An Implementation of OWL-S to Support Semantic Web Services Discovery

Luís Antonio Rodriguez [1] and José Maria Parente de Oliveira [1]

[1] Aeronautics Institute of Technology (ITA), Pça Mal Eduardo Gomes S/N, São José dos Campos, Brazil

Abstract

Current industry standards for describing web services focus on interoperability across different development platforms but fail to provide a consistent foundation to be able to acquire some automation, like to make intelligent searches or service compositions. Effective web services discovery should be defined as the process of matching user requests with some available web services and must focus on semantics, depending directly on the ability to make semantic specifications about web services at a machine-readable level. This paper proposes an implementation of OWL-S to support semantic discovery for Brazilian Air-Traffic Management Web Services.

Keywords

OWL, Semantic Web Services

1. Introduction

[1]An important initiative in the evolution of the World Wide Web has been the development of standardized Web Services (WS) technologies [8]. WS can primarily provide users with information but they can also make available more complex operations [9] like booking hotels or travel tickets, offering a basis for interoperability between providers and consumers using a reliable exchange of messages, which is the major reason for the spread of the use of this technology.

Current industry standards for describing web services focus on interoperability across different development platforms but fail to provide a consistent foundation to be able to acquire some automation [8], like to make intelligent searches or service compositions. Most recent works about semantic discovery are strongly coupled to some existing technology [4] like the XML-based Simple Object Access Protocol (SOAP) [9] and the Web Services Description Language (WSDL) [9] and they identify, among a set of services, the ones which get closer to satisfy requirements by using matching techniques and algorithms.

The integrated vision of Semantic Web should enable full access not only to content, but also to WS [3] since they are largely used to achieve human ordinary goals. The idea is about users and software agents being able to discover, invoke, compose and execute WS with a high degree of automation. OWL-S ("OWL for Services") is an ontology for WS which makes these actions possible describing a set of representational notations for expressing WS specifications. It is based on the Web Ontology Language (OWL) syntax [2]. It is a powerful mechanism to describe semantic WS registries [9], artifacts that could be queried semantically and these query results could present objects which would define precisely what is the existing service which is able to accomplish a set of user requirements.

The implementation of OWL-S enables the development of software programs that can interpret those semantic descriptions and then employ the WS to satisfy any user goals [8]. Some research works are focused on mixing OWL specifications with other artifacts like WSDL [7], aiming

[1]*FOMI2022: 12th International Workshop on Formal Ontologies meet Industry, September 12-15, 2022, Tarbes, France*.
Luís Antonio de Almeida Rodriguez and José Maria Parente de Oliveira
rodriguezlaar@gmail.com. parente@ita.br

to support some level of automation based on semantics to identify with a higher level of precision features about WS. It is a huge job since developers must learn and manage both technologies. Effective web services discovery should be defined as the process of matching user requests with some available web services [4] and must focus on semantics, depending directly on the ability to make semantic specifications about web services at a machine-readable level, allowing information systems to interact with those artifacts.

This paper proposes an implementation of OWL-S to support semantic discovery for a Brazilian Air-Traffic Management use case. Section 2 presents the implementation of the use case, Section 3 presents the obtained results and Contributions. Section 4 presents the conclusions and further works.

2. An Implementation of OWL-S to Support Semantic Web Services Discovery

The problem addressed in this work is the lack of semantic descriptions in the Brazilian Aeronautical WS implementations offered by aeronautical providers. Brazilian aeronautical authorities are running to create models which can provide data exchange with the United States and Europe aviation, aiming to reach the ability to integrate information systems with their standards. The authors have proposed a solution with the objective: *to create an ontology based on the OWL-S implementation, using the original ".owl" files released by the official distribution [3] which could be customized to adapt to Brazilian aeronautical laws and book of rules*. The resulting product should be a customized OWL-S ontology, the **OWL-S_BR**, adapted to Brazilian vocabulary with new entities and filled with a set of OWL instances [2] which were created aiming a use case meaning real types of Brazilian aeronautical WS.

The first task was to create a set of *Competency Questions* (CQ) [5] to define the functional requirements to be implemented by the Brazilian ontology developers following the specifications of the original OWL-S [3]. The purpose of these CQ is to define a multi criteria way to find a single service by searching by different parameters, which represent all the entities that are connected to the Class *Profile,* as illustrated in Figure 1. The answers must find precisely the requested web service which is an instance of the Class *Service*. An example of those CQ is:"What is the *Service* instance advertised by the ***Profile_Gama*** instance?".

After that a set of instances was implemented starting by the *Service.owl* [3] ontology. This ontology contains the upper level of the OWL-S and the same amount of instances was created for the *Service.owl* Classes: *Service, Service Profile, Service Model* and *Service Grounding*. These instances were connected among themselves using original OWL-S Object Properties [2]:(what the service does) *presents* (Profile), (how service works) *describedBy* (Process (Model)) and (how to access it) *supports* (Grounding). After this, the *Profile* part of the OWL-S ontology was implemented with the creation of instances which represent the whole set of classes surrounding the Class *Profile* with their specific relationship.

The Classes connected to *Profile* have different characteristics which are related to: inputs and outputs of the service, preconditions, results, product delivered, provider, process, parameter, name, text description and others. This is a way to relate an instance of one of these classes to an instance of the *Profile* Class and make a *triple*, so, if we search for a *Profile* instance by an *input*, after finding the *Profile* instance, we can find the exact *Service* instance that is connected to that *Profile* just querying again. This way is possible to navigate among the implemented instances relationships and reach the focus point of the semantic search proposed here: A *Profile* instance.

After the ontology implementation, the CQs were translated to SPARQL queries [5] and after programmed, these queries were executed [6] on the OWL-S_BR using Protegé [10] with the purpose to check if the functional requirements were accomplished. The whole set of CQs and their translations were created targeting one specific *Service* instance: *Full_FlightPlan_**Service***, and one specific *Profile* instance: *Full_FlightPlan_**Profile***. All the SPARQL queries have found the same *Profile* instance and the same *Service* instance, presenting the ability to make semantic search and find precisely the requested WS in a multi criteria way.

Figure 1 presents the final implementation of one of the set of instances of *Service*. It is possible to see a standardized set of relationships among the entities which are connected by Object

Properties or Datatype Properties (OWL, 2022). This patterns presents a multi criteria way to reach a *Profile* instance, which is connected to a *Service* instance. This way first of all SPARQL queries were executed to find the *Profile*, and, after that, another query were executed to find the respective instance of *Service* and then it was possible to conclude the semantic search with a precise result. All the CQ were translated to SPARQL and executed and the results present a set of answers which has validated the ontology engineering.

3. Results and Contributions

The authors consider the results were very satisfying in the sense of the adaptation of using the OWL-S technology in a customized extension aligned with the original semantic descriptions but aiming to support Brazilian vocabulary of terms, laws and rules. The main results obtained were:- the implementation of a new extension of OWL-S representing Air-Traffic Management domain through the creation of new OWL Classes, Properties, Individuals (instances), Taxonomies and Axioms, which were built to support ATM domain descriptions;- a new ontology to serve as a generic data representation model that can be used to share information between ATM systems;- a triple repository full of axioms and semantic descriptions made to define relationships among the WS's implemented instances, which were created to represent business rules and other features of ATM domain;- a generic approach to put in practice the semantic search and match precisely any requested WS through the SPARQL protocol using a multi criteria set of semantic search parameters.

Figure 1 presents a vision of only one of the whole set of instances of the Class *Profile*. It is possible to see the pattern of relationships supported by all the instances of *Profile*. If we have had a registry containing twenty instances of Aeronautical WS, we should have 20 (twenty) portions of the implementation like in Figure 1. The instances were implemented following the essence of the *Profile* Class which is to provide the advertisement feature of the Semantic Search process. It is possible to reach the instance of *Profile* by several different criteria which are also connected among themselves, becoming possible to find the instance following anyone of the ways it is connected.

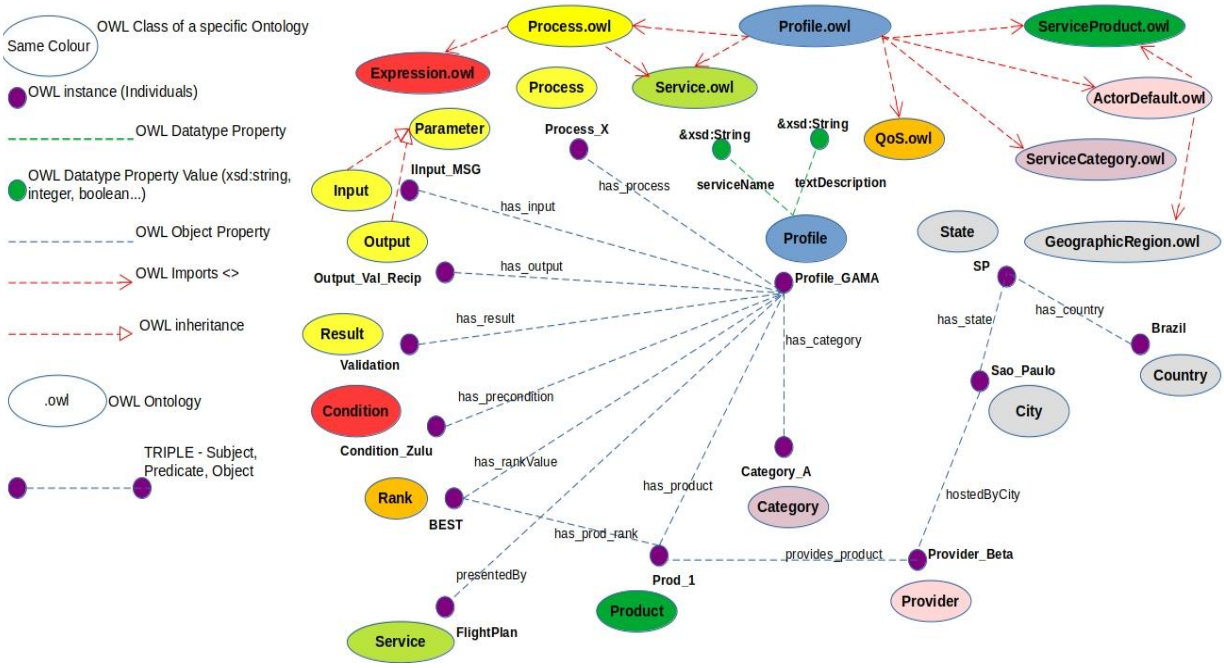

Figure 1: A single instance of Profile Class and its relationships

The main contributions are: - a new OWL-S ontology which is an extension of OWL-S original technology; - the ontology serves as a new formal vocabulary and as a data exchange model for ATM systems;- a new approach to create a triple repository using a formal ATM WS vocabulary and creating relationships that represent business rules and domain features;- a new approach to put in practice the semantic search and match precisely any requested WS from the OWL-S ontology through the SPARQL protocol, using a multicriteria set of semantic search parameters. The authors also realized that any domain can be implemented like this with a low level of customization. Besides the creation of this WS repository, it was possible to discuss and implement a brand new Brazilian ATM vocabulary which can define a mechanism to integrate Brazilian ATM systems to the USA and Europe aviation organizations.

4. Conclusions and Future works

The main conclusion of this work is that it has provided more results than the authors have planned. The implemented new OWL-S extension can serve as a formal vocabulary for ATM domain, as a reference model to make data exchange among ATM systems, as a new approach to build a triple repository of aeronautical WS and as a new way to make semantic discovery of these WS. The OWL-S technology is extensible for any kind of knowledge domain at a low level of customization, so it is possible to say it is really adaptable to any domain and scalability level, allowing different communities of developers to take advantage of these new approaches.

As future works we intend to develop a software to make use of this OWL-S extension and SPARQL queries results to serve as messages exchanged between the core of the system and the ontology to make it possible to take decisions and execute software actions to reach user goals. The OWL-S_BR extension would provide a way to reduce the code lines about business rules, normally programmed in Java, Python or any other language. The implementation is scalable in terms of number of services or amount of different metadata developers want to create to specify any specific domain.

5. Acknowledgments

We would like to thank all the institutions involved in this work for their support: Aeronautics Institute of Technology - ITA, Air Navigation Institute (ICEA), Airspace Control Department (DECEA), and the *National Council for Scientific and Technological Development* – CNPq.

6. References

[1] ICAO - System Wide Information Management (SWIM), the International Civil Aviation Organization. www.icao.int. Access 06/1st/2022.
[2] OWL - Web Ontology Language. W3C recommendation. https://www.w3.org/OWL/. Access 06/1st/2022.
[3] OWL-S - Semantic Markup for Web Services. https://www.w3.org/Submission/OWL-S/. Access 06/01/2022.
[4] Priyadharshini, G.; Gunasri R.; Saravana B. - A Survey on Semantic Web Service Discovery Methods. International Journal of Computer Applications (0975 – 8887) Volume 82 – No 11, November 2013. Tamil Nadu, India.
[5] Potoniec J.; Wi D.; Ławrynowicz A.; Keet M. - Dataset of ontology competency questions to SPARQL-OWL queries translations. Data Article. Elsevier. Data in brief 29 (2020) 105098. Contents lists available at ScienceDirect - Data in brief: journal homepage: www.elsevier.com/locate/dib.
[6] Marco L.S., David M., Claude M. - Discovering Semantic Web Services using SPARQL and Intelligent Agents - Hewlett-Packard Italy Innovation Center, Corso Trapani 16, 10139 Torino, Italy. 2012.
[7] Paolucci M., Julien Soudry J., Srinivasan N. and Sycara K. - A BROKER FOR OWL-S WEB SERVICES. The Robotics Institute of Carnegie Mellon University 5000 Forbes ave Pittsburgh, PA USA. paolucci,jsoudry,naveen,katia@cs.cmu.edu. 2004.
[8] Martin, D. et al. Bringing Semantics to Web Services with OWL-S. Conference Paper at World Wide Web (2007) 10:243–277. DOI 10.1007/s11280-007-0033-x.
[9] W3C. World Wide Web Consortium. http://www.w3.org. Access 06/1st/2022.
[10] Protegé. A free, open-source ontology editor and framework for building intelligent systems.https://protege.stanford.edu/. Access 06/1st/2022.