Appendix A - ePrOnto: OWL-Based Ontology for ePrints Information Management

Joanna Isabelle Olszewska
School of Computing and Engineering, University of Huddersfield,
Queensgate, Huddersfield, HD1 3DH, United Kingdom
{j.olszewska@hud.ac.uk}

1. Introduction

Interoperability is the challenge of getting processes to share and exchange information effectively. Service orientation relates to creating self-contained, self-describing, accessible, and open, computer services. Both these challenges relate to the representation of the data being exchanged/manipulated. There are various existing sources of research information in the University, for example, ePrints – the publications repository. Research information is complex, structured data, and the future requirements of it are only partially known. If we commit to one encoding, or even one representation language, later it may turn out to be inadequate or obsolete. Current work [10] on these issues points to representing the data in an ontology.

More specifically, an ontology is a notion defined by Gruber as an explicit specification of a conceptualization [8]. The term (from the Greek, *ontos*: of being and *logia*: study) is borrowed from Philosophy and it refers to the subject of existence. In Artificial Intelligence (AI), an ontology is constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary [7]. Thus, the ontology describes a formal specification of a certain domain: a shared understanding of a domain of interest as well as a formal and machine understandable model of this domain.

In the e-business context [6], a mechanism to improve system usability, maintenance, efficiency and interoperability could reside in the formal description of the semantic of the document-based framework for business collaborations. The formal descriptions could be provided through the definition of an ontology that represents the implicit concepts and the relationships that underlie the business vocabulary.

2. Aims and Objectives

The aims of this work are:

- to examine appropriate knowledge representation schemes within the context of ontologies, and an available tool support, for research information, for example, ePrints;
- to ensure that the representation is consistent with the process models developed in WP2;
- to develop representations of knowledge for research information in an ontological form, as identified above.

3. Methodology

The creation of an ontology requires specialized skills and involves various stakeholders. The ontology development process depends on a variety of factors like the choice of the software tool used to build and edit the ontology, the language in which the ontology is implemented, the methodology which will be followed to develop it, the applications in which it will be used, the type of the ontology under construction, the available formal and informal existing knowledge resources, such as lexicons, existing ontologies, etc, and may include a large number of necessary activities.

There is no an established and unique procedure to develop ontologies despite several methodologies have been proposed over time [4], [5]. However, four general tasks to build ontologies have been identified:

- selection (includes selection of the available resources – related literature, existing ontologies, group of expert to the domain under description, selection of the appropriate tool and language);
- analysis (includes analysis of selected resources, of the classes and the properties of the selected ontology);
- definition (includes definition of what is important for the description of a specific domain through the competency questions, definition of the purpose and the domain of the ontology, the definition of the classes, the class hierarchy, the properties and the instances of the ontology);
- evaluation (includes evaluation of the selected resources, evaluation of the technical quality of the ontology and evaluation of the overall quality of the obtained results).

These tasks are distributed into different phases:

- the specification phase (answers why the ontology is being built, what its intended uses are, who the end-users are);
- the conceptualization phase (conceptualizes the domain knowledge);
- the implementation phase (transforms the conceptual model into a formal computable model);
- the evaluation phase (assesses the resulting ontology).

These phases correspond roughly to the main steps of software engineering methodologies like the *IEEE Standard 1074-1995 for Developing Software Life Cycle Processes* [5].

Moreover, the ontology development methodologies could be classified into two categories:

- methodologies focused on building a single ontology for a specific ontology for a specific domain of interest;
- methodologies focused on the construction of ontology networks.

The single ontologies could be further distinguished among those aiming at building ontologies:

- from scratch;
- by reusing pre-existing ontologies;
- by using non-ontological resources.

These single ontologies are also divided in collaborative and non-collaborative, according to the degree of participation of the involved ontology engineers, users, knowledge engineers and domain experts in the ontology engineering process.

They are also described as application dependent, semi-application dependent and application independent, according to the degree of dependency of the developed ontology on its final application.

The single-ontology capture approach could vary according to the adopted strategy for identifying concepts, and could be bottom-up (from the most concrete to the most abstract), top-down (from the most abstract to the most concrete), or middle-out (from the most relevant to the most abstract and most concrete).

We can further distinguish manual, semi-automatic and automatic ontology construction, according to the degree of human involvement in the building process.

However, the above mentioned criteria are not standards. Furthermore, none of the methodologies proposed in the literature [5] are fully mature and they need then to be adapted to the project needs. Hence, we have followed our original methodology based on the criteria previously enumerated, while developing the new ontology called ePrOnto (from the contraction of **ePr**ints and **Onto**logy), and we have carried out the actions as described below.

4. Implementation

As noted above, we decided to investigate the application of ontology using the publications repository. Capturing this information within a standard ontology language would make it universally accessible throughout the web, and allow it to be analysed, queried and compared using powerful, open tools.

We interviewed the Head of Computing Library Services of the University of Huddersfield to understand the mechanism of ePrints publication repository in order to capture ePrints knowledge to build system interoperability services. We thus identified the actors interacting with the ePrints system as well as the procedures by which the actors interact with this system. The resulting UseCase diagram (Fig. 1) uses the standard Unified Modeling Language (UML) [3] and shows the functionality of the system as well as its dependencies at a high level viewpoint.

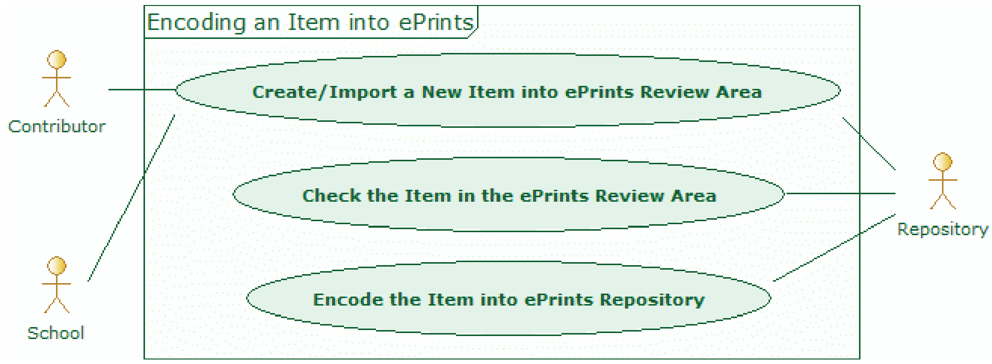

Figure 1: UseCase Diagram describing the action of encoding an Item into ePrints

Next, we have modelled the business process using Buisness Process Modeling Language (BPML) [1] and generated the flowchart shown in Fig. 2. The UseCase diagram as well as the Buisness Process Modeling Notation (BPMN) has been designed with Modelio Free Edition v1.2 [1], mainly because this software supports UML and business modelling while being a user-friendly and free tool. These steps have helped us in answering to the competency questions to determine the domain and the scope (“what we do with it”) of the ontology.

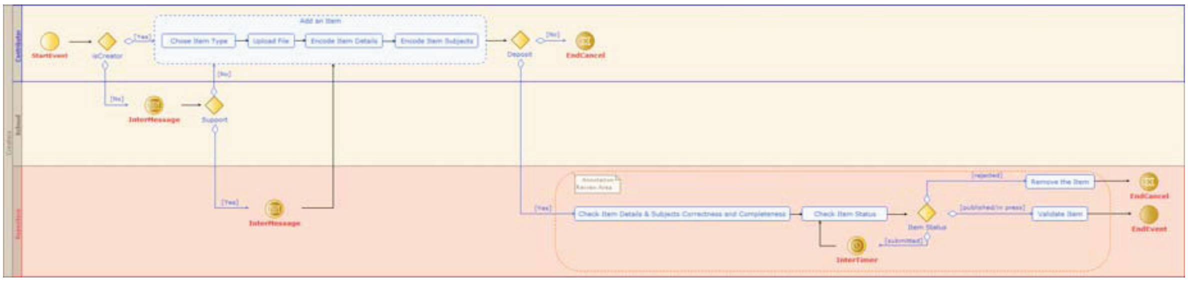

Figure 2: BPMN Diagram describing the action of adding/removing an Item in ePrints

For a Pilot implementation we developed ePrOnto, the covered domain is ePrints and its scope is to enable interoperability/sharing knowledge, efficient maintenance, and also question queries. Moreover, ePrOnto is application independent, that means that it is the same ontology e.g. for maintenance as well as for the query purpose.

Then, in order to build the domain ontology, we have selected and analyzed the ePrints vocabulary. The key related words have been identified when logging into ePrints system and doing the task of adding an Item to the repository.

To capture the ontology, we have chosen Protégé v4.1 [2], running on a Windows platform. This choice is motivated by the fact that this tool [9] facilities the interoperability with other knowledge-representation systems and has a user-friendly, configurable interface.

The adopted language to express the ontology is the Web Ontology Language (OWL) [12], according to the World Wide Web Consortium (W3C) recommendation. In particular, we have adopted OWL-DL specie because, on one hand, it is more expressive than OWL-Lite – an OWL sub-language only adapted for simple situations. On the other hand, OWL-DL is based on Description Logics (DL). Thus, it is possible to perform automated reasoning on OWL-DL-based ontology like in [11]. That is not the case for OWL-Full-based one, where OWL-Full is union of OWL syntax and Resource Description Framework (RDF)'s data representation. Moreover, OWL-DL enables reasoner use to compute the inferred ontology class hierarchy and to perform the consistency check.

To develop ePrOnto, we have identified specific basic concepts used as cornerstones for the ontology design. Next, we have mapped these concepts to a set of OWL main classes, which represent the roots of a set of corresponding subclasses together with their relationships with other classes. Each of these identified concepts represents also a starting point for the browsing of the ontology.

Protégé	**OWL**
Instances	Individuals
Slots	Properties
Classes	Classes

Table I: Definition equivalence between Protégé and OWL

A Protégé ontology consists of classes, slots, facets, and axioms as mentioned in Table I. Classes are concepts in the domain of discourse and constitute a taxonomic hierarchy. Slots describe properties or attributes of classes and instances. Facets describe properties of slots. Axioms specify additional constraints. A Protégé knowledge base includes the ontology and individual instances of classes with specific values for slots.

OWL ontology has similar components to Protégé-based one. However, the terminology used to describe these components is slightly different (see Table I).

OWL classes are interpreted as sets that contain individuals. An example of ePrOnto class is shown in Fig. 3. The class is called *Contributor* and contains four subclasses (*Contribution*; *Contributor_Email*; *Contributor_Family_Name*; *Contributor_Given_Name/_Initials*).

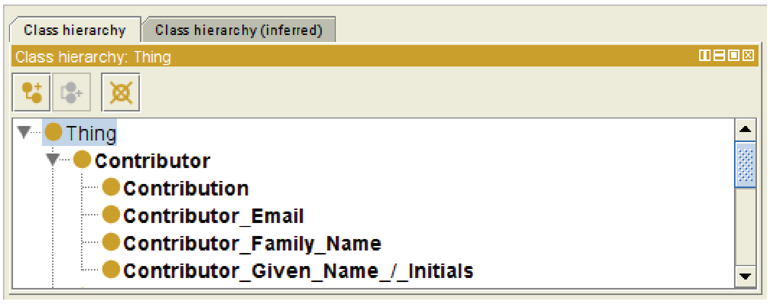

Figure 3: Example of Class (*Contributor*) implemented into ePrOnto ontology

Individuals represent objects in the domain that we are interested in. Instances can be referred to as being "instances of classes". In our example, the subclass *Contribution* has individuals as partially shown in Fig. 4.

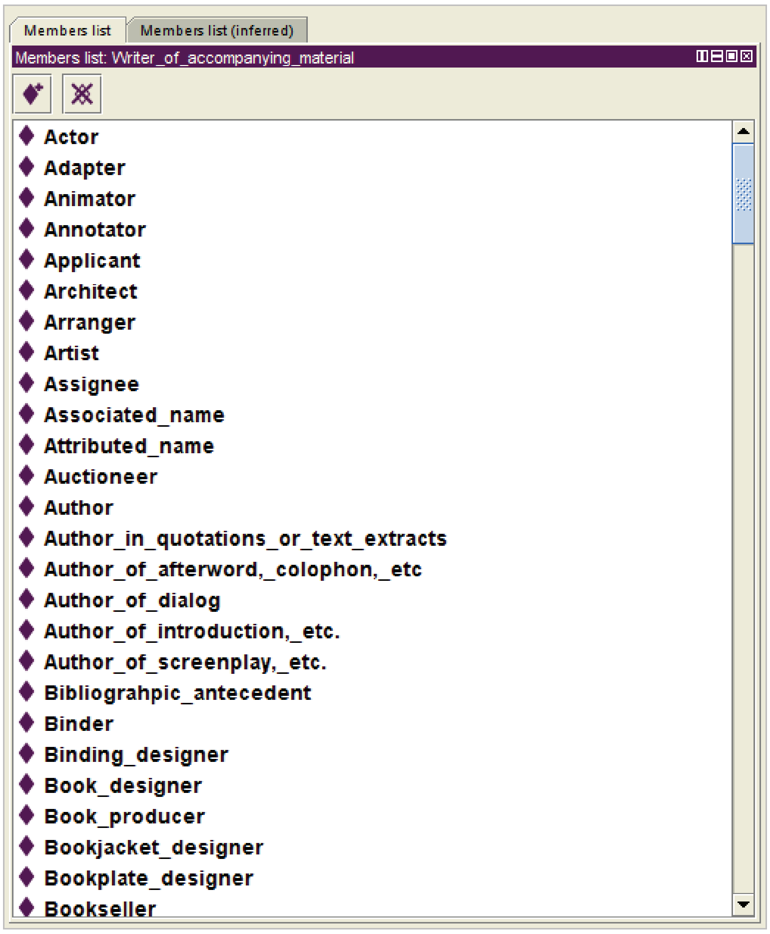

Figure 4: Example of Individuals of *Contribution* Sub-Class implemented into ePrOnto ontology

Properties are binary relations on individuals – i.e. properties that link two individuals together. They can have inverses. Properties can be limited to having a single value – i.e. to being functional. They can also be either transitive or symmetric.

Properties are also used to create restrictions in OWL. The latter ones could be of three categories: quantifier restrictions, cardinality restrictions, and *hasValue* restrictions. These quantifier restrictions are composed of a quantifier, a property, and a filler. The two quantifiers that may be used are: the existential quantifier (read as *at least* or *some* in OWL speak) and the universal quantifier (read as *only* in OWL speak). Hence for a set of individuals, an existential restriction ($\exists$) specifies the *existence of a* (i.e. at least one) relationship along a given property to an individual that is a member of a specific class. For example, $\exists$ *hasContribution Contributor* describes all of the individuals that have at least one (some) relationship along the *hasContribution* property to an individual that is member of the class *Contributor*.

5. Outputs and Results

The ontology for ePrints Information Management we called ePrOnto could be characterized in regards to the classification presented in the methodology section. Hence, ePronto ontology, developed with Protégé OWL, could be considered as a semi-automatic single ontology with multiple layers (different levels of hierarchy). Some of the layers could be seen in left part of Fig. 5 as well as the automatically generated OWL code. An overview of the whole ePrOnto ontology structure is demonstrated in the right part of Fig. 5.

Moreover, ePrOnto was designed in a one-step collaborative way from scratch and is application independent. The adopted approach for the knowledge capture is a middle-out strategy.

As none of the ontology development methodologies described in the literature were directly suitable, ePrOnto was developed according to a unique scheme.

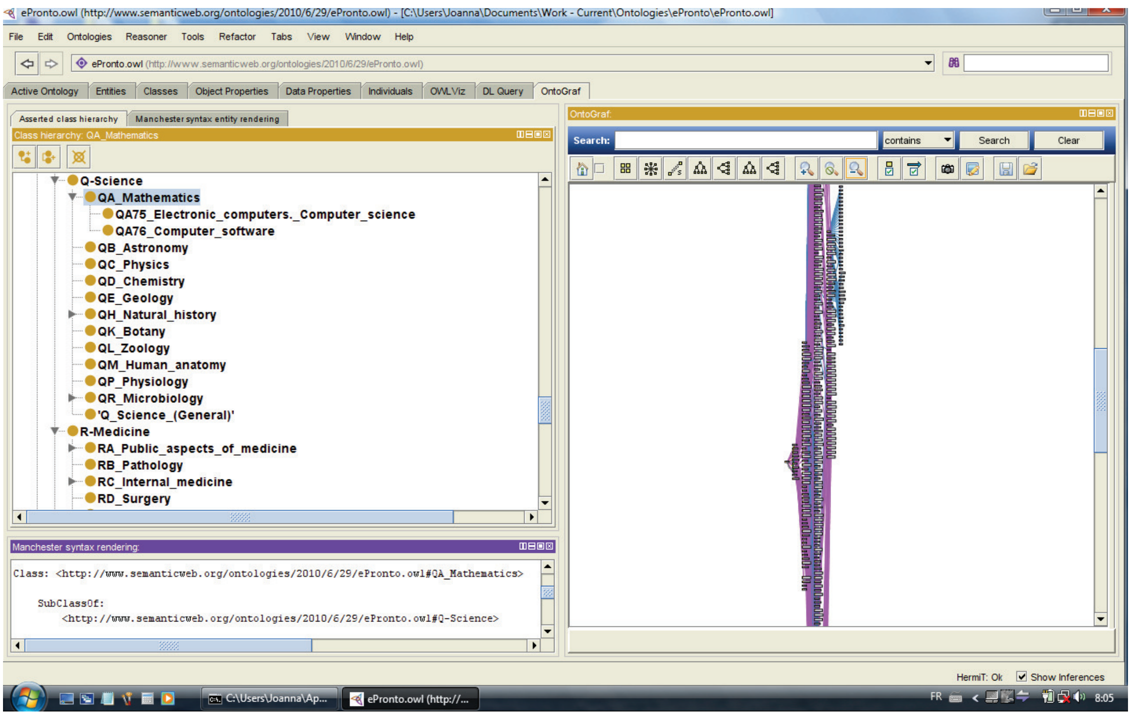

Figure 5: Illustration of ePrOnto ontology

Some of the Items (publications) from ePrints have been encoded into ePrOnto to validate the interoperable services of our designed and implemented ontology. Hence, the proposed ontology is a first ontology developed for ePrints repository management, and which could be compatible with query process.

6. Conclusions and Perspectives

This study was focused on the modelling and the efficient management of the publication repository (ePrints) system of the University of Huddersfield, in an interoperable way. Hence, the related complex data and information have been transformed into structured knowledge through the use of the ontological approach. This has lead to the design and the implementation of ePrOnto – the OWL-based ontology for ePrints information management.

A very recent work [13] has proposed to automatically extract topics from text corpus. Following this direction, our future work will be the development of an innovative method to automatically update the developed ontology (ePrOnto) in order to provide a fully automatic interoperable service.

7. References

[1] http://www.modeliosoft.com/

[2] http://protege.stanford.edu/

[3] S. W. Ambler, *The Elements of UML 2.0 Style*, Cambridge University Press, New York, USA, 2005.

[4] N. Dahlem, J. Guo, A. Hahn and M. Reinelt, “Towards an user-friendly ontology design methodology”, In *Proceedings of the IEEE International Conference on Interoperability for Enterprise Software and Applications (IESA’09)*, pp. 180-186, April 2009.

[5] M. Fernandez Lopez, “Overview of methodologies for building ontologies”, In *Proceedings of the IJCAI Workshop on Ontologies and Problem Solving Methods (KRR5'99)*, pp. 4.1-4.13, August 1999.

[6] N. Gessa, M. Busanelli, P. De Sabbata, and F. Vitali, “Extracting a semantic view from an ebuisness vocabulary”, In *Proceedings of the IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services (EEE’06)*, pp. 57-57, June 2006.

[7] P. Green and M. Rosemann, *Buisness Systems Analysis with Ontologies*, Idea Group Publishing, Hershey, PA, USA, 2005.

[8] T. R. Gruber, “Towards principles for the design of ontologies used for knowledge sharing”, *International Journal of Human-Computer Studies*, 43(5-6):907-928, November 1995.

[9] M. Horridge, *A Practical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools*, Ed. 1.2, March 2009.

[10] S. Jain and J. Pareek, “Automatic topic(s) identification from learning material: An ontological approach”, In *Proceedings of the IEEE International Conference on Computer Engineering and Applications (ICCEA'10)*, pp. 358-362, March 2010.

[11] M. Peim, E. Franconi, N. W. Paton, and C. A. Goble, “Query processing with description logic ontologies over object-wrapped databases”, In *Proceedings of the IEEE International Conference on Scientific and Statistical Database Management (SSDBM’02)*, pp. 27-36, July 2002.

[12] S. Suwanmanee, D. Benslimane, and P. Thiran, “OWL-based approach for semantic interoperability”, In *Proceedings of the IEEE International Conference on Advanced Information Networking and Applications (AINA’05)*, vol. 1, pp. 145-150, March 2005.

[13] W. Wei, P. Barnaghi, and A. Bargiela, “Probabilistic topic models for learning terminological ontologies”, *IEEE Transactions on Knowledge and Data Engineering*, 22(7):1028-1040, July 2010.