A Guideline for Evaluating Semantic Web Service Discovery Approaches

Keyvan Mohebbi, Suhaimi Ibrahim, Norbik Bashah Idris, Sayed Gholam Hassan Tabatabaei

Abstract

Currently, most enterprises deploy their services on the Web. This augments the request for tools to discover, select, compose and invoke Web services, effectively and efficiently. Discovery is considered as the foremost and most important step in attracting a vast number of researches. In this paper, a taxonomy of Web service discovery systems is provided. Different points of view are taken into consideration to provide such a taxonomy which covers various existing approaches. With the focus on Semantic Web service discovery, the most prominent approaches of this category are also classified and described. In addition, a comprehensive set of criteria for evaluating Web service discovery approaches are presented. The Semantic Web service discovery approaches are compared with respect to some criteria. The findings of this study can help researchers in both academia and industry in deciding on the criteria that should be used for evaluating the existing Web service discovery approaches or implementing new ones as well as knowing the comparison result of these approaches, in order to select the most appropriate one.

Keywords: Web Services, Semantic Web Services, Web Service Discovery, Matchmaker

1. Introduction

After the emergence of Service Oriented Architecture (SOA), the popularity of Web services has increased from day to day. According to the World Wide Web Consortium (W3C), “A Web service is a software system identified by a URI [1], whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols” [2].

Recent researches concentrate on the whole process of the life cycle of Web services. They study how to specify, discover, select, compose, and invoke Web services. Nowadays, because most of the organizations are attempting 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 [3]. Due to this phenomenon, finding an appropriate Web service which is in agreement with the user’s desire is a challenge that emphasizes the need for effective and efficient Web service discovery approaches. A considerable amount of research has targeted such an improvement.

W3C has defined Web service discovery as “the act of locating a machine-processable description of a Web service-related resource that may have been previously unknown and that meets certain functional criteria. It involves matching a set of criteria with a set of Web service descriptions. The goal is to find an appropriate Web service” [4].

During discovery process, the description of formalized goals of service requesters and semantic annotations of formalized Web services need to be compared in order to recognize common elements in these descriptions. A discovery service is responsible for the process of performing discovery. Service descriptions may be found by a requester during the development of a system as static, or during execution of a system as dynamic. Semantic Web service discovery attempts to make the process of discovery run automatically in all of the above cases.

In this paper, we focus on the Web service discovery and offer a survey of recent approaches that provides solutions to Semantic Web service discovery problem. Firstly, we introduce a taxonomy to categorize Web service discovery systems. The taxonomy is based on three different points of view, namely Architectural View, Automation View, and Matchmaking View. Then, focusing on the...
Semantic-based subtype of Matchmaking View, the most prominent approaches to Web service discovery are classified into Logic-based, Non-Logic-based, Logic-&Non-Logic-based, and Logic-&Syntax-based categories. Next, we introduce and describe a set of evaluating criteria. Based on some parameters, these criteria are further divided into two subsets, namely Common, and Uncommon Criteria. Finally, the proposed matchmaking categories are evaluated and compared by means of their prominent approaches, with respect to Common Criteria. We also evaluate Uncommon Criteria in order to study their application feasibility.

Researchers in both academia and industry may apply the results of this work. Especially the proposed criteria for evaluating and comparing the Web service discovery approaches can serve as a basis for either implementing a new or selecting and reusing an existing Web service discovery approach by determining its strengths and weaknesses.

The remainder of this paper is structured as follows. We review existing surveys on Web service discovery systems in Section 2. A taxonomy of Web service discovery systems where the stress is on the matchmaking viewpoint is introduced in Section 3. We present the classification of Semantic Web service discovery approaches in Section 4. After the introducing of some criteria, we evaluate and compare the categories of our proposed classification with respect to these so-called Common Criteria in Section 5. Also, some so-called Uncommon Criteria are introduced and evaluated in this section. Finally, Section 6 concludes our paper.

2. Related Works

According to our investigation, a few numbers of research works concerning the classification or comparison of the Semantic Web service discovery approaches have been published.

In a survey performed by Garofalakis et al. [5], several approaches mainly using two different points of view were discussed. Their work only focused on the architectural aspects and the data models that facilitate discovery and adequate attention was not paid to the usage of semantics. Recently, Crasso et al. [6] have published a survey of approaches to Web service discovery. They propose several criteria based on the requirements for discovering services within common service-oriented environments. Again, their criteria do not specifically characterize semantic approaches of Web service discovery. On the contrary, Le et al. [7] presented a survey of Web service discovery systems which focuses on methods of using semantics. They introduce a taxonomy of discovery systems and highlight the advantages and disadvantages of various systems. However, their taxonomy does not include the semantic systems which use matchmaking techniques other than logic based reasoning. A classification of Web service discovery and matchmaking is proposed in [8]. In addition, authors evaluate and compare the categories with respect to some qualitative criteria, but their approach to perform such an analysis is unknown. In [9] various criteria are suggested in to classify the discovery approaches. The considered criteria are: matching elements, support for degree of match, indirect matching, and the type of matchmaking algorithm i.e., using logical reasoning or other non-logic and hybrid techniques. It has also selected some typical algorithms that belong to each classification and described their characteristics regarding to the criteria given above. Undoubtedly, one of the most comprehensive surveys on Web service discovery systems is [10]. It classifies both the discovery architectures and existing approaches to semantic service matching. The semantic service matching category is further classified based on the parts of service semantics that are considered for matching and the way that matching is performed in terms of logic- or non-logic-based techniques or a combination of both. However, neither [9] nor [10] considered other important criteria such as multi stage matching, support for UDDI, and support for different ontologies for classifying the approaches.

This work is mostly inspired by [7], [9], and [10]. It aims to integrate the most important criteria proposed by those works with the focus on semantic service matching to categorize precisely the existing approaches. In addition, it attempts to introduce some qualitative criteria and compare the Semantic Web service discovery approaches with respect to these criteria. The proposed set of criteria, as well as the conducted comparative evaluation, is not addressed by any of the aforementioned works.
3. Taxonomy of Web Service Discovery Systems

This section introduces a taxonomy of Web service discovery systems. The taxonomy is based on three different points of view namely: Architecture View, Automation View and Matchmaking View as described in the following. Figure 1 illustrates this taxonomy.

![Figure 1. Taxonomy of Web service Discovery Systems](image)

3.1. Architecture View

This view concerns the way that discovery systems handle the storage of service information and its location in an environment. Using this view, the systems can be categorized as Centralized and Decentralized [5], [10]. Centralized discovery systems rely on one single, global directory service, e.g. a repository or registry, supported by a certain super-peer or middle agent like a matchmaker. Decentralized discovery systems rely on distribution technologies such as Peer-to-Peer (P2P) to publish service descriptions over several nodes in a network.

3.2. Automation View

This view concerns the way that discovery systems are invoked. Using this view, the systems can be divided to Manual and Automatic [4], [5]. Manual service discovery systems are invoked by a human user and typically at design time to search for his goal. Automatic service discovery systems are invoked by a system agent either at design time or at run time to search for a service.

3.3. Matchmaking View

This view concerns the matching algorithm used to determine whether the capabilities of a requested service and an advertised service conform to each other. Since the matching algorithm used in matchmaker module is at the core of each Web service discovery system, we focus more on this category. Using this view, the systems can be broadly categorized as Syntax-based and Semantic-based.

3.3.1. Syntax-based

The discovery mechanism supported by the legacy UDDI standard [11] is the best example of Syntax-based service matching. In short, the Web service textual description is stored in the UDDI. In the retrieval stage, a query is entered, either manually or automatically. The keywords included in query are matched against the services descriptions stored in the registry. Typically, services can be located by category, name, location, business, bindings or UDDI technical models (tModel). Finally, the matched Web services are returned as a candidate answer set and the user browses through them in order to select the most suitable one [5], [9].

As an extension to this approach, a dictionary of synonyms, such as WordNet [12] can be used to construct a query with the same meaning. Because of the different keywords used (synonyms of previous ones), more services are found which possibly comply with the user’s request. Liang et al. [13] use this approach as part of their work to construct the list of service categories that are relevant to the service request.
Li et al. [14] combine keyword (Syntax-based) matching with P2P storage and present a system that uses distributed hashing to map the XML service descriptions over a P2P network.

3.3.1.1. Advantages of Syntax-based Approaches

- They enable a huge number of available services to be filtered rather quickly.
- They work on no particular annotations for services rather than on simple WSDL descriptions.

3.3.1.2. Disadvantages of Syntax-based Approaches

- They are limited by the ambiguities of natural languages. This imposes some restrictions for example it prevents retrieval of Web services with different textual descriptions and similar functionality.
- They preclude fully automatic discovery, composition, invocation, and monitoring of services, due to the lack of semantic understanding.

For further reference, a more complete list of shortcomings of the Syntax-based approaches can be found in [9].

3.3.2. Semantic-based

The semantic service annotations were invented in order to realize the vision of (semi) automate the whole lifecycle of a service, i.e. from advertisement to invocation and execution. Their important role especially is in service discovery that affects every aspect of this process including its architectures, algorithms, tools and effectiveness of service retrieval. Semantic-based Web service discovery approaches perform similarity matchmaking based on the semantic service annotations and the ontology matching. Although semantic matchmaking is able to overcome the inadequacies of Syntax-based discovery but generally, its matching algorithms are more complex than those of the Syntax-based approaches [9].

4. Classification of Semantic Web Service Discovery Approaches

The scope of this research is restricted to Semantic-based Web service discovery approaches. The existing approaches are classified mainly based on their matchmaking algorithm, as explained in the following paragraphs. Figure 2 shows this classification.

![Figure 2. Classification of Semantic Web Service Discovery Approaches](image)

4.1. Logic-based Approaches

This category of matchmakers exploits standard logic inferences. Based on logical reasoning on the semantic annotations of services, they aim to determine semantic relations between services. Logic-based approaches are more accurate than the Syntax-based approaches, because of their solid mathematical base.

Most of the Semantic Web service matchmakers perform logic-based matching. One of the earliest and most important works in this category is that introduced by Paolucci et al. [15]. For matchmaking, their approach takes into account only the inputs and outputs of the service profiles. They also describe
the notion of degree of match (DoM) between two outputs or two inputs which depends on the relationship between the domain ontology concepts which relate to the outputs and inputs. A Logic-based semantic matchmaker for OWL-S services is presented by Jaeger et al. [16] focusing on service IO matching. The importance of their approach is that it divides the matching process into several stages. This especially increases the accuracy of matchmaking. Another Logic-based approach introduced by Srinivasan et al. [17] performs matchmaking between inputs, outputs, pre-conditions, and effects (IOPE) of the services.

4.2. Non-Logic-based Approaches

Using formal logic for modeling query and service semantics can be prohibitively difficult and the complexity of system behavior significantly affects this difficulty. In addition, using logical reasoning for the Semantic Web service discovery can be very time consuming and the computational complexity makes it worse [18].

The Non-Logic-based Semantic Web service discovery aims to overcome the above disadvantages of Logic-based approaches. This category of matchmakers exploits semantics that are implicit in patterns or relative frequencies of terms in service descriptions. They do not perform logical reasoning to determine the degree of similarity between two service descriptions, rather, generally rely on techniques such as graph matching, linguistics, data mining, or Information Retrieval (IR). Among those, the IR techniques are at the centre of interest. They aim to use IR data models (e.g. document vectors) for representing service descriptions. IR techniques can then be applied (e.g. similarity measures, clustering, latent semantic indexing) to retrieve the information, i.e. service descriptions [9], [10].

One of the most recent Non-Logic-based discovery approaches introduced by Plebani and Pernici [19]. They present an approach for Web service retrieval called URBE (Uddi Registry By Example). Their algorithm is able to evaluate the degree of similarity between a pair of Web services by comparing the related WSDL descriptions. It takes into account the relationships between the main elements composing a WSDL description, and if available, the annotations included in a SAWSDL file to improve the performance of semantic matching. As another work, Li et al. [20] design an ontology-based Semantic Web services discovery framework based on WSMO. They define semantic similarity between two terms based on the graph theory. Finally, iRDQL [21] is another Non-Logic-based service matchmaker proposed for OWL-S annotations.

4.3. Logic-&Non-Logic-based Approaches

Logic-based approaches are inadequate as in order to assess semantic similarity of services, they use only the explicit semantics that are described by the domain ontologies. In this way, some relative services may be discarded from the answer set. A way to overcome such limitation is to use the Non-Logic-based approaches which exploit implicit semantics of the services, besides the Logic-based approaches. This combination is termed as Logic-&Non-Logic-based. The main idea in the Logic-&Non-Logic-based approaches is that in case logic-based matching fails, non-logic-based matching techniques could be applied [9].

The first Logic-&Non-Logic-based matchmaker for OWL-S services is OWLS-MX [22]. It adds approximated IR-based matching to logic-based reasoning. This way, in some cases, the failure results of the logic-based matching is tolerated which results in more services to be returned. iMatcher2 [23] determines the degree of semantic matching between two OWL-S service profiles with the use of logic-based reasoning and multiple text similarity metrics. As a Logic-&Non-Logic-based matchmaker for WSMO services, WSMO-MX [24] similarly applies different filters including logic-based and text similarity matching to retrieve and rank the relative services.
4.4. Logic-&Syntax-based Approaches

This category makes use of both pure Logic-based matchmaking and Syntax-based (e.g., traditional keyword-based) discovery approach. The discovery approach in FUSION [25] combines the means of logic-based and syntax-based matching. Its Discovery Manager module first aims to retrieve service advertisements or service provider records through keyword-based search, then based on the requirements of the service consumer, it continues to search for proper services via logic-based reasoning. Lumina [26] essentially provides a logic-based matchmaker for WSDL-S services, however to give more flexibility to search for the non-semantic services, it also provides a general UDDI discovery through syntax-based (keyword) searching.

5. Evaluation of Semantic Web Service Discovery Approaches

In this section, a comprehensive set of criteria are proposed. Based on some parameters, we divide such criteria into two subsets, namely: **Common** and **Uncommon Criteria**. Both of such criteria could help evaluating the existing Semantic Web service discovery approaches or serve as the abilities to be considered in implementing new approaches. In the following subsections, each of these criteria is described and discussed in detail. Figure 3 outlines the evaluation criteria.

![Figure 3. Proposed Criteria for Evaluating Semantic Web Service Discovery Approaches](image)

5.1. Comparative Evaluation based on Common Criteria

In the following, some common criteria are proposed. They are termed as common because they are more or less supported by the current approaches and they are also not dedicated to specific matchmaking models or categories. After proposing the common criteria, our quadruplet Semantic Web service discovery categories from the previous section as well as some prominent approaches within each category are compared with respect to these criteria. In order to simplifying the evaluation process, we further classify the common criteria into three categories as follows.
5.1.1. Accuracy-related Criteria

We define *Accuracy* as the ability of an approach to accurately capture all the requirements that are sought by requesters and offered by services. It is estimated based on the evaluation of the following criteria. The result can be seen in Table 1.

### Table 1. Evaluation of Semantic Web Service Discovery Approaches Based on Accuracy

<table>
<thead>
<tr>
<th>Prominent Approaches</th>
<th>Matchmaking Algorithm</th>
<th>Accuracy-related Criteria</th>
<th>Overall Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Matching Elements</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>IO</td>
<td>PE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Kifer+ 04) [31]</td>
<td>Logic-based</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>OWL-S IDE (Srinivasan+ 06) [17]</td>
<td>“ ”</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(Somasundaram+ 06) [32]</td>
<td>“ ”</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(Li+ 07) [20]</td>
<td>Non-Logic-based</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>URBE (Plebani+ 09) [19]</td>
<td>“ ”</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>WSMO-MX (Kaufer+ 06) [24]</td>
<td>Logic-&amp;Non-Logic-based</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>OWLS-MX (Klusch+ 09) [22]</td>
<td>“ ”</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>iMatcher (Kiefer+ 08) [23]</td>
<td>“ ”</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SAWSDL-MX2 (Klusch+ 09) [27]</td>
<td>“ ”</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Lumina (Li+ 05) [33]</td>
<td>Logic-&amp;Syntax-based</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>FUSION (Kourtesis+ 08) [25]</td>
<td>“ ”</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

#### 5.1.1.1. Matching Elements

They represent the parts of service advertisement and service request which are considered for matching. Possible options are:
- **IO**: Service Inputs and Outputs.
- **PE**: Service Pre-conditions and Effects or Post-conditions.
- **Other**: Either non-functional aspects such as the name, business category, cost, reliability, and quality of service, or other parts of service such as WSDL descriptions, interface parts (operation names, operation numbers, parameter names, parameter numbers, parameter types) and textual descriptions of service functionality.

#### 5.1.1.2. Support for DoM

Sometimes, services and requests are matching partially. Using rigid logic-based matchmaking ignores these services. However, these partially matched services might perfectly match the requestor’s intention. The concept of DoM (Degree of Match) is introduced to handle this problem and represents the similarity level between two services, IOPE attributes or specific service operations [9]. *Support for DoM*, indicates that the approach enables ranking of the matching results according to their relevance to the request of the user.

#### 5.1.1.3. Multi Stage Matching

Indicates that the discovery approach performs the matching process in several stages, sequentially or concurrently between different elements and finally aggregate the result of each stage to reach a more fine-grained and effective result of the matched services. Multi stage matching leads to more
meaningful results in discovery but with the cost of matching complexity which in turn increases the query response time. Therefore, it is necessary to optimize the trade-off between quality and duration of matching. Some approaches allow users to manage the trade-off between quality and response time of matching by selecting the stages of a matching process [7].

5.1.1.4. Intelligency

Recently, machine learning techniques from Artificial Intelligence (AI) are being applied to the problem of Web service matching and classification. As an example, iMatcher2 [23] can be trained over a given test collection by means of a regression model. After training, iMatcher2 learns which of its text similarity measures performs best for a given query. Experimental evaluation of this matchmaker shows that the combination of logical inference and regression-based learning on text similarities outperforms logic-based only matching [22]. Similarly, SAWSDL-MX2 [27] and iSeM [28] have implemented a machine learning approach that trains a classifier based on the basic matching results. For aggregating the results of their various matching filters, these matchmakers employ a machine learning strategy using a Support Vector Machine (SVM) with a non-linear kernel. Apart from application of machine learning techniques in the process of matchmaking, they also have been utilized to solve the problem of automatically generating predefined taxonomies for Web services (e.g. [29] and [30]).

In summary, several approaches based on machine learning techniques have been used to leverage Web service discovery without involving an additional level of the semantic markup. All of them report enhancements in Precision of automated service matchmaking [30], which in turn cause the Performance of the Web service discovery approaches to improve (Performance and its related parameters are discussed in 5.1.3.2). We define Intelligency as the dependency of an approach to the AI mechanisms.

5.1.2. Scalability-related Criteria

We define Scalability by means of two criteria. It is estimated based on the evaluation of the criteria explained as follows. The result can be seen in Table 2.

Table 2. Evaluation of Semantic Web Service Discovery Approaches Based on Scalability

<table>
<thead>
<tr>
<th>Prominent Approaches</th>
<th>Matchmaking Algorithm</th>
<th>Scalability-related Criteria</th>
<th>Overall Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Kifer+ 04) [31]</td>
<td>Logic-based</td>
<td>✔</td>
<td>Average</td>
</tr>
<tr>
<td>OWL-S IDE (Srinivasan+ 06) [17]</td>
<td>&quot; &quot;</td>
<td>×</td>
<td>Low</td>
</tr>
<tr>
<td>(Somasundaram+ 06) [32]</td>
<td>&quot; &quot;</td>
<td>×</td>
<td>Average</td>
</tr>
<tr>
<td>(Li+ 07) [20]</td>
<td>Non-Logic-based</td>
<td>✔</td>
<td>High</td>
</tr>
<tr>
<td>URBE (Plebani+ 09) [19]</td>
<td>&quot; &quot;</td>
<td>×</td>
<td>Average</td>
</tr>
<tr>
<td>WSMO-MX (Kaufer+ 06) [24]</td>
<td>Logic-&amp;Non-Logic-based</td>
<td>×</td>
<td>Low</td>
</tr>
<tr>
<td>OWLS-MX (Klusch+ 09) [22]</td>
<td>&quot; &quot;</td>
<td>×</td>
<td>Low</td>
</tr>
<tr>
<td>iMatcher (Kiefer+ 08) [23]</td>
<td>&quot; &quot;</td>
<td>×</td>
<td>Low</td>
</tr>
<tr>
<td>SAWSDL-MX2 (Klusch+ 09) [27]</td>
<td>&quot; &quot;</td>
<td>×</td>
<td>Low</td>
</tr>
<tr>
<td>Lumina (Li+ 05) [33]</td>
<td>Logic-&amp;Syntax-based</td>
<td>✔</td>
<td>Average</td>
</tr>
<tr>
<td>FUSION (Kourtesis+ 08) [25]</td>
<td>&quot; &quot;</td>
<td>×</td>
<td>Average</td>
</tr>
</tbody>
</table>
5.1.2.1. Support for Different Ontologies

Indicates the approach supports matching Web services whose requesters and providers use different ontologies. Since Web services are autonomous, heterogeneous, and developed independently, using the same ontology between the service requester and the provider is a major limitation. Support for Different Ontologies enables a Web service provider to provide an exact service to the requester even though they do not use similar ontology.

5.1.2.2. Support for UDDI

Generally, all Syntax-based approaches rely on UDDI for discovery. However, using XML to describe its data model prevents UDDI from representing semantics. Support for UDDI indicates that the semantic discovery approach can support matching non-semantic as well as Semantic Web services by combining the matchmaker and UDDI [7].

5.1.3. Qualitative Criteria

We are able to compare our proposed matchmaking categories with respect to two qualitative criteria, namely Query Response Time and Performance. These criteria are described as follows. The results can be seen in Table 3.

Table 3. Evaluation of Semantic Web Service Matchmakers

<table>
<thead>
<tr>
<th>Matchmaking Algorithm</th>
<th>Prominent Approaches</th>
<th>Qualitative Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Query Response Time</td>
</tr>
<tr>
<td>Logic-based</td>
<td>(Kifer+ 04) [31]</td>
<td>Average</td>
</tr>
<tr>
<td></td>
<td>OWL-S IDE (Srinivasan+ 06) [17]</td>
<td>Average</td>
</tr>
<tr>
<td></td>
<td>(Somasundaram+ 06) [32]</td>
<td></td>
</tr>
<tr>
<td>Non-Logic-based</td>
<td>(Li+ 07) [20]</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>URBE (Plehani+ 09) [19]</td>
<td>Low</td>
</tr>
<tr>
<td>Logic-&amp;Non-Logic-based</td>
<td>WSMO-MX (Kaufer+ 06) [24]</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>OWLS-MX (Klusch+ 09) [22]</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>iMatcher (Kiefer+ 08) [23]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SAWSDL-MX2 (Klusch+ 09) [27]</td>
<td></td>
</tr>
<tr>
<td>Logic-&amp;Syntax-based</td>
<td>Lumina (Li+ 05) [33]</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>FUSION (Kourtesis+ 08) [25]</td>
<td>Average</td>
</tr>
</tbody>
</table>

5.1.3.1. Query Response Time

This criterion determines how long a query for Web service discovery takes to process [17]. Comparing the query response time of Logic-based, Non-Logic-based and Logic-&Non-Logic-based approaches is conducted based on the results of the evaluation experiments provided in [34] and [27]. No evaluation has been done or published yet on the comparison of the query response time of Logic-&Syntax-based category with other approaches.

5.1.3.2. Performance

The discovery of Web services can be seen as a special Information Retrieval (IR) problem [35]. The most important used performance measures for evaluating IR systems are Recall and Precision.
Recall is defined as the fraction of the relevant documents that are retrieved. It indicates the retrieval of all items a user is interested in. Precision is defined as the fraction of the answer set (retrieved by the matchmaker) that is relevant. It indicates the retrieval of only the items a user is interested in.

A discovery service with high performance should provide both high Recall and high Precision [18].

In order to evaluate precisely the performance of the matchmaking approaches and comparing them with each other, a common test bed containing proper number of Web services is needed. Also, it is necessary to have many queries involving in the process of evaluation, because usually a few queries are not sufficient to make a statistically significant statement.

5.1.4. Discussion

As illustrated in Table 1, the Accuracy of an approach is estimated based on the overall result of four criteria as outlined below:

- **Matching Elements**: The more elements that an approach takes into account for matching, the more accurate it is. For each option (i.e. IO, PE, and Other) of Matching Elements that is supported by the approach, one score is assigned to it.
- **Support for DoM, Multi Stage Matching, and Intelligency**: If an approach supports each of these abilities, it is more accurate than the approach which does not. For each of such abilities that is supported by the approach, one score is assigned to it.

The overall scores of the above parameters determine the degree of Accuracy. We evaluate total scores of one or two as “Low”, total scores of three or four as “Average”, and total scores of five or six as “High”.

As shown in Table 2, the Scalability of an approach is estimated based on the overall result of two criteria as stated in the following:

- **Support for Different Ontologies**: If a matchmaking approach has the ability to support different ontologies for the service provider and the service requester, then it is more scalable than the approach which does not have this ability.
- **Support for UDDI**: Currently, most of the Web services are non-semantic. Obviously, it is more beneficial to reuse these preexisting components in the new semantic systems. On the other hand, a considerable effort is required to semantically model Web services using formal logic. Supporting UDDI by a semantic approach means that it can also support matching non-semantic Web services, as it is more scalable than the approach which does not have this ability.

To determine the degree of Scalability, if an approach supports both of the above parameters, we evaluate it as “High” and if it supports only one of them, it is evaluated as “Average”, otherwise it is deemed as “Low”.

Finally, as illustrated in Table 3, our proposed matchmaking categories are compared by means of Query Response Time and Performance. For this comparison, we mainly rely on the results of the evaluation experiments provided by some of the prominent approaches, as explained below:

- **Query Response Time**: Based on the experimental evaluations in [27], and [34], the Query Response Time for Logic-&Non-Logic-based matching variant of WSMO-MX [24] is slightly slower than its pure Logic-based matching but significantly slower than its Non-Logic-based matching variant. Thus, Non-Logic-based matching is faster than the Logic-based matching variant. These results are in line with that of SAWSDL-MX2 and thus fortify the proposition that Non-Logic-based matching outperforms both Logic-based and Logic-&Non-Logic-based matching in terms of Query Response Time. Those approaches which combine logic-based and syntax-based matchmaking, e.g. [25], are more effective than pure logic-based approaches, since they provide a filtering mechanism to reduce the set of advertisements that are potential candidates for matchmaking. This way the discovery process is faster. However, we are not able to compare Logic-&Syntax-based category with other approaches, by just basing on such theoretical preferences and so far no evaluation has been done or published as yet in comparing this category with others. Based on the above discussion, we consider the Query Response Time level of the whole Non-Logic-based category as “Low”, which is the best, Logic-based category as “Average”, Logic-&Non-Logic-based category as “High”, and Logic-&Syntax-based category as “N/A”, i.e. not available.
Performance: For comparing the Performance of the categories, we rely on [22], [23], [36] and [27]. All these references have evaluated the retrieval performance based on the Recall and Precision measures. Their agreeing results show that incorporating non-logic-based techniques into the logic-based algorithms improves the performance of the matchmaking. Since it causes some syntactically similar, but logically disjoint services be truly included in the answer set of some queries. Also, generally Logic-based matchmakers result in higher Recall and Precision compared to Non-Logic-based ones. Finally, in most cases, using the Syntax-based matching in addition to the Logic-based determines the sets of services or service providers [25], i.e. it only restricts the searching domain. Therefore, it makes no difference with the pure Logic-based matching in terms of the performance measures. Based on the above discussion, we consider the Performance level of the whole Logic-Non-Logic-based category as “High”, that is the best, both Logic-based and Logic-Syntax-based categories as “Average”, and Non-Logic-based category as “Low”. It is worth noting that in situation where high Performance is needed, Logic-Non-Logic-based matchmakers do not have any competitors.

An indirect result of the above discussion is that increasing the Performance will cause the Query Response Time to increase. Consequently, with the exception of special cases in which high level of accuracy is needed, there should be a tradeoff between these two criteria.

In a nutshell, Table 3 can be used to compare our proposed matchmaking categories, while Table 1, 2 are especially used in evaluating the approaches of each category and then comparing them with one another. The results and designation of such tables may be used to select or promote an approach or a category, by determining its weaknesses and strengths.

5.2. Evaluation of Uncommon Criteria

In the following, some uncommon but still important criteria are proposed. They are termed as uncommon and are treated separately because either they are not supported by the selected prominent approaches of the previous section or they are meaningful in the specific matchmaking models, i.e. not to be supported by all of the approaches. Consequently, after proposing such criteria, they are evaluated (instead of approaches) and based on that we could study their application feasibility in Web service discovery approaches. The results of such evaluation can be seen in Table 4.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Supporting Approaches</th>
<th>Supported Matchmaking Model</th>
<th>Possible Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Behavior Matching</td>
<td>(Bansaill+ 03) [39],</td>
<td>Independent</td>
<td>▼ ▲ ▲</td>
</tr>
<tr>
<td></td>
<td>(Hu+ 05) [38],</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Vaculin+ 07) [40],</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Günay 08) [37]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support for Indirect</td>
<td>(Bose 08) [41]</td>
<td>Independent</td>
<td>▼ ▲ ▲</td>
</tr>
<tr>
<td>Matching</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support for IoM</td>
<td>(Keller+ 05) [42],</td>
<td>Set-based</td>
<td>▲</td>
</tr>
<tr>
<td></td>
<td>WSMO-MX (Kaufner+ 06)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[24]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support for Filter Selecting</td>
<td>WSMO-MX (Kaufner+ 06)</td>
<td>Hybrid and Multi Stage</td>
<td>▲ ▲ ▲</td>
</tr>
<tr>
<td></td>
<td>[24],</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OWLS-MX (Klusche + 09)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[22]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support for Different</td>
<td>N/A</td>
<td>Hybrid and Multi Stage</td>
<td>▲</td>
</tr>
<tr>
<td>Execution Orders</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accessibility</td>
<td>Accessible via UI</td>
<td>All</td>
<td>▲</td>
</tr>
<tr>
<td></td>
<td>Accessible via API</td>
<td>WSMX [43], ALIVE [44]</td>
<td></td>
</tr>
<tr>
<td>Indepency</td>
<td>N/A</td>
<td>Independent</td>
<td>▲</td>
</tr>
</tbody>
</table>

Table 4. Evaluation of Uncommon Criteria
5.2.1. Process Behavior Matching

Most of the current Semantic Web service discovery systems perform matchmaking in the level of service profile. In service profile matching (so-called “black-box” service matching), the semantic similarity between the services is determined based on the description of their profiles. The profile of a service describes its functionality in terms of the service signature that is aforementioned IOPE quadruple in addition to the non-functional properties of the service. On the other hand, a few approaches perform matchmaking in the service process level. The process behavior matching (so-called “glass-box” service matching) determines the extent to which the operational behavior of two services matches with each other. The operational behavior of a service is described in terms of its process control and data flow [10].

Some researchers state that service profile matching methods suffer from low precision due to lack of use of internal process knowledge of services in matchmaking. The inadequacy in precision of matchmaking cannot guarantee compatibility of the Web service composition, nor can it make the Web services easy to complete a dead-lock free and bounded process transaction [37], [38], [39].

In order to enable the discovery according to process behavior, in [38] Web services are modeled as Annotated Deterministic Finite State Automata (ADFSA), which help to match the different business processes. So, Web services are published via ADFSA that is mapped from a process description. Based on this model, two kinds of service matching have been defined, namely Exact and Plugin matching. There are other existing approaches and standards in order to define the behavior of business processes and interactions among them, such as BPEL4WS and OWL-S. However, they require a much higher computational complexity compared to FSA. Besides, ADFSA constitutes a foundation for the purpose of automatic composition of services at the level of executable processes.

As another example, [37] proposes two matchmaking methods based on the use of the internal process models of services. Both of the proposed methods use FSA techniques to model the internal processes of services. In order to enhance the model, the related elements are associated with the semantic concepts from an ontology.

As an alternative method to perform process behavior matching, [40] and again [37] make use of the model checking techniques for matchmaking. In this method a process specification language is used to represent the details of the internal process of the services. On the other hand, the service requests are being modeled by a set of logical formula. Model checking techniques will then be applied using these service models and service requests. The result of this model checking could be used for the process oriented matchmaking.

As mentioned before, the service process model provides a far richer description of a service than the service profile. Consequently, matching based on the service process behavior has more precision than service profile matching. This will in turn lead to better Performance. Such improvement might inversely affect the Query Response Time, especially when resolving complex processes. Therefore, researchers should consider inventing and using more optimized algorithms.

5.2.2. Support for Indirect Matching

In [9] a twofold classification of matchmakers was introduced. This classification is based on the ability of the matchmaker to perform matching whether against a single service advertisement or against composite services, constructed from multiple service advertisements. These are called Direct and Indirect Matching, respectively. So, Support for Indirect Matching is defined as the ability of an approach to perform composite service matchmaking. Most of the current approaches are Direct matchmakers. Even the same reference does not mention any concrete discovery approach performing Indirect Matching. One of the rare works that supports Indirect Matching is proposed in [41]. The author devised a three-phased Semantic Web service discovery methodology and implemented it. After finding semantically similar Web services for a user query in the first phase, if a single Web service is unable to fully satisfy his request, a composition of multiple inter-related Web services is presented to the user. The task of checking the possibility of linking multiple Web services, namely the link analysis is done in the second phase.

It is reasonable to suppose that Indirect Matching increases the potential of retrieval of all Web services a user is interested in i.e. Recall. Accordingly, increases in this parameter will enable the Performance of the discovery approach to improve.
5.2.3. Support for IoM

Some approaches represent Web services by means of sets of objects that can be delivered. In this case, a modeler might want to express that either all of the elements that are contained in the set, or only some of them are requested or can be delivered. Therefore, he has to explicitly specify his intention when describing the set of relevant objects for a Web service. The concept of IoM (Intention of Matching) refers to intuitive notions of match a requester or a provider has in mind when requesting or advertising a service [42].

We define **Support for IoM** as the ability of considering intentions in set-based approaches. This ability has been supported in very few discovery approaches so far (e.g. [42] and WSMO-MX [24]). However, we believe that such ability, gives the modeler additional freedom in modeling and results in delivering more accurate Web services.

5.2.4. Support for Filter Selecting

In the discovery systems with multiple matching filters, such as the hybrid matchmakers (e.g. Logic-&Non-Logic-based and Logic-&Syntax-based approaches) and multi stage matchmakers, it is desirable if the system allows the user to choose from various filters. This will activate/deactivate applying each matching filter, on demand.

We define **Support for Filter Selecting** as the ability of an approach to allow selecting between its matching filters. Currently, only a small number of hybrid matchmakers (e.g. OWLS-MX [22] and WSMO-MX [24]) allow partial selecting of their matching filters. Discarding some of the matching filters, reduce the time to responding to a query, but also might inversely affect the **Precision** of the result set.

5.2.5. Support for Different Execution Orders

In discovery systems where there are several existing matching filters, it is of interest if the system allows the user to define the order of execution of its various filters. Different execution orders could be: 1) **Parallel**, i.e. each of the matching filters executes independently. All intermediate outputs will be integrated to achieve the final results. 2) **Sequential**, i.e. matching filters execute sequentially. The output of each step will be the input for the next step.

We define **Support for Different Execution Orders** as the ability of an approach to allow selecting between different execution orders.

To the best of our knowledge, none of the current Web service discovery systems supports the proposed ability. However, wise selection of an execution order to apply, might cause one or both of the **Performance** parameters, i.e. **Precision** and **Recall** to improve.

5.2.6. Accessibility

To enable the reuse of software and automated processing, we should be able to invoke a matchmaker as a component via an Application Programming Interface (API) as well. We define **Accessibility** as the ability to invoke a matchmaker via different interfaces. If an approach supports both application programming and user interfaces, it has maximum **Accessibility**. Currently, almost all of the matchmakers are accessible via their User Interfaces (UIs), while only WSMX [43] and recently ALIVE [44],[45] allow their discovery components to be programmatically invoked.

5.2.7. Independency

A large number of Semantic Web services have been developed based on different description languages, like OWL-S, WSML, and SAWSDL. Existing Semantic Web services discovery systems only support matching based on specific description languages. Suppose a Web service uses OWL-S to describe the service. A request uses WSML-Rule for service description. OWL-S uses OWLJessKB while WSML-Rule uses IRIS as a reasoner. The service which is described by OWL-S can meet the requirements of the Web service request which is described by WSML-Rule. However, these two services cannot be matched using the current Semantic Web services discovery systems because their
reasoners do not interact with each other [7]. We define Independency as the ability for matching Semantic Web services that use different description languages. Consequently, Independency will improve both the potential of the retrieval of all Web services matching a service request and the usage domain of a matchmaker in terms of different description languages.

5.2.8. Discussion

As illustrated in Table 4, our proposed Uncommon Criteria was evaluated based on their Supporting approaches, Supported Matchmaking Models, and Possible Effects as follows:

Wherever possible, some concrete Semantic Web service matchmakers supporting each criterion have been mentioned in Supporting Approaches. Such concrete approaches act as proof-of-concept for each criterion. While “N/A” means there is no available approach, “All” sits in place of all current approaches. Regarding the criteria with no supporting approach, based on their Supported Matchmaking Models and Possible Effects, they might be considered when implementing future matchmakers.

Supported Matchmaking Model shows the dependency of criteria to a matchmaking model. While Independent criteria can be evaluated for all of the approaches, others necessitate specific underlying matchmaker models. For instance, Support for IoM can be evaluated just for Set-based approaches. In these matchmakers, Web services and requests are represented as sets of objects that can be delivered by the services or desired by the requesters, respectively. As another example, Support for Filter Selecting and Support for Different Execution Orders are meaningful to be evaluated just for Hybrid and Multi Stage approaches. These include either hybrid (Logic-&Non-Logic-based and Logic-&Syntax-based) or multi stage matchmakers which have multiple matching filters.

Finally, possible effects of supporting each criterion by an approach on its Usability, Query Response Time, and Performance have been evaluated in Possible Effects. These parameters could improve, worsen, or remain neutral as are shown by the respective symbols of “▲”, “▼”, or “▬”. Here, Usability is defined as the extent of the usage domain of a matchmaker. For Query Response Time and Performance, we rely on definitions in 5.1.3.1 and 5.1.3.2, respectively.

Results of the above evaluation could help in studying the applicability of Uncommon Criteria which is useful in evaluating existing as well as considering them in new Web service discovery approaches.

6. Conclusion and Future Works

In this paper, we aimed to provide an overview and evaluate and compare recent progress in Web service discovery approaches. Also, our proposed list of criteria may serve as the features that may be considered in developing the Semantic Web service discovery approaches. This list has been extracted based on the analysis of the requirements of such approaches.

After introducing a taxonomy of Web service discovery systems, we presented the classification of Semantic Web service discovery approaches. We also proposed a comprehensive set of criteria which are mainly used in evaluating Web service discovery approaches. Finally, the categories of our classification are evaluated and compared with respect to the criteria that are more commonly used, while some other more specific criteria have been evaluated in order to study the feasibility of their applicability. Noteworthy, the results of our comparison show that there is no single superior Semantic Web service discovery approach in all cases. Therefore, deciding which classification or approach to use in a certain scenario should be done based on both its specifications and preferred qualitative criteria.

It should be noted that although the conducted comparison is based on the most prominent approaches, the outcome of this research is not restricted to such approaches. In other words, our classification of the approaches as well as our considered criteria could serve as a guideline for evaluating or developing Web service discovery approaches.

Finally, we identify the subsequent lines of research to be followed more in the field of Web service discovery. Most of the current approaches perform matching on functional aspects of Web services such as input, output, pre-condition, post-condition, and operation. A few approaches, for example [46], [47], [48], and [49] rely on Quality of Service (QoS) attributes that are defined as non-functional
aspects of Web services. These attributes include availability, response time, reliability, performance, security, price, and so forth. There is a need for matchmaking mechanisms that support functional elements as well as QoS properties. This will make the discovery systems more precise and feasible. Another interesting line of research could be incorporating advanced Information Retrieval (IR) techniques especially those adopt Natural Language Processing (NLP) in non-logic-based matchmaking. Last but not least is constructing hybrid discovery systems by putting together the Web service profile and the process matchmakers. In such systems, a control component should be designed to decide on the use of an appropriate matchmaker when applicable.

7. Acknowledgments

This research is supported by the RMC-RU grant of Universiti Teknologi Malaysia (UTM) under the Vot no. 00H74. We would like to thank UTM, Malaysian Ministry of Higher Education (MOHE) and individuals who have been involved either directly or indirectly in this project.

8. References


