Journal of Theoretical and Applied Information Technology
15th October 2022. Vol.100. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A MULTI-AGENTS SYSTEM BASED APPROACH TO WEB SERVICE DISCOVERY

MOHAMED HALIM[1], NOUHA ADADI[2], MOHAMMED BERRADA[3], DRISS CHENOUNI[4]

[1, 2, 4] IPI Laboratory, Sidi Mohamed Ben Abdellah University –Fez, Morocoo

[3] IASSE Laboratory, Sidi Mohamed Ben Abdellah University -Fez, Morocoo

E-mail: [1]mohamed.halim@usmba.ac.ma, [2]nouhaadadi@gmail.com, [3]mohammed.berrada@gmail.com, [4]d_chenouni@yahoo.fr

ABSTRACT

A growing number of companies are using web services to make their expertise and data available through the network. The current problem is that the content of these web services remains inaccessible to machine processing. Only humans can interpret their contents. The Semantic Web is the new vision of the Web that promises to overcome this difficulty. The concept of semantic web services, is the result of the convergence of the field of web services with the semantic web, indeed its objective is to automate the discovery, selection and composition of web services. In this work, we are interested in the semantic discovery of Web services. The main problem is to automate the discovery of web services to respond to a request from a client. In this sense, firstly we present a conceptual framework and architecture to carry out our approach. The originality of the proposed solution lies in the use of mixed technical tools ranging from semantic models to multi-agents systems, including Matchmaking algorithms. Afterwards, we implement our proposed architecture. To validate our work, we conduct tests with a variety of user queries and a panel of Web services. As part of a case study we consider an online travel organization problem. This problem is a typical web services discovery scenario to apply the concepts of our approach.

Keywords: *Semantic Web, Web services, Semantic discovery, Multi-agents systems, Matchmaking algorithms.*

1. INTRODUCTION

The semantic web, invented by Tim Berners-Lee [1] (director of the W3C), is to define a new generation of the web. It designates a set of technologies aimed at making today's web a vast space for the exchange of human and machine resources through an explicit representation of the semantics of data, programs, web pages and web services.

This infrastructure allows the use of formalized knowledge in addition to the current informal content of the Web and to locate, identify and transform resources in a robust and healthy way by relying on a system of formal metadata, representation languages developed by the W3C and the ontologies that represent the key technology of the semantic web.

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-process able format (specifically WSDL). Currently, descriptions of Web services are published in UDDI registers. The aim of the latter is to facilitate the search for services published by the various companies. However, given the large number and diversity of Web services, their discovery remains a daunting task. The discovery of Web services is an emerging area of research. Initially, the discovery is made in the UDDI registry, it is based primarily on research syntactic WSDL descriptions of Web services. But with the development of Semantic Web technologies, the techniques for discovery have become essentially semantic. This semantics is provided through one of ontologies important technologies of the Semantic Web. Thus, software agents can be developed to reason about these ontologies, making the discovery of Web services dynamic and automatic.

In this work, we propose an approach to discovery of semantic web services using agent technology and ontologies.

The layout of this paper is as follows. The second section presents an overview of web services discovery approaches and Multi Agent Systems. In the third section we focus on the presentation of the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

proposed approach. The fourth section is devoted to results of the work and discussions. The conclusion and future work is presented in the fifth section.

2. LITERATURE REVIEW

3.2. Web services Discovery

Web service discovery is a process that aims to compare a user request with the capabilities of a web service, and sorts the results according to some mechanism. Capabilities can be functional (interface descriptions: syntactic or semantic, behavioral descriptions) or non-functional (quality of service). Several criteria can be used to categorize discovery approaches [2], we have:

- ✓ The criterion of centralization/distribution of directories.
- ✓ The principle of the matching algorithm (syntactic, semantic (logical, non-logical), hybrid, behavioral, non-functional, etc.)
- ✓ The automation criterion

Different approaches have been proposed to achieve dynamic service discovery. In figure 1, we present a classification of discovery approaches.

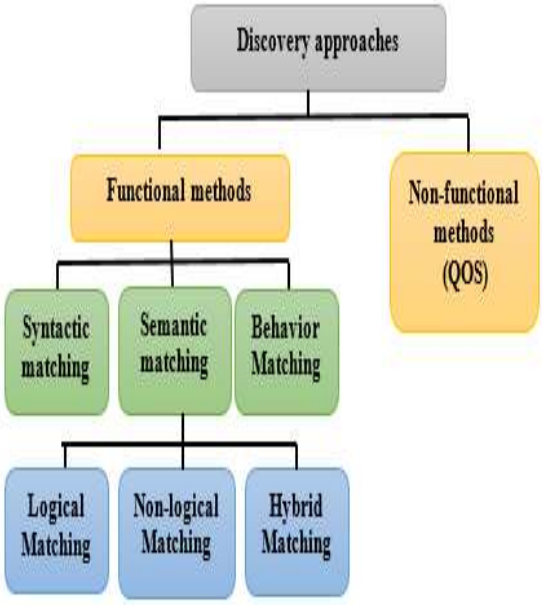

Figure 1: Classification of web services discovery approaches

The first service discovery approaches proposed in the literature were syntactic approaches, they are based on the use of a register of Web service descriptions and on the comparison of the keywords sent by the request. With the emergence of the new generation of the web, semantic approaches have appeared. These adopt ontologies for the matching of the request with the services. They use logic, datamining, and even similarity measures to compare semantic interfaces such as OWLS [3], SAWSDL [4] and WSMO [5]. We also find in the literature Non-Functional approaches (QOS) [6] which are based on reputation as a means of discovering services, as well as behavioral approaches which support the notion of behavior of processes, (or execution paths) , this behavior must be modeled with formal means such as finite state automata and algebraic processes.

3.3. Multi-Agents Systems

An agent is an autonomous entity, real or abstract, which is able to act on itself and on its environment, which, in a multi-agent universe, can communicate with other agents, and whose behavior is the consequence of observations, knowledge and interactions with other officers. Indeed Multi-agent systems (MAS) offer a new approach for the development of composite, distributed and complex information systems.

A Multi-Agent Reactive Decisional System (MARDS) [7] is a software structure characterized by a set of Decisional Reactive Agents (DRA), interconnected by communication interfaces. The MARDS concept was applied in different domains like the automated systems of production, the mobile systems [8], organizational system [9], and the composition of Web services [10].

The internal structure of a MARDS is based on a two-level tree (Figure 2), composed in parallel of a Supervisor DRA (DRAS), two or more possible sub-agents, which can be in turn MARDS and interfaces communication between the supervisor and his sub-agents. For a MARDSj, it can be either a simple DRA or a MARDS built recursively.

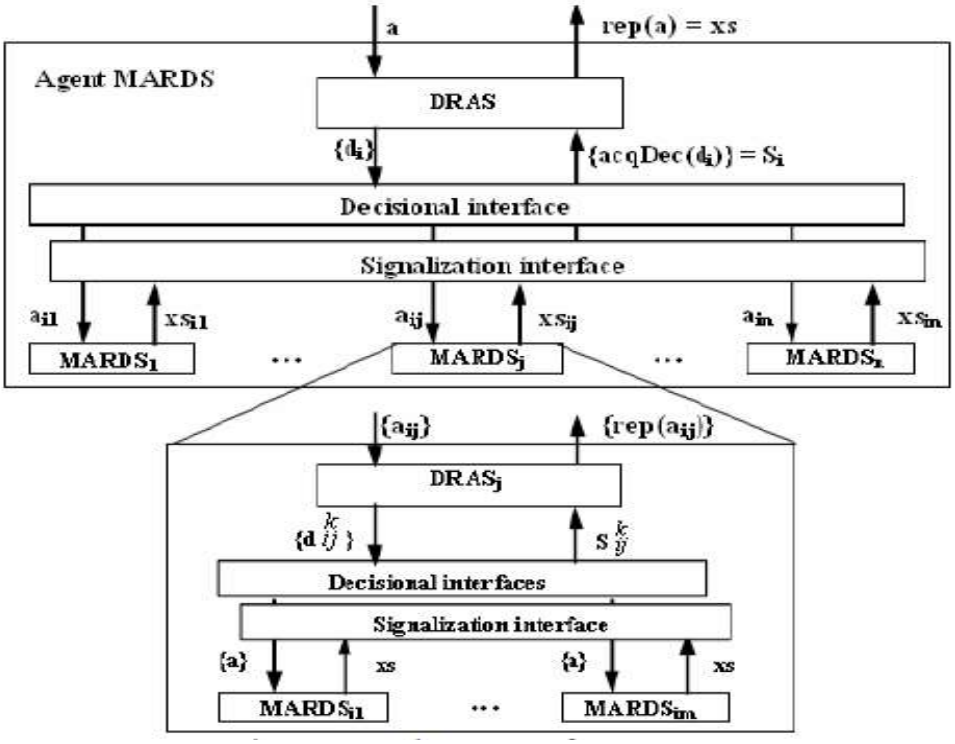

Figure 2: Internal Structure of a MARDS

3.4. Summary

The approaches, standards and languages described in this section emphasize the mechanism for comparing the request with the services, as well as its degree of automation. According to our point of view, service discovery aims to compare a user request with the capabilities of a web service, and sorts the results according to a certain mechanism. Capabilities can be functional (interface descriptions: syntactic or semantic, behavioral descriptions) or non-functional (quality of service). In our approach, we have adopted the semantic approach and more precisely that based on the OWL-S anthologies and matching algorithm.

The MARDS model constitutes an approach among the newest and most useful ones for the composing and modeling of complex system such as the automated systems of production, the mobile systems and organizational system. We will use this system for discovering web services. We propose using this model for the dynamic discovery of web services.

3. PROPOSED APPROACH

3.1. Architecture for discovery of web services

We propose an architecture (Figure 3) based on a multi-agent system and ontologies, of web services discovery approach.

Journal of Theoretical and Applied Information Technology
15th October 2022. Vol.100. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

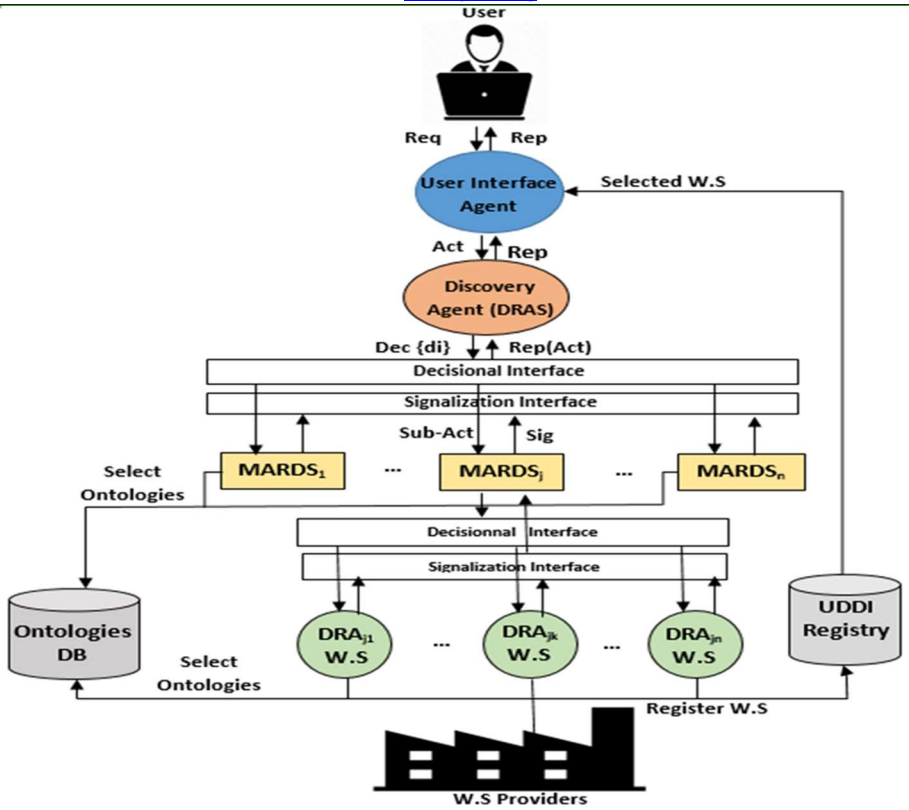

Figure 3: Proposed architecture of Web services discovery approach based on MARDS

As illustrated in the figure above, the architecture is composed of six main elements: The user; the user interface agent; the Discovery Agent which is the Supervisor (DRAS) of MARDS system, Decisional Reactive Agent (DRA); the ontologies data base and the UDDI registry.

- ✓ **User**: is the one who expresses the request, it can be a human or software.
- ✓ **User interface agent**: the user interface agent uses processing and analysis rules to translate the request expressed in natural language into a semantic request respecting the domain ontology. Indeed, it is the gateway of external requests to the system which obtains a semantic and formal representation of the user's request. This request management work, ultimately, results in a list of semantically described action with input-output parameters sent to the Decisional Reactive Agent Supervisor (DRAS) which present the Discovery Agent. The reverse operation is also available, i.e. User interface agent receives the results from the discovered agent and then selects the relevant services from UDDI registry to present them to the user.
- ✓ **Discovery Agent**: It is an agent that allows the discovery of descriptions of Web services satisfying the request sent by the user on the semantic level. This agent present the supervisor of Multi-Agent Reactive Decisional System (DRAS).Indeed, it has the role of generating several decisions {di, i = 1 ... m} on receipt of the initial action {a}. Each of these decisions will be translated by the decision interface to one or more actions {aik, k = 1 ...n} appropriate to the lower level agents {MARDSk}. The external states issued by these agents will be translated by the signaling interface to a single signaling {si}, which represents the acquittal of the decision {di}. The MARDS agents will subsequently generate the decision {di + 1} or just the final external state {e} considered by the MARDS agent as a response to the initial action. This response returned to the supervisor contains the description of the list of relevant services which respond to the user's request.
- ✓ **Decisional Reactive Agent**: Each DRA agent presents abstract instance of a Web services. Concrete instances of Web services are

registered by the provider in the UDDI directory and possessing a domain ontology in the ontology database. This Decisional Reactive Agent has the task of deciding whether the web service linked with it can respond to the user's request or not. For making this decision, firstly, the DRA receives the sub-actions from his supervisor. Then, it selects the corresponding domain ontology. Afterwards, it performs the search algorithm which allows comparing the inputs and outputs of the services published in the OWL-S ontology with those defined in the request. Ultimately, it sends the response to the supervisor (discovery Agent) in the form of an external state {xs}.

- ✓ **Ontologies DB**: Data Base that stores and contains the concepts and ontologies of various domains, their hierarchy and the relationships between them.
- ✓ **UDDI**: is a directory that allows the registration of web services and facilitates their discovery by offering an XML-based data structure and an integration API. The use of the UDDI directory allows the supplier to present itself and publish its services to accelerate their exchanges via an operator on the web. It behaves itself like a web service whose methods are called via the SOAP protocol. The primary purpose of a UDDI registry is to provide a basic infrastructure for publishing, discovering, and invoking services.

3.2. The matching procedure

The matching procedure is composed of two essential phases:

- ✓ **The analysis phase:** selects the domain ontology corresponding to the request from the ontology base, extracts the classes and their links and builds the corresponding tree structure. Each vertex of this tree corresponds to a class of the ontology and each arc corresponds to a subclass relation. This tree structure makes it possible to deduce subsumption relations between the concepts, i.e. the fact that one concept is more general than another. A concept C subsume a concept C' if the extension of C' is included in that of C. We will then say that C is more general than (or subsume) C'. This principle allows us to make flexible comparisons between offers and requests, i.e. to associate offers with a request which do not correspond exactly to the needs expressed but which are close to them.
- ✓ **The comparison phase:** allows to compare a request and offers of services by considering the ontology and this in accordance with the four main modes of comparison defined in [11] by using a matchmaking algorithm: Exact mode, PlugIn mode, Subsume mode and Fail mode.

  - The Exact mode selects an offer if it corresponds exactly to an offer (demand = offer) i.e. the inputs and outputs of the request are equivalent to the inputs and outputs of the offer (exact matching).
  - Plug-In mode returns a supply if it encompasses a demand (demand<supply) i.e. demand inputs encompasses supply inputs and demand outputs are encompassed by supply outputs offer in the domain ontology (inclusive matching).
  - Subsume mode returns an offer if it is include in a request (request>supply) (the Inverse of Plug-In mode) (partial matching)
  - Fail mode returns false, if no match between supply and demand (request # supply) (matching failure).

The comparison algorithm used both in Plug-In mode and in Subsume mode uses the Subsume function [12] given below (figure 4).

The agent applies a subsumption test on the outputs (figure 5) then, a score is assigned for each matching mode: Exact (score=3), PlugIn (score=2), Subsume (score=1), Fail (score=0) (Figure 6).

```
Function Subsume (E1: string, E2: string): Boolean
% This function returns true if E1 subsume E2 false otherwise
% E1 is an element of the Input or Output clause of the Offer
%E2 is an element of the Input or Output clause of Request
% A represents the ontology (in tree form)
% The following high-level functions are used:
% Father (E): returns the father of E in A
% Root (A): returns the root of A
Variables
CurrentVertex: AVertex % Vertex of A being examined
Ancestors: SetofVertices % The ancestors of E2
Beginning
Ancestors ← Ø
If E2 = root (A) Then
% E2 has no ancestor and cannot be subsumed
Ancestors ← Ø
Otherwise
CurrentVertex ← Father(E2)
Ancestors ← Father(E2)
While (CurrentVertex < >Root(A)) Do
VertexCurrent ← Father(VertexCurrent)
% "+" denotes adding a new item
% in the set Ancestors
Ancestors ← Ancestors + CurrentVertex
End While
End if
Subsume (E1 ∈ Ancestors)
End
```

Journal of Theoretical and Applied Information Technology
15th October 2022. Vol.100. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 4: Function Subsume

```
Procedure degreeOfMatch(OutD,OutO : string )
% This Procedure returns comparison result
% OutD, OutO are the output of demand and Offer respectively
Variables Ch: string
Beginning
If OutO = OutD Then ch ← "Exact"
If Subsume(OutO, OutD) Then ch ← "PlugIn"
If Subsume(OutD, OutO) Then ch ← "Subsume"
Otherwise ch ← "Fail"
End if
Return ch
End
```

Figure 5: Output matching procedure

```
GetScore(rel: string) function: Integer
% This Function returns the matching score
Value =0
Beginning
If rel = "Exact" Then val = 3
If rel = "PlugIn" Then val=2
If rel = "Subsume" Then val=1
If rel = "Fail" Then val=0
End if
GetScore value
End
```

Figure 6: Function returns the matching score

3.3. Synthesis

The following diagram (Figure 7) formalizes the process, described above, of the proposed WS discovery approach in a BPM notation.

- ✓ The user interface agent presents a form to the system user for filling (input, output, etc.).
- ✓ Once a request is received, the semantic information of the request is transferred to DRAS in the form of an action.
- ✓ The MARDS multi-agent system transmits this action through the decision interface to the underlying DRA agents, which process the OWL-S ontologies of services related to them by applying matching algorithms.
- ✓ The result is a set of pertinent Web services based on the level of semantic correspondence that satisfies the user request.
- ✓ The results are sent to the Discovery agent via the MARDS system.
- ✓ The final result is presented to the user by UI agent which invoke the web service (s) of interest.

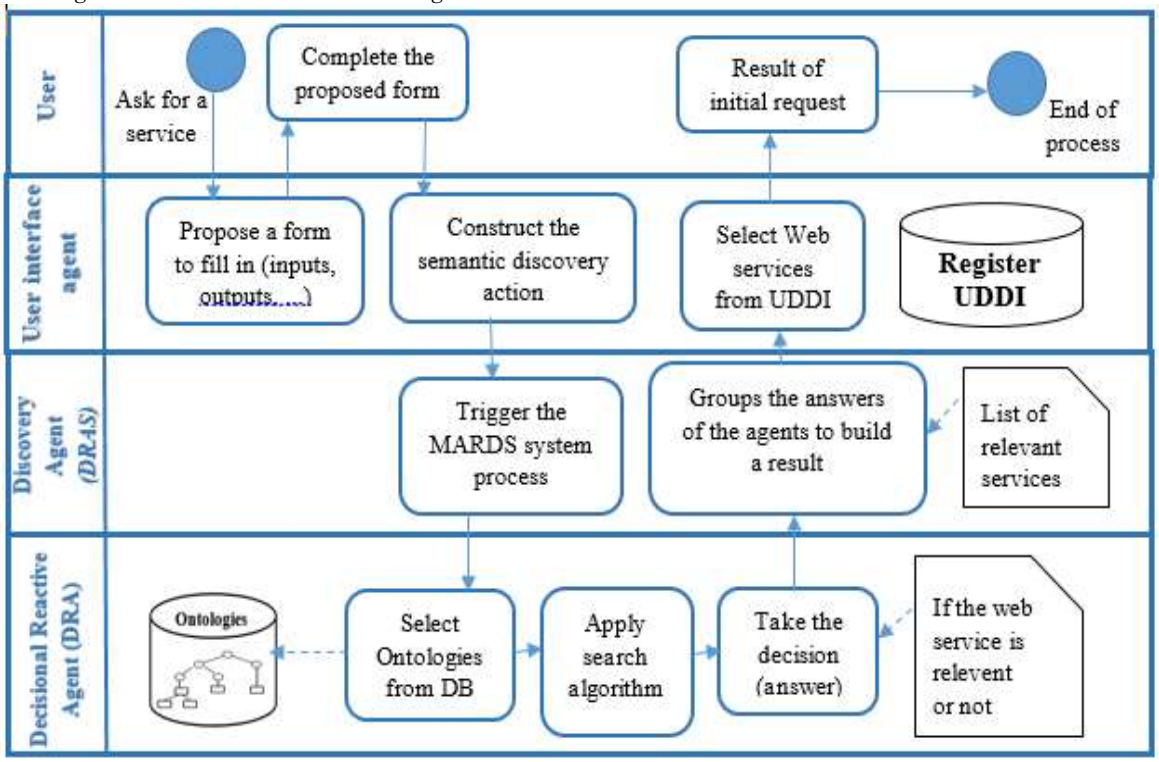

Figure 7: Interaction between the components of the proposed architecture

Journal of Theoretical and Applied Information Technology
15th October 2022. Vol.100. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. RESULTS AND DISCUSSION

4.1. Results:

As an illustrative example, we will consider in this work an online Travel Organization problem.

Assume that there are three Web services: Hotel Service, Flight Service, and Car Rental Service published on the Web. Functional parameters (inputs, outputs) are presented in figures 8, 9 and 10.

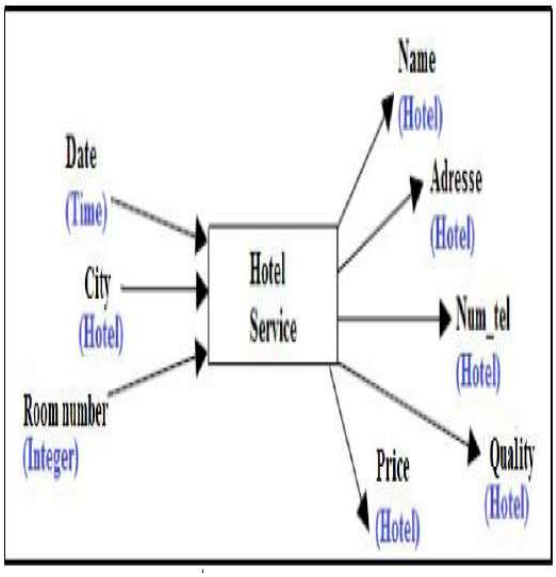

Figure. 8: Hotel Service

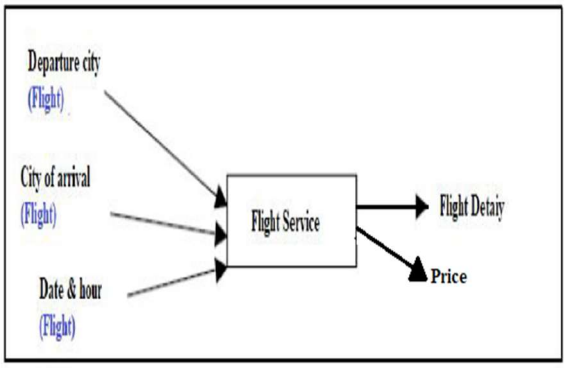

Figure 9: Flight Service

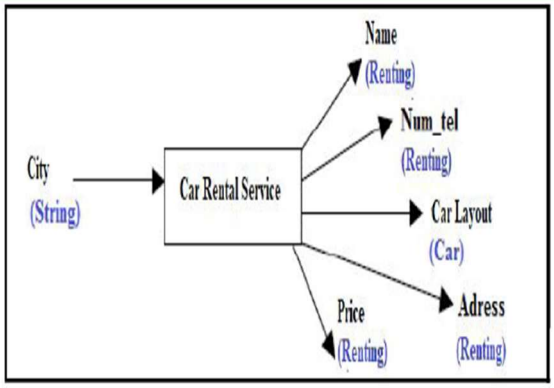

Figure 10: Car Rental Service

Consider a user request R contains one output "Price" and two inputs "City of arrival" and "Date".

Given the ontology fragment displayed in figure 11.

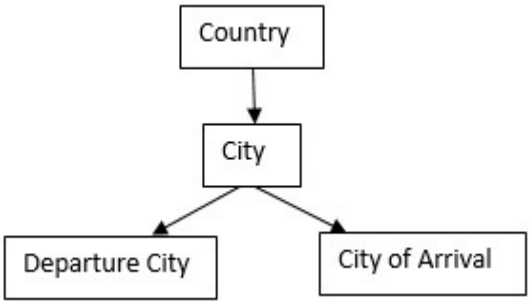

Fig. 11: A fragment of the City ontology

Applying the matching algorithm gives the following results:

✓ **Hotel Service**

- *Inputs comparison:*

City of arrival □□City, Founded Input Relation is = Plug-in, Their score = 2, Total Inputs Score is : 2
City of arrival □□Date, Founded Input Relation is = Fail, Their score = 0, Total Inputs Score is: 2
City of arrival □□Room Number, Founded Input Relation is = Fail, Their score = 0, Total Inputs Score is: 2
Date □□Date, Founded Input Relation is = Exact, Their score = 3, Total Inputs Score is: 5
Date □□City, Founded Input Relation is = Fail, Their score = 0, Total Inputs Score is: 5
Date □□Room Number, Founded Input Relation is = Fail, Their score = 0, Total Inputs Score is: 5

- *Outputs comparison:*

Price □□Price, Founded Output Relation is = Exact, Their score = 3, Total Outputs Score is :3
Price □□Name, Founded Output Relation is = Fail, Their score = 0, Total Outputs Score is :3
Price □□Adress, Founded Output Relation is = Fail, Their score = 0, Total Outputs Score is :3

Price □□Num_Tel, Founded Output Relation is = Fail, Their score = 0, Total Outputs Score is : 3
Price □□Quality, Founded Output Relation is = Fail, Their score = 0, Total Outputs Score is : 3

- *Global matching:*

Total score (Hotel Service) = 5 + 3 = 8

✓ **Flight Service**

- *Inputs comparison:*

City of arrival□□City of arrival, Founded Input Relation is = Exact, Their score = 3, Total Inputs Score is: 3
City of arrival □□Departure city, Founded Input Relation is = Fail, Their score = 0, Total Inputs Score is: 3
City of arrival □□Date, Founded Input Relation is = Fail, Their score = 0, Total Inputs Score is: 3
City of arrival □□Time, Founded Input Relation is = Fail, Their score = 0, Total Inputs Score is: 3
Date □□ City of arrival, Founded Input Relation is = Fail, Their score = 0, Total Inputs Score is: 3
Date □□ Departure city, Founded Input Relation is = Fail, Their score = 0, Total Inputs Score is: 3
Date □□Date, Founded Input Relation is = Exact, Their score = 3, Total Inputs Score is: 6
Date □□Time, Founded Input Relation is = Fail, Their score = 0, Total Inputs Score is: 6

- *Outputs comparison:*

Price □□Price, Founded Output Relation is = Exact, Their score = 3, Total Outputs Score is: 3
Price □□Flight detaiy, Founded Output Relation is = Fail, Their score = 0, Total Outputs Score is: 3

- *Global matching:*

Total score (Flight Service) = 6 + 3 = 9

✓ **Car Rental Service**

- *Inputs comparison:*

City of arrival □□City, Founded Input Relation is = Plug-in, Their score = 2, Total Inputs Score is: 2
Date □□City, Founded Input Relation is = Fail, Their score = 0, Total Inputs Score is: 2

- *Outputs comparison:*

Price □□Price, Founded Output Relation is = Exact, Their score = 3, Total Outputs Score is: 3
Price □□Name, Founded Output Relation is = Fail, Their score = 0, Total Outputs Score is: 3
Price □□Adress, Founded Output Relation is = Fail, Their score = 0, Total Outputs Score is: 3
Price □□Num_Tel, Founded Output Relation is = Fail, Their score = 0, Total Outputs Score is: 3
Price □□Car Layout, Founded Output Relation is = Fail, Their score = 0, Total Outputs Score is: 3

- *Global matching:*

Total score (Hotel Service) = 2 + 3 = 5

Comparing the global matching of different services, it is concluded that, the Flight Web service is regarded as the best corresponding to the request.

4.2. Discussion:

The majority of state-of-the-art works model the functional aspect by the signature of operations i.e. (inputs and outputs of services). As we have already mentioned in the literature review part, these solutions are classified into three categories:

- Logical semantic approaches
- Non-logical semantic approaches
- Hybrid semantic approaches.

The major drawbacks of logical approaches are:

- The exponential complexity of concept matching (non-scaling).
- The presence of false positives and false negatives
- We cannot compare services with the same logical score. (and therefore numerical scores are more adaptive and flexible)

Hybrid approaches have improved the performance of logical approaches, but they still remain below the performance (recall and precision) of some non-logical approaches such as [13].

The primary problem of non-logical approaches lies in the choice of concept matching techniques (such as similarity measures) and/or algorithms for optimizing these matchings.

To overcome the difficulties mentioned above, we have chosen algorithms belonging to the non-logical class, and more precisely we have chosen a simple similarity measure to compare the concepts of the request and the services. This similarity measure is described in equation 1. Assuming there are m concepts in a service description and there are m corresponding concepts in a service request, the similarity or global match between the request R and the service S can be derived by summing up the match scores between the a concept pair.

$$\text{Similarity}(D, O) = \sum_{i=1}^{m} \text{Match}(C_i^D, C_i^O) \quad (1)$$

5. CONCLUSIONS

Web services are the latest technology adopted for the integration and interoperability of distributed systems. They are characterized by their independence from platforms and operating systems, which has implied their adoption by the various commercial and industrial organizations offering

their services through the Web, and consequently the increase in the number of services offered.

The discovery of Web services constitutes an emerging research axis. Various approaches have been proposed. These approaches have moved from keyword based research (syntactic discovery) to semantic based methods. In this work, an agent-based approach for modeling semantics web services discovery is proposed.

Our approach of discovery of web services is based on standardized and powerful languages, and technologies (MARDS model, OWL-S Ontologies, Matching algorithm) therefore it can solve problems of the web services discovery in different application domains and at all levels of complexity. As part of a case study, we chose the Travel online problem to better explain, illustrate and help to understand proposed approach.

For the prospects, we aim to add the phase of composition of different web services to obtain new functionalities.

REFERENCES:

[1] T. Berners-Lee, J. Hendler, O. Lassila (2001).The Semantic Web. In Scientific American, 284(5): 35 – 43.

[2] K. Mohebbi, S. Ibrahim, M. Khezrian, K. Munusamy, and S-G-H Tabatabaei (2010). A Comparative Evaluation of Semantic Web service Discovery Approaches. In Proceedings 12th International Conference on Information Integration and Web-based Applications & Services. Paris, France.

[3] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan and K. Sycara (2004). OWL-S: Markup for Web services. W3C Member. http://www.w3.org/Submission/OWL-S/.

[4] J. Farrell and H. Lausen (2007). Semantic annotations for WSDL and XML schema. W3C recommendation, W3C, http://www.w3.org/TR/2007/REC-sawsdl-20070828/

[5] D. Roman, U. Keller, H. Lausen, J. Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel (2005). Web Service Modeling Ontology. Applied Ontology. pp.77 - 106.

[6] F. Yu, Z. Chen, D. Xie (2011). A Method for Semantic Web Service Selection Based on QoS Ontology. JOURNAL OF COMPUTERS, 6(2).

[7] B. Bounabat (2000). Méthode d'analyse et de conception orientée objet décisionnel. Application aux langages synchrones et aux systèmes répartis. PhD Thesis, Cadi Ayyad University, Faculty of sciences, Marrakech, Morocco.

[8] A. Aaroud, S. E. Labhalla, and B. Bounabat (2005). Modelling the handover function of global system for mobile communication. The International Journal of Modelling and Simulation, 25(2).

[9] M. Berrada, B. Bounabat, and M. Harti (2007). Modeling and simulation of Multi-Agent reactif decisionnal systems using business process management concepts, International Review on Computers and Software, 2(2): 159-169.

[10] N. ADADI, M. BERRADA, M. HALIM and D. CHENOUNI (2019). Modeling and Implementation of Web Services Composition Based on MARDS. International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), 8(6): 3381 – 3388.

[11] Paolucci, M., Kawamura, T., Payne, T., Sycara, K. (2002). Semantic matching of web services capabilities. In: Proceedings of the First International Semantic Web Conference, LNCS 2342, Springer-Verlag, pp. 333 – 347.

[12] Bouzguenda Lotfi (2006) Coordination Multi-Agents pour le Workflow Inter Organisationnel Lâche. PhD Thesis, Université de Toulouse. pp. 100 - 102.

[13] P. Plebani & B.Pernici, URBE: Web Service Retrieval Based on Similarity Evaluation. IEEE Transactions on Knowledge and Data Engineering, 21(11). (2009).