Journal of
Computer Science & Systems Biology

Research Article Open Access

B-WSD Approach and Validation: Use Satisfaction Survey

Messaoud WB*

Universite de la Manouba, Ecole Nationale des Sciences de l'Informatique, Tunisia

Abstract

Web service discovery deals with the satisfaction problem facing consumer requirements. In literature, some approaches have addressed the issue of discovery without considering the consumer requirements or the runtime environment. Faced with the complexity of consumer needs and diversity of responses on the web, the consumer hardly selects the service that satisfies him. Thus, B-WSD (Behavioral Web Service Discovery) approach is the solution that we provide in this context, the principle is to conceive a discovery system that meets consumer expectations and filters provider offerings to achieve a satisfaction compromise between these two actors.

Keywords: Web service; Web service discovery; Behavioral aspect; B-WSD approach

Introduction

The web service technology has the most important role in the development of applications and distributed systems on the internet. It is the most agile way of interaction between the various actors of web such as; web service consumer, web service provider, service agency and their various platforms. The Web service is an emergent and promising technology for the development, the deployment and the integration of the internet applications. The description of the web services allows the customers to search, to discover and to invoke a service.

One of the contentious issues in literature is the web service discovery. It refers to the process of search which implements the wished functionality by interviewing service directories to know which web service is available. The discovery process follows the launching of the consumer query that contains a set of functional and not functional requirements. This process takes place in two phases the first one serves to search the service via services directories; the second selects the service among several candidates respecting consumer needs.

In front of the complexity of his needs and the diversity of the answers on the web, the consumer has difficulty to select the service which satisfies him. So, a system of analysis and translation of needs would allow assuring a compromise of satisfaction between the service consumer and the service provider. It is in this context that we proceeded in the implementation of a new approach of web service discovery based on the behavior aspect by considering the requirements of the consumer as a basic criterion in the process of discovery.

Web Service

To reach web, most of people use a web browser, which supplies an interface dedicated to the presentation of the information and the services centered on the user such as online auctions and retail shops. When a user asks for a web page, the request is handled by a distant web server, which sends back the information in hypertext marking language (HTML).

Definitions

To understand the concept of web service (SW), we present several definitions adopted by standards and certified works.

W3C definition: The web service is a software system identified by an URI, whose public interfaces and bindings are defined and described in XML. Its definition can be dynamically discovered by other software systems. These other systems can then interact with the web service in a manner described by its definition, using XML messages transported by the Internet protocols [1].

Other definitions: Web services are distributed software components published by providers, which transmit information flow to the applications across an interface. They enable interoperability between different software systems over a computer network. Web services are accessed via the web by well-known protocols; described in XML, interact via XML; very loosely coupled, localizable in registers, fully transversal to the platforms [2].

From these definitions, a main idea emerged, namely that a service can expose one or more functionalities offered by a provider to potential customers. The web service can be defined as methods that can be called remotely via a network; they deliver data to computers in contrast to services based on the content that deliver web pages (generated statically or dynamically) intended for human consumption.

Proposition: The web service is a software application developed and published by a service provider in a service directory, described in a WSDL file, invoked by a consumer application that requires its functional and non-functional properties. Once used, the service consumer can express his opinion by a feedback which will be analyzed to overcome the service issues in future developments.

Characteristics

Interoperability: Interoperability presents the main characteristic of web services. Indeed, regardless of the platform used (Windows, UNIX, etc.) and the used development language, web services are based on XML standards to simplify the construction of distributed systems and cooperation between them.

Simplicity of use: Using standards such as XML and HTTP highlighted the simplicity of handling web services and encouraged the strong market actors such as Microsoft and IBM to integrate new products.

***Corresponding author:** Messaoud WB, Universite de la Manouba, Ecole Nationale des Sciences de l'Informatique, Tunisia, Tel: +21671600444; E-mail: ben.m.wala@gmail.com

Received September 21, 2017; **Accepted** October 04, 2017; **Published** November 09, 2017

Citation: Messaoud WB (2017) B-WSD Approach and Validation: Use Satisfaction Survey. J Comput Sci Syst Biol 10: 098-105. doi:10.4172/jcsb.1000257

Loosely coupled applications: Web Services is an exchange intermediate structured documents that pass through the access controls in a heterogeneous environment. Collaboration between different applications to exchange documents is a direct way between objects.

Web service concepts

The standard specifications used by Web services are:

XML (Extensible Markup Language): XML is a generic markup language for structuring data to make them readable by both humans and programs. It is indeed a language to get fit documents using tags (markup). XML strength lies in its ability to describe any data domain through its scalability. It provides structure, pose the vocabulary and syntax of the data that will contain.

SOAP (Simple Object Access Protocol): SOAP is a protocol described in XML and standardized by the W3C; it sets the general framework for the exchange of data structured in XML. His goal is to invoke remote service regardless of how the service is implemented [3]. SOAP is used to call an RPC method (Remote Procedure Call) and send messages to remote machines via HTTP.

WSDL (Web Services Description Language): A WSDL document is a collection of items in a root element definition. These elements describe the service and how to implement an endpoint whose service is available [4]. Each WSDL comprises two parts, an abstract and concrete. The abstract part describes the messages sent and received. The transaction combines a model of message exchange via one or more messages. For the concrete part, it specifies the transport and format details for one or more interfaces. A port (an endpoint), which combines a network address with a binding agent and service that includes the terminal implement a common interface. An operation is similar to a function in a high level programming language. A message exchange is also designated as an operation. The operations are the focal point of interaction with the service.

UDDI (Universal Description Discovery and Integration): UDDI is an oriented Business directory created by IBM, Microsoft and Ariba in three versions; V1 (September 2000), V2 (June 2001) and V3 (July 2002) [5]. UDDI is a central element of the group of related standards that constitute the stack of web services. The UDDI specification defines a standard method for publishing and discovering network-based software components in a service-oriented architecture (SOA). Its development is led by the OASIS consortium of enterprise software providers and customers.

Interaction between web service concepts

Web services interact through a set of core technologies mentioned previously that share a common architecture. They were designed to be made on many systems developed and deployed independently. These technologies interact with each other according to Figure 1.

Web service invocation: Web services are software programs that allow communication and data exchange between heterogeneous systems and applications in distributed environments. They are intended to be independent of the basic components that collectively represent an application environment [6]. With the growth of web services, the interoperability issues are evolved, which emphasizes the need to improve existing technologies and approaches such as selection, composition, choreography and discovery.

Web service discovery: The web service discovery refers to the research process that implements the desired functionality by querying service books to find out which web service is available. The similar features of different web services pushed consumers to find new techniques to research their services according to their requirements. The authors of [7] claim that the web service discovery approaches have been emerged respectively as follows: syntactic approaches, semantic approaches, contextual approaches and behavioral approaches.

Web service selection: In the UDDI registry, the web services can offer similar functionality, so the research should be more detailed in order to ensure the quality of services sought. A selection phase is required to filter the most relevant web service suitable to the consumer requirements.

Hence the selection phase is based on functional and non-functional requirements extracted from the consumers query to fully meet their preferences. Non-functional requirements are generally expressed with the QoS requirements (Quality of service) such as availability, performance, summoning cost, reliability. Functional requirements reflect the operation of the service; they mean the operations that the web service can provide.

Web service composition: The composition of web services is defined as a set of sequenced services where the invocation and the execution provide a response to a need. Two main approaches are proposed; dynamic approach, in which a service is a composition process that is initiated by a request. It leads to discover, select and dynamically assemble services to meet the request. The dynamic approach meets the needs of flexibility and adaptation. It allows building a new composition for every request taking account of available services for the composition.

In contrast, the static approach considers a services composition as a predefined artifact. This artifact describes the planning of services to be invoked. This approach adapted to stable applications, including business logic, is fully defined during design and does not take into account specificities of the initial request [8]. Static composition techniques are defined using business process; orchestration and choreography.

SOA

The SOA is a software design approach that dissolves business applications into separate services that can be used independently of the applications (which they are part) and computing platforms (on which they run). SOA is an open, scalable and modular architecture that promotes to structure an application in contract-services in order to overcome the problems of interoperability, reuse and application continuity.

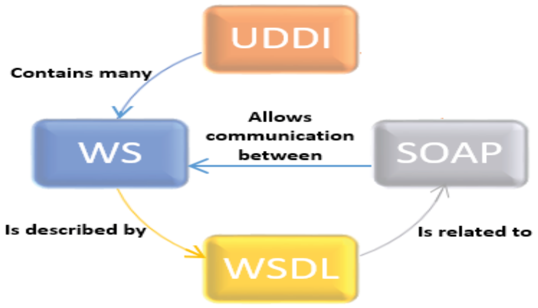

Figure 1: Interaction between web service concepts.

Definitions

Several definitions have been given in the literature:

- A service-oriented architecture is a multi-tier style of software architecture that helps organizations to share business logic and data between multiple applications and multiple usage models [9].
- Service-oriented architecture is a paradigm for organizing and using distributed expertise that can be of various domains. This provides a uniform means to offer, discover, interact and use expertise to produce the desired result with preconditions and measurable goals [10].
- The service-oriented architecture enables integration of applications and resources in a flexible manner; by representing each application or resource as a service exposing a standardized interface, enabling a service to exchange structured information (messages, documents, "business objects") and coordinating and organizing the services in order to ensure that they can be invoked, used and effectively changed [11].

SOA and web services

In the service-oriented architecture, web service takes a key role in defining software respecting these architectures dealing with a specific aspect (discovery, selection, composition, publication). Four service properties arise:

- A service is autonomous: Services can be completely autonomous, but can also depend on the availability of other services, or the existence of a resource.
- A service must exercise strong control over its execution environment underlying. Over this control is stronger, the execution of a service is predictable [12].
- A service is stateless: stateless, is a service that does not remember anything between two calls. The operating without state has many advantages; the stateless does not retain the information (reserve resources on indefinite periods), no risk of forgetting to close a session and each invocation of the service is completely independent of the foregoing.
- A service exposes a Contract: All services of the same technical system are exposed through contracts following the same standardization rules [12].
- The services communicate through messages.
- The services are loosely coupled: The software components can find the information they need when they need them. The contract of a service must impose a weak coupling of its customers.

According to this architecture, the life cycle of a web service can be divided into four phases [13]:

- Identification: This role is associated with the business analyst who defines business processes, identifies the business services and optimizes processes through simulation.
- Specification: This is the second phase aggregated to the architect who used to model services.
- Development: This phase is affiliated with the developer who implements the services.
- Management: The manager publishes services, manages the life cycle and controls the quality of service.

Actors

SOA is based on a basic three actors interacting with each other.

Service provider: The service provider implements the web service and makes it available on the internet. He offers the service implementation, provides a service description through a WSDL document depicting its properties, its operation and the conditions of its use and publishes this specification in a directory. The provider is the agent responsible for the quality of service. He can improve the quality of his services by considering the consumers' views that has already used the service.

Service consumer: The service consumer is an application running on a server and comprising a software module accessed in XML. It a client program launching a service request containing the required needs and aiming to find a clear answer. The consumer can be satisfied or not by search results. Thus, it can send feedback to the service provider mentioning the improvements that can be successful in the next service developments.

Service directory: The service directory is a service book. It provides a central location where providers can publish and consumer can seek services. It offers the service description and its location.

Interaction between actors

The interactions between the three actors of SOA follow several steps:

• Definition, deployment and service description: This interaction is defined by WSDL descriptions that answer the question what is the functionality provided and how to access it.

• Publication: The provider diffuses descriptions of its web services in the directory.

• Composition: The provider has the possibility to recover several descriptions of different services and dial them for advanced functionality.

• Discovery: The customer is looking for a particular service in a directory that will provide descriptions and URLs of the required service.

Web Service Discovery

The web service discovery is the service search process, triggered by a request, aimed to meet the needs of the consumer by interviewing the services agency. Each discovery approach defines a description model for the sought services and evokes its representation manner by a well-defined formalization. The service discovery process can adopt a syntactic and/or semantic matching. It provides information on matching between the concepts used to describe the web service properties and the properties described in the service request of the consumer.

Discovery phases

To properly apply the process of web service discovery, two critical phases will be defined; the basic phase named ''Research phase" where the service consumer seeks the service using the functional aspect (the sense of service, its inputs/outputs, the production parameters, preconditions, and effects). This phase ensures that services meet the basic requirements of service consumption. A secondary phase named "Selection phase" that allows selecting the most appropriate service if several candidate services met the needs of consumers (business requirements and/or non-functional requirements).

Web service discovery approaches

The majority of discovery approaches are listed in four categories; syntactic, semantic, contextual and behavioral. Syntactic approaches are generally based on the comparison between the query based on keywords and syntax descriptions of the web service (WSDL), such as UDDI [14] and AASDU (Agent Approach for Service Discovery and Utilization) [15].

The proposed semantic approaches as iMatcher1 [16], OWLS-iMatcher2 [17] and PSWSD, are all based on the same technique of calculating the level of semantic correspondence between the functional parameters of services and functional parameters mentioned in the consumer's request. PSWSD approach adopts WSMO ontology while the approaches iMatcher1 and OWLS-iMatcher2 adopt OWL-S ontology. This ontology, according to [18] is more mature in some aspects such as the definition of the process model of the web service.

However, WSMO has significant advantages compared to OWL-S; its conceptual model has better separation between consumer and provider, it includes web service orchestrating enabling static or dynamic reuse of services in case of composition and provides formal semantics for the choreography of the web service.

Contextual approaches are based on the notion of context. The context is the set of all the information that can be used to characterize the situation of an entity, an entity is a person, place or object that may be relevant to the interaction between the user and the application, including the user and the application themselves [19]. We cite the CASD (Context Aware Service Discovery) approach [20]. It is based on the context in which a semantic discovery module is designed to determine the categories of services, which have a semantic relationship with the consumer query based on domain ontologies. This approach admits service storage as an OWL graphs.

The behavioral approaches are based on web service behavior which is described by sequences of messages, data types, constraints and data properties that specify the time within which the messages are exchanged [21].

Many approaches can be cited such as Behavioral Synchronization [22] that defines two types of behavior; the first is the control behavior that demonstrates the business logic and the second is the operational behavior which regulates the progress of the control behavior. The idea is to coordinate the two behaviors at the execution time by developing the conversation messages and transmitting information between these two behaviors.

Hence, web service discovery approaches cited in literature do not ensure the behavior criterion that defines the execution enchainment of web service operations. These approaches do not ensure reliable results satisfying the requirements of service consumer; they neglect the integration of non-functional properties with the functional properties in the selection phase. They do not use the notion of ontology in the discovery phase and the selection phase to fully translate the query of the service consumer in terms of functional and non-functional properties. So, it is mandatory to conceive a new approach that attenuates these issues.

B-WSD approach

This section highlights the importance of the behavioral aspect in web service discovery by supporting that the execution path of the service operations becomes a quality criterion which was not treated in the approaches mentioned above (Table 1).

Behavioral aspect: The behavioral aspect is defined by the execution manner of the web service operations. In fact, a consumer who requires searching a web service with a well-defined list of the requirements cannot be met by either the syntactic or semantic discovery. The consumer requires a new aspect that allows a functional phase defining the basic sought functionality and a non-functional phase aiming to select the most appropriate service if more than one service satisfies the query exigency.

Architecture: The B-WSD approach follows SOA formalism. The interaction between the three layers spread out the service discovery process beginning with launching the consumer query and reaching the search mechanism, the selection and the consumer satisfaction (Figure 2).

Development layer: It is the layer that provides the service. The provider conceives and publishes web services in UDDI to be available with their QoS descriptions and WSDL descriptions (1). Once the consumer demands the service, the discovery system analyzes the request of the consumer as a first step and then translates each keyword to be understood by the service developer. This translation specifies the functional requirements of those non-functional (3).

The developer manages Web service discovery by querying the UDDI intermediate extended to collect information on Web services available based on the functional and non-functional requirements (4). If the result of the research phase is a set of services, a selection phase will be required to filter the most relevant service answering the requirements outlined in the consumer query, the service developer applies filtering services. The result of the selection is a set of cooperated and inter-operated web services forming a composite web service (5).

The service developer then is the responsible for deploying the composite web service and executing in a web services runtime. The

	Ordinary discovery method	B-WSD approach
FP	51%	71%
NFP	18%	69%
Execution cost	20%	68%
Quality/Price	16%	68%
Waiting time	28%	55%
Service duration	18%	74%
Service buyback	18%	63%

Table 1: Satisfaction comparison with and without B-WSD approach.

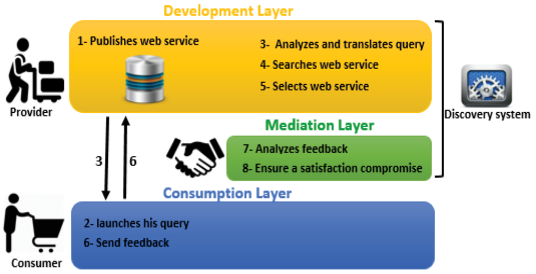

Figure 2: B-WSD in layers.

result of the execution will later be transferred to the consumer.

Consumption layer: It is the layer that requests the service. The consumer is the agent responsible for triggering discovery process. He launched a query as keywords describing its functional and non-functional requirements (2). This query will be subsequently considered by the service developer. Once the system discovery sends discovery results to the consumer, this latter may leave a message describing his complaint (6).

Mediation layer: It is the middle layer that holds the service information between the consumption layer and the development layer. It serves to analyze and translate the consumer request in a language understood by the service developer. The consumer feedbacks will be analyzed by the discovery system and sent to provider to improve his future conceptions of services (7). The mediation layer represents the intermediary between the consumer demands and the provider offers, it maintains a satisfaction compromise between the two layers (8).

Implementation

Query analyze: The step 1 of the discovery process handles the separation between functional properties (FP) and non-functional properties (NFP) from the consumer query. According to B-WSD approach, the consumer query is a set of keywords that define its requirements. If this query is not well specified, the risk of falling in case of linguistic polysemy is almost obvious. So we had to search for synonyms at all entrances of the service query to properly filter the service concerned.

Each entrance of the service query belongs to WordNet file saved in a location specified by the developer containing all possible synonyms. The authors of [23] and [24] defines WordNet as one of the sets of electronic dictionaries compiled manually, it is limited to a specific area and covers most English nouns, adjectives, verbs and adverbs.

Keywords translation in automaton: The step 2 of the discovery process treats the translation of the query keywords to a set of functional operations sequenced in an automaton to facilitate the satisfaction of the consumer needs. The automaton is represented as a directed graph known as state graph which consists of a finite set of vertices known as nodes, together with a set of directed links between pairs of vertices called arcs. Vertices are represented by circles and arcs by arrows. We can also represent an automaton with a state-transition table [25].

Before designing the automaton, functional properties should be divided in two categories; Functional properties type Functionality (FPF) and Functional properties type Quality (FPQ).

To conceive the automaton, we should define a set of operations basing on the FPF and the FPQ. These operations can be grouped in three categories [26]:

- Normal operation: is a FPF, its order is very important in the execution path. At each location in the automaton, we must calculate the FPQ and the NFP to select the most suitable WS according to the service consumer requirements.
- Free operation: is a FPF, it can be placed anywhere in the automaton.
- Parallel operations: Two normal operations that have the same functionality but may provide different results.

Semantic research: The step 3 is to search the web service semantically basing on the functional properties [16,17,27]. The discovery system launches a semantic search of services via the service directory by mentioning the FPF as search keywords. The service directory returns a well-defined list of candidate services. For each service returned, the discovery service extracts the execution path of its operations.

If it is a simple service, the execution path is extracted from WSDL and if it is a composite service, the execution path is extracted from BPEL4WS (Business Process Execution Language for Web Service) file. The extracted sequence is converted into a word that will be accepted or rejected by the automaton already defined. The words recognized by the automaton correspond to the web service accepted [26].

Web service selection: After designing the automaton, we extract the languages to accept. Hence we can apply a simple comparison between the sequenced operations of the offered service and the extracted languages of the automaton [28]. The last step is filtering the most relevant web service basing on the non-functional properties if more than one candidate service is accepted by the automaton. Thus, the service offering the best values of NFP required by the consumer is the most appropriate service.

In B-WSD approach, we apply a local selection; more precisely the Euclidean distance [29]. The principle is to calculate the distance between the consumer requirements and the provider offers basing on Q_oS parameters. The authors in [29] propose as a first step a matrix representation of the web service such that each service is represented by a set of Q_oS parameters.

- S={WS_1, WS_2, WS_3, …. WS_n}: a set of web services.
- QPc={qpc_1, qpc_2, … qpc_k}:QoS parameters required by the consumer.

Each Q_oS line represents a web service and each column represents the values of Q_oS parameters (Figure 3).

The second step in [29] is to calculate the Euclidean distance between the QoS parameters required by the service consumer and those of the web services offered by the provider.

$$dist(QPc, QP_i) = \sqrt{\sum_{j=1}^{k} \left(qpc_j - qp_{ij}\right)^2}$$

Once the distance is calculated, we end by choosing the web service with the minimum value.

Guiding environment: This section presents the aspects and technological choices related to the implementation of the techniques presented in the previous sections as part of achieving the B-WSD approach. It seemed essential to develop a guiding environment for the B-WSD approach to make available to users the results we have achieved. This guiding environment is designed through a succession of interfaces dedicated to a set of actors according to their roles and privileges.

An evaluation of the approach is implemented to position our results compared to results of other approaches. The guiding environment should guide the user as a service consumer to formulate his query, as a service provider to develop and publish his service, as discovery system to arrange queried and responses and as a service directory to manage the different actors and services.

In the diagram of the Figure 4, we present all the modules of the platform, and the components present in each module. We organized the components relative to design pattern MVC 2. We propose a brief description of the various modules and components:

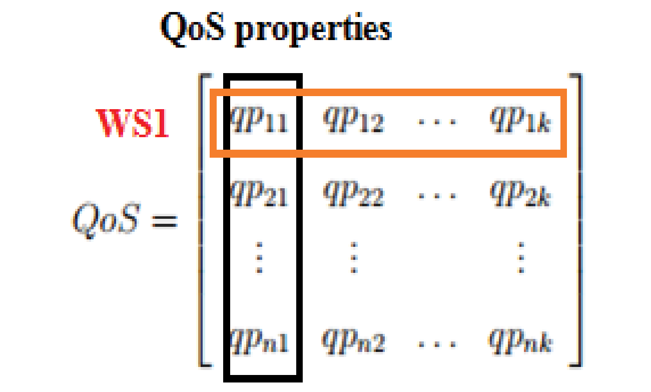
QoS properties
WS1
$QoS =$
qP_{11} qP_{12} … qP_{1k}
qP_{21} qP_{22} … qP_{2k}
qP_{n1} qP_{n2} … qP_{nk}

Figure 3: Matrix representation of the web service.

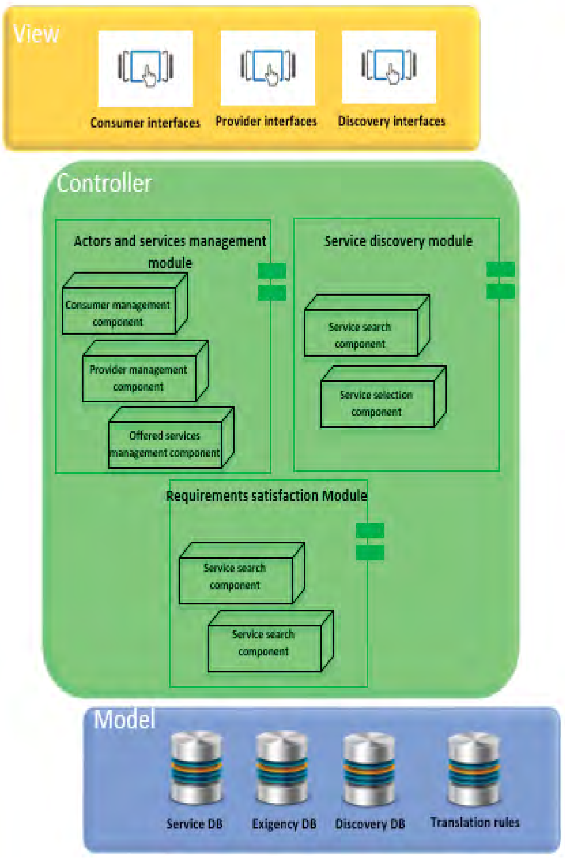

Figure 4: General architecture of the guiding environment.

Actors and services management module

- **Consumer management component:** Consumer management is the task performed by the service directory. The principle is to add, delete or update the consumer information.
- **Provider management component:** Provider management is the task performed by the service directory. The principle is to add, delete or update the provider information.
- **Offered services management component:** Offered services management is the task performed by the service directory. The principle is to list the services according to consumer demand and send it to the discovery system.

Service discovery module

- **Service search component:** Considering the query analysis result, the discovery system develops the discovery automation then launches semantic search within the service directory.
- **Service selection component:** It defines the selection phase of the most relevant web services. Once the automaton is developed and the result of the semantic search is ready, the selection is made by trying the services operations sequencings by the automaton. The accepted sequencings are filtered as the most relevant web services.

Requirements satisfaction module

- **Consumer query analysis component:** The discovery system analyzes the consumer query by translating consumer inputs to functional and non-functional requirements. As for functional requirements, the system separates the functionality requirements from quality requirements.
- **Feedback component:** Every consumer has the possibility of leaving a service satisfaction message. The discovery system analyses this message and sends the result of the analysis to provider to improve his future designs of web service.

Validation: The validation is to assess the adequacy of the system developed compared to the needs expressed by its future consumers [3]. Validation methods are techniques to rigorously reason about computer programs or electronic equipment, to demonstrate their validity relative to a certain specification. As a method, the satisfaction survey is an important part in the quality management. From the moment that the organization assigns a priority to the consumer's opinion, it is essential to organize the way in which we will collect and analyze the elements relating to expectations and satisfaction of the consumer. The Satisfaction is defined as the consumer opinion resulting from the difference between the perception of the service consumed and the consumer expectations.

In order to validate the B-WSD approach, we opted for an assessment. First of all, we apply a satisfaction survey containing a number of questions dependent on satisfaction criteria chosen by the service consumer in the formulation of his query. We choose 51 service consumer called "Reservation stay" we apply the questionnaire without B-WSD approach then we re-apply with the B-WSD approach. Questions are developed based on the NFP and FPQ required by the consumer. The result of this questionnaire assesses whether the consumer is satisfied by the choice of web services offered.

Questionnaire development: The questionnaire is divided into three parts; a first part is devoted to consumer identification, a second part defines a set of satisfaction criteria that may be required by the service consumer and a third part contains the different issues of satisfaction based on the criteria selected in the second part:

Consumer identification

Q1: What is your sector (industry, services, education, other)?

Q2: Can you define your professional category (student, unemployed, businessman, frame, artist, trader, worker, retired, other)?

Q3: Can you specify the interval of your incomes (1..7)?

Less than 300 TND	1
[300 - 449 TND]	2
[450 - 799 TND]	3
[800 -1199 TND]	4
[1200 - 1799 TND]	5
[1800 - 2999 TND]	6
3000 TND and more	7

Q4: How often do you use this service (All the time, very often, quite often, not often, never)?

Satisfaction criteria

Q5: What are the services criteria required in your query (multiple choice answers) (simplicity of use, availability, ergonomics, cost, time, robustness, security)?

Satisfaction

Based on satisfaction criteria required in your service request:

Q6: Are you satisfied with the analysis of your functional requirements (very satisfied, satisfied, slightly satisfied, dissatisfied)?

Q7: Are you satisfied with the analysis of your non-functional requirements (very satisfied, satisfied, slightly satisfied, dissatisfied)?

Simplicity of use

Q8: What is your opinion about the difficulty of your request formalization (very easy, easy, medium, and difficult?

Q9: Overall, what do you think of the simplicity of the service request procedure (Extremely simple, simple, simple enough, and complicated)?

Availability

Q10: What is your opinion about service availability (readily available: found after a simple search, hardly available: found after multiple searches)?

Ergonomics

Q11: What is your general impression about the ergonomics of our service (excellent, good, good enough, and not good at all)?

Cost

Q12: Are you satisfied by the execution cost of the service offered (very satisfied, satisfied, slightly satisfied, and dissatisfied)?

Q13: How do you rate the quality/price ratio of this service (excellent, good, fair, not good)?

Time

Q14: What do you think about the waiting time before being served (Extremely short, slightly short, some long, long)?

Q15: What is your level of satisfaction about the service duration (very satisfied, satisfied, slightly satisfied, dissatisfied)?

Robustness

Q16: If mishandled, is the service interface remains stable (stable system, slightly stable, sometimes stable sometimes unstable, slightly unstable, unstable system)?

Security

Q17: What is the level of security of your data (high, medium, low)?

Fidelity

Q18: Do you buy back this service (yes, maybe, no)?

Complaint

Q19: Do you have claims to announce to the service provider (yes, no)? Which:

Recommendation

Q20: Do you recommend this service to a friend (yes, no)?

Questionnaire results: The target population is a set of 51 consumers. Firstly, we ask to apply their usual method of a search service "Reservation stay," then we have provided their questionnaire to answer the various questions. Secondly, we rebroadcast their questionnaire but this time after the application of our approach B-WSD. In this section, we examine the results of the questionnaire evaluation with and without the B-WSD approach. We established a set of criteria satisfaction as the majority choice of the target population.

We note that the ordinary discovery method meets the consumer FP to 51% while the B-WSD approach guarantees a satisfaction rate of 71% which explains a good understanding of the requirements of launched requests.

From the results obtained, the NFP in the ordinary discovery method are almost neglected (18%) compared to their importance (69%) in the results of the B-WSD approach.

More specifically, the services offered by the ordinary discovery does not meet consumer expectations in terms of execution cost, they provide a satisfaction rate of 20%. While in the same conditions, we notice that our approach provides a satisfaction rate of 68%. This one tells us about the importance of B-WSD approach explained by the relevant web service selection methods.

Regarding the quality/price ratio, the results are incomparable; a 16% of the target population is satisfied by the ordinary research compared to 68% satisfied by B-WSD approach.

Turning now to the time criterion, it is considered as a basic criterion for the majority of the target population. As for the waiting period, most consumers lose a considerable time before being served which does not meet their basic needs (28%). While B-WSD approach addresses these needs with a most to least interesting satisfaction rate (55%).

As for the service duration, in our example, it set the time lost in moving to reach the stay booking offer. Our approach considers this criterion as a challenge. It provides an ideal satisfaction rate equal to 74% while, the usual method provides only 18% as a satisfaction rate. This is explained in our work by integrating the scheduling process of service operations arranged by the automaton conception.

Finally, we deduce from Figure 5 that B-WSD approach has met the functional and non-functional requirements of consumers and this is spread by the service redemption percentage equal to 63% compared to 18% for the service redemption filtered by the ordinary discovery method. So service buyback rate judges the quality of service offered and predicts improvements to make, to ensure the total satisfaction of the consumer.

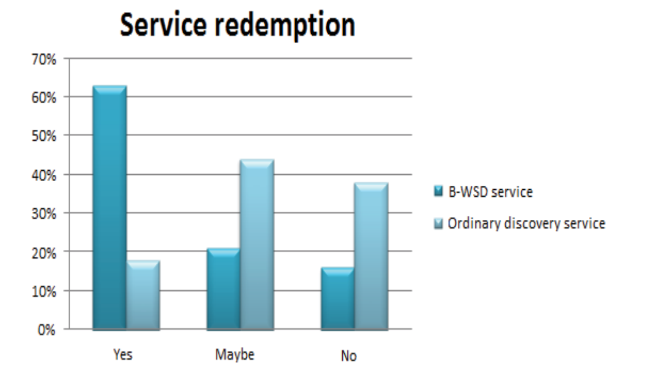

Figure 5: Comparison between redemption services.

Conclusion

Design and validation approaches of software architectures have changed cope with the complexity and the explosion of the web. To ensure the interoperability of distributed systems, software architectures have been developed. However most of these technologies, although having given a fairly satisfactory answer to interoperability in the context of distributed systems, were poorly suited for interoperability on the internet. This middleware is replaced by solutions overcoming these problems which interact and communicate via well-defined standards named today service-oriented architectures.

In this context, we attacked a problem in the theme of web services discovery. Hence the services discovery addresses the satisfaction problem cope with the consumer demands. In fact, resolve this problem amounts to solving certain problems related to the unconformity observed in a service discovery activity where the selected services may not meet consumer needs. Some approaches have addressed the issue of discovery without considering the requirements of the consumer or the runtime environment.

In this paper, we proposed a web service discovery approach based on behavioral aspects called B-WSD whence behavior aspect has been defined as the sequence of operations of a web service. It offers an extracting system for the functional and non-functional properties which is based on ontological modeling in order to properly analyze consumer needs and therefore the translation of consumer needs on technical needs understandable by the service developer. Once the problem is analyzed, the B-WSD approach proposes the modeling of a discovery system allowing services semantic search developed according to data from the consumer query.

References

1. Hugo H, Brown A (2004) Web Services Glossary. Available at: https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/
2. Curbera F, Cardoso J, Miller JA (2001) Web Services Overview. University of Georgia, USA.
3. Plumejeaud C. Service Web (SOAP). Available at http://www.youscribe.com/catalogue/ressources-pedagogiques/education/fiches-de-lecture/service-web-soap-1389492
4. Albreshne A, Fuhrer P, Pasquier J (2009) Web Service Orchestration and Choreography: Enabling Business Processes on the Web. Politecnico deo Milano, Italy.
5. http://www.uddi.org/pubs/uddi-tech-wp.pdf
6. Erl T (2004) Introduction to Web Services technologies: SOA, SOAP, WSDL and UDDI. Service-Oriented Architecture.
7. Messaoud WB, Ghedira K, Halima YB (2015) Web service discovery based on behavioral aspects. 2015 2nd World Symposium on Web Applications and Networking (WSWAN), IEEE, Tunisia.
8. Halima YB (2014) Une Méthode prédictive pour la composition de services basée intentions et exigences en vue d'une exécution sur des environnements hétérogènes. RIADI.
9. Schulte RW, Natis YV (1996) Service oriented architectures, Part 1. Gartner, SSA Research Note SPA-401-068.
10. MacKenzie CM, Laskey K, McCabe F, Brown PF, Metz R (2006) Reference model for service oriented architecture 1.0. OASIS Standard, p: 12.
11. Dodani MH (2004) From Objects to Services: A Journey in Search of Component Reuse Nirvana. Journal of Object Technology 3: 49-54.
12. Erl T (2008) SOA: Principles of service design. Upper Saddle River: Prentice Hall.
13. https://www.scribd.com/presentation/307096160/cours-SOA-AO-FB
14. Clement L, Hately A, von Riegen C, Rogers T (2004) UDDI Version 3.0.2. UDDI Spec Technical Committee Draft. OASIS UDDI.
15. Palathingal P, Chandra S (2004) Agent approach for service discovery and utilization. 37th Annual Hawaii International Conference on IEEE, USA.
16. Schumacher M, Helin H, Schuldt H (2008) CASCOM: Intelligent service coordination in the semantic web. Springer Science & Business Media.
17. Kiefer C, Bernstein A (2008) The creation and evaluation of isparql strategies for matchmaking. European Semantic Web Conference, Springer, Berlin, Heidelberg.
18. Abowd GD, Dey AK, Brown PJ, Davies N, Smith M (1999) Towards a better understanding of context and context-awareness. International Symposium on Handheld and Ubiquitous Computing, Springer, Berlin, Heidelberg.
19. Doulkeridis C, Loutas N, Vazirgiannis M (2006) A system architecture for context-aware service discovery. Electronic Notes in Theoretical Computer Science 146: 101-116.
20. Kari J (2011) Automata and formal languages. University of Turku.
21. Maamar Z, Sheng QZ, Yahyaoui H, Bentahar J, Boukadi K (2009 A new approach to model web services' behaviors based on synchronization. International Conference on Advanced Information Networking and Applications Workshops, United Arab Emirates.
22. Lara R, Polleres A, Lausen H, Roman D, de Bruijn J (2005) A conceptual comparison between WSMO and OWL-S. WSMO Final Draft D.
23. Elabd E (2011) Compliance of Web services over a high level specification. Université Claude Bernard-Lyon I, France.
24. Fellbaum C (1998) *WordNet*. Blackwell Publishing Ltd.
25. Bhattacharjee A, Purkayastha BS (2014) A Novel Approach to Construct Deterministic Finite State Automata. Inter J Eng and Comp Sci 3: 3700-3703.
26. Messaoud WB, Ghedira K, Halima YB (2016) Towards Behavioral Web Service Discovery Approach: State of the Art. Procedia Computer Science 96: 1049-1058.
27. Le Hung V, Hauswirth M, Aberer K (2005) Towards p2p-based semantic web service discovery with qos support. International Conference on Business Process Management, Springer, Berlin, Heidelberg.
28. Taher L, El Khatib H, Basha R (2005) A framework and QoS matchmaking algorithm for dynamic web services selection. The Second International Conference on Innovations in Information Technology (IIT'05)), Dubai, UAE.
29. Horowitz E, Boehm BW (1975) Practical strategies for developing large software systems. Addison-Wesley Publishing Company, Massachusetts, USA.