Ld: DF(x, y) = S_{max} - SF(x, y) \quad (7)$$

Smax is the maximum value which can be obtained by the similarity measure.

Several similarity measures were proposed in various applications fields, we cite:

Similarity for textual data [26];

Similarity for intrusion detection [27];

Similarity for Web services [28].

The next section presents a new quantitative similarity measure for a context-based service discovery system.

Toward a New Framework for Context-Based Web Services Discovery

Nowadays, semantic Web services discovery is based on Input parameters and Output results [29]. However, the emergence of new means of access to information, such as wireless networks (Wi-Fi, Bluetooth) and the emergence of new communication terminals like laptops and PDAs made that the access to the service is not carried out any more in the same way, nor by the same devices.

In addition to devices heterogeneity, the diversity of contents, offered to users, made that other parameters must be considered during the service discovery. All these parameters form a particular environment of discovery.

In various environments, users reaching the same services can receive different answers. Thus, discovery systems take into account these parameters and filter the desired services, adapted to the environment constraints expressed by the user.

Thus, in Web services, the adaptation is based on the determination and the management of all these characteristics for the selection and the presentation of suitable services. These characteristics represent the context.

To have a context sensitive based discovery system, we propose a discovery system which allows the publication and the discovery of services in a UDDI directory by using a similarity measure.

We will take into account the user and service contexts during the discovery process in order to offer services having contexts corresponding to user requirements. This discovery solution will be deployed in our services delivery platform [30].

II.5) The Framework Objectives

The work presented in this paper is an extension of the adaptation system proposed in [30]. New functionalities and improvements are proposed. Among the objectives to reach, we cite:

- **Management of services:** The platform must allow, to providers, the addition and the deletion of services and also the update of their characteristics.
- **Proposing a search engine:** In order to define users' needs, expressed via a request (in natural language), it is necessary to translate this request to be comprehensible by the system.
- **Offering to users the best service:** The system should satisfy the users' requests as well as possible, according to their needs and contexts.
- **Defining methods to implement the discovery process:** These functions help the system to generate services offers satisfying the user request (syntactic and contextual level). This includes functions, such as similarity measure, used to obtain the correspondence rate between the user and the suggested services contexts.
- **Management of WSDL documents and creation of CC/PP files:** Contrary to the approaches already presented (section II), which refer to WSDL files via the Tmodel [31] and define only the functional parameters of the service, we add a new reference in the Tmodel in order to establish a link towards the service CC/PP (*Composite Capabilities/Preference Profiles*) repertory. This repertory contains CC/PP files representing the non-functional parameters of a service version.
- **Defining methods to implement the publication process:** In order to make efficient the search processing, the publication process should integrate functions allowing the service publication in UDDI. Moreover, it integrates also an enriched representation of services allowing their publication on a contextual level.

Before presenting the discovery system, it is necessary to define the various parameters forming the two contexts: user and service context.

II.6) Context Elements

The suggested context contains mainly several dimensions able to represent information characterizing a service (QoS, devices… etc.). It makes it possible to model the user's context, requesting the service, as well as the offered service.

We propose a multidimensional-semantic representation which allows the contexts representation by creating a concepts hierarchy instead of independent unit fields.

The defined structure is flexible; various characteristics can be extended thanks to the hierarchical character of the proposed model.

The model is arborescence and is composed of one or more components, each component contains one or more attributes (value, weight).

The context is formed of various parameters such as the service cost, the localization (for geographical restriction), supported terminals, Quality of service (QoS), etc.

The user context is characterized by the following parameters: user identity, his preferences, type of terminal, localization, etc.

Various models for context representation were proposed [26]. We chose CC/PP format for the context representation [32]. CC/PP [33] is a W3C (*World Wide Web Consortium*) standard, which allows an expressive description of the discovering environment constraints by using a general structure which takes a standard form for all users.

The suggested context makes it possible to model the user and the service contexts by using a common vocabulary to represent relative data of the two entities.

This context is formed of 4 components (**Figure 1**) according to their types:

- Service Identity (service name, service functionalities…),
- Service QoS preferences as well as service language: this parameter allows the provider to define the following aspects:
  - Required QoS parameters;
  - Language preferences;
  - Execution time parameters: The attributes relating to the execution time such as capacity, performance, reliability, availability, flexibility, exactitudes.
- Terminal characteristics: This parameter contains all the details concerning the user terminal as well as the client localization. This information is represented by:
  - The terminal description: Includes the description attributes of the terminal, thus providing basic information on which the service will carry out or will have to be adapted (terminal type, model, manufacturer…).
  - Software parameters: Contains the software details of the terminal such as the type and the version of the operating system and the navigator.
  - Hardware parameters: Contains the material details of the terminal such as the size and the resolution of the screen.
- Localization.

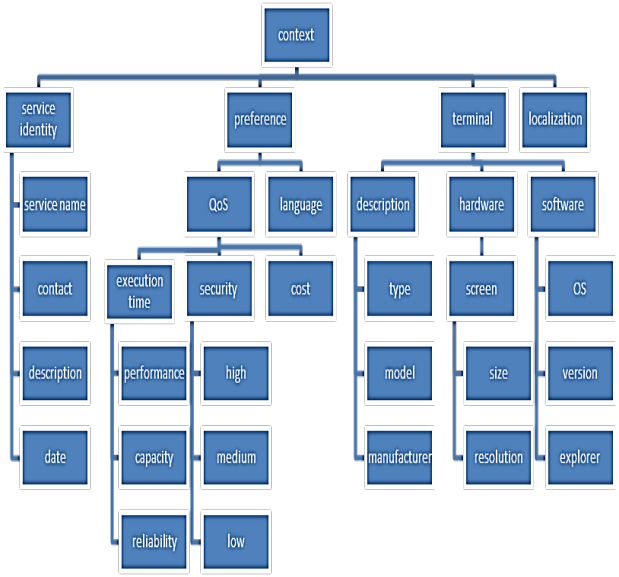

Fig. 1: General representation of the context model

II.7) System Agents

The proposed system is composed of five agents (**Figure 2**):

- **Publication Agent:** This agent is an interface between the system and the services providers. The functionalities ensured by this agent are:
  - Managing information about the provider and the service, services accesses, services type;
  - Saving syntactic descriptions of services in UDDI;
  - Making automatic updates of Web services descriptions;
  - And in order to make more efficient the search and selection steps, the publication process integrates an enriched representation of Web services.
- **Registration Agent:** This agent is an interface between the user and the system. Its role is:
  - To save services syntactic descriptions in UDDI;
  - To save the static characteristics of the user in the database.

- **Subscription Agent:** This agent allows clients to subscribe into the system.
- **Discovery Agent:** The role of this agent is the discovery of services descriptions according to the user request (syntactic and contextual level). The client can choose the type of request to be carried out.
  - The request can be syntactic (formulated in natural language), with a basic search;
  - Or advanced; in this case the agent enriches the request with client's context information.
- **Contexts Agent:** Its role is:
  - To recover the whole context characteristics of the desired/provided Web service;
  - To standardize user/service contexts in CC/PP;
  - To save the standardized contexts;
  - To allow automatic updates of contexts.

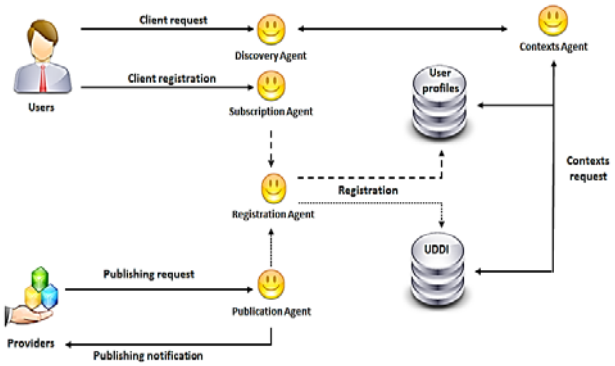

Fig. 2: System agents

After the authentication step, the system loads one of the two following agents: the publication or the discovery agent according to the client profile (user or provider).

The publication agent allows providers to publish, in a public register, all their described services.

Users can search and select from existing registers, services via the discovery agent.

a) Services publication process

The publication process assumes the following tasks:

- **Management of publishers:** Each organization wishing to host a service, in order to publish it, must have a single identifier PID (Publisher ID) to login. The suggested system must be able to manage:
  - The registration of new publishers in the UDDI server database. Following each registration, a repertory will be associated to the publisher.
  - Deletion of publisher which must automatically imply the deletion of its services and their CC/PP contextual descriptions.

- **Management of Companies registration:** Authenticated publishers can register one or more companies within UDDI directory, by introducing information to identify it (Company-Name, Category, Company-Description, URL…).

- **Management of services:** A provider announces his new service by publishing it description. Therefore, the publisher is invited to fill a form representing the context of the published service. These characteristics are used during the discovery stage in order to approach the user's needs as much as possible.

The provider (Publisher) has, via the publication form, the full context attributes. The provider selects the context attributes which are appropriate to him.

The management of services must assume the following tasks:

- Service registration and its description in the UDDI directory: in this stage, service's description is high level (no technical information is described). Among stored information, we have: Service-Name, Service-Description, links to services (URL of the service), reference towards its WSDL description and a reference towards the CC/PP repertory, which contain all contextual descriptions of the various versions of the service.
- Registration of the service and its description in the server: The service description contains contextual information about the various versions of the service already created on the UDDI server.
- Deletion of the service, which automatically implies the deletion of all its versions.

- **Management of versions:** the management of versions must deal with the following operations:
  - Registration of versions: for each service a version at least must be registered (CC/PP format) in the service repertory referenced by the Tmodel [31] according to the following path (*PublisherID/CompanyName/ServiceName*) during the publication of the service (**Figure 3**).
  - Creation of versions: a service provider can add new versions to his service already published.
  - Updating data: a service provider can carry out modifications on the services versions.
  - Removing one or several versions of the service.

- **Deletion functionalities:** In the suggested architecture, we propose various functionalities of

deletion, such as deletion of publishers' accounts as well as deletion of information relating to companies and services.

b) Services discovery process

In this stage, we aim to search for services, corresponding as well as possible, to the user request.

Thus, we propose two types of searches: syntactic search and advanced search. To allow a flexible use, the user is invited to carry out a search in natural language (syntactic search) [28].

The syntactic search is an elementary search for services which can be done according to the name of the service. The novelty proposed by our approach is that we also propose to draw up the search for a service according to a set of contextual concepts (service name, service description…).

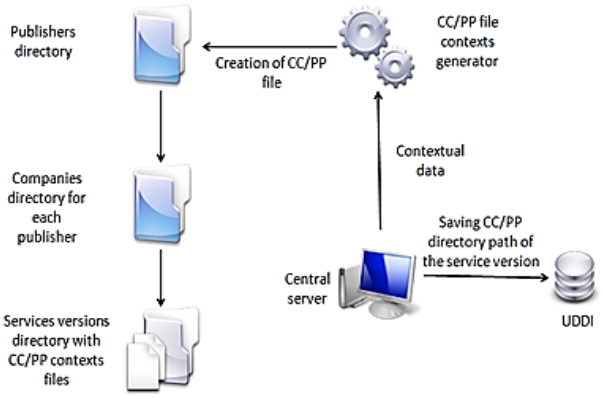

Fig. 3: Publication process

- Syntactic search: During this phase, the user is invited to express his request in natural language. When the server receives the request, a syntactic analysis will be started in order to extract the various concepts, using treatments presented in [28][34]. At the end of this step, a SOAP request is transmitted to the UDDI server to search for services according to the various concepts of the received request. At the end of this phase, the result is sent to the user.

- Advanced search: The advanced search is based on the concept of contexts. The client context is recovered from his profile, saved automatically by the server as a CC/PP file.

The client request will be formed by the various attributes values.

In SOA (*Service-Oriented Architecture*) architectures, a Web service is proposed by a provider which publishes the service description in a specific register (generally UDDI). The service description is realized with WSDL.

When a user searches for Web services, he uses UDDI. The user takes note of available services in UDDI thanks to WSDL definitions diffused by providers. When a user chooses a service, the connection between these two entities is established via SOAP.

This solution does not take into account the user context as well as the service context during the discovery process.

Our goal is to propose, to users, services according to their contexts. In other words, the service must be able to fulfil the user requirements while being compatible with the various characteristics composing its context.

Before presenting our architecture it is necessary to remind that in SOA architectures, services descriptions are made only at syntactic level. In the proposed solution, Web services are described with parameters, taking into account the service context, by using a common vocabulary to describe contexts of the two entities: the user and the service.

Figure 4 presents the discovery process.

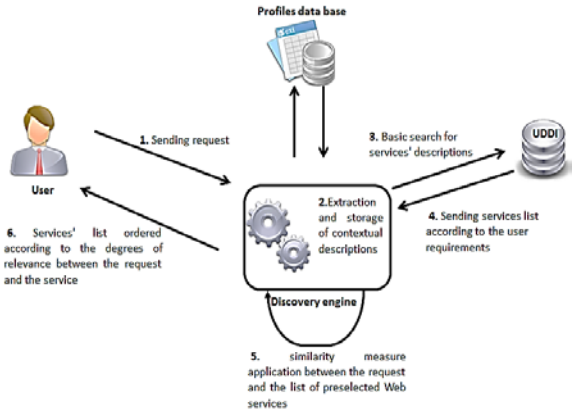

Fig. 4: Global view of the discovery process

Services discovery is based on the functional parameters of the services defined in WSDL files and referred by the Tmodel. However, other non-functional parameters are not considered in the discovery process. In order to have a services discovery system, based on contextual information, we propose to consider the non-functional parameters of the service, defined previously, and referred by the Tmodel, in addition to the functional parameters, during the discovery process.

Then, we establish a comparison between user and services contextual information:

1. First, we extract contextual information from the user CC/PP profile: The user context is an important parameter to consider during the service selection. During the interaction between the user and the discovery system, the user recovers his context in an automatic way. The user context is then sent by the *main agent* to the *searcher agent*.

2. Then we extract the contextual information from service's descriptions.

3. Finally, we compare the two contexts (user and service contexts) in order to calculate the correspondence rate.

In the proposed solution, Web services are described with parameters, taking into account the service's context, by using a common vocabulary to describe contexts of the two entities: the user and the service.

When the user submits his/her request, the *Discovery agent* uses the data XML representation to build a formal request modelling the user request. The *Discovery agent* returns a list of significant words; these words will be filtered to keep only a set of concepts constituting the request: concepts of the main segment and concepts of the optional segment. The *Discovery agent* stores the description of the requested service in the annotations database (Demand), and extracts the users' profile concepts to stores it in the users' profiles database (CC/PP format).

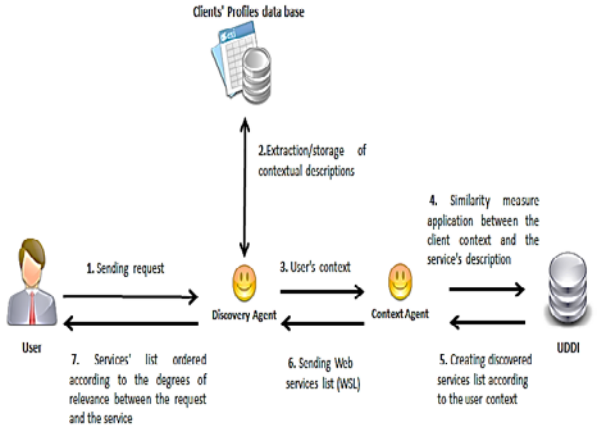

Fig. 5: The discovery process

Our work highlights the lack of taking into account the users' and services' contexts during the discovery process in SOA architectures. Our goal is to find Web services corresponding to user context. In other words, the service proposed by the system must be able to fulfil the requirements of the user while being in adequacy with the various characteristics of his context.

In order to attain our objective, we propose the use of a quantitative similarity measure.

III. Web Services Similarity Evaluation Using the QSim Measure

Services discovery is based on services descriptions (properties required and provided), to draw up a comparative study.

Gathering information (relating to the user or service contexts), is made by filling of a form (manual approach) or collected in an automatic way (localization…) by using mechanisms quite specific (GPS…).

The advanced search aims to:

- Fill the flaw of basic search: Services discovery is based on the functional parameters of the services defined in WSDL files and referred by the Tmodel [31]. However, other non-functional parameters are not considered in the discovery process. In order to have a services discovery system, based on contextual information, we propose to consider the non-functional parameters of the service, defined previously, and referred by the Tmodel, in addition to the functional parameters, during the discovery process. We associate to each parameter it weight.
- Establish a comparison between user and services contextual information: After the determination of the group of services satisfying the user request, in terms of functional parameters, in the next step the system filters this group by selecting services where contextual parameters (QoS, localization, terminal…) are adequate with user context. The selection is made in three phases:

4. Extraction of contextual information from the user CC/PP profile: The user context is an important parameter to consider during the service selection. During the interaction between the user and the discovery system, the user recovers his context in an automatic way. The user context is then sent by the *Context Agent* to the *Discovery Agent* via the user request.

5. Extraction of contextual information from services versions CC/PP file: For each preselected service, satisfying the user request in terms of functional parameters, we extract contextual descriptions of each version of the service.

6. Comparing contexts (user and service version contexts) in order to calculate the correspondence rate: For the quantification of resemblances, we choose a numerical similarity measure. Its use is extremely flexible. Indeed, the evaluation of the resemblances by a value implies that it is always possible and easy to compare context attributes (user and services' versions contexts).

To compare two characteristics forming a context, we compare their values. However, when we have several characteristics to compare, the previous method is insufficient. Therefore, we have proposed a new similarity measure to determine the degree of similarity between the two contexts while taking into account all the characteristics forming the context.

The proposed measure is the quantitative similarity measure (***QSim***).

III.1) The QSim Similarity Measure

QSim was proposed in order to fill in the weaknesses of the ***Jaccard*** measure, used in our previous work [28].

The *Jaccard* similarity measure was used to evaluate the similarity between two objects (service and user).

Let **N** be a set of objects (documents, users, services…). Each object is described by **m** characteristics. Thus, for each object X is associated a binary vector (X_1, X_2,…, X_m) such as:

$$X_i = \begin{cases} 1 & if\ the\ feature\ i\ exist\ in\ X \\ 0 & else \end{cases} \quad (8)$$

$$Where\ i \in \{1,2,3 \dots, m\}$$

Let us note by:

- **a**: characteristics suggested by the service and required by the client;
- **b**: characteristics suggested by the service but not requested by the client;
- **c**: characteristics requested by the client but not suggested by the service.

The *Jaccard* similarity measure used in our previous work [28] for services discovery state as follows:

$$S = \frac{a}{a+b+c} \quad (9)$$

This measure gave good performances, however it presents some disadvantages.

Among these weaknesses, we have noticed that the ***Jaccard*** measure checks only the existence of a characteristic (we have a binary result: 1 if the parameter exists, 0 else).

Let us take the following example: We consider two objects X and Y with the following properties:

X= (1, 1, 0, 0, 1, 0, 1)
Y= (1, 1, 0, 0, 1, 0, 1)

The ***Jaccard*** similarity measure used in [28] allows to measure the similarity of these two objects and to turn over the value 1 because the same characteristics are present.

Let us consider now a quantitative parameter (we suppose that the first parameter is the connection rate). Let X be a user having a 1 MØ connection rate and Y a service requiring a 2MØ connection rate to carry out.

Even if with the ***Jaccard*** similarity measure the similarity degree between the two objects is of 100%, it is obvious that the two objects are not completely identical.

Our new quantitative similarity measure is formalized as follows:

- Let **P** be a set of profiles (users, documents, services …). An object is described by **m** contextual characteristics X= (X_1, X_2, X_3…, X_m).

In our context, let **N** be a set of services. Each service is composed of a set of concepts, and each concept is described by **n** characteristics.

- Let **X**= (X_1, X_2,…, X_n) be a profile belonging to **P** and (w_1, w_2,…,w_n) is a set of weights associated to each characteristic where $\sum_{i=1}^{i=n} w_i = 1$.

- We define a threshold in order to present only services that have a similarity rate with the user profile higher than the threshold defined.

The similarity measure ***QSim***: PxP →[0,1] is defined as follows:

$$\boldsymbol{QSim}(X,Y) = \frac{a \sum_{i=1}^{i=n} w_i * \mathbf{ASim}(X_i, Y_i)}{a \sum_{i=1}^{i=a} w_i + b \sum_{i=1}^{i=b} w_{a+i} + c \sum_{i=1}^{i=c} w_{a+b+i}} \quad (10)$$

Where:

- X and Y are two profiles belonging to P;
- a is the set of common characteristics of X and Y;
- b is the set of characteristics existing in X and not existing in Y;
- c is the set of characteristics existing in Y and not existing in X;
- ***ASim*** is the atomic similarity between each characteristic of X and Y. ***ASim*** is defined as follows:

$$\boldsymbol{ASim}: R^+ \times R^+ \rightarrow [0,1]$$

$$\boldsymbol{ASim}(X_i, Y_i) = \begin{cases} 1 & if\ (X_i = Y_i) \\ 0 & if\ ((X_i \neq Y_i) \wedge (\text{type} = \text{Qualitative})) \\ \frac{\min(X_i, Y_i)}{\max(X_i, Y_i)} & if((X_i \neq Y_i) \wedge (\text{type} = \text{Quantitative})) \end{cases} \quad (11)$$

ASim checks the properties of similarity measures [35].

The distance (dissimilarity) corresponding to ***QSim*** is defined as follows:

$$\boldsymbol{Dist}: P \times P \rightarrow [0,1]$$

$$\boldsymbol{Dist}(X,Y) = 1 - \boldsymbol{QSim}(X,Y) \quad (12)$$

$$\boldsymbol{Dist}(X,Y) = 1 - \frac{a\sum_{i=1}^{i=n} w_i * ASim\ (X_i,Y_i)}{a\sum_{i=1}^{i=a} w_i + b\sum_{i=1}^{i=b} w_{a+i} + c\sum_{i=1}^{i=c} w_{a+b+i}} \qquad (13)$$

III.2) Services Discovery Result

The result of the discovery process is a list of services. The presentation of this list, provided by the system, can be made in three different ways:

- **Presentation of the required companies:** In this presentation, we present to the user all the companies answering its request with their descriptions and their activities. While clicking on one of the results (company name), the system presents all the services belonging to the selected company.

- **Presentation of the results for syntactic search:** We can present to the user services corresponding to his syntactic search. We can list the related services to a concept with other information describing the exposed services.

- **Presentation of the results for advanced search:** In this view, we present all the services answering the advanced request of the user associated with links to reach the service description. The list of services is ordered according to their relevance. Services are classified in a decreasing order, indicating the correspondence, calculated by ***QSim***, between the user context and the services contexts.

IV. QSim versus Jaccard

In order to evaluate the relevance of our similarity measure, we compared the results given by ***QSim*** and the results given by the ***Jaccard*** similarity measure (***JSM***) used in our previous work [28].

IV.1) Tests Environments

Let **S** be a service having 10 contextual versions (**CV1** to **CV10**).

We suppose that we have five clients with five different contexts of use. The clients search for the same service.

Table 3 represents the different contexts attributes with various contextual versions of the service sought by the user.

Table 3 Contextual attributes of the service version.

Context attributes	CV1	CV2	CV3	CV4	CV5	CV6	CV7	CV8	CV9	CV10
Performances	10	66	40		86	15		100	23	
Capacity		23	2		90	60	40		32	57
Reliability	No	Yes		Yes	Yes	No		Yes	No	No
Availability	Yes	Yes	No	No			Yes	No	No	No
Flexibility	No	No	No	Yes	Yes	No	No	Yes		No
Security	Low		High	High	Moderate	Low		High	Moderate	
Cost	50	44	100	150	60	65	80	100		90
Language (ISO 639-2 Code)	Chi		Hin	Chi	Chi	Chi	Ara		Eng	

Table 4 User Contextual attributes

Context attributes	U1	U2	U3	U4	U5
Performances	High	Moderate			Low
Capacity	Moderate	Low	High	High	Moderate
Reliability	Reliable	Unreliable	Reliable		Unreliable
Availability	Not available	Available	Not available	Not available	Not available
Flexibility	Inflexible	Flexible	Flexible		Inflexible
Security	Low	High	Moderate		Low
Cost	50	70	200	30	100
Language	Dut	Zha	Chi	Chi	Chi

The objective of the tests is to confront the two similarity measures (***JSM*** and ***QSim***) and to compare the obtained results with the distance of ***Manhattan (DM)***.

This distance indicates the dissimilarity between users' contexts and services' contexts. The distance of *Manhattan* is calculated with the following formula:

$$\boldsymbol{DM}(X,Y) = \frac{1}{2} * \sum_i |X_i - Y_i| \qquad (14)$$

IV.2) Tests Results

Below, the results obtained by the two similarity measures (***QSim*** and ***JSM***) during the discovery process according to users contextual attributes.

Figure 6, **Figure 7**, **Figure 8**, **Figure 9** and **Figure 10** represent the similarity degrees obtained between each user request and the various contexts of services.

Figure 11 indicates the number of discovered services for each request. In the bar chart we represent only services which have a similarity degree higher than 50%, i.e. that half of the user needs are satisfied by the service.

U1	Ctxt1	Ctxt2	Ctxt3	Ctxt4	Ctxt5	Ctxt6	Ctxt7	Ctxt8	Ctxt9	Ctxt10
JSM	**8 7%**	**7 5%**	**8 7%**	**7 5%**	**8 7%**	**8 7%**	**6 2%**	**6 2%**	**7 5%**	**62 %**
QSim	**6 0%**	**5 6%**	4 5%	4 9%	**5 4%**	**6 1%**	4 1%	4 1%	4 2%	**56 %**
DM	5 3%	4 7%	5 7%	5 4%	4 2%	3 3%	3 4%	5 9%	6 5%	24 %

U2	Ctxt1	Ctxt2	Ctxt3	Ctxt4	Ctxt5	Ctxt6	Ctxt7	Ctxt8	Ctxt9	Ctxt10
JSM	**8 7%**	**7 5%**	**8 7%**	**7 5%**	**8 7%**	**8 7%**	**6 2%**	**6 2%**	**7 5%**	**62 %**
QSim	**5 9%**	4 9%	4 5%	**5 2%**	4 7%	**5 4%**	**5 8%**	4 5%	4 6%	**51 %**
DM	5 6%	4 2%	3 3%	4 0%	4 5%	3 6%	3 5%	4 3%	6 8%	9%
U3	Ctxt1	Ctxt2	Ctxt3	Ctxt4	Ctxt5	Ctxt6	Ctxt7	Ctxt8	Ctxt9	Ctxt10
JSM	**7 5%**	**6 2%**	**7 5%**	**7 5%**	**7 5%**	**7 5%**	**6 2%**	5 0%	**6 2%**	**62 %**
QSim	**5 3%**	4 0%	4 0%	**8 7%**	**8 2%**	**5 6%**	**5 5%**	**5 3%**	**6 3%**	**60 %**
DM	4 3%	5 0%	4 3%	2 5%	3 3%	2 3%	1 6%	4 9%	5 4%	2 1%

U4	Ctxt1	Ctxt2	Ctxt3	Ctxt4	Ctxt5	Ctxt6	Ctxt7	Ctxt8	Ctxt9	Ctxt10
JSM	3 7%	5 0%	**6 2%**	5 0%	3 7%	3 7%	**8 7%**	1 2%	**6 2%**	**62 %**
QSim	2 4%	3 7%	3 3%	4 9%	2 7%	2 3%	**7 2%**	3 %	**5 0%**	**65 %**
DM	4 7%	6 8%	8 1%	5 1%	3 2%	2 3%	2 8%	8 5%	3 8%	4 7%

U5	Ctxt1	Ctxt2	Ctxt3	Ctxt4	Ctxt5	Ctxt6	Ctxt7	Ctxt8	Ctxt9	Ctxt10
JSM	**8 7%**	**7 5%**	**8 7%**	**7 5%**	**8 7%**	**8 7%**	**6 2%**	**6 2%**	**7 5%**	**62 %**
QSim	**7 9%**	4 4%	**5 2%**	**6 0%**	**5 9%**	**9 1%**	**5 5%**	3 6%	**7 1%**	**69 %**
DM	2 8%	4 4%	5 2%	3 2%	2 3%	8 %	1 2%	5 5%	3 9%	36 %

The first remark relates to the results obtained by the ***JSM***. Indeed we note according to the histograms that the similarity degree is always high even if there is not a big correspondence between the user request and the service context.

That is due principally to the fact that the ***JSM*** tests only the existence of the requested attribute in the service context, without taking into account its value. In this case, we have some examples: User 1 with Context 5 (an 87% similarity degree with only one common attribute), User 1 with Context 3 (an 87% similarity degree with two common attributes), User 2 with Context 1 (an 87% similarity degree with two common attributes), User 3 with Context 3 (an 75% similarity degree with one common attribute)...

Contrary to the ***JSM*** curves, we note that the values represented by the ***QSim*** curves are more realistic if we compare the set of services contexts and the users' requests.

We also note that the values given by the quantitative measure are in inverse proportion to the values given by the distance measure and reflect the real degree of correspondence. Indeed, we notice that the more important the values given by our measurement are (high degree of similarity) and the more we have small values concerning the distance (low dissimilarity) thanks to the quantification of the context attributes by ***QSim*** measure.

As a result, we note that the number of discovered services (**Figure 11**) with the ***QSim*** measure is noticeably lower than the number of discovered services with the ***JSM*** (with a threshold of similarity of 50%). Thus, we can say that our quantitative measure makes it possible to refine services search and to propose to users only services which really correspond to their needs.

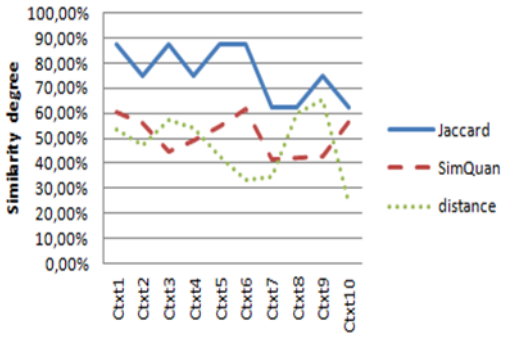

Fig. 6: Similarity results for User 1 request

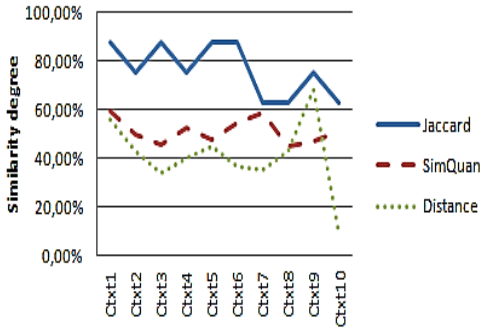

Fig. 7: Similarity results for User 2 request

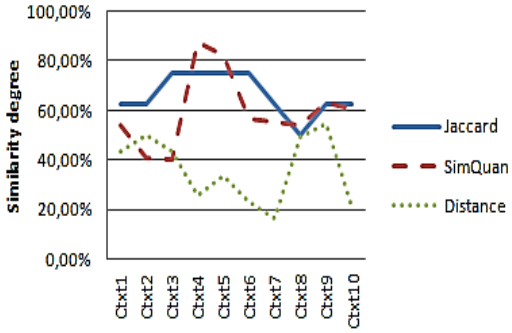

Fig. 8: Similarity results for User 3 request

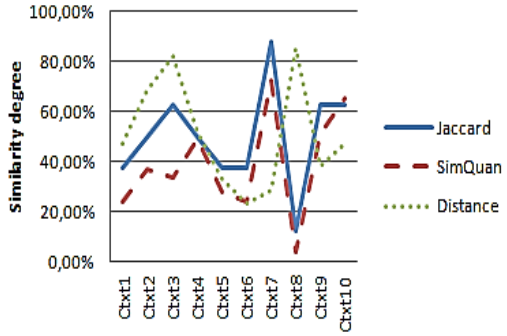

Fig. 9: Similarity results for User 4 request

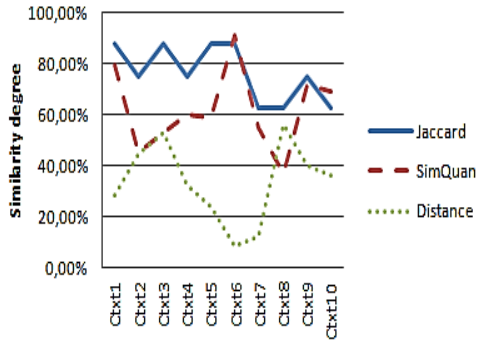

Fig. 10: Similarity results for User 5 request

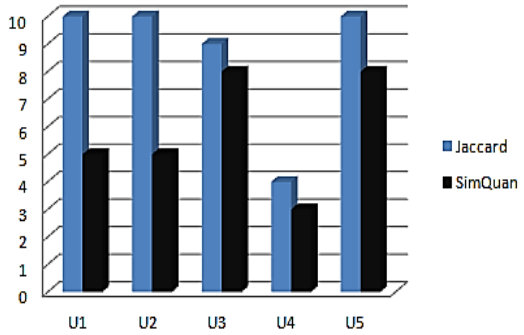

Fig. 11: Number of discovered services satisfying at least 50% of the users' needs

V. Conclusion and Perspectives

Web services search returns, generally, several results. Most of these results are likely to be non-appropriate to users' requirements. The discovery process must be able to increase the relevance of the results in order to approach the user requirements as much as possible.

In this paper, we proposed a framework for context-based web services discovery. Users' and services' contexts are modelled by a common vocabulary for the representation of the relative data of the two contexts. The various contexts are represented by a set of components.

Moreover, we proposed a discovery process according to users' contexts. The evaluation of the correspondence degree, between the client context and the service context, is carried out using a quantitative similarity measure. This new measure is more adapted, to services discovery, than the ***Jaccard*** similarity measure used previously.

Despite the fact the objectives were achieved, several perspectives proved to be promising and can bring more to our work.

The aim of these perspectives is to improve the performances of the system by:

- Exploiting a multi-agents system for a distributed UDDI architecture;
- Simplification of the search process;
- And optimization of Web services storage space.

These are the research directions that will guide our future work.

References

[1] Newcomer, E.: *Understanding Web Services: XML, WSDL, SOAP and UDDI*, Addison Wesley Professional, Canada, 2002.

[2] Christensen, E. Curbera, F. Meredith, G. and Weerawarana, S.: Web services description language (wsdl) 1.1. [Online] W3C publishing. Available from http://www.w3.org/tr/wsdl, 2001.

[3] Clement, L. Hately, A. Riegen, C.V. and Rogers,T.: Universal description discovery and integration. [Online]OASIS standard. Available from http://uddi.org, 2004.

[4] Gudgin, M. Hadley, M. Mendelsohn, N. Moreau, J.J. and Nielsen, H.F: Simple object access protocol (soap) version 1.2. [Online] W3C publishing. Available from http://www.w3.org/tr/soap/, 2003.

[5] Booth, D. Haas, H. McCabe, F. Newcomer, E. Champion, M. Ferris, C. and Orchard, D.: Web services architecture. [Online] Working Group Note produced by the W3C Web Services Architecture Working Group. Available from http://www.w3.org/TR/ws-arch/, 2004.

[6] Paolucci, M. Kawamura, T. Payne, T.R. and Sycara, K.: Semantic matching of web services capabilities. In ISWC 2002: Proceedings of the first International Semantic Web Conference on The Semantic Web, Italy, 2002.

[7] Beatty, J. et al.: Web services dynamic discovery (ws-discovery). [Online] Microsoft Corporation. Available from http://schemas.xmlsoap.org/ws/2005/04/discovery/, 2005.

[8] Chuanchang, L. Yong, P. and Junliang, C.: Web Services Description Ontology-based Service Discovery Model. In WI 2006: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, China, 2006.

[9] Grimm, S.:Discovery: Identifying relevant services. In Andreas (Eds), The Semantic Web Services: Concepts, Technologies and Applications, Springer-Verlag, Germany, 2007.

[10] M'Bareck, N. and Tata, S.: How to consider requester's preferences to enhance web service discovery. In ICIW 2007: Proceedings of the second International Conference on Internet and Web Applications and Services, Mauritius, 2007.

[11] Rompothong, P. and Senivongse, T.: A query federation of UDDI registries. In ISICT 2003: the proceedings of the first International Symposium on Information and Communication Technologies, Ireland, 2003.

[12] Palathingal, P. and Chandra, S.: Agent approach for service discovery and utilization. In HICSS-37: Proceedings of the 37th Annual Hawaii International Conference on System Science, Hawaii, USA, 2004.

[13] Martin, D. Burstein, M. Hobbs, J. Lassila, O. McDermott, D. McIlraith, S. et al.: Owl-s: Semantic markup for web services. [Online] W3C Member Submission. Available from http://www.w3.org/Submission/OWL-S/, 2004.

[14] Motta, E. Domingue, J. Cabral, L. and Gaspari, M.: Irs-II: A framework and infrastructure for semantic web services. In ISWC2003: Proceedings of the Second International Semantic Web Conference, USA, 2003.

[15] Vu, L. Hauswirth, M. and Aberer, K.: Towards p2p-based semantic web service discovery with qos support. In BPM 2005: Proceedings of the International Workshop on Business Process Management, France, 2005.

[16] Verma, K. Mulye, R. Zhong, Z. Sivashanmugam, K. and Sheth, A.: Speed-R: Semantic p2p environment for diverse web service registries. [Online] W3C Technical Report. Available from http://webster.cs.uga.edu/~mulye/SemEnt/Speed-R.html, 2004.

[17] Dey, A.K. and Abowd, G.D.: Towards a better understanding of context and context-Awareness. In CHI 2000: Proceedings of the Workshop on The What, Who, Where, When, and How of Context-Awareness, as part of the 2000 Conference on Human Factors in Computing Systems, Netherlands, 2000.

[18] Abowd, G.D. Dey, A.K. Brown, P.J. Davies, N. Smith, M. and Steggles, P.: Towards a better understanding of context and context-awareness. In HUC 2009: Proceedings of the first international symposium on Handheld and Ubiquitous Computing, Germany, 1999.

[19] Pokraev, S. Koolwaaij, J. and Wibbels, M.: Extending UDDI with Context-Aware Features Based on Semantic Service Descriptions: In ICWS 2003: Proceedings of the International Conference on Web Services, USA, 2003.

[20] Keidl, M. and Kemper, A.: Toward context/aware adaptable web services. In W3C: Proceedings of the 13th World Wide Web Conference, USA, 2004.

[21] Kouadri-Mostéfaoui, S. and Kouadri-Mostéfaoui, G.: Towards a contextualization of service discovery and composition for pervasive environments: In WSABE'2003: Proceedings of the AAMAS workshop on Web Services and Agent-Based Computing, Australia, 2003.

[22] Bruandet, M.F and Chevallet, J.P.: Utilisation et construction de base de connaissances pour la recherche d'information. In Hermès Sciences (Eds) Assistance Intelligente à la Recherche d'Information, pp 85-118, 2003.

[23] Vosniadou, S. and Ortony, A.: Similarity and Analogical reasoning, Cambridge University Press, USA, 1989.

[24] Slimani, T. Ben Yaghlane, B. and Mellouli, K.: Une extension de mesure de similarité entre les concepts d'une ontology. In SETIT 2007: Proceedings of the 4th International Conference on Sciences of Electronic, Technology of Information and Telecommunications, Tunisia, 2007.

[25] Bisson, G.: KBG: Induction de Bases de Connaissances en Logique des Prédicats. Unpublished PhD thesis, Paris XI, France, 1993.

[26] Rajman, M.: Similarités pour données textuelles. In JADT 1998: Proceedings of the 4th Journées Internationales d'Analyse Statistique des Données textuelles, France, 1999.

[27] Belkhirat, A. Bouras, A. Belkhir, A.: A new similarity measure for the anomaly intrusion detection. In NSS 2009: Proceedings of the Third International Conference on Network and System Security, Australia, 2009.

[28] Bouyakoub, F.M. and Belkhir, A.: A similarity measure for the negotiation in web services. Multimedia Tools and Applications, Vol 50, pp.279 -312, 2009

[29] Shercliff, G. Stockreisser, P.J. Gray, W.A. Deora, V. Shao, J. and Fiddian, N.J.: Incorporating qos specifications in service discovery. In WISE 2004: Proceedings of the Web Information Systems Workshops, Australia, 2004.

[30] Bouyakoub, F.M. and Belkhir, A.: AdaMS: an adaptation multimedia system for heterogeneous environments. In NTMS 2008: Proceedings of the Second International Conference on New technologies, Mobility and Security, Morocco, 2008.

[31] Maesano, L. Bernard, C. Le Galles, X.: Services Web avec J2EE et .NET: conception et implémentation, Eyrolles edition, 2003.

[32] Bouyakoub, F.M. and Belkhir, A.: Automatic generation of user's profiles for location-based adaptation of multimedia documents. In GenCWiNets'08: Proceedings of the First IEEE International Workshop on Generation C Wireless Networks, USA, 2008.

[33] Klyne, G. Reynolds, F. Woodrow, C. Ohto, H. Hjelm, J. Butler, M.H. and Tran, L.: Composite Capability/Preference Profiles (CC/PP) : Structure and Vocabularies 1.0. [Online] World Wide Web Conference (W3C). Available from http://www.w3.org/TR/CCPP-struct-vocab/, 2004.

[34] Bouyakoub, F.M.: Négociation de services pour les téléphones de 3éme génération. Unpublished PhD Thesis, University of Sciences and Technology Houari Boumediene USTHB, Algeria, 2010.

[35] Belkhirat, A. and Belkhir, A.: A new similarity measure for the profiles management. In UKSIM 2011: Proceedings of the Tenth International Conference on Computer Modeling and Simulation, England, 2011.

Authors' Profiles

Bouyakoub Fayçal M'hamed is an assistant professor at the computer science department (University of Sciences and Technology Houari Boumediène USTHB - Algiers) and a member of the VAAL research team (LSI Lab).

He received his PhD degree in computer science from USTHB University in 2009.

His research activities focus on Web services, multimedia systems, quality of service and services adaptation and negotiation.

Belkhir Abdelkader: is a professor at the computer science department, faculty of electronic and computer science, USTHB University, Algiers. He leads a research team whose theme is multimedia and computer security (LSI laboratory). His research field is multimedia and computer security.