Proposal for an e-learning system model based on the invocation and semantic discovery of web services

Mohamed Halim, Abdelmajid Tahiri, Yassir El Ghzizal, Nouha Adadi, Driss Chenouni
IPI Laboratory, Sidi Mohamed Ben Abdellah University, Fez, Morocoo

Article Info

Article history:

Received Dec 1, 2023
Revised Mar 13, 2024
Accepted Mar 27, 2024

Keywords:

E-learning domain ontology
Matching algorithm
Semantic descriptions
Service-oriented computing
Web ontology language
Web services the discovery

ABSTRACT

Service-oriented computing (SOC) provides a new framework for designing distributed web applications and software in a flexible, scalable, and cost-effective manner. Its use is widespread to efficiently integrate existing Web services and create high value-added applications. This model, proven in various fields such as e-commerce, also shows significant advantages in the field of e-learning. This approach highlights the discovery and use of Web services listed in specialized directories. In fact, this paper proposes a framework for exploring web services associated with education. This approach is based on the application of a matching algorithm to select the services best suited to the needs of users of the online learning system, as well as the ontology of the e-learning domain and the semantic descriptions of the web services via web ontology language for web services (OWL-S).

Corresponding Author:

Mohamed Halim
IPI Laboratory, Sidi Mohamed Ben Abdellah University
Fez, Morocoo
Email: Mohamed.halim@usmba.ac.ma

1. INTRODUCTION

Nowadays, e-learning environments are distributed and based on service-oriented architectures [1], [2]. The components and resources of these systems are implemented using web services. However, their use creates problems in discovering the services that best meet the needs and specificities of the various users of the e-learning system.

The first service discovery approaches proposed in the literature were syntactic approaches [3]-[6] based on the web services description language (WSDL) description and the universal description, discovery, and integration (UDDI) directory, the discovery is based on the syntactic match of the query keywords with the WSDL description of the web services. With the emergence of the new generation of the Web, semantic approaches have appeared. They present a semantic description of web services using a new generation of web languages, such as resource data framework (RDF) [7] and web ontology language (OWL) [8]. This semantic description can be interpreted by an application to ensure a high degree of automation.

Semantic approaches [9] adopt ontologies for matching queries with services. They employ logic, data mining, and even similarity measures to compare semantic interfaces such as OWLS [10], semantic annotations for WSDL (SAWSDL) [11] and web service modeling ontology (WSMO) [12]. It is observed that ontology [13] is a fundamental concept of the semantic web; it produces a good description of the information contained in the Web. Ontology is similar to a dictionary or a glossary but with a detailed and large structure that allows machines to process its content.

We note that semantic approaches are more complex and more reliable than syntactic techniques. We distinguish three classes of semantic approaches: logical, non-logical and hybrid approaches. Logical approaches [14], [15] exploit inferences to check the compatibility between the query and the service annotation (subsumption and consistency test). Logical approaches are simple in terms of implementation, but the algorithms used in these approaches have low recall and low precision and subsumption does not cover all semantic links. On the other hand, the high complexity of the reasoning weakens the chances of scalability. Finally, we note that the majority of current web services are not annotated with formal logic languages. Non-logical approaches [16], [17] exploit the implicit or informal semantics of services and process them with other techniques, such as data mining, graph matching, information retrieval techniques and similarity measures. Hybrid approaches [18]-[20] mix techniques from the two previous classes of approaches (logical and non-logical). Thus, they implement several filters, some of them are purely logical while others are based on information search techniques, or data mining. Hybrid approaches combine logical and non-logical methods for matching web services with the aim of overcoming certain limitations of each of these two classes through different hybrid combinations by aggregating the matching results.

For synthesize, the semantic web [21] is a vision towards the automation of web resource management. Its intersection with web services technology gives rise to semantic web services, characterized by their intelligence. Then, it is determined beforehand by the discovery of this type of advanced service, using in particular the OWL-S semantic modelling language and ontologies. Indeed, a dynamic approach to the discovery of Web services is proposed by projecting the architecture of this approach in the context of e-learning.

The objective of this work is to offer an e-learning application based on web services to take advantage and reuse existing services in the web. The issue is that the services associated with the e-learning system, cannot be discovered by the UDDI directory, because it does not allow the storage of information relating to the semantics of the teaching content; hence, they need a semantic description by the OWL-S language. This proposal consists of developing a hybrid service discovery approach adapted to the needs and specificities of users of e-learning systems. It is based on a matching algorithm that allows the filtering of the most relevant web services to respond to the user’s request as well as two types of ontologies: E-learning domain ontology which characterizes knowledge, concepts, and properties of the domain, and service ontology which gives a semantic description of web services associated with the e-learning system.

The layout of this paper is as follows. The second section presents the conception of e-learning system based on web services and the architecture of the proposed approach of discovering and invocation of web services. The third section is devoted to the implementation of proposed approach by formalization of domain ontology and services ontologies. The fourth section is devoted to conclusion and future work.

2. METHOD

2.1. Design of e-learning system

Learner autonomy is one of the major challenges of e-learning. Current computer resources make it possible to define teaching that adapts to the behaviors, results, and preferences of learners. It is therefore noted that the exploitation of adaptive e-learning platforms is necessary to avoid the risk of failure and abandonment that always remains an obstacle to the successful use of e-learning. In this work, we propose a design of an e-learning system that adapts to learner profiles. This design is divided into two parts. The first one presents a model as shown in Figure 1 of an adaptive e-learning system without the integration of web services. The second model as shown in Figure 2 includes the use of web services in the same e-learning system, it is therefore a service-oriented adaptive e-learning system.

The adaptive e-learning system model shown in Figure 1 is a unified modeling language (UML) diagram consisting of many actors such as administrator, learner, trainer, and adapter. Each of these actors performs one or different functionalities. The service-oriented adaptive e-learning system model is similar to that of the system which does not integrate the use of services; except there are some additions which are described in the Figure 2. Example: the addition of the expert actor whose role is to design the ontology of the domain, the definition of the concepts and the semantic links, the ontological description and the publication of the web services associated with the pedagogy, as well as their update. The other actors remain the same by adding some use cases such as “search for services”; “invoke a service”.

2.2. Proposed approach

The proposed approach as shown in Figure 3 offers the various actors of the e-learning platform the following two functionalities, which are the publication of web services of the e-learning system with a semantic and pedagogical description and the discovery of web services that best meet the needs and specificities of different users. Publication allows the expert to publish internal (i.e., accessible by learners and registered users) or external (i.e., usable by other e-learning platforms) web services. By publishing a service, a WSDL description is created containing the information necessary for the invocation of this service and registered in the UDDI directory, as well as a semantic description registered as an instance of the OWL-S ontology in the directory of semantic services.

The discovery and invocation of web services are based, in our approach, on two ontologies; the ontology of the e-learning domain allows to exploit the semantic coverage of the concepts of the learning domain and the ontology of the OWL-S services allows us to determine the services semantically related to the request. A filtration is then carried out using a matching algorithm on the results of the discovery according to the semantic density. This mechanism thus makes it possible to select the services that best meet the needs and specificities of users.

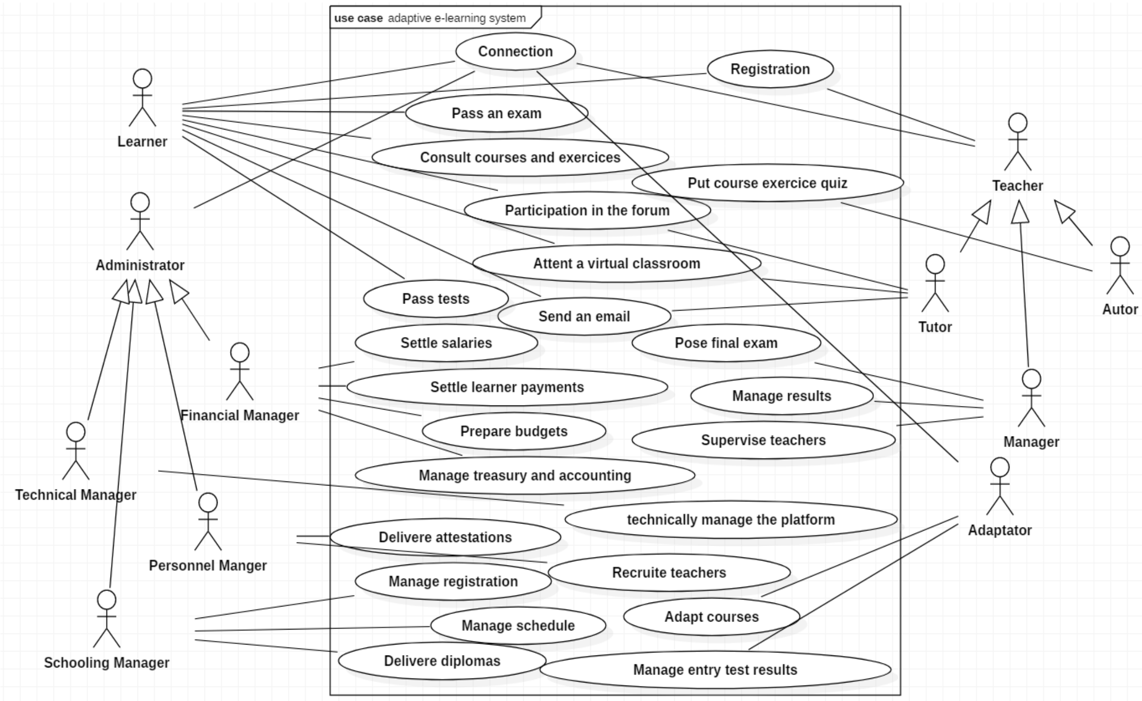

Figure 1. Adaptive e-learning system use case diagram

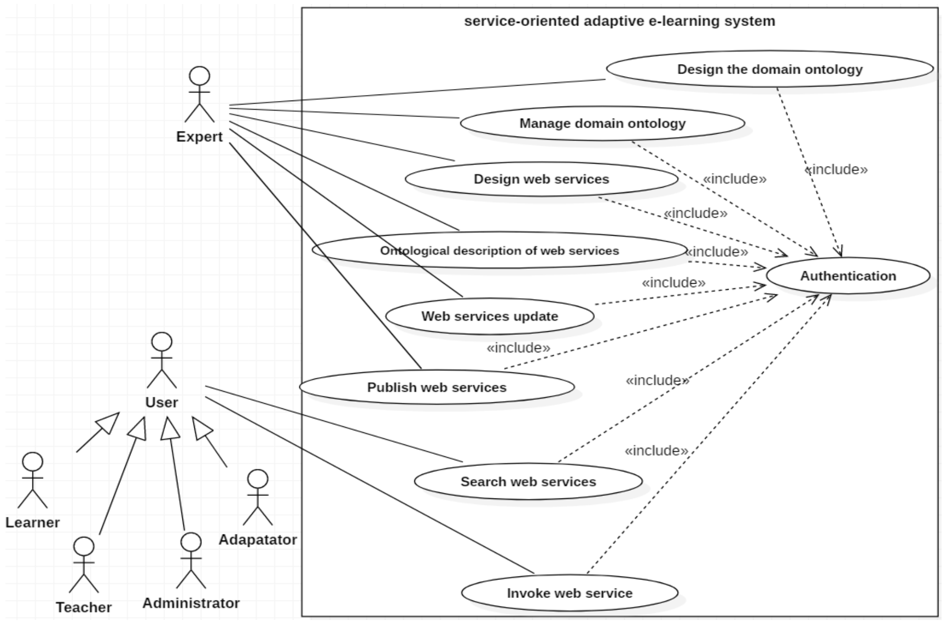

Figure 2. Service-oriented e-learning system use case diagram

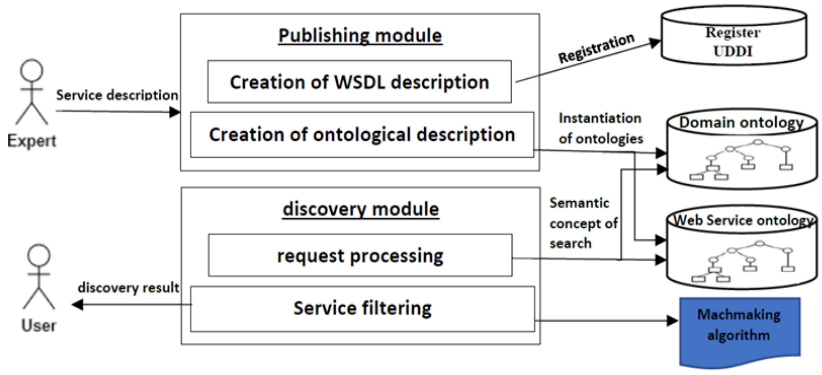

Figure 3. Architecture of the proposed system

2.3. The proposed matchmaking algorithm

In this algorithm, we model services and requests by two concept vectors Vi and Vo. Vi represents the conceptual indexing of the entries of the request R or the service S, for example if S has the entries C1, C2 and C3, then Vi= {C1, C2, C3}. Similarly, Vo represents the conceptual indexing of the outputs of the request R or the service S. for example if S has the inputs C1 C4 and C5, then Vo= {C1, C4, C5}. In follows, to differentiate between the designation of the request and the designation of a service we call the two vectors of input and output concepts of the request respectively Ri and Ro, and for the services we call them calls Si and So. The inputs of the proposed algorithm constitute a query R and a service base B formatted in OWLS, as long as the outputs are a set of retained services A.

The objective of this algorithm is to select (filter) a number n of services (set A) from S (the services of base B), even if the search space is restricted, it is still possible to construct the best plan. In order to achieve this objective and based on the functions and procedures of the matchmaking algorithm described previously, we define a function "Inclusion function (Vx, Vy)" which makes it possible to determine whether a vector of concepts Vx is included in a vector of Vy concepts. Vx and Vy are respectively the input or output concept vectors of services X and Y. The algorithm for the Inclusion (Vx, Vy) function as shown in Figure 4 is based on the following equivalence (1). For Vx to be included in Vy, it is necessary that for every concept of Vx, there exists a concept of Vy such that the similarity score of the two concepts is strictly greater than 1, which means their degree of matching is either "exact" or "plugin".

$$Vx \subset Vy \quad <=> \quad \forall\, Cx \in Vy, \exists Cy \in Vy \text{ such that } Match(Cx, Cy) > 1 \qquad (1)$$

```
Function inclusion (Vx, Vy) : boolean
% This function returns true if Vx is included in Vy and false otherwise
% Vx is the vector of service input or output concepts
% Vy is the vector of service input or output concepts Y
% Cx is the concept of inputs or outputs of the vector Vx
% Cy is the concept of inputs or outputs of the vector Vy
Variables
Integer Cpt←0 ;
Boolean val←Faux
Chain Cx, Cy;
Begin
%for Vx to be included in Vy it is necessary that for every concept of Vx, there exists a concept of Vy %such
that the similarity score of the two concepts is strictly greater than 1 which means their %degree of matching is
either " Exact" or "Plugin"
For each (Cx of Vx) do
        For each (Cy of Vy) do
                if (match(Cx, Cy)>1)) do
                Cpt++
                End if
        End for
End for
if (Cpt = Vx .Length) do
Val←true
End if
Return Val
End
```

Figure 4. Algorithm of the inclusion function

The main objective of the proposed matchmaking algorithm as shown in Figure 5 is to determine a list of compatible and composable services. The composable property means that the list of services found must be able to construct a graph whose nodes are the candidate services and the root is the query. This graph is divided into three parts:

- Community input: presents the community of services whose input parameters correspond directly or indirectly to the input parameters of the request.
- Community output: presents the community of services whose output parameters correspond directly or indirectly to the output parameters of the request.
- The result set: presents the intersection of the two parts 1 and 2 (i.e., the inputs and outputs of the services of this community correspond semantically to the inputs and outputs of the query).

```
Inputs: a request R, a service database B formatted in OWLS
Outputs: a set of retained services A
Begin
The indexing phase
1.  For (each service S∈ Base B) do
    Extract, from OWLS documents, the inputs and outputs in order to construct the Si and So vectors.
    End For
2.  Construct the query input and output concept vectors Ri and Ro

The Matching phase
R is the root node of the graph to be constructed
A= Ø // we initialize all the results empty.
1.  Building the Input Community CE
    CE= Ø // we initialize the Input Community empty.
    For (each service S∈ Base B) do
      If (Inclusion (Si, Ri)) then
      CE= CE ∪ {S}
      //we add the services whose input parameters correspond directly to the input parameters of the request.
      Child(R)=S //S is a successor node of R in the graph
      End if
    End for
    For (each service S ∈ Base B) do
        For (each service SE ∈ Base CE) do
          if (Si∩SEo≠Ø) then
          CE= CE ∪ {S}
        //we add the services whose input parameters correspond indirectly to the input parameters of the request.
          Child (SE)=S
          End if
      End for
    End for
1.  Building the Output Community CS
    CS= Ø // we initialize the Community Output empty.
    For (each service S∈ Base B) do
      If (Inclusion (So, Ro)) then
      CS= CS ∪ {S}
     //we add the services whose output parameters correspond directly to the output parameters of the request.
      Child(R)=S //S is a successor node of R in the graph
      End if
     End for
     For (each service S ∈ Base B) do
        For (each service SS ∈ Base CS) do
          If (So∩SSi≠Ø) then
          CS= CS ∪ {S}
          //we add the services whose output parameters correspond indirectly to the output parameters of the request.
          Child (SS)=S
          End if
        End for
     End for
2.  Construct result set A
    A= CE ∩CS
    // The result set A is initially constituted by the intersection of the two CE Input and CS Output Communities.
    For (each service SE ∈ CE) do
      For (each service S ∈ Base A) do
        If (Child (SE)=S) then
         A= A ∪ {SE}
         End If
      End For
     End For
    For (each service SS ∈ CS) do
      For (each service S ∈ Base A) do
        If (Child (SS)=S) then
        A= A ∪ {SS}
        End if
      End For
    End For
Return A
END
```

Figure 5. Proposed semantic service discovery algorithm

2.4. Creation of the domain ontology

The domain ontology describes the vocabulary related to a given domain. By using this ontology, it is aimed to integrate a layer of knowledge from heterogeneous sources. Designing ontologies is a complex task. Faced with such a mission, two approaches are possible: designing the ontology from scratch or adapting an existing ontology.

Albeit, it is generally accepted that formalizing a new ontology by reusing existing ontologies is "more advantageous" in terms of time, cost, and effort, it is not always the right route to take. Indeed, most existing ontologies, open sources, in particular, are difficult to handle and customize. The benefit of re-engineering an ontology, thus becomes uncertain, since the investment required to understand existing ontologies could be even greater than building an ontology from scratch.

Given this, since we are working on a case study, we have chosen to design and develop a partial e-learning ontology with a level of granularity adapted to our case study. An ontology is always linked to a construction methodology, a construction tool, and an ontology representation language. As part of this prototype, we adopt method which is recognized as the most mature and complete method. At the level of the development language, we adopt OWL as an ontological specification language.

The creation of the domain ontology is divided into two principales phases the design phase and formalization and implementation phase which is presented in the next section. The objective of the design stage is to extract knowledge from several sources and then organize and structure it using an external representation independent of the language and the implementation environment. It is therefore a question of constructing a concept diagram to specify the support classes and the subclasses.

The ontology of the e-learning domain is made up of several concepts, attributes, relations, and instances. Indeed, it can be organized in a set of sub-ontologies that revolve around structuring concepts (administration, teaching, and adaptation). The diagram as shown in Figure 6 presents an excerpt from the diagram of the e-learning domain ontology concerning the teaching sub-ontology.

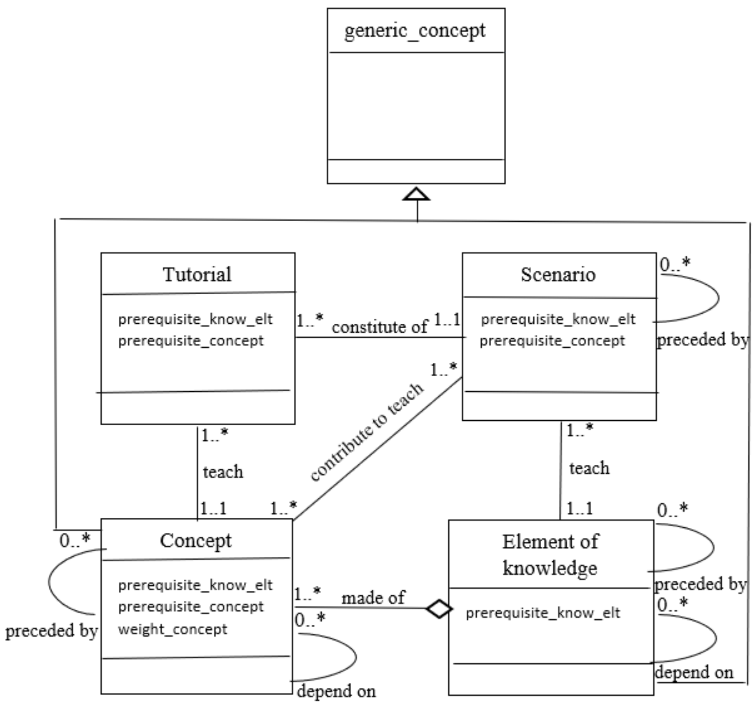

Figure 6. An excerpt from the e-learning domain ontology diagram

2.5. Creation of semantic web services ontologies

The creation of the domain ontology is divided into two principales phases the design phase and formalization and implementation phase which is presented in the next section. This implementation step consists in creating the different OWL-S descriptions of the services supposed to be discovered. Several services can be extracted from the design of the e-learning system produced above. In this manuscript we propose to present the ontology of the "pedagogical resource" service.

The ontology of the "pedagogical resource" service is defined as a sub-ontology of the "ServiceProfile" ontology. The classes of this ontology give a description of the pedagogical and semantic content of the service as well as the technical and financial characteristics deemed necessary for the description of the pedagogical resources. The main classes of this ontology are therefore linked to the pedagogical, technical and financial aspects. Figure 7 schematizes these classes with their subclasses and attributes.

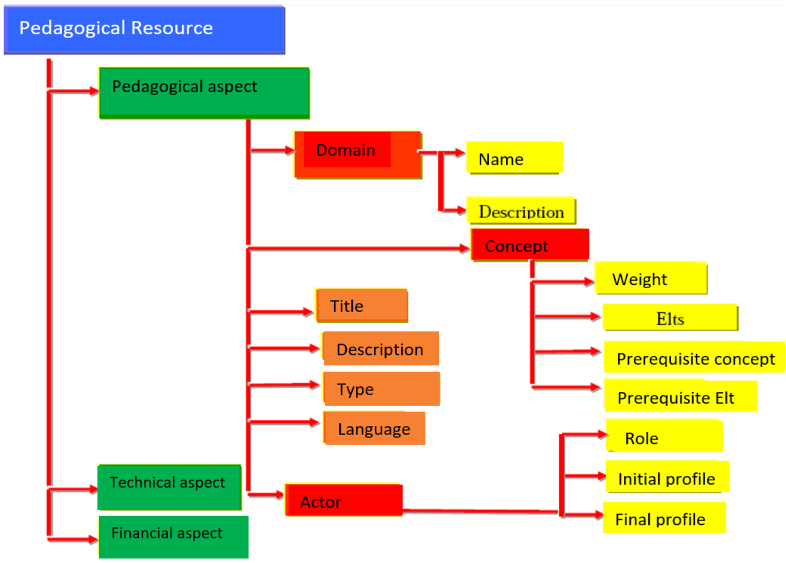

Figure 7. Description of the classes and attributes of the service ontology "pedagogical resource"

3. RESULTS AND DISCUSSION
3.1. Formalization and implementation of e-learning domain ontology

Based on the results of the conceptualization phase, the ontology was formalized with the protege tool Figure 8 by transforming the conceptual model into an implemented model. The translation into ontological specification language of the modeling carried out in the design phase is automatically generated with the protégé tool in the form of an OWL file. A much more detailed ontology would be needed for using the system with real-world settings. In fact, the purpose of this experiment is to demonstrate the technical feasibility of the approach. We then judge that the current size is sufficient.

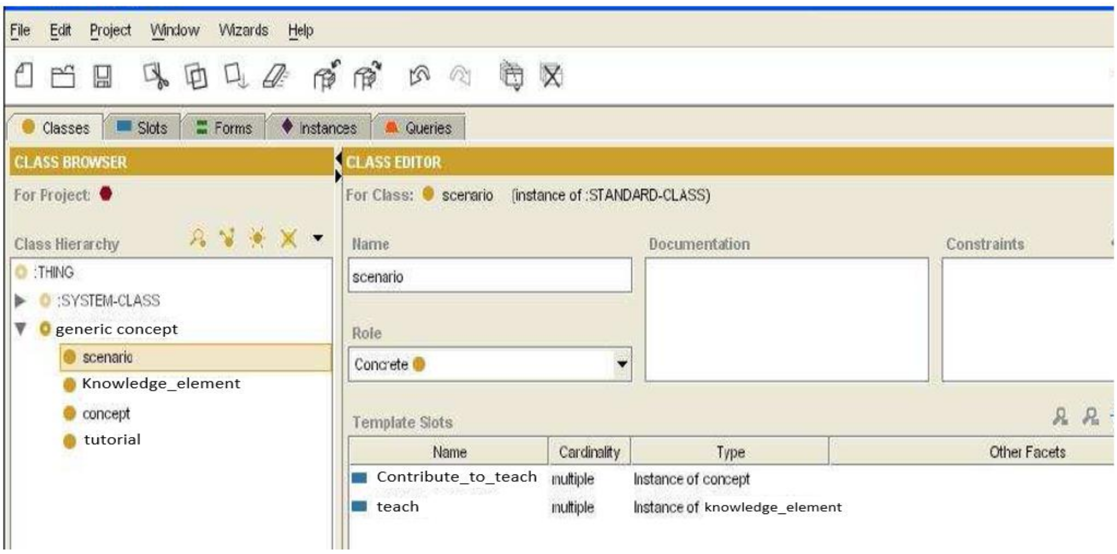

Figure 8. Formalization of domain ontology with protege

3.2. Implementation of semantic web services ontologies

This step is mainly about manipulating semantic Web services described in OWL-S language. For this need, we used the development environment "Protégé." Using this tool, the work done in this step is to create the classes and attributes of the "pedagogical resource" ontology to generate the OWL-S description of this service ontology. Figure 9 presents the implementation of this ontology. This ontology includes the sub-ontologies: service profile, process model, and service grounding. Indeed, the OWL-S description of the "pedagogical resource" service is made up of the files: PedagogicalResourceProfil.owl, PedagogicalResourceProcess.owl, and PedagogicalResourceGrounding.owl.

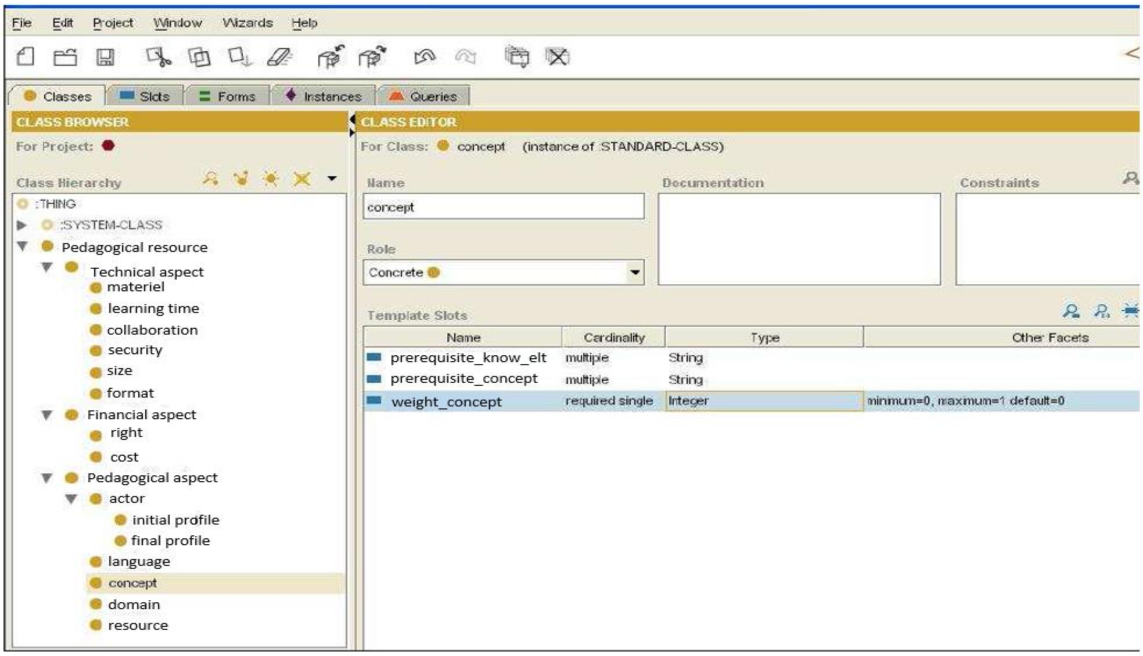

Figure 9. Implementation of the service ontology "pedagogical resource"

3.3. Discussion

Web services technology has brought several benefits to the e-learning field such as the integration of content into learning as a-service system which paves the way for easier maintenance since it is stored once on the provider server and learners get connected to the same content. Updates to this content are performed only once, and a physical distribution of the content to all the learning management system (LMSs) that have imported it is not necessary, the configuration of the platforms according to user needs, and their functionalities can be reused in different platforms. This justifies the continued growth in the number of e-learning services on the web, which signifies, the need for automated research.

In other parts, the idea pursued with web services is to better exploit Internet technologies (semantic web, domain ontology, and services ontologies) by replacing, as much as possible, humans who currently carry out a certain number of tasks. With machines in order to allow an automatic and dynamic management of services on the web. Automation and dynamics are therefore key concepts that must be taken into consideration in the process of designing and implementing a system based on service-oriented architecture such as e-learning system.

Among the objectives of our work is to develop a dynamic and automatic approach to discovery of web services, so we should compare our approach to giving more importance to the study of semantic approaches. Table 1 shows a comparison between our approach and a set of semantic service discovery approaches. Several criteria are used in this comparison, they are defined as follows:

- IO: taking into account inputs-outputs
- PE: taking into account preconditions and effects
- Non-functional properties: taking into account quality of service
- Others: taking into account properties such as the number of operations, the number of parameters, and their types.

- DoM support (degree of matching): this criterion is verified if partial matching of the service with the request is supported.
- Multi-stage matching: this criterion is verified if we aggregate several matching sub-scores (of several descriptors) into a single final score.
- Accuracy: this criterion is satisfied if we consider several matching descriptors (IOPE), several matching steps, as well as support for partial matching.
- Support of several ontologies: this criterion is satisfied if the client and the supplier use several types of ontologies.
- UDDI support: this criterion is satisfied if we support syntactic matching.
- Scalability: scalability is ensured if we can support several ontologies at the same time, as well as syntactic matching.
- Response time: this is the time required to process a request.
- Performance: it is measured with recall, and precision, the better these two metrics are, the better the performance.

The symbol √ means that the criterion is satisfied by the approach, while * means the opposite.

Table 1. Comparison of the proposed web service discovery approach with existing approaches

Category	Approaches	Criteria: Elements used: PE	Elements used: Non-functional properties	Response time	Multi-stage matching	Accuracy	Support of several ontologies	Performance	Scalability
Logic	[14]	√	*	Moyen-	*	Weak	√	Average	Average
	[15]	*	*	high	*	Weak	*		Weak
Not	[16]	*	*	Weak	√	Average	√	Weak	High
logic	URBE [17]	*	*		√	Average	*		Average
Hybrid	WSMO-MX [18]	√	*		√	High	*		Weak
	OWLS-MX2 [19]	*	*		√	Average	*		Weak
	SAWSDL-MX2 [20]	*	*		√	Average	*		Weak
	Proposed approach	*	√	Weak	√	Average	*	Average	Average

The objective of this manuscript is to create domain and service ontologies. These will be used by a matchmaking algorithm [22] which makes it possible to search a service register for services and detect the most relevant services at the functional level. These services must be composable because they will be brought to the composition stage [23]-[25].

4. CONCLUSION

In this article, we proposed the conceptual and architectural framework of an approach for the publication and discovery of web services related to the e-learning system. The approach developed is based on two ontologies: the e-learning domain ontology, and the OWL-S semantic descriptions of web services. The use of these ontologies makes it possible to improve the process of discovering web services linked to the e-learning system by selecting those that best meet the needs and specificities of the different users. Our future works consist of the implementation of our approach in an e-learning platform, as well as the implementation of principles making it possible to locate and compare services with a view to selecting services entering into the composition of another service. In fact, web services are limited to relatively simple functionality. However, for certain types of applications such as e-learning, it is necessary to combine a set of simple web services into a service meeting more complex requirements. Which poses a new problem in this subject which is the composition of Web services.

REFERENCES

[1] N. El Ghouch, E. M. En-Naimi, and M. Kouissi, "Implementation of an adaptive learning system based on agents and web services," *International Journal of Emerging Technologies in Learning*, vol. 15, no. 14, pp. 162–178, Jul. 2020, doi: 10.3991/ijet.v15i14.13837.

[2] H. A. Hamzah and M. S. A. Seman, "The role of web engineering in e-learning application development: a review study," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 27, no. 3, pp. 1576–1588, Sep. 2022, doi: 10.11591/ijeecs.v27.i3.pp1576-1588.

[3] I. Lizarralde, C. Mateos, J. M. Rodriguez, and A. Zunino, "Exploiting named entity recognition for improving syntactic-based web service discovery," *Journal of Information Science*, vol. 45, no. 3, pp. 398–415, Aug. 2019, doi: 10.1177/0165551518793321.
[4] A. Abdelli, W. Serrai, and L. Mokdad, "A novel and efficient index based web service discovery approach," *Computer Standards and Interfaces*, vol. 80, p. 103586, Mar. 2022, doi: 10.1016/j.csi.2021.103586.
[5] H. Amal, Y. Gamha, C. B. Njima, and L. B. Romdhane, "SemLinkWS: Collaboration social network of web services to aid service discovery," *Procedia Computer Science*, vol. 192, pp. 1237–1244, 2021, doi: 10.1016/j.procs.2021.08.127.
[6] B. Pourghebleh, V. Hayyolalam, and A. A. Anvigh, "Service discovery in the internet of things: review of current trends and research challenges," *Wireless Networks*, vol. 26, no. 7, pp. 5371–5391, Jun. 2020, doi: 10.1007/s11276-020-02405-0.
[7] M. A. A. Di Luca and N. G. Benítez, "Comparative study of RDF and OWL ontology languages as support for the semantic web," in *Communications in Computer and Information Science*, vol. 1193 CCIS, Springer International Publishing, 2020, pp. 3–12.
[8] Z. Faycal and T. Abdelkamel, "Building a semantic web services ontology in the pharmaceutical field using the OWL-S Language," Dec. 2021, doi: 10.1109/ICISAT54145.2021.9678408.
[9] M. Fariss, N. El Allali, H. Asaidi, and M. Bellouki, "A semantic web services discovery approach integrating multiple similarity measures and k-means clustering," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 24, no. 2, pp. 1228–1237, Nov. 2021, doi: 10.11591/ijeecs.v24.i2.pp1228-1237.
[10] D. Roman *et al.*, "Web service modeling ontology," *Applied ontology*, vol. 1, no. 1, pp. 77–106, 2005.
[11] L. Z. Varga and Á. H. Sztaki, "Semantic web services description based on web services description," *In W3C workshop on frameworks for semantics in web services*, 2005, [Online]. Available: https://www.w3.org/2005/04/FSWS/Submissions/5/SZTAKI_FSWS_050420.htm.
[12] J. Kopecký, T. Vitvar, C. Bournez, and J. Farrell, "SAWSDL: Semantic annotations for WSDL and XML schema," *IEEE Internet Computing*, vol. 11, no. 6, pp. 60–67, Nov. 2007, doi: 10.1109/MIC.2007.134.
[13] J. Li, X. Wu, C. Lu, W. Deng, and Y. Zhao, "Computing views of OWL ontologies for the semantic web," in *The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021*, Apr. 2021, pp. 2624–2635, doi: 10.1145/3442381.3449881.
[14] C. Kiefer and A. Bernstein, "The creation and evaluation of iSPARQL strategies for matchmaking," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5021 LNCS, Springer Berlin Heidelberg, 2008, pp. 463–477.
[15] M. Paolucci, T. Kawamura, T. R. Payne, and K. Sycara, "Semantic matching of web services capabilities," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2342 LNCS, Springer Berlin Heidelberg, 2002, pp. 333–347.
[16] F. Li, F. Yang, K. Shuang, and S. Su, "Q-peer: A decentralized QoS registry architecture for Web services," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4749 LNCS, Springer Berlin Heidelberg, 2007, pp. 145–156.
[17] P. Plebani and B. Pernici, "URBE: Web service retrieval based on similarity evaluation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 11, pp. 1629–1642, Nov. 2009, doi: 10.1109/TKDE.2009.35.
[18] F. Kaufer and M. Klusch, "WSMO-MX: A logic programming based hybrid service matchmaker," in *Proceedings of ECOWS 2006: Fourth European Conference on Web Services*, 2006, pp. 161–170, doi: 10.1109/ECOWS.2006.39.
[19] M. Klusch, B. Fries, and K. Sycara, "OWLS-MX: a hybrid semantic web service matchmaker for OWL-S services," *Web Semantics*, vol. 7, no. 2, pp. 121–133, Apr. 2009, doi: 10.1016/j.websem.2008.10.001.
[20] M. Klusch, P. Kapahnke, and I. Zinnikus, "SAWSDL-MX2: A machine-learning approach for integrating semantic web service matchmaking variants," in *2009 IEEE International Conference on Web Services, ICWS 2009*, Jul. 2009, pp. 335–342, doi: 10.1109/ICWS.2009.76.
[21] A. Rhayem, M. B. A. Mhiri, and F. Gargouri, "Semantic web technologies for the internet of things: systematic literature review," *Internet of Things (Netherlands)*, vol. 11, p. 100206, Sep. 2020, doi: 10.1016/j.iot.2020.100206.
[22] M. Halim, N. Adadi, M. Berrada, and D. Chenouni, "Web service discovery approach: application in e-healing domain," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 30, no. 1, pp. 557–566, Apr. 2023, doi: 10.11591/ijeecs.v30.i1.pp557-566.
[23] M. Halim, N. Adadi, M. Berrada, and D. Chenouni, "A multi-agents system based approach to service discovery," *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 19, 2022.
[24] M. Halim, N. Adadi, M. Berrada, A. Tahiri, and D. Chenouni, "Proposition of web services discovery and composition approach: application in e-learning platform," *International Journal of Emerging Technology and Advanced Engineering*, vol. 12, no. 9, pp. 49–62, Sep. 2022, doi: 10.46338/ijetae0922_06.
[25] K. D. Hanabaratti and R. Patil, "Efficient algorithm for replanning web service composition," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 31, no. 1, pp. 491–500, Jul. 2023, doi: 10.11591/ijeecs.v31.i1.pp491-500.

BIOGRAPHIES OF AUTHORS

Mohamed Halim received the Ph.D. degree in Computer Sciences from the faculty of Science and Technology, Sidi Mohammed ben Abdellah University (USMBA), Fez, Morocco, in 2023. He is currently an administrator at ENS and member of IPI Laboratory in USMBA Fez, Morocco. He can be contacted at email: mohamed.halim@usmba.ac.ma.

Abdelmajid Tahiri received the Ph.D. degree in Computer Sciences from the faculty of Science and Technology, USMBA, Fez, Morocco, in 2021. He is currently a member of IPI Laboratory and professor of ICTE and Educational Engineering at ENS. He can be contacted at email: abdelmajid.tahiri@usmba.ac.ma.

Yassir El Ghzizal received the Ph.D. degree in Computer Sciences from the faculty of Science and Technology, USMBA, Fez, Morocco, in 2022. He is currently an administrator at ENS and member of IPI Laboratory in USMBA Fez, Morocco. He can be contacted at email: yassir.elghzizal@usmba.ac.ma.

Nouha Adadi received the Ph.D. degree in Computer Sciences from the Faculty of Science and Technology, USMBA, Fez, Morocco, in 2018. She is currently a member of IPI Laboratory and a professor of Computer Sciences at ENS, Fez, Morocco. She can be contacted at email: nouha.adadi@usmba.ac.ma.

Driss Chenouni received the Ph.D. degree in physics from the University of Montpellier II, France, in 1989. He is currently the director of IPI laboratory and the director of ENS Fez, Morocco. He can be contacted at email: driss.chenouni@usmba.ac.ma.