FCA-Based Concept Detection in a RosettaNet PIP Ontology

Jamel Eddine Jridi and Guy Lapalme

DIRO, Université de Montréal, Canada,
{jridijam, lapalme}@iro.umontreal.ca

Abstract. This paper presents an FCA-based methodology for concept detection in a flat ontology. We apply this approach to an automatically generated ontology for a RosettaNet Partner Interface Process (PIP) which does not take advantage of some important OWL semantic relations like `subClassOf`. The goal of our approach is to regroup ontology classes sharing a set of properties (Data and Object Property) in order to improve the quality, readability and the inheritance richness of a flat ontology.

Keywords: Formal Concept Analysis, Ontology, RosettaNet PIP Ontology, Inheritance Richness.

1 Introduction

Ontologies are widely used in knowledge management, information integration, natural language processing, information retrieval, business-to-business (B2B), e-commerce, etc [4].

Research is now envisioning the adoption of semantic web technologies in the business domain. Lytras et al. [8] analyse the practical requirements in terms of interoperability or knowledge representation. The use of ontologies is not only for communication between different applications but also to provide reasoning support to infer, integrate information and to extract meaning.

After an ontology is constructed, it is important to assess the quality of an ontology to detect design defects and to automatically recognize parts that cause problems and might need more work. In this paper, we try to improve the inheritance richness of an OWL ontology based on the balance between depth and height of the inheritance tree which, according to Tatir et al. [13], play a role in a quality assessment. In Software Engineering, Sheldon et al. [10] claim that when the hierarchy in the inheritance tree is shallow, better will be the maintainability, readability and understanding.

In this paper, we propose a method to verify, regroup concepts sharing properties to improve the quality and readability of an automatically generated ontology. We use Formal Concept Analysis in the context of concept detection in OWL ontologies using OWL artifacts (*Class*, *DataProperty* and *ObjectProperty*).

The remainder of this paper is structured as follows. Section 2 states the problem. In Section 3, we describe the principles of the Formal Concept Analysis algorithm underlying our approach and the adaptation of its principles for the detection of concepts in a RosettaNet Ontology. Section 4 presents and discusses the validation results. A summary of the related work in ontology-based attempts of B2B standards and the use of Formal Concept Analysis in Ontological Engineering is given in Section 5. We conclude and suggest future research directions in Section 6.

2 Background and Problem Statement

In this section, we describe the problem of concepts detection and the importance of the inheritance tree to represent a domain knowledge. We start by defining important notions. Then, we detail the specific problems that are addressed by our approach.

2.1 Basic notions

Ontology in Semantic Web technology provides a shared and common vocabulary for a domain of interest and a specification of the meaning of its terms [3]. It allows users to organize information into a hierarchy of concepts, to describe relationships between them, and to make semantics machine processable, not just readable by a human.

RosettaNet B2B Standard is a consortium which provides a global forum for suppliers, customers, and competitors to do business and collaboration in an efficient and profitable manner. To manage business activities, RosettaNet formalizes Partner Interface Processes (PIP) with either Data Type Definition (DTD) format or XML Schema, and define business processes between trading partners. PIPs are organized into eight groups of core business processes called clusters, themselves further grouped into segments. Each segment includes several PIPs. In section 3, we will use the PIP3A4 as running example of our methodology. But we managed to apply the same technique on all published PIPs.

The RosettaNet architecture contains a Business Dictionary to define the properties for basic business activities and Technical Dictionaries to provide properties for products [14]. The RosettaNet Implementation Framework (RNIF) describes the packaging, routing, and transport of all PIP messages and business signals.

2.2 Problem Statement

There are few works to measure quality of an ontology using metrics. But, to our knowledge, no work has yet proposed methods to verify, maintain concept

hierarchy representation and improve the quality and readability of an ontology from inheritance point of view using OWL artifacts (*Class*, *DataProperty* and *ObjectProperty*).

Tatir et al. [13] and Sicilia et al. [11] propose an *Inheritance Richness* metric defined as the average number of subclasses per class and represent the distribution of information across different levels of the ontology inheritance tree. Values close to zero indicate horizontal ontologies representing perhaps more general knowledge while large values represent vertical ontologies describing detailed knowledge of a domain.

In our case study, we have chosen an ontology describing a detailed domain knowledge. Some approaches are proposed to ontologize some of the famous B2B standards like RosettaNet[1] and ebXML[2].

In this paper, we apply our methodology on a RosettaNet Ontology described in our paper published in the 15th International Conference on Enterprise Information Systems [5]. RosettaNet focuses on a supply chain domain and defines processes and business documents in detail. This automatically generated ontology, described in Section 4, has some drawbacks because it does not take advantage of some important OWL semantic relations like `subClassOf`. It is quite flat and does not describe in details the supply chain knowledge provided by RosettaNet, although it deals with a specific domain having several concepts with a common semantics. Although approaches have been previously proposed to ontologize a few B2B standards like RosettaNet and ebXML [2,6,7], we argue in [5] that our methodology is the first to deal with the complete set of RosettaNet PIPs.

During the transformation process, it is difficult to automatically detect elements with a common semantics which are described in separate OWL files. But, we noticed that many of those share some properties and lexemes in their compounded names. For this reason, we use FCA as a classification approach to detect concepts by regrouping classes in an ontology to improve its quality and readability.

3 FCA-Based Concept Detection Approach

The goal of our approach is to regroup ontology classes sharing a set of properties (Data and Object Property) and then maintain the hierarchy representations.

The core of our system has three main parts: Ontology Processing, FCA System and Regrouping concepts. The Ontology Processing step builds a cross-table from ontology artifacts (Class, Data Property and Object Property) without dealing with the property type or occurrence restrictions. This table describes relationships between objects (ontology classes are represented by rows) and attributes (Data and Object properties correspond to columns).

Taking the example in Table 1, $o_{1..k}$ represents ontology classes with k is the number of classes in the ontology. $p_{1..m}$ represents the set of Data and Object

[1] http://www.rosettanet.org/
[2] http://www.ebxml.org/

Properties with m being the total number of data and object properties. Element $<i,j>$ of the table is marked with "x" if the domain of property p_j is class o_i. We use FCA to create a hierarchy of concepts displayed as an inheritance tree. According to [10], maintainability, readability and understanding of a hierarchy are better when it is shallow. For this reason, we consider only two levels in the hierarchy of the concept lattice.

Table 1. Example of cross-table.

R	p_1	p_2	p_3	....	p_m
o_1	x	x	x		x
o_2		x	x		x
...					
o_k	x		x		x

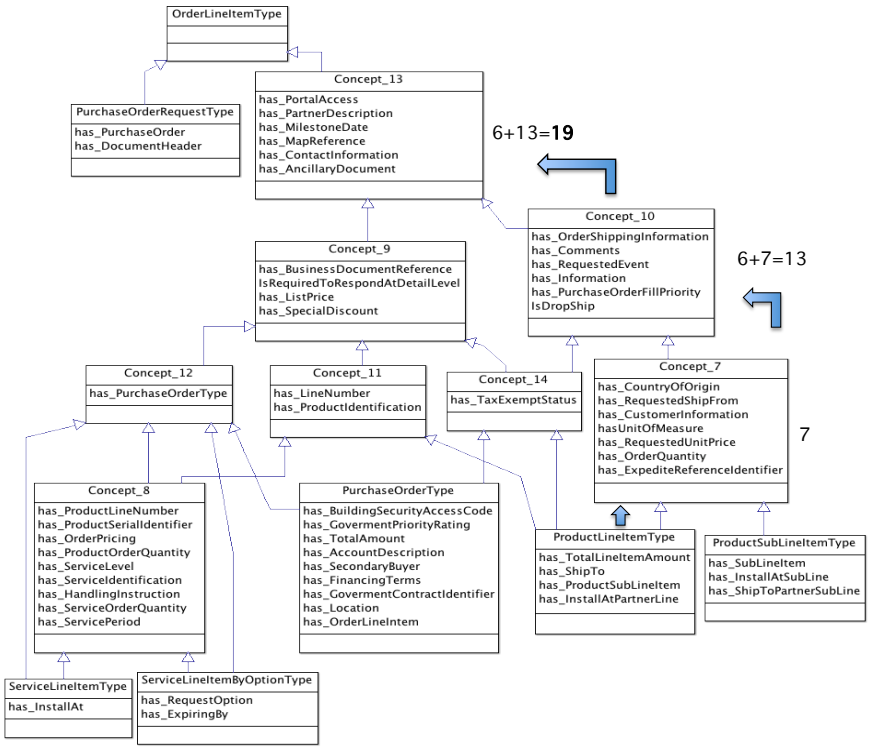

Fig. 1. An example of a `concept lattice` describing `RosettaNet PIP3A4 Purchase Order Request`. The process to build this `concept lattice` starts by extracting ontology artifacts (Class, Data Property and Object Property) from the OWL File of PIP3A4 which are added to a cross-table as CSV file. This file serves as input to the FCA system to build this concept lattice.

The `concept lattice` in Figure 1, which is the output of our FCA application, describes the RosettaNet PIP3A4 (Purchase Order Request). Intermediate concepts (of the form Concept_NN) were generated by the algorithm and represent the shared set of properties. The others represent the original ontology classes.

Each class will be associated with the concept having the largest number of shared properties. As shown in Figure 1, we note that the `Concept_7` combines the two classes `ProductSubLineItemType` and `ProductLineItemType` with 19 shared properties; `Concept_8` joins `ServiceLineItemByOptionType` and `ServiceLineItemType` because of 22 shared properties.

The combined classes share some semantics consistent with the concept definitions provided by RosettaNet. They represent one concept (`Concept_7` and `Concept_8`) that has been generated by our approach. As the regrouped classes share tokens in their compound names, we will use these to rename the FCA generated names. So, `Concept_7` becomes `ProductLineItem` and `Concept_8`, `ServiceLineItem`.

Using the generated group of classes, we update the original ontology. To do this, we used the OWL API[3] for manipulating, developing and maintaining the ontology.

4 Experimentation

The goal of our study is to evaluate the efficiency of our approach for concept detection in an OWL ontology. In this section, we describe our experimental setup and results.

In order to test our methodology, we use the RosettaNet Ontology, proposed in our `ICEIS 2013` paper [5]. It is the result of mapping the full set of RosettaNet Partner Interface Process (PIP) descriptions, currently defined with DTD or XML Schemas format, to an ontological representation using an OWL/XML rendering. Among the 132 PIPs, 112 PIPs are available for download from the PIP Directory in RosettaNet website from which we generated 138 OWL documents, valid according to the XML Schema for OWL/XML serialization from syntactical point of view. These OWL documents were also checked for consistency with an OWL reasonner.

In Table 3, we evaluate our RosettaNet Ontology using state of the art ontology metrics defined in Table 2. Only basic metrics related to the main elements of ontologies have been used [11]. Although, we use two metrics that indicate the relationship and inheritance richness of an ontology schema. The **Relationship Richness metric** (rr) reflects the diversity and placement of relations in the ontology [13]. It is defined as the ratio of the number of properties (nop) divided by the sum of the number of subclasses ($nosc$) plus the number of properties. Also, the **Inheritance Richness metric** (ir) represents how knowledge is grouped across different levels of the ontology inheritance tree [13]. It is the average number of subclasses per class.

[3] http://owlapi.sourceforge.net/

Table 2. Ontology metrics.

Metrics		Definition
Number of	classes (*noc*)	number of classes (|C|) in the ontology.
	data properties (*nodp*)	number of data properties.
	object properties (*noop*)	number of object properties.
	properties (*nop*)	sum of *nodp* and *noop* metrics.
	subclasses (*nosc*)	number of subclasses (|sC|) in the ontology.
	root classes (*norc*)	number of root classes (without superclasses). The range of this metric is between 1 and |C|.
	leaf classes (*nolc*)	classes without subclasses. The range of this metric is between 1 and |C|.

Table 3. Empirical analysis of our RosettaNet Ontology using some Ontology Metrics: Before and After applying our FCA-based Concept Detection Approach.

Metrics	All PIPs (Before)	All PIPs (After)
noc	1252	1252
nodp	3045	3045
noop	2607	2607
nop	5652	5652
nosc	0	384
norc	1252	1050
nolc	1252	868
rr	1	0.93
ir	0	0.31

We notice in Table 3 that the values of *ir* and *nosc* metrics are zero and the values of *norc* and *nolc* are equal to the number of classes |C| in the ontology. These metric values indicate that our original RosettaNet PIP Ontology as generated from the DTD and XML Schemas is a flat or horizontal ontology representing a general knowledge despite the fact that RosettaNet represents a specific domain of interest with many elements sharing a common semantics.

The application of our FCA-based Concept Detection approach extracts 182 groups of concepts comprising 384 classes from the 1252 in the ontology, bringing the inheritance metric *ir* from `0` to `0.31`. On average, each concept contains 2 classes.

We extracted 90 groups of concepts because several concepts are shared between PIP files e.g. the concept, combining `RegionalBusinessTaxIdentifier` and `NationalBusinessTaxIdentifier` classes, is detected in 3 PIPs (`PIP3A5`, `PIP3A11` and `PIP3B6`).

We also performed a manual validation of all detected concepts. We noticed that among the 90 concepts detected, 79 concepts have effectively a common semantics. So, we have a detection precision of 87%. From Table 3, we can see that the value of *nosc* and *ir* metrics increase, *norc* and *nolc* decrease after applying our FCA-based approach.

5 Related Work

Ontologies and Formal Concept Analysis (FCA) aim at modeling concepts [1]. For this reason, we use FCA to regroup concepts in a flat ontology representing general knowledge to improve its inheritance richness. To our knowledge, no previous work has addressed the problem of flat ontology using FCA techniques.

Some FCA-based proposals in Ontology Engineering differ from our own strategy with respect to the nature of the problem. Cimiano et al. [1] proposed a benchmark to discuss how FCA can be used to support Ontology Engineering and how ontology can be exploited in FCA applications. The FCA can support the building of the ontology and the constructed ontology can be analyzed and navigated using FCA techniques [1].

Stumme [12] presents an Ontology Merging approach based on FCA, named `FCA-MERGE`, for organizing business knowledge. It takes as input one or more source ontologies and returns a merged ontology between the given source ontologies.

Obitko et al. [9] propose an approach to improve ontology design by discovering the need for new objects (or classes) and relations (properties). They argue for the necessity of a better description of concepts and relations than just ordering them in taxonomy. In our case, we consider instead regrouping concepts from existing ontology classes for improving the taxonomy representation in a flat ontology.

6 Conclusion

We have presented an approach for concepts detection in a flat ontology using a Formal Concept Analysis and applied this methodology on a RosettaNet PIP Ontology which was automatically generated ontology.

Our goal is to improve the readability, maintainability and inheritance richness of an ontology. We used FCA to detect groups of concepts sharing properties and having a common semantics. Through the use of ontology metrics, we have shown that our FCA-based concept detection methodology can improve inheritance richness.

As the results using RosettaNet are promising, we suggest as future work to test the efficiency of our concept detection approach to other ontologies dealing with other domains.

Acknowledgements

We would like to thank RosettaNet Group for allowing us to download the RosettaNet Partner Interface Processes (PIPs) from the RosettaNet website. This work has been partially funded by Tunisian Government and NSERC.

References

1. Cimiano, P., Hotho, A., Stumme, G., Tane, J.: Conceptual knowledge processing with formal concept analysis and ontologies. In: Concept Lattices, pp. 189–207. Springer (2004)
2. Dogac, A., Kabak, Y., Laleci, G.B.: Enriching ebXML registries with OWL ontologies for efficient service discovery. Proceedings of the 14th International Workshop on Research Issues on Data Engineering: Web Services for E-Commerce and E-Government Applications (RIDE) (2004)
3. Euzenat, J., Shvaiko, P.: Ontology Matching, vol. 18. Springer Heidelberg (2007)
4. Gómez-Pérez, A., Corcho, O., Fernandez-Lopez, M.: Ontological Engineering. Springer-Verlag, London, Berlin (2002)
5. Jridi, J.E., Lapalme, G.: Adapting RosettaNet B2B standard to Semantic Web Technologies. vol. 2, pp. 484–491. 15th International Conference on Enterprise Information Systems, Angers, France (July 2013)
6. Kotinurmi, P., Haller, A., Oren, E.: Ontologically Enhanced RosettaNet B2B Integration. Semantic Web for Business: Cases and Applications (2008)
7. Kotinurmi, P., Haller, A., Oren, E.: Global Business: Concepts, Methodologies, Tools and Application, vol. 4, chap. Ontologically enhanced RosettaNet B2B Integration, p. 27. USA (2011)
8. Lytras, M., García, R.: Semantic Web applications: a framework for industry and business exploitation–what is needed for the adoption of the semantic web from the market and industry. International Journal of Knowledge and Learning 4(1), 93–108 (2008)
9. Obitko, M., Snasel, V., Smid, J., Snasel, V.: Ontology design with formal concept analysis. Concept Lattices and their Applications, Ostrava: Czech Republic pp. 111–119 (2004)
10. Sheldon, F.T., Jerath, K., Chung, H.: Metrics for maintainability of class inheritance hierarchies. Journal of Software Maintenance and Evolution: Research and Practice 14(3), 147–160 (2002)
11. Sicilia, M., Rodríguez, D., García-Barriocanal, E., Sanchez-Alonso, S.: Empirical findings on ontology metrics. Expert Systems with Applications 39(8), 6706–6711 (2012)
12. Stumme, G.: Using ontologies and formal concept analysis for organizing business knowledge. In: In Proc. Referenzmodellierung 2001. Citeseer (2001)
13. Tartir, S., Arpinar, I.B., Moore, M., Sheth, A.P., Aleman-Meza, B.: OntoQA: Metric-based ontology quality analysis. In: IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources. vol. 9 (2005)
14. Wang, J., Song, Y.: Architectures supporting RosettaNet. In: Software Engineering Research, Management and Applications, 2006. Fourth International Conference on. pp. 31–39. IEEE (2006)