Evaluation of RDF(S) and DAML+OIL Import/Export Services within Ontology Platforms

Asunción Gómez-Pérez and M. Carmen Suárez-Figueroa

Laboratorio de Inteligencia Artificial
Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo sn.
Boadilla del Monte, 28660. Madrid, Spain
asun@fi.upm.es
mcsuarez@delicias.dia.fi.upm.es

Abstract. Both ontology content and ontology building tools evaluations play an important role before using ontologies in Semantic Web applications. In this paper we try to assess ontology evaluation functionalities of the following ontology platforms: OilEd, OntoEdit, Protégé-2000, and WebODE. The goal of this paper is to analyze whether such ontology platforms prevent the ontologist from making knowledge representation mistakes in concept taxonomies during RDF(S) and DAML+OIL ontology import, during ontology building and during ontology export to RDF(S) and DAML+OIL. Our study reveals that most of these ontology platforms only detect a few mistakes in concept taxonomies when importing RDF(S) and DAML+OIL ontologies. It also reveals that most of these ontology platforms only detect some mistakes in concept taxonomies during building ontologies. Our study also reveals that these platforms do not detect any taxonomic mistake when exporting ontologies to such languages.

1 Introduction

Ontology content should be evaluated before using or reusing it in other ontologies or software applications. To evaluate the ontology content, and the software used to build ontologies are important processes to take into account before integrating ontologies in final applications. Ontology content evaluation should be performed during the whole ontology life-cycle. In order to carry out such evaluation, ontology development tools should support content evaluation during the whole process.

The goal of ontology evaluation is to determine what the ontology defines correctly, what it does not define or defines incorrectly. Up to now, few domain-independent methodological approaches [4, 8, 11, 13] have been reported for building ontologies. All the aforementioned approaches identify the need for ontology evaluation. However, such evaluation is performed differently in each one of them.

The main efforts on ontology content evaluation were made by Gómez-Pérez [6, 7] and by Guarino and colleagues with the OntoClean method [9].

In the last years, the number of tools for building, importing, and exporting ontologies has increased exponentially. These tools are intended to provide support

R. Monroy et al. (Eds.): MICAI 2004, LNAI 2972, pp. 109–118, 2004.

for the ontology development process and for the subsequent ontology usage. Examples of such platforms are: OilEd [2], OntoEdit [12], Protégé-2000 [10], and WebODE [3, 1].

Up to now, we do not know of any document that describes how different ontology platforms evaluate ontologies during the processes of import, building and export. In this paper we study whether the previous ontology platforms prevent the ontologist from making knowledge representation mistakes in concept taxonomies.

We have performed experiments with 24 ontologies (7 in RDF(S)[1, 2] and 17 in DAML+OIL[3]) that are well built from a syntactic point of view, but that have inconsistencies and redundancies. These knowledge representation mistakes are not detected by the current RDF(S) and DAML+OIL parsers [5]. We have imported these ontologies into the previous ontology platforms. We have also built 17 ontologies with inconsistencies and redundancies using the editors provided by the previous platforms. After that, we have exported such ontologies to RDF(S) and DAML+OIL.

This paper is organized as follows: section two describes briefly the method for evaluating taxonomic knowledge in ontologies. Section three gives an overview of the ontology platforms used. Section four exposes the results of importing, building and exporting RDF(S) and DAML+OIL ontologies with taxonomic mistakes in the ontology platforms. And, section five concludes with further work on evaluation.

2 Method for Evaluating Taxonomic Knowledge in Ontologies

Figure 1 shows a set of the possible mistakes that can be made by ontologists when modeling taxonomic knowledge in an ontology [6].

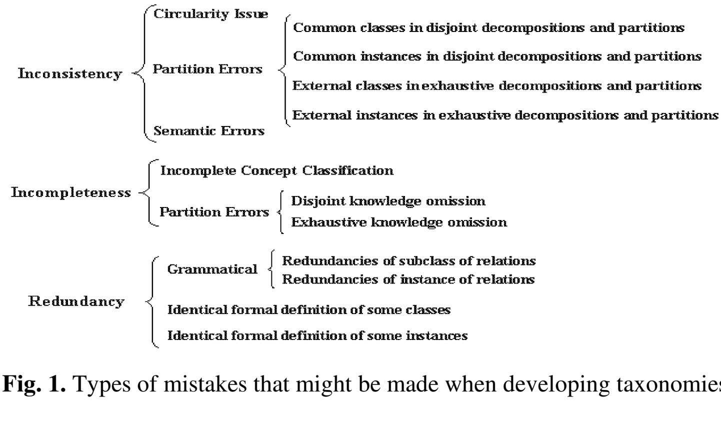


Fig. 1. Types of mistakes that might be made when developing taxonomies

In this paper we have focused only on inconsistency mistakes (circularity and partition) and grammatical redundancy mistakes, and have postponed the analysis of the others for further works.

---

[1] http://www.w3.org/TR/PR-rdf-schema
[2] http://www.w3.org/TR/REC-rdf-syntax/
[3] http://www.daml.org/2001/03/daml+oil-walkthru.html

We would like to point out that concept classifications can be defined in a disjoint (*disjoint decompositions*), a complete (*exhaustive decompositions*), and a disjoint and complete manner (*partitions*).

3 Ontology Platforms

In this section, we provide a broad overview of the tools we have used in our experiments: OilEd, OntoEdit, Protégé-2000, and WebODE.

OilEd[4] [2] was initially developed as an ontology editor for OIL ontologies, in the context of the IST OntoKnowledge project at the University of Manchester. However, OilEd has evolved and now is an editor of DAML+OIL and OWL ontologies. OilEd can import ontologies implemented in RDF(S), OIL, DAML+OIL, and in the SHIQ XML format. OilEd ontologies can be exported to DAML+OIL, RDF(S), OWL, to the SHIQ XML format, and to DIG XML format.

OntoEdit[5] [12] was developed by AIFB in Karlsruhe University and is now being commercialized by Ontoprise. It is an extensible and flexible environment and is based on a plug-in architecture, which provides functionality to browse and edit ontologies. Two versions of OntoEdit are available: Free and Professional. OntoEdit Free can import ontologies from FLogic, RDF(S, DAML+OIL, and from directory structures and Excel files. OntoEdit Free can export to OXML, FLogic, RDF(S, and DAML+OIL.

Protégé-2000[6] [10] was developed by Stanford Medical Informatics (SMI) at Stanford University, and is the latest version of the Protégé line of tools. It is an open source, standalone application with an extensible architecture. The core of this environment is the ontology editor, and it holds a library of plug-ins that add more functionality to the environment (ontology language import and export, etc.).

Protégé-2000 ontologies can be imported and exported with some of the back-ends provided in the standard release or provided as plug-ins: RDF(S, DAML+OIL, OWL, XML, XML Schema, and XMI.

WebODE[7] [3, 1] is an ontological engineering workbench developed by the Ontology Engineering Group at Universidad Politécnica de Madrid (UPM). It is an ontology engineering suite created with an extensible architecture. WebODE is not used as a standalone application but as a Web application. Three user interfaces are combined in the WebODE ontology editor: an HTML form-based editor for editing all ontology terms except axioms and rules; a graphical user interface, called OntoDesigner, for editing concept taxonomies and relations; and the WebODE Axiom Builder (WAB) [3], for creating formal axioms and rules.

There are several services for importing and exporting ontologies: XML, RDF(S), DAML+OIL, OIL, OWL, XCARIN, FLogic, Jess, Prolog, and Java.

---

[4] http://oiled.man.ac.uk
[5] http://www.ontoprise.de/com/start_downlo.htm
[6] http://protege.stanford.edu/plugins.html
[7] http://webode.dia.fi.upm.es/

4 Comparative Study of Ontology Platforms

At present, there are a great number of ontologies in RDF(S) and DAML+OIL, and most of the RDF(S) and DAML+OIL parsers are not able to detect knowledge representation taxonomic mistakes in ontologies implemented in such languages [5]. Therefore, we have decided to analyze whether ontology platforms presented in section 3 are able to detect this type of mistakes during RDF(S) and DAML+OIL ontology import, ontology building, and ontology export to RDF(S) and DAML+OIL.

The results of our analysis are shown in the tables using the following symbols:

- ✓ The ontology platform detects the mistake.
- ☑ The ontology platform allows inserting the mistake, which is only detected when the ontology is verified.
- ✗ The ontology platform does not detect the mistake.
- 😐 The ontology platform does not allow representing this type of mistake.
- -- The mistake cannot be represented in this language.
- ⊖ The ontology platform does not allow inserting the mistake.

4.1 Detecting Knowledge Representation Mistakes during Ontology Import

To carry out this experiment, we have built a testbed of 24 ontologies (7 in RDF(S) and 17 in DAML+OIL), each of which implements one of the possible problems presented in section 2. In the case of RDF(S) we have only 7 ontologies because partitions cannot be defined in this language. This testbed can be found at *http://minsky.dia.fi.upm.es/odeval*. We have imported these ontologies using the import facilities of the ontology platforms presented in section 3. The results of this experiment are shown in table 1. Figure 2 shows the code of two of the ontologies used in this study: circularity at distance 2 in RDF(S) and external instance in a partition in DAML+OIL.

```
<rdfs:Class rdf:ID="ClassA">
   <rdfs:subClassOf rdf:resource="#ClassB" />
</rdfs:Class>
<rdfs:Class rdf:ID="ClassB">
   <rdfs:subClassOf rdf:resource="#ClassC" />
</rdfs:Class>
<rdfs:Class rdf:ID="ClassC">
   <rdfs:subClassOf rdf:resource="#ClassA" />
</rdfs:Class>
```

a) Loop at distance 2 in RDF(S)

```
<daml:Class rdf:ID="ClassA" />
<daml:Class rdf:ID="ClassP1" />
<daml:Class rdf:ID="ClassP2" />
<ClassA rdf:ID="Instance_A" />
<daml:Class rdf:about="#ClassA">
  <daml:disjointUnionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#ClassP1"/>
    <daml:Class rdf:about="#ClassP2"/>
  </daml:disjointUnionOf>
</daml:Class>
```

b) External instance in partition in DAML+OIL

Fig. 2. Examples of RDF(S) and DAML+OIL ontologies

The main conclusions of the RDF(S) and DAML+OIL ontology import are:

Circularity problems at any distance are the only problems detected by most of ontology platforms analyzed in this experiment. However, OntoEdit Free does not detect circularities at distance zero, but it ignores them.

Table 1. Results of the RDF(S) and DAML+OIL ontology import

			OilEd		OntoEdit Free		Protégé-2000		WebODE	
			RDF(S)	DAML+OIL	RDF(S)	DAML+OIL	RDF(S)	DAML+OIL	RDF(S)	DAML+OIL
Inconsistency: Circularity Problems	At distance zero		✓	✓	✗	✗	✓	✓	✓	✓
	At distance one		✓	✓	✓	✓	✓	✓	✓	✓
	At distance n		✓	✓	✓	✓	✓	✓	✓	✓
Inconsistency: Partition Errors	Common classes in disjoint decompositions	Direct	--	✗	--	✗	--	✗	--	✓
		Indirect	--	✗	--	✗	--	✗	--	✓
	Common classes in partitions		--	✗	--	😐	--	✗	--	✓
	Common instances in disjoint decompositions	Direct	--	✗	--	✗	--	😐	--	😐
		Indirect	--	✗	--	✗	--	😐	--	😐
	Common instances in partitions		--	✗	--	😐	--	😐	--	😐
	External classes in exhaustive decompositions		--	✗	--	😐	--	✗	--	😐
	External classes in partitions		--	✗	--	😐	--	✗	--	✓
	External instances in exhaustive decompositions		--	✗	--	😐	--	✗	--	😐
	External instances in partitions		--	✗	--	😐	--	✗	--	✓
Redundancy: Grammatical Problems	Redundancies of subclass-of relations	Direct	✗	✗	✗	✗	✗	✗	✗	✗
		Indirect	✗	✗	✗	✗	✗	✗	✓	✓
	Redundancies of instance-of relations	Direct	✗	✗	✗	✗	✗	😐	😐	😐
		Indirect	✗	✗	✗	✗	😐	😐	😐	😐

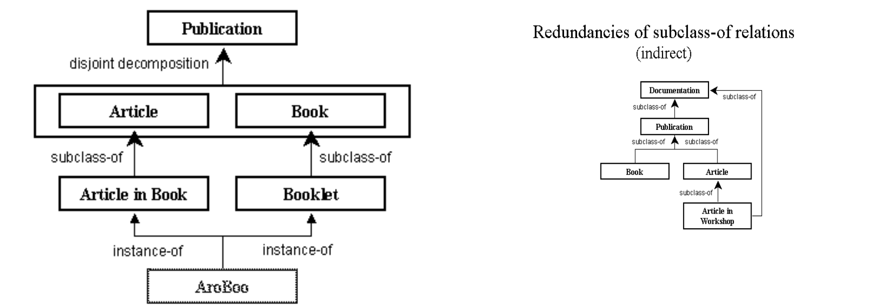
Regarding *partition errors*, we have only studied DAML+OIL ontologies because this type of knowledge cannot be represented in RDF(S). Most of ontology platforms used in this study cannot detect partition errors in DAML+OIL ontologies. Only WebODE using the ODEval[8] service detects some partition errors.

Grammatical redundancy problems are not detected by most of ontology platforms used in this work. However, some ontology platforms ignore direct redundancies of ‘subclass-of’ or ‘instance-of’ relations. As in the previous case, only WebODE using the ODEval service detects indirect redundancies of ‘subclass-of’ relations in RDF(S) and DAML+OIL ontologies.

4.2 Detecting Knowledge Representation Mistakes during Ontology Building

In this section we analyze whether the editors of the ontology platforms detect concept taxonomy mistakes. We have built 17 ontologies using such ontology platforms. Each of which implements one of the problems presented in section 2.

Figure 3 shows two of the ontologies used in this study: the first represents an indirect common instance in a disjoint decomposition and the second represents an indirect redundancy of ‘subclass-of’ relation.



a. Indirect common instance in disjoint decomposition

b. Indirect redundancy of ‘subclass-of’ relation

Fig. 3. Examples of ontologies built in the ontology editors

The results of analyzing the editors of the ontology platforms are shown in table 2. The main conclusions of this study are:

Circularity problems are the only ones detected by most of ontology platforms used in this study. However, OntoEdit Free detects neither circularity at distance one nor at distance ‘n’. Furthermore, OntoEdit Free and WebODE have mechanisms to prevent ontologists from inserting circularity at distance zero.

As for *partition errors*, WebODE detects only external classes in partitions. OilEd and Protégé-2000 detect some partition errors when the ontology is verified, but these types of mistakes can be inserted in those ontology platforms. Most of partition errors are not detected by the platforms or cannot be represented in the platforms.

[8] http://minsky.dia.fi.upm.es/odeval

Table 2. Results of the ontology building

			OilEd	OntoEdit Free	Protégé-2000	WebODE	
						HTML form-based	OntoDesigner
Inconsistency: Circularity Problems	At distance zero		✓	⊖	✓	⊖	✓
	At distance one		✓	×	✓	✓	✓
	At distance n		✓	×	✓	✓	✓
Inconsistency: Partition Errors	Common classes in disjoint decompositions	Direct	☑	×	☑	×	×
		Indirect	×	×	☑	×	×
	Common classes in partitions		☑	😐	☑	×	×
	Common instances in disjoint decompositions	Direct	×	×	😐	😐	😐
		Indirect	×	×	😐	😐	😐
	Common instances in partitions		×	😐	😐	😐	😐
	External classes in exhaustive decompositions		×	😐	☑	😐	😐
	External classes in partitions		×	😐	☑	✓	✓
	External instances in exhaustive decompositions		×	😐	×	😐	😐
	External instances in partitions		×	😐	×	×	😐
Redundancy: Grammatical Problems	Redundancies of subclass-of relations	Direct	×	⊖	✓	✓	✓
		Indirect	×	×	✓	×	×
	Redundancies of instance-of relations	Direct	×	×	😐	😐	😐
		Indirect	×	×	😐	😐	😐

Table 3. Results of the RDF(S) and DAML+OIL ontology export

			OilEd		OntoEdit Free		Protégé-2000		WebODE	
			RDF(S)	DAML+OIL	RDF(S)	DAML+OIL	RDF(S)	DAML+OIL	RDF(S)	DAML+OIL
Inconsistency: Circularity Problems	At distance zero		×	×	⊖	⊖	⊖	⊖	⊖	⊖
	At distance one		×	×	×	×	⊖	⊖	⊖	⊖
	At distance n		×	×	×	×	⊖	⊖	⊖	⊖
Inconsistency: Partition Errors	Common classes in disjoint decompositions	Direct	--	×	--	×	--	×	--	×
		Indirect	--	×	--	×	--	×	--	×
	Common classes in partitions		--	×	--	😐	--	×	--	×
	Common instances in disjoint decompositions	Direct	--	×	--	×	--	×	--	😐
		Indirect	--	×	--	×	--	×	--	😐
	Common instances in partitions		--	×	--	😐	--	×	--	😐
	External classes in exhaustive decompositions		--	×	--	😐	--	×	--	😐
	External classes in partitions		--	×	--	😐	--	×	--	×
	External instances in exhaustive decompositions		--	×	--	😐	--	×	--	😐
	External instances in partitions		--	×	--	😐	--	×	--	×
Redundancy: Grammatical Problems	Redundancies of subclass-of relations	Direct	×	×	⊖	⊖	⊖	⊖	×	×
		Indirect	×	×	×	×	×	×	×	×
	Redundancies of instance-of relations	Direct	×	×	×	×	😐	×	😐	😐
		Indirect	×	×	×	×	😐	×	😐	😐

Regarding *grammatical redundancy problems*, direct redundancies of ‘subclass-of’ relations are detected by Protégé-2000 and WebODE, but are forbidden by OntoEdit Free. Protégé-2000 also detects indirect redundancies of ‘subclass-of’ relations. Other grammatical problems are not detected or cannot be represented in the platforms.

4.3 Detecting Knowledge Representation Mistakes during Ontology Export

To analyze whether the export facilities of the ontology platforms detect concept taxonomy mistakes, we have exported to RDF(S) and DAML+OIL the 17 ontologies built in the previous experiment. After exporting these ontologies, we have analyzed 7 RDF(S) files and 17 DAML+OIL files. Since RDF(S) cannot represent partition knowledge, this type of knowledge is lost when we export to RDF(S).

The results of analyzing the RDF(S) and DAML+OIL export facilities of these ontology platforms are shown in table 3. The main conclusions of this study are:

Circularity problems are not detected by RDF(S) and DAML+OIL export facilities of ontology platforms. Furthermore, some ontology platforms do not allow inserting this type of problems, therefore the ontologies exported do not contain these mistakes.

With regard to *partition errors*, no ontology platforms detect these mistakes. Furthermore, some partition errors cannot be represented in ontology platforms.

Grammatical redundancy problems are not detected by the ontology platforms used in this study. OntoEdit Free and Protégé-2000 do not allow inserting direct redundancies of ‘subclass-of’ relations; therefore, neither RDF(S) nor DAML+OIL exported files can contain this type of mistake. Furthermore, some grammatical problems cannot be represented in the ontology platforms studied.

5 Conclusions and Further Work

In this paper we have shown that only a few taxonomic mistakes in RDF(S) and DAML+OIL ontologies are detected by ontology platforms during ontology import. We have also shown that most editors of ontology platforms detect only a few knowledge representation mistakes in concept taxonomies during ontology building. And we have also shown that current ontology platforms are not able to detect such mistakes during ontology export to RDF(S) and DAML+OIL.

Taking into account these results, we consider that it is necessary to check possible anomalies that can be made during ontology building in ontology platforms. Therefore it is important that these platforms help the ontologist build ontologies without making knowledge representation mistakes. We also consider that it is necessary to evaluate ontologies during the import and export processes.

We also consider that we need tools for giving support to the evaluation activity during the whole life-cycle of ontologies. These tools should not only evaluate concept taxonomies, but also other ontology components (relations, axioms, etc.).

Acknowledgements. This work has been supported by the Esperonto project (IST-2001-34373), the ContentWeb project (TIC-2001-2745), and a research grant from UPM ("Beca asociada a proyectos modalidad B"). We are very greatful to Óscar Corcho, Rosario Plaza and José Ángel Ramos for their revisions and intelligent comments.

References

1. Arpírez JC, Corcho O, Fernández-López M, Gómez-Pérez A (2003) *WebODE in a nutshell*. AI Magazine 24(3): 37-48. Fall 2003.
2. Bechhofer S, Horrocks I, Goble C, Stevens R (2001) *OilEd: a reason-able ontology editor for the Semantic Web*. In: Baader F, Brewka G, Eiter T (eds) Joint German/Austrian conference on Artificial Intelligence (KI'01). Vienna, Austria. (Lecture Notes in Artificial Intelligence LNAI 2174) Springer-Verlag, Berlin, Germany, pp 396–408.
3. Corcho O, Fernández-López M, Gómez-Pérez A, Vicente O (2002) *WebODE: an Integrated Workbench for Ontology Representation, Reasoning and Exchange*. In: Gómez-Pérez A, Benjamins VR (eds) 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'02). Sigüenza, Spain. (Lecture Notes in Artificial Intelligence LNAI 2473) Springer-Verlag, Berlin, Germany, pp 138–153.
4. Fernández-López M, Gómez-Pérez A, Pazos-Sierra A, Pazos-Sierra J (1999) *Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment*. IEEE Intelligent Systems & their applications 4(1) (1999) 37-46.
5. Gómez-Pérez A, Suárez-Figueroa MC (2003) *Results of Taxonomic Evaluation of RDF(S) and DAML+OIL Ontologies using RDF(S) and DAML+OIL Validation Tools and Ontology Platforms Import Services*. Evaluation of Ontology-based Tools (EON2003) 2nd International Workshop located at the 2nd International Semantic Web Conference (ISWC 2003) Sundial Resort, Sanibel Island, Florida, USA. PP: 13-26.
6. Gómez-Pérez A (1996) *A Framework to Verify Knowledge Sharing Technology*. Expert Systems with Application. Vol. 11, N. 4. PP: 519-529.
7. Gómez-Pérez A (1994) *Some Ideas and Examples to Evaluate Ontologies*. Technical Report KSL-94-65. Knowledge System Laboratory. Stanford University. Also in Proceedings of the 11th Conference on Artificial Intelligence for Applications. CAIA94.
8. Grüninger M, Fox MS (1995) *Methodology for the design and evaluation of ontologies*. In Workshop on Basic Ontological Issues in Knowledge Sharing (Montreal, 1995).
9. Guarino N, Welty C (2000) *A Formal Ontology of Properties* In R. Dieng and O. Corby (eds.), Knowledge Engineering and Knowledge Management: Methods, Models and Tools. 12th International Conference, EKAW2000, LNAI 1937. Springer Verlag: 97-112.
10. Noy NF, Fergerson RW, Musen MA (2000) *The knowledge model of Protege-2000: Combining interoperability and flexibility*. In: Dieng R, Corby O (eds) 12th International Conference in Knowledge Engineering and Knowledge Management (EKAW'00). Juan-Les-Pins, France. (Lecture Notes in Artificial Intelligence LNAI 1937) Springer-Verlag, Berlin, Germany, pp 17–32.
11. Staab S, Schnurr HP, Studer R, Sure Y (2001) *Knowledge Processes and Ontologies*, IEEE Intelligent Systems, 16(1). 2001.
12. Sure Y, Erdmann M, Angele J, Staab S, Studer R, Wenke D (2002) *OntoEdit: Collaborative Ontology Engineering for the Semantic Web*. In: Horrocks I, Hendler JA (eds) First International Semantic Web Conference (ISWC'02). Sardinia, Italy. (Lecture Notes in Computer Science LNCS 2342) Springer-Verlag, Berlin, Germany, pp 221–235.
13. Uschold M, Grüninger M (1996) *ONTOLOGIES: Principles, Methods and Applications.* Knowledge Engineering Review. Vol. 11; N. 2; June 1996.