



How to Validate Ontologies with Themis

Alba Fernández-Izquierdo^(✉) and Raúl García-Castro

Ontology Engineering Group, Universidad Politécnica de Madrid, Madrid, Spain
{albafernandez, rgarcia}@fi.upm.es

Abstract. Validating ontologies regarding the requirements they need to satisfy is a crucial activity during ontology development in order to assure, both to domain experts and ontology developers, that the ontologies are complete regarding their needs. The aim of this work is to present Themis, a web-based tool for validating ontologies by means of test expressions, which represent the desired behaviour expected in an ontology if a requirement is satisfied. The purpose of these test expressions is to ease the formalization of the requirements into test cases and, therefore, the validation process.

Keywords: Ontology testing · Ontology requirements · Ontology development

1 Introduction

The validation of ontologies, which deals with assessing the correct conceptualization that the ontologies need to specify [6], is a crucial activity for assuring the quality of the ontologies delivered and published on the Web. One of the main concerns in the validation of ontologies is to check whether the ontology requirements that need to be covered are satisfied, with the aim of assuring both the domain experts and the ontology developers that the ontologies they are building or using are complete regarding their needs.

The main problem in validating ontologies regarding their requirements is the ambiguity of such ontological requirements, which sometimes are difficult to formalize into tests and to translate into axioms. There are some testing tools, such as TDDOnto [5] or OntologyTest [4], that execute tests on an ontology. TDD is focused on checking the presence of axioms and added them to the ontology if they are absent while OntologyTest supports several types of tests related to the instances of an ontology.

In this context, the goal within this paper is to present a testing tool to help ontology practitioners and users to validate ontologies regarding their functional requirements, i.e., the requirements which define the knowledge the ontology has to represent [7]. This tool is called Themis¹ and can be used independently of the ontology development platform without any installation. Themis supports a

¹ <http://themis.linkeddata.es/>.

collection of test expressions [2] which are extracted from a collection of lexico-syntactic patterns (LSPs) [1] collected by CORAL Corpus [3] in order to ease the requirements formalization into tests, and that represent the expected knowledge that should be added to an ontology.

The remainder of this paper is structured as follows. Section 2 presents the main features of Themis while Sect. 3 describes its architecture. Section 4 describes the demonstration that will be performed. Finally, Sect. 5 outlines some conclusions and future steps to improve this testing tool.

2 Themis Features

Themis supports and implements a set of 15 test expressions which were extracted from the existing LSPs related to ontological requirements collected by the CORAL corpus. These 15 test expressions allows to validate every type of requirement identified in such corpus. Among others, these test expressions include the validation of subsumption relations or cardinalities in the ontology. The complete test expression catalogue is available in the Themis web site, along with the requirement templates that are associated to each test expression² in order to help the translation of requirements into test cases. Each of these test expressions has an associated implementation in order to check whether the ontology satisfies a test. The implementation process of the test expressions is presented in [2]. Themis allows to execute the same test expressions on several ontologies, which can be used to check if there are ontological commitments between them.

By loading an ontology or a set of ontologies, Themis checks whether the test expressions are passed or not. Four possible results can be returned for each test expression and each ontology:

1. *Undefined terms*: The terms in the test expression are not defined in the ontology to be analysed.
2. *Passed*: The ontology includes the terms and relations defined in the test.
3. *Absent relation*: The expected relation is not modelled in the ontology.
4. *Conflict*: The addition of the axioms related to the test expression leads to a conflict with the information included in the ontology.

Figure 1 shows the Themis home page, where the users can enter the ontologies and the test expressions. Themis also allows to load test suites that are generated and published on the Web as RDF files following the Verification Test Case ontology³ by entering the URL of such file.

After executing such test expressions, Themis generates a table that includes: (1) the test executed, (2) the result of each test and (3) the problem found after the execution of the test. Figure 2 shows the Themis interface with some examples of possible results for an ontology. Since each test expression can be

² <http://themis.linkeddata.es/tests-info.html>.

³ <https://w3id.org/def/vtc#>.

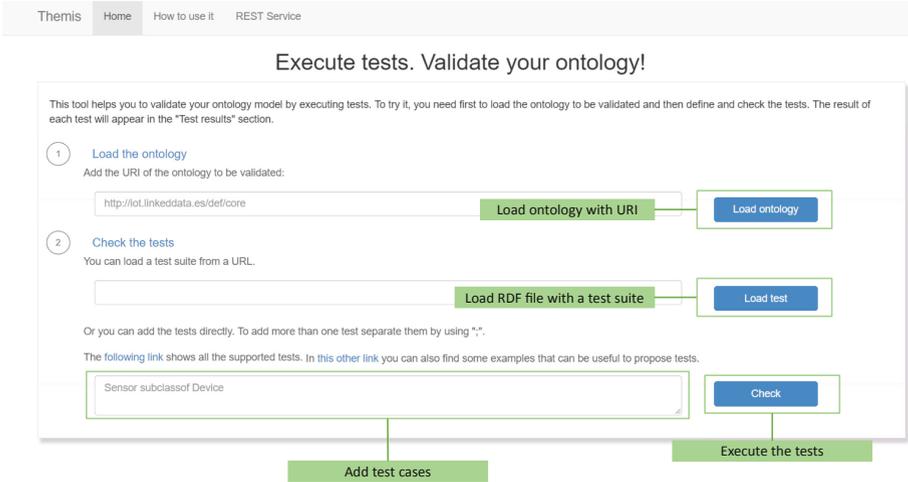


Fig. 1. Themis home page

The screenshot shows the "Tests results" section with three columns: "Executed tests", "Result of each test", and "Problem founded". The table contains the following data:

Executed tests	Result of each test	Problem founded
Device subclassOf core#Sensor	Conflict	The ontology has a relation which causes a conflict with the one define in the test
Sensor subclassOf Device	Absent relation	The ontology does not implement the requirement associated to the test
User type Class	Undefined terms	The terms in the test are not defined in the ontology
Sensor type Class	Passed	None

At the bottom right, there is a blue button labeled "Export test suite".

Fig. 2. Example of results for an ontology

executed on more than one ontology, Themis displays each test expression result for each of the analysed ontologies. The collection of executed test expressions can be exported to an RDF file to allow its reuse on the same or other ontologies.

3 Architecture

Along this section the architecture of Themis, which is summarized in Fig. 3, is presented. Themis is a web-based tool based on JAVA,⁴ HTML,⁵ CSS,⁶ and

⁴ <https://www.oracle.com/technetwork/java/javae/overview/index.html>.

⁵ <https://www.w3.org/html/wg>.

⁶ <https://www.w3.org/Style/CSS>.

jQuery⁷ technologies. The user interface consists on a view where the user enters the URIs of the ontologies to be analysed, as well as the test expressions associated to a requirement. Then, the ontologies are parsed with the OWL API⁸ and the associated implementation of the test expression is generated. To execute the test implementation on an ontology, Themis needs a glossary of terms to associate the generic terms in the test expression with a term in the ontology. For each ontology to be analysed, Themis automatically generates a glossary where the terms are the fragments of the URIs of each concept. For example, the term *http://iot.linkeddata.es/def/core#Device* will be represented as the term *Device* in the glossary. This glossary of terms provided by Themis can be modified by the users, since it is also displayed in the web user interface. Finally, the test execution results are displayed by means of the web user interface.

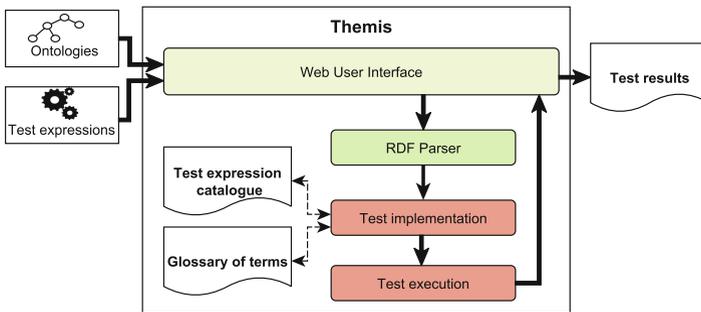


Fig. 3. Themis architecture

4 Demonstration

During the demonstration, two use cases will be analysed in order to show the participants all the features of Themis: (1) the validation of an ontology regarding its requirements and (2) the validation of two ontologies simultaneously to check whether there is an overlap of ontological commitments between them.

First, an ontology will be loaded in Themis and analysed by executing different types of test expressions. Additionally, such test expressions will be exported to an RDF file. Finally, a test suite that is publicly available on the Web in RDF will be loaded and executed on the analysed ontology.

Afterwards, another ontology will be loaded in Themis and several test expressions will be executed, which will allow the users to check whether the two loaded ontologies satisfy the same requirements.

⁷ <http://jquery.com>.

⁸ <http://owlapi.sourceforge.net>.

5 Conclusions

In this paper Themis is presented, along with its functionalities and its architecture, as a tool to validate ontologies through ontological requirements. This tool allows to: (1) execute a set of tests adding them directly in the web interface, (2) load a test suite in RDF and execute all its tests, (3) export the test suite to RDF, and (4) execute a test suite on one or several ontologies simultaneously.

Themis is currently being used by ontology engineers and it was also used in two European research projects, namely, the VICINITY⁹ and the DELTA projects¹⁰, where several ontologies are under development.

As part of the continuous process of improving Themis, if new types of requirements or LSPs are found, new tests associated to them are going to be included in the test expression catalogue supported by Themis. Furthermore, it is also planned to provide more types of results to the users, in addition to the four types that are provided so far. Finally, future work will be directed both to the addition of a feature to allow uploading a local ontology and to the integration of Themis in other ontology engineering tools, such as OnToology.¹¹

Acknowledgments. This work is partially supported by the H2020 project VICINITY: Open virtual neighbourhood network to connect intelligent buildings and smart objects (H2020-688467) and by a Predoctoral grant from the I+D+i program of the Universidad Politécnica de Madrid.

References

1. Aguado de Cea, G., Gómez-Pérez, A., Montiel-Ponsoda, E., Suárez-Figueroa, M.C.: Natural language-based approach for helping in the reuse of ontology design patterns. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 32–47. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87696-0_6
2. Fernández-Izquierdo, A., García-Castro, R.: Requirements behaviour analysis for ontology testing. In: Faron Zucker, C., Ghidini, C., Napoli, A., Toussaint, Y. (eds.) EKAW 2018. LNCS (LNAI), vol. 11313, pp. 114–130. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03667-6_8
3. Fernández-Izquierdo, A., Poveda-Villalón, M., García-Castro, R.: CORAL: a corpus of ontological requirements annotated with lexico-syntactic patterns. In: European Semantic Web Conference (2019)
4. García-Ramos, S., Otero, A., Fernández-López, M.: OntologyTest: a tool to evaluate ontologies through tests defined by the user. In: Omatu, S., et al. (eds.) IWANN 2009. LNCS, vol. 5518, pp. 91–98. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02481-8_13
5. Lawrynowicz, A., Keet, C.M.: The TDDonto tool for test-driven development of DL knowledge bases. In: Description Logics (2016)

⁹ <http://vicinity.iot.linkeddata.es>.

¹⁰ <http://delta.iot.linkeddata.es>.

¹¹ <http://ontoology.linkeddata.es>.

6. Suárez-Figueroa, M.C., Gómez-Pérez, A.: First attempt towards a standard glossary of ontology engineering terminology. In: International Conference on Terminology and Knowledge Engineering (2008)
7. Suárez-Figueroa, M.C., Gómez-Pérez, A., Villazón-Terrazas, B.: How to write and use the ontology requirements specification document. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2009. LNCS, vol. 5871, pp. 966–982. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-05151-7_16