

# WEB SERVICE DISCOVERY METHODS AND TECHNIQUES: A REVIEW

Soodeh Pakari and Esmaeel Kheirkhah and Mehrdad Jalali

Department of Computer Engineering, Mashhad Branch, Islamic Azad University,  
Mashhad, Iran

## **ABSTRACT**

*Web Services are independent software systems which offer machine-to-machine interactions over the Internet to achieve well-described operations. With the advent of Service-Oriented Architecture (SOA), Web Services have gained tremendous popularity. As the number of Web Services is increased, finding the best service according to users requirements becomes a challenge. The Semantic Web Service discovery is the process of finding the most suitable service that satisfies the user request. A number of approaches to Web Service discovery have been proposed. In this paper, we classify them and determine the advantages and disadvantages of each group, to help researchers to implement a new or to select the most appropriate existing approach for Semantic Web Service discovery. We, also, provide a taxonomy which categorizes Web Service discovery systems from different points of view. There are three different views, namely, architectural view, automation view and matchmaking view. We focus on the matchmaking view which is further divided into semantic-based, syntax-based and context-aware. We explain each sub-group of it in detail, and then subsequently compare the sub-groups in terms of their merits and drawbacks.*

## **KEYWORDS**

*Semantic Web Service, Service Discovery, Ontology, Context-aware*

## **1. INTRODUCTION**

Service-Oriented Architecture (SOA) describes interaction between the service provider and the service consumer through provision, discovery, and usage of services over the Internet. The provider introduces the core functionality that will be used by requesters. The term “service” will be used to refer to the Software Engineering community, i.e., the computational parts of concrete services. The Business community, on the other hand, is used to refer to the whole process, including actual interactions, as a service [1]. Recent research works focus on the whole process of the Web Services’ (WS) life cycle. They study how to specify, discover, select, compose, secure, ensure the correctness, and invoke Web Services. Nowadays, because most of the organizations have attempted to implement their Business-to-Business (B2B) and Business-to-Customer (B2C) transactions in the form of Web Services, the number of available Web Services has increased dramatically. Therefore, finding an appropriate Web Service which is in agreement with the user’s requirements is a challenge. This emphasizes the need for effective and efficient Web Service discovery approaches.

In this paper, we focus on the Web Service discovery and offer a review of recent approaches that offer solutions to the Semantic Web Service discovery problem. Furthermore, we introduce a taxonomy for Web Service discovery systems. According to Mohebbi [2], there are three different

views, namely architectural view, automation view, and matchmaking view. We focus on the latter, which is further divided into semantic-based, syntax-based and context-aware.

## 2. RELATED WORK

In a survey performed by Mydhili, et al. [3] they divide Web Service discovery into two categories: functional requirements of the Web Service and non-functional requirements or Quality of Service (QoS). They, then, divide functional requirements of the Web Service into five categories: Google WSIL<sup>1</sup>, StikeIron Registry<sup>2</sup>, Google WSDL search<sup>3</sup>, Crawl WSDL search<sup>4</sup>, UDDI Keyword Query. Non-functional Requirements or Quality of Service (QoS) includes Quality Metrics Typology, Quality Models, and Quality Ontologies. In another survey performed by Netra Patil et al. [4], Web Service discovery methods are divided into two categories: Web Service discovery based on decentralized approach, and Web Service discovery based on centralized approach. Many researchers have focused on centralized Web Service discovery. Although centralized methods have some advantages for the discovery of Web Services, they suffer from problems associated with having centralized systems such as a single point of failure, and bottlenecks. Mohebbi, et al. [2] suggested a taxonomy based on three different points of view, namely: architecture view, automation view, and matchmaking view. Each view, in turn, is divided into further sub-categories. Architecture view is divided into centralized and decentralized, automation view is divided into manual and automatic, and matchmaking view is divided into syntax-based and semantic-based. Semantic-based, itself, is divided into logic-based and non-logic-based approaches. Malaimalavathani [5], classified Web Service discovery methods into four categories: Taxonomy-based matchmaking, structural ontological knowledge, role-oriented matchmaking, and filtering.

## 3. A TAXONOMY OF WEB SERVICE DISCOVERY SYSTEM

This section presents a taxonomy of Web Service discovery, based on three different points of view namely: architecture view, automation view, and matchmaking view, with our focus being on matchmaking view. Figure1 illustrate this taxonomy. Additionally, we focus on semantic-based approach which is further divided into domain ontology approaches, public ontology approaches, syntax- and semantic-based approaches. Another group is context-aware approaches, as pointed out in Introduction.

---

<sup>1</sup> This Google Web Service API and developer kit is used to extract Web Service Inspection Language Document WSIL, which can be parsed for WSDL locations.

<sup>2</sup> Search a registry not associated with a public Universal Business Registry (UBR) like Microsoft. Instead, StrikeIron Registry is used.

<sup>3</sup> Use Google Web Service API and developer kit and extract the WSDL references using the Google Search Engine.

<sup>4</sup> Use Web crawling to try and locate a WSDL from a domain.

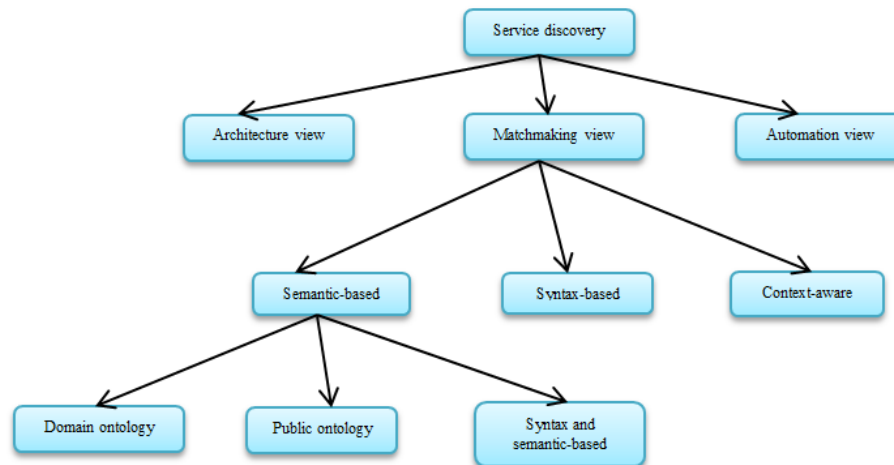


FIGURE 1. Taxonomy of Service Discovery

### 3.1. Domain Ontology-based Approach

Some authors have suggested semantic-based approaches with a domain ontology, of which we discuss a few. In [6], they consider ontologies as knowledge structures that specify attributes of services, their properties and relations among them to enable finding semantic similarity between service descriptions and service requests. They use knowledge from ontologies to enhance the both user service requests and service descriptions by adding concepts that are not presented in the original descriptions, and use them in a comparison process. It results in more precise matching, since they consider also implicit concepts. Thus, services and requests that do not contain exact matching attributes can be found semantically matching on some abstraction level. In [7], they propose a Web Service discovery with non-explicit service description semantics that match a specific service request. They suggest an approach, including semantic-based service categorization and semantic enhancement of the service request. They propose a solution for achieving functional level service categorization based on an ontology framework. Additionally, they cluster Web Services based on service functionality. This clustering is performed offline at the Universal Description Discovery and Integration (UDDI). The semantic enhancement of the service request causes to find more relevant services. The service request enhancement involves expansion of additional terms, retrieved from ontology to the request services. Then the matching will be done between the enhanced service request and the retrieved service descriptions. In [8], a framework of Semantic Web Service Discovery based on Ontology Mapping (SWSDOM) is proposed. Their framework performs mapping of input and output parameters in functional properties of Semantic Web Services expressed by different ontologies, the final similarity is obtained by aggregating linguistic similarity, structural similarity and instance similarity.

In [9], the semantics-based Web Service matching model is proposed to improve the performance of Web Service discovery. Semantics-based Web Service matching is a method, including semantic similarity matching and qualitative filtering. Function-based semantic similarity matching is used by matching algorithm in order to find the most appropriate services. It is, also, used for selecting the best service among the results of qualitative filtering.

In [10], they propose a semantic based Web Service discovery and a prototypical tool based on syntactic and structural schema matching. Their method is based on matching an input ontology, describing a service request, to Web Service descriptions at the syntactic level through Web

Service description language or at the semantic level through service ontologies described with languages such as Ontology Web Language for Services (OWL-S), Web Service Modeling Ontology (WSMO), Semantic Web Service Framework and Web Service Description Language Semantics (WSDL-S). The different input schemas, WSDL descriptions, Ontology Web Language (OWL) ontologies, OWL-S, WSMO, SWSF and WSDL-S components are represented in a directed rooted graphs. In this uniform internal representation, a number of matching algorithms operate, including structural-based algorithms (children matcher, leaves matcher, graph and sub-graph isomorphism) and syntactical ones (edit distance (levenshtein distance or LD) and synonym matcher (through the WordNet synonyms thesaurus)).

In [11], they propose a two-fold Semantic Web Service matchmaking approach, consisting of (a) a general-purpose semantic-level mediator and (b) comparison and matchmaking of Semantic Web Service capabilities. Their Semantic-level mediation approach enables the implicit representation of similarities across distinct Semantic Web Service by grounding service descriptions in so-called Mediation Spaces (MS). Given a set of Semantic Web Service and their respective grounding, a Semantic Web Service matchmaker automatically computes instance similarities across distinct Semantic Web Service ontologies and matches the request to the most suitable Semantic Web Service.

In [12], they propose a Web Service discovery framework using OWL-S advertisements. The purpose of this approach is to determine fast an initial set of candidate Web Services for a specific request. This set can then be used in more fine-grained discovery approaches, based on richer Web Service descriptions.

In [13], proposes a novel Semantic Web Service discovery method based on user preference cluster. Firstly, the method optimizes the design of unmixed semantic UDDI model and then makes use of the clustering technology to preprocess the user preference from the standpoint of user demand before service discovery. For that reason, a user could receive demanded services quickly, which have been thought highly of other users, whose preferences similar to, reducing processing semantic information. Verified by actual test environment, this method can shorten the service discovery response time on condition of not affecting the service discovery accuracy, thereby enhancing the performance of unmixed semantic UDDI. In [14], Web Service ontology is defined. They use Web Service function and progress to calculate similarity. Petri net is used to describe Web Service process. This similarity is based on an accurate concept semantic similarity of the domain ontology. A domain ontology hierarchy is defined to describe the concept semantic information. The concept semantic similarity is discussed from several aspects, such as the path between two concepts, the path weight, the density of concepts, and the antisense relationship. Finally, they use the Web Service similarity to cluster services, and prove the similarity efficiency.

[15], proposes a weighted ontology-based semantic similarity algorithm for Web Service to support a more automated and veracious service discovery and rank process in the Semantic Web Service framework. In [16], they propose a Semantic Web Services discovery method. In this approach, they use vector space model for calculating the semantic distance at first. Then match Web Services with ontology hierarchically; finally match Web Services with QoS. In [17], they present a self-adaptive Semantic Web Service matching method, which improves the precision and recall of service discovery. In this method, they transform the requirement document and service profile ontology in to the ontology trees respectively. Conception similarity, attribute similarity and structure similarity of corresponding nodes in trees are calculated through taxonomic and hierarchical methodology. Then a serial of constraints are defined according to the relationship between conception similarity and structure similarity, to get the corresponding

restructure rules. By restructuring requirement ontology tree in self-adaptive way, they achieve more accurate destination service collections.

In [18], they proposed a new QoS-aware framework to improve the Semantic Web Service discovery based on broker by using ontology concepts. Due to having real-time values of QoS attributes of Web Services, composite monitoring mechanism in which various Web Service-related applied reports continuously can be collected. These reports are compiled through Web Service monitoring agent, feedback from user, and provider advertisement. This problem is solved by combining Semantic Web and Web Service technologies.

In [19], a new architecture of SOA is proposed which incorporates a new adaptive technique called social learning that improves service provider's domain ontology from service consumer's concept contributions and thus eventually makes the service more semantically discoverable. The proposed architecture contains new similarity measure and automatic merging algorithms on weighted ontology. The proposed architecture reduces overlapping concepts and thus more relevant services are discovered.

In [20], they use semantic matchmaking for discovery and composition of Semantic Web Services to find more relevant service candidates. Among these candidates, best ones are chosen to build up the composition, or for substitution in the case of an execution failure. This matchmaker performs semantic matching of Web Services on input, output, precondition and effect.

There are some advantages and disadvantages of domain ontology based approach that we mention them briefly:

Advantages:

- The aim of Semantic Web Service technology is to reduce the manual discovery and achieve automatic identification, integration, and execution of these Web Services [21].
- Increasing number of similar Web Services, one of the important issues is to discover most relevant services to the user request. Using semantics can improve the relevancy of Web Service discovery [22].
- Pledge the automation of Web Service discovery process [23].
- Effective and reliable technique [23].

Disadvantages:

- Some proposed discovery methods are based on a user request that is expressed in a specific semantic description language like OWL-S, WSMO, or WSDL-S. As a result, end user should have intimate knowledge of Semantic Web Services and related description and implementation details which makes their usage difficult for end users [21].
- The problem is that, the requestor may not be aware of all the knowledge that constitutes the domain ontology. Specifically, the service requestor may not be aware of all the terms related to the service request. As a result, we miss many services relevant to the request during service discovery process [21].
- Another limitation of some proposed methods consist on their semantic matching approaches. In fact, both service provider and service requester use domain ontologies to build semantic service description file. Most of proposed approaches assume that both service provider and service requester use the same ontology domain to describe service but is not applicable in real-world scenario. To overcome this ontology heterogeneity, it

is needed to utilize ontology mapping techniques to coordinate the differences between these ontologies to support interoperability [21].

- More complex technique [23].
- In these approaches semantic tagging of Web Services may be needed [23].

### 3.2. Public Ontology-based Approach

In [24], they present a new way of enhancing Web Services semantically using WordNet concepts (synsets). The important advantage of the solution is that it allows developers to enhance Web Services with semantic information without semantic annotation against an ontology. This is different from traditional, ontology-based researches, which require significant cost and effort for semantic annotation and ontology management. The proposed solution allows associating semantic tags on the message parts of Web Services. They solve the semantic service discovery problem with a domain neutral service annotation technique. In [25], a two-stage filtering approach based on implicit semantics of Web Service description and WordNet is proposed to identify candidate services using semantic reasoners. In the first stage, after filtering out irrelevant services, the candidate services relevant to the given service query are identified. In the second stage, the trust score of each candidate service is compared against a user-defined trust score using Trust Rank algorithmic tool. Only those candidate services with trust score greater than the user-defined trust score are considered as the services which necessarily require complete semantic matching of service capabilities using a semantic reasoner. In order to give service semantic information, service ontology should be build. There are two kinds of service ontology: self-defined ontology and public ontology. By way of improving the usability of the method, they prefer public ontology. [26], proposes two levels Semantic Web Service discovery method. This method uses semantic word in WordNet to annotate service and service interface, and then discovers target service within two levels. At the first level, compute service similarity degree; at the second level, compute service interface similarity degree. Threshold is used to eliminate unsatisfied service in every level. In [27] they present Themis-S, a prototype of an ontology-based natural language service discovery engine. In a series of experiments, they evaluate the retrieval effectiveness of Themis-S in comparison with other information retrieval models. The experiments show that Themis-S, using WordNet as a general purpose ontology, can outperform systems applying syntactic information retrieval models such as the Vector Space Model (VSM) or the Probabilistic Relevance Model (PRM).

Now we explain some advantages and disadvantages of these approaches.

Advantages:

- The important advantage of these approaches is that it allows developers to enhance Web Services with semantic information without semantic annotation against an ontology. This is different from traditional, ontology-based researches, which require significant cost and effort for semantic annotation and ontology management [24].
- There are two kinds of service ontology: self-defined ontology and public ontology. By way of improving the usability of the method, public ontology is used. WordNet is a well-known public dictionary, terms and concepts in WordNet has its precise semantic. Therefore, we take WordNet as our service ontology, and use terms and concepts in WordNet to annotate service library, service and service interface [28].
- Mostly all semantic service annotations are based on ontologies. Despite being theoretically ideal solution for semantic annotation, ontologies have some drawbacks. First, they are domain dependent. In case of mash-ups where services from different domains are involved, domain dependent annotations introduce ontology matching

problems. Second, the cost of ontologies is relatively high and requires special expertise for creation and maintenance. WordNet is aggregation of dictionary and thesaurus that is perfect for text analysis and AI use-cases [24].

- WordNet, It is not domain specific and nearly eliminates the semantic annotation cost of services [24].

Disadvantages:

- Different parts of WordNet have different granularity for the description of word senses. In general, WordNet is too fine-granular for many purposes [29].
- There are WordNet versions for a large number of languages, but there is no real multi-lingual WordNet. The different WordNet differ in coverage, format, and availability [29].
- WordNet focuses on paratactic semantic relations between single words [29].

### 3.3. Syntax- and Semantic-based Approaches

[30], proposes an ontological Web Service discovery approach based on semantic matching process of functional and non-functional requirements. This Web Service discovery process matches functional and non-functional requirements, ranks according to preferences, considering three related ontologies: 1. An ontology for integration of quality standards and retrieve properties, 2. An ontology for modelling the relations between these standards and preferences to rank functionality and/or qualities, 3. An ontology relating quality models to Web Service functionality. The main activities of the proposed Web Service discovery process, combining syntactic and semantic issues are: First activity: matching of FR. The Web Service query using FR is expressed as an OWLS file; the matching is performed syntactically using a Web server, containing the information on the Web Service functionality; the matching results are the URLs with the descriptions of the services represented also by OWLS files, responding to the functional requirements of the client. Second activity: matching on NFR. This search is also based on a matching to retrieve QoS metrics from the properties defined in the quality model for the specific Web Service functionality, that are specified in the QStdOnt ontology; an inference engine of Protégé is used, and the queries are expressed in SQWRL; for example, search for a transactional Web Service that requires efficiency, with a certain throughput, measured by a resource-utilization metric. Third activity: Web Service ranking according to expert-assigned common preferences. The services, found after the matching of NFR, are ranked according to their common preferences specified in Onto-Relation-ComPref Ontology. Generally, levels (high, medium, and low) are associated to each quality property in order to rank them. Domain experts assign these preferences to quality properties. For example, Web Service with high-level rate of internet access are frequently demanded. Fourth activity: Web Service ranking according to user-assigned functional preferences and/or priorities.

In [31], their efficient technique has two stages including 'input parameter search' and 'synonym based search'. For the service discovery in the proposed approach, they search the Semantic Web Service description syntactically. Semantic description of Web Services plays a very important part in Semantic Web Service discovery. They proposed an approach for service discovery in Semantic Web Services. This approach searches Web Services correctly. A Web Service contains two types of parameters; Input parameters and Output parameters. The other information about Web Service what are the main tasks done by that Web Service. The functionality which is performed by Web Service is the task performed by that service. They use input parameters, their synonyms, goal description and goal description synonym matches. In [32], they select the Web Service that best matches user contexts on a mobile device. They use OWL-S to conclude the

semantic similarity between the request and advertisement's parameters. In this approach, the filtering of Web Services is done in three stages:

- Functional matching
- Contextual matching
- QoS matching

[33], present a hybrid Semantic Web Service discovery framework that exploits both the signatures and specifications of a Web Service, whilst adopting logical and non-logical service matching methods. For signature level service matching, they have developed two techniques: (a) logical similarity measures applied to the services' input/output concepts; and (b) non-logical matching based on a Structure Preserving Semantic Matching algorithm. For specification level service matching, they have applied a unique short sentence matching approach on the textual-description of a Web Service. They evaluated the performance of the S5 Web Service Matchmaker using the OWLS-TC dataset, and furthermore compared its performance with the OWLS-MX discovery model. Their results indicate that S5 Web Service Matchmaker offers an improved Web Service matching performance with a significant increase in recall and subtle improvements in precision. Web Services are independent software systems designed to offer machine-to-machine interactions over the WWW to achieve well-described operations. The description of a Web Service entails (a) a syntactic component detailing the service's operations and data structures in terms of the Web Services Description Language (WSDL), and (b) a semantic component that offers a semantic description, in terms of an ontology, of the services' data and operations. Typically, service providers expose their services to the public by providing brief descriptions of the service's operations; the challenge is to discover the right service based on rather sparse service descriptions in response to a specific service request.

In [34], the authors present an interoperable discovery platform that describes an efficient matching and ranking algorithm able to process service descriptions and discovery requests from both semantic and syntactic SDP<sup>5</sup>s. The proposed discovery platform leverages the advanced communication capabilities provided by the PLASTIC middleware to discover services in multi-network environments. An evaluation of the prototype implementation shows that multi-protocols service matching can be achieved in ambient computing environments. In [35], The OWLS-X matchmaker selects OWL-S1.1 services that are relevant to a given service request by means of logic-based matching complemented with syntactic similarity measurement. In [36], they implemented a matchmaker OWLS-MX2 with improved precision in average. The WSMO-MX service matchmaker applies different matching filters to retrieve Semantic Web Services written in a dialect of the prominent service description language WSML-Rule. For this purpose, WSMO-MX recursively computes logic-based and syntactic similarity-based matching degrees and returns a ranked set of services that are semantically relevant to a given request

Both semantic and syntax based approaches have some advantages and some disadvantages, with semantic and syntax based approaches we want to use advantages of both and decrease their disadvantages, we mention some disadvantages of them here.

Disadvantages of syntax based approaches:

- It can't retrieve the Web Services with similar functionality [23].
- Not suited for automatic processing [23].
- Still required human interaction [23].

---

<sup>5</sup> Service discovery protocols



- The main problem is that, there are different words can be used in different terms inside different domains, so we often receive some irrelevant information [31].
- However, UDDI service descriptions have some advantages and can provide some meta-data on the service and are often even natural language based, but it is difficult for automated algorithms to understand the exact semantics and capabilities of a service and selecting the correct service [22].
- To describe Web Service by Web Service Description Language (WSDL) based on XML, guarantees syntactic interoperability, but lacking consideration of semantic information. Moreover, service registration and discovery mechanism based on Universal Description, Discovery and Integration (UDDI) that only support operation of syntactic level has two weaknesses. From one hand, to describe service accurately cannot be provided in service description process. From the other hand, required service is only obtained through keyword matching in service discovery process. Therefore, it is difficult to satisfy user functional and non-functional requirement. With rapid increasing of Web Services, traditional Web Service is powerless to discover user or agent required service accurately and efficiently from mass of Web Service [17].
- Syntax-based approaches do not allow re-planning a Web Services workflow on the way in case a service fails [22].
- They can't find the most similar service among a large set of available and semantically similar services [22].
- They cannot determine the exact capabilities of a service and decide whether the service is good in combination with other services to solve a problem [22].
- Many approaches to service discovering are based on matching algorithms that try to identify whether service description contains keywords from user service request query. More generally, we need to identify service descriptions that semantically match (in some degree) content of user request. However, most current service discovering algorithms are based on syntactic matching of keywords. Using service discovering mechanism based on keywords matching leads to low precision and low recall of the discovery results. This obviously results in imprecise discovery mechanism since occurrences of keywords do not mean that service is what user needs and non-occurrences of keywords do not mean that service does not exist or inappropriate [6].
- The idea behind the keyword search is that the keyword involved in the search query which matches them with Web Service description. Since the keyword based searching unable to match the underlying semantics of Web Service, they may miss the relevant results and returns irrelevant service to the users [37].
- Another limitation of keyword search is that user can't describe the search request more precisely than keyword. Furthermore, keywords do not suffice for accurately specifying user's information needs [36].

Advantages of syntax based approaches:

- Simple and widely used technique [23].
- Standards like UDDI exist [23].
- They offer a simple syntax in terms of a list of key word phrases that users can use their own words to express their information requirement. Also, keyword-based search is more familiar to the user [21].

### 3.4. Context-aware Approaches

[37], proposes Conceptual Situation Spaces (CSS) that are aligned to established Semantic Web Service standards. CSS enables the description of situation characteristics as members in geometrical vector spaces, following the idea of Conceptual Spaces. Semantic similarity between situations is calculated regarding their Euclidean distance within a CSS. To prove its feasibility, they apply their approach to the E-Learning and E-Business domains and provide a proof-of-concept prototype.

[38], introduces an approach for selecting the most suitable service within a SOA-based collaboration system, where suitability depends on the user's context. The approach includes context modeling, generation of context-aware selection criteria, and a suitable service selection methodology. [39], proposes a context modeling approach which can dynamically handle various context types and values. More specifically, they use ontologies to improve the meaning of a user's context values and automatically identify the relations among different context values. Based on the relations among context values, they suggest the services which the user might need. [40], presents a new compound context, public-ness, and uses it to filter out inappropriate services. Public-ness is decided by multiple contexts; those that related to place, user, and information. With public-ness context, proposed service discovery scheme filter out inappropriate services which cannot be achieved by previous approaches, with negligible overhead. In addition, as user context changes, this scheme cannot filter out precisely in some situations. Therefore, this service discovery scheme performs post processing for accuracy.

[41] presents a fuzzy rough set theory based context-aware dynamic service matchmaking approach that composes an application through combining semantic information and context information. The proposed approach consists of formalized service description model with semantics and context attributes, and fuzzy rough set based service matchmaking. By describing the context attributes, the proposed approach is capable of composing context-aware application. Through a transformation technique, the incomplete information system is converted into a simpler system and then reducts are obtained from the transformed system based on the fuzzy rough set theory. Afterwards, the candidate service sets are selected by the function of the degree of keyword match, and ranked through the function of the degree of service match. This paper describes the design and mechanism of the proposed approach that is expected to increase users' satisfaction in Pervasive environments.

[42] presents the perspective of a context-aware service platform which is based on the idea of utilizing network information as services that is delivered via application programming interfaces. Effectively, it proposes a fuzzy MADM method and a context similarity measure. They take into account the quality of contextual information in aggregating contextual information from different sources. [43], proposes a middleware-level approach to support user-centric semantic service discovery. This middleware, called AIDAS, exploits context-awareness based on user/device/service profile metadata to provide personalized views on services of interest, and supports semantic-based matchmaking between requested and offered service capabilities. Semantic support services, such as ontology repositories and inference engines, typically require a large amount of computational and memory resources that might not fit the properties of all mobile devices. AIDAS addresses this issue by transparently and dynamically adapting semantic-based discovery support to the properties of different access devices.

Context-aware approaches also have some advantages and disadvantages, explained next.

#### Advantages of context-aware approaches:

- Semantic Web Services enable the automatic discovery of Web Services based on comprehensive semantic representations. However, although Semantic Web Service technology supports the automatic allocation of Web Services for a given well-defined task, it does not entail their discovery according to a given situational context. Whereas tasks are highly dependent on the situational context in which they occur, Semantic Web Service technology does not explicitly encourage the representation of domain situations. Moreover, describing the complex notion of a specific situation in all its facets is a costly task and may never reach sufficient semantic expressiveness. Particularly, following the symbolic Semantic Web Service approach leads to ambiguity issues and does not entail semantic meaningfulness. Apart from that, not any real-world situation completely equals another, but has to be matched to a finite set of semantically defined parameter descriptions to enable context-adaptability. [37]
- Given the large amount of existing services and the diversified needs nowadays, it is time-consuming for end-users to find appropriate services. To help end-users obtain their desired services, context-aware systems provide a promising way to automatically search and recommend services using a user's context. [39]
- Context-awareness is a key ingredient in any Ubiquitous and Pervasive system and provides intelligence to the system, allowing computing devices to make appropriate and timely decisions on behalf of users. Context-awareness in Mobile Computing refers to internal and external adaptation of the environment and applications to the context state of each other. Such systems should adapt to the changes and variations of user's context, such as location, device status, connectivity, etc. [42]

#### Disadvantages of context-aware approaches:

- The concept of context is too broad, as context could be anything related to the user and applications. It is, unfortunately, too complicated to be modeled easily and then employed in the applications. [44]
- Due to the complexity of the context, it is infeasible to properly model context for all applications in one way. [44]

TABLE 1. merits and shortcomings of service discovery methods

	<i>merits</i>	<i>shortcomings</i>
<b>Domain ontology-based approaches</b>	1-Minimize the manual discovery and usage of Web Service by allowing software agents to automatically and dynamically discover Web Services 2-Effective and reliable technique 3-Relevancy of Web Service discovery can be improved by augmenting semantics through expressive formats like OWL	1-they require the end user to have intimate knowledge of Semantic Web Services and related description and implementation details which makes their usage difficult for end users 2- The discovery scope of these approaches is often limited to some Web Services that are published in a specific description standard. 3- The service requestor may not be aware of all the terms related to the service request. 4- We need to use ontology mapping techniques to coordinate the differences between these ontologies to support interoperability
<b>Public ontology-based approaches</b>	1-It allows developers to enhance Web Services with semantic information without semantic annotation against an ontology. 2- WordNet, It is not domain specific and eliminates the semantic annotation cost of services. 3- Terms and concepts in WordNet has its precise semantic	1-Different parts of WordNet have different granularity for the description of word senses. In general, WordNet is too fine-granular for many purposes. 2- There is no real multi-lingual WordNet
<b>Syntax and semantic-based approaches</b>	1-Simple and widely used technique. 2- Standards like UDDI exist. 3- keyword-based search is more familiar to the user.	1- Still required human interaction 2- can't choose most similar service among a large set of available and semantically similar services
<b>Context-aware approaches</b>	1-Semantic Web Services enable the automatic discovery of distributed Web Services based on comprehensive semantic representations 2- Context-awareness is a key in-gredient in any Ubiquitous and Pervasive system and provides intelligence to the system	1- The concept of context is too broad 2- Due to the complexity of the context, it is infeasible to properly model context for all applications in one way

#### 4. CONCLUSIONS

In this paper, we provided an overview of the recent progress made in Web Service discovery approaches. Furthermore, we performed an analysis over these approaches and highlighted some of their merits as well as shortcomings. After introducing a taxonomy of Web Service discovery systems, we presented the advantages and disadvantages of each group which mention them in TABLE1 briefly. We, finally, argued that this taxonomy and classification is novel and useful for researchers in the Semantic Web Service discovery.

## REFERENCES

- [1] H. Saboohi, S. Abdul Kareem, "world-alerting semantic web service discovery and composition techniques-a survey," in 7th International Conference on Semantic Web and Web Services, Las Vegas, Nevada, USA, July 2011.
- [2] K. Mohebbi, S. Ibrahim, M. khezrian, K. Munusamy, Seyed G. H. Tabatabaei, "a comparative evaluation of semantic web service discovery approaches," in 12th international conference on information integration and web based applications and services, ACM New York, USA, 2010.
- [3] M. K. Nair, V. Gopalakrishna, "look before you leap: a survey of web service discovery," international journal of computer applications, vol. 7, no. 5, pp. 22-30, September 2010.
- [4] N. Patil, A. Gopal, "comparative study of mechanisms for web service discovery based on centralized approach focusing on UDDI," international journal of computer applications, vol. 14, no. 1, pp. 28-31, january 2011.
- [5] M. Malaimalavathani, "a survey on semantic web service discovery," in international conference on information communication and emmedded systems, feb 2013.
- [6] V. Oleshchuk, "ontology-based service matching and discovery," in 6th international conference on intelligent data acquisition and advanced computing systems, sept 2011.
- [7] A. V. Paliwal, "semantic-based automated service discovery," IEEE transaction on service computing, vol. 5, no. 2, pp. 260-275, April-June 2012.
- [8] Y. Shi, G. Li, J. Li, "framework of semantic web service discovery based on ontology mapping," in international conference on research challenges in computer science, Dec 2009.
- [9] K. Zamanifar, A. Zohali, N. Nematbakhsh, "matching model for semantic web service discovery," in international conference on advanced information networking and applications workshops, May 2009.
- [10] B. D. Martino, "semantic web service discovery based on structural ontology matching," international journal of web and grid services, vol. 5, no. 1, pp. 46-65, 2009.
- [11] Stefan Dietze, Neil Benn, John Domingue, Alex Conconi, Fabio Cattaneo, "two-fold service matchmaking-applying ontology mapping for semantic web service discovery," the semantic web lecture notes in computer science, vol. 5926, pp. 246-260, 2009.
- [12] G. Meditskos, N. Bassiliades, "structural and role-oriented web service discovery with taxonomies in OWL-S," IEEE transactions on knowledge and data engineering, vol. 22, no. 2, pp. 278-290, 2009.
- [13] M. Wang, X. Li, X. Qiao, "semantic web service discovery based on user preference cluster," in 3rd IEEE international conference on broadband network and multimedia technology, 2010.
- [14] L.I. Xie, F.Z. Chen, J.S. Kou, "ontology-based semantic web service clustering," in 18th international conference on industrial engineering and engineering management, Sept 2011.
- [15] M. Liu, W. Shen, Q. Hao, J. Yan, "an weighted ontology-based semantic similarity for web service," Export systems with applications, vol. 36, no. 10, pp. 12480-12490, May 2009.
- [16] L. Zhou, "an approach of semantic web service discovery," in international conference on communications and mobile computing, April 2010.
- [17] Ch. KE, Zh. Huang, "self-adaptive semantic web service matching method," in Knowledge-based systems, Nov 2012.
- [18] A. Yousefipour, A. G. Neiat, M. Mohsenzadeh, M. a. Seyyedi, "a new broker-based semantic web service discovery framework for selecting and ranking suggested web services," in IEEE international conference on intelligent computer communication and processing, Aug 2010.
- [19] Sajib Kumar Mistry, Mosaddek Hossain Kamal, Dilip Mistry, "semantic discovery of web services through social learning," in The 2012 Iberoamerican conference on electronics engineering and computer science, 2012.
- [20] A. B. Bener, V. Ozadali, E. S. Ilhan, "semantic matchmaker with precondition and effect matching using SWRL," Expert systems with applications, vol. 36, no. 5, pp. 9371-9377, July 2009.
- [21] Asma Adala, Nabil Tabbane, Sami Tabbane, "A Framework for Automatic Web Service Discovery Based on Semantics and NLP Techniques," advanced in multimedia-special issue on web services in multimedia communication, vol. 2011, no. 1, January 2011.
- [22] R. Suganyakala, M. Aarthilakshmi, G. R. Karpagam, S. Maheswari, "ontology based Qos driven web service discovery," international journal of computer science issues, vol. 8, no. 4, pp. 191-198, July 2011.

- [23] A. Pradnya Khutade, B. Rashmi Phalnikar, "Qos-based web service discovery using OO concepts," international journal of advanced technology & engineering research, vol. 2, no. 6, pp. 81-86, Nov. 2012.
- [24] R. Karimpour, F. Taghiyareh, "conceptual discovery of web services using WordNet," in Service Computing conference, Singapore, Dec. 2009.
- [25] G. Ganapathy, C. Surianarayanan, "an approach to identify candidate services for semantic web service discovery," in IEEE international conference on service-oriented computing and applications, Perth, WA, Dec. 2010.
- [26] Y. Peng, "two levels semantic web service discovery," in seventh international conference on fuzzy systems and knowledge discovery, Yantai, Shandong, Aug. 2010.
- [27] Joerq Becker, Oliver Mueller, Manuel Woditsch, "an ontology-based natural language service discovery engine-design and experimental evaluation," in the proceeding of the European conference on information systems, Pretoria, South Africa, 2010.
- [28] Yanbin Peng, Chunming Wu, "automatic semantic web service discovery based on assignment algorithm," in 2nd international conference on computer engineering and technology, Chengdu, April 2010.
- [29] M. Pinkal, A. Koller, "semantic theory," 2005.
- [30] N. Levy, F. Losavio, A. Matteo, A. Ramdane-Cherif, H. Hadj Salem, "quality standards for ontology web service discovery," in international conference with peer:XXXVII CLEI, Quito, Ecuador, October 2011.
- [31] Amjad Farooq, Rabia Arshad, "an efficient technique for web services identification," international journal of multidisciplinary sciences and engineering, vol. 2, no. 1, pp. 26-30, March 2011.
- [32] C. B. Merla, "context-aware match-making in semantic web service discovery," international journal of advanced engineering sciences and technologies, vol. 9, no. 2, pp. 243-247, 2011.
- [33] M. S. Farukh, "Exploiting Semantics and Syntax for Service Specification and Signature Matching: The S5 Web Service Matchmaker," Faculty of graduate thesis, 2011.
- [34] Robert Speicys Cardoso, Sonia Ben Mokhtar, Pierre-Guillaume Raverdy, "interoperable semantic and syntactic service discovery for ambient computing environments," international journal of ambient computing and intelligence, vol. 2, no. 4, pp. 13-32.
- [35] M. Klusch, P. Kapahnke, B. Fries, "hybrid semantic web service retrieval: a case study with OWLS-MX," in IEEE international conference on semantic computing, Santa Clara, CA, Aug. 2008.
- [36] Jeberson Retna Raj, Dr. T. Sasipraba, "web service discovery based on computation of semantic similarity distance and Qos normalization," Indian journal of computer science and engineering, vol. 3, no. 2, pp. 235-239, May 2012.
- [37] Stefan Dietze, Michaël Mrissa, John Domingue, Alessio Gugliotta, "context-aware semantic web service discovery through metric-based situation representations," 2010.
- [38] Hong Qing Yu, Stephan Reiff-Marganiec, "automated context-aware service selection for collaborative systems," in proceedings of the 21st international conference on advanced information systems engineering, Berlin, Heidelberg, 2009.
- [39] Hua Xiao, Ying Zou, L. Ng, L. Nigul, "an approach for context-aware service discovery and recommendation," in IEEE international conference on web services, Miami, FL, July 2010.
- [40] Yong-Jin Shin, Young-Hee Lee, rule-based context-aware service discovery based on service publicness, a thesis submitted to the faculty of th KAIST, 2010.
- [41] Shan Liu, Yichao Yang, Wenfeng Zheng, Xiaolu Li, "context-aware dynamic service matchmaking," in 12th international conference on computer and information technology, Chengdu, 2012.
- [42] Peyman TalebiFard, Victor C.M. Leung, "a data fusion approach to context-aware service delivery in heterogeneous," in 8th international conference on mobile web information systems, 2011.
- [43] Alessandra Toninelli, Antonio Corradi, Rebecca Montanari, "semantic-based discovery to support mobile context-aware service access," computer communications, vol. 31, no. 5, pp. 935-949, 2008.
- [44] Wenge Rong, Kecheng Liu, "a survey of context aware web service discovery: from user's perspective," in SOSE '10 proceedings of the 2010 fifth IEEE international symposium on service oriented system engineering, Washington, DC, USA, 2010.