Information Reuse and Interoperability with Ontology Patterns and Linked Data

First Experiences from the ExpertFinder Project

Karl Hammar, Feiyu Lin, and Vladimir Tarasov

School of Engineering at Jönköping University
P.O. Box 1026
551 11 Jönköping, Sweden
{haka,life,tavl}@jth.hj.se

Abstract. Semantic web technologies show great promise in usage scenarios that involve information logistics. This paper is an experience report on improving the semantic web ontology underlying an application used in expert finding. We use ontology design patterns to find and correct poor design choices, and align the application ontology to commonly used semantic web ontologies in order to increase the interoperability of the ontology and application. Lessons learned and problems faced are discussed, and possible future developments of the project mapped out.

Keywords: semantic web, ontology design patterns, ontology alignment, linked data.

1 Introduction

The semantic web extends the current web, using semantic web ontologies to solve the problem of defining common meanings of terms and concepts. This enables automatic integration of information between different platforms and information systems utilizing the common definitions. Furthermore, it allows for reasoning agents to collate such information and from it infer knowledge that is greater than the sum of its parts.

In the context of information logistics the benefits of semantic web technologies are many. By allowing users to query for concepts rather than keywords, the precision of information queries can be substantially improved. Common concepts allow for services to be created in which users can query for information, rather than simple data, originating from a multitude of sources, interlinked using standardized vocabularies.

The potential gains of using semantic web technologies for information logistics purposes are not limited to enterprises or large organizations. In [1] a picture is painted of a future in which the use of semantic web technologies can realize ubiquitous computing scenarios, allowing individuals immediate access to information they need in order to plan and execute their every day life with a minimum of fuzz.

W. Abramowicz, R. Tolksdorf, K. Węcel (Eds.): BIS 2010 Workshops, LNBIP 57, pp. 168–179, 2010.

The contribution of this paper is an experience report on applying ontology design patterns and connecting existing linked data sources when developing a semantic web ontology for use in an expert finding system. The paper offers a perspective on two aspects of reuse in ontologies: reusing modeling knowledge via design patterns, and designing for reuse by aligning to commonly used standard ontologies.

The rest of this paper is structured as follows. First, we introduce the theoretical background of the areas and the ExpertFinder project. In section 3, we discuss how reuse is actually performed within our project. Our experience and lessons learnt from this are presented in section 4. Section 5 summarizes the work and gives an outlook on future work.

2 Background

In this section, we discuss the theoretical background of the technologies and methods touched upon in the paper, and the project to which these were applied.

2.1 Linked Data Interoperability

Linked data refers to data and information that has been published online using semantic web standards (ontologies and semantically marked up XHTML). For such linked data to be interoperable and truly form one "web of data" rather than multiple incompatible "islands of data", the ontologies used for defining concepts need to be aligned to one another. Automatically matching and mapping ontology concepts is an unsolved problem as yet, meaning that ontology alignment is still a rather labor intensive process if one does not design one's ontologies with interoperability in mind from the very beginning.

If, however, one can align with the commonly used ontologies of the semantic web, the potential interoperability and integration benefits achieved without specialized integration software or technologies can be significant. Some examples of semantic web integration and interoperability are mentioned in [2], such as the integration of the various types of heterogeneous datasets in the life sciences, and the enabling of exploitation of public data in different formats by citizens. Another example including the manufacturing industry and efficient supply chain management is discussed in [3].

2.2 Ontology Design Patterns

Ontology design patterns have been proposed (by for instance [4]) as a method of simplifying ontology development. The term "ontology design pattern" has multiple definitions in literature. In this paper we adopt the definition used for the similar term *ontology pattern* in [5], which states:

An ontology pattern is a set of ontological elements, structures or construction principles that intend to solve a specific engineering problem and that recurs, either exactly replicated or in an adapted form, within some set of ontologies, or is envisioned to recur within some future set of ontologies.

Recently a lot of work has been done on such patterns, both with regards to construction and to the various uses of them. A general method for constructing and reusing content ontology design patterns are presented in [6]. The use of a web portal to hold a repository of patterns is presented in [7], an idea that has been realized in the creation of the site OntologyDesignPatterns.Org. Ontology patterns and case-based reasoning is used in [5] to perform semi-automatic construction of ontologies from input texts.

Experiments on using ontology design patterns are detailed in [8], which concludes that the use of patterns improve the quality and coverage of the resulting ontologies, and that they help modelers avoid common mistakes.

2.3 The ExpertFinder Project

ExpertFinder is an internal research project at Jönköping University. The driving motivation behind the project is the possibility of using semantic technologies to help university staff in finding experts within various fields. Potential experts are to be found among all of the university researchers and teachers, whose profiles are stored in a semantic web ontology. Two main usage scenarios have been used to focus the project:

– **Internal Expert Finding.** A course manager needs a guest lecturer on IT in health-care and wants to find the best suited person in the university. The course manager writes a simple keyword query, which is matched against the ontology, and information about the persons related to the matching concepts are retrieved. Rules are used for ranking the results, for instance weighing prior teaching experience in the subject higher than being a co-author of a paper remotely concerned with the subject.
– **External Expert Finding.** A company needs help to understand how to setup an ad-hoc sensor network in their facilities. A company representative can use the system for finding relevant competencies in the university. The user writes a simple keyword query, and if terms are not present in the ontology, semantically similar terms are suggested by the system. If no ontology term matches a certain keyword, the keyword is used by a standard IR system, as a complement to the ontology-based query. To help the user to see if the experts are relevant for the task, a motivation of why they were retrieved is also presented in the ranked result list.

The ExpertFinder ontology is implemented in the OWL language. Both the data import logic that populates the ontology and the query interfaces for accessing it are implemented in Java, and the intention is to deploy the latter as a web application on the university web site in the future.

3 Reuse in the ExpertFinder Ontology

This paper deals with reuse in the design of the ExpertFinder ontology. Apart from this, software components and frameworks have also been reused successfully within the actual application that populates and queries the ontology, but such reuse is considered to be outside of the scope of this paper.

During the development of the ExpertFinder ontology, the methodology for ontology construction proposed in [9], was followed. This methodology consists of four distinct phases: requirements analysis, building, implementation, and evaluation and maintenance. The ontology is in accordance with this methodology being developed iteratively.

The ontology resulting from the first iteration did not make use of any ontology patterns, and was not aligned to any standard semantic web ontologies. For the purpose of increasing interoperability and compatibility as well as finding suboptimal design choices made in the first version, these features were implemented in a second iteration. It is primarily the experiences in performing this second iteration that this paper covers, although the section on content reuse refers to the initial design.

In the current version, the domain has been restricted to researchers and teachers of the department of computer and electrical engineering. However, the ambition is that the general design should be general enough that future iterations will be able to accommodate knowledge about staff, projects, courses and programs of all schools and departments of the university.

3.1 Linked Data Use

In the initial iteration, ontology search engines such as Swoogle, Ontosearch, and Watson were used to find academic ontologies describing universities and research. However, after analyzing the ExpertFinder requirements, these ontologies were deemed too complex and not suitable for the project, and the initial construction was therefore performed from a blank slate without any reuse. When the second iteration began, the first task was thus to find semantic web ontologies to align that had a reasonable scope for this project, i.e. that were general enough to further interoperability but still specific enough to be applicable to the application. Also, since aligning ontologies can be rather time consuming, the ontologies found had to be easy to tie in to the first iteration version.

BIBO. The first such ontology that was found to be appropriate was the Bibliographic Ontology, commonly known as BIBO ([10]). Since the ExpertFinder ontology covers scientific publications, standardizing how such publications are represented makes a lot of sense. It further allows easier implementation of import from public data sources that use the BIBO standard (including for instance the US Library of Congress). Implementing BIBO compatibility consisted of importing the ontology and connecting the defined ExpertFinder classes and properties to their BIBO equivalents by defining them to be linked via owl:equivalentClass statements. Many of the concepts were found to map easily in this manner, but some were not found to have any natural counterparts in BIBO. In these cases, the most common superclass was mapped to instead, providing at least some interoperability. For instance, the concept “Conference Papers” had no direct equivalent, and was therefore mapped as a subclass of bibo:Document which is a more general term.

Implementing BIBO compatibility enabled us to notice and fix shortcomings in our own initial design. One such shortcoming was that our ontology made frequent use of datatype properties to define values of publications where using object properties could provide better reasoning capacity and semantics. To take one example, the term “peer reviewed conference paper” was often used as a comment or status string for individual publications, whereas BIBO instead models publication review statuses semantically, making it possible to query for all peer reviewed items in the knowledgebase in a much simpler manner.

FOAF. BIBO itself depends upon the Friend-of-a-Friend vocabulary, FOAF ([11]), that was also imported. FOAF provides a standard way of talking and reasoning about people, groups and organizations. This is of course useful in many situations, and very appropriate for a system such as ExpertFinder that deals primarily with finding people. FOAF compatibility was implemented in the same manner as BIBO, by connecting existing classes and properties to their equivalents or nearly related concepts in the foaf: namespace. We were in this manner able to connect both our concepts of experts (as persons) and universities (as organizations) to standard classes. For a listing of the classes (not including object or data properties) aligned in this manner, see table 1.

Table 1. ExpertFinder classes mapped to BIBO and FOAF

FOAF/BIBO Class	ExpertFinder Class	Mapping relation
bibo:Document	ef:ConferencePaper	Subclass of
bibo:Chapter	ef:BookChapter	Equivalent
bibo:AcademicArticle	ef:JournalArticle	Equivalent
bibo:Report	ef:Report	Equivalent
foaf:Organization	ef:UniversitySchool	Subclass of
foaf:Person	ef:Expert	Subclass of

W3C Time. The third standard ontology that was aligned to was the W3C Time ontology originally presented in [12]. While this standard is formally listed as being retired, it is nonetheless still in active use in a number of ontologies, including BIBO. Discussing time is something that falls within the domain of ExpertFinder (for instance, modeling the times when a certain expert participated in a certain project or course), and being able to do it in a standardized interoperable way motivates the use of this standard, in spite of its retired status.

Aligning to W3C Time required more work than aligning to BIBO and FOAF, since the previous date representation was not compatible with the imported time: namespace definitions. New properties had to be defined, using the W3C Time definitions, and all existing time references from projects, courses, etc. in the ontology had to be manually updated to use the new properties. Since the amount of data in the ontology was relatively small this was not a huge undertaking, but for larger ontologies such a refactoring could be quite time consuming if not possible to automate.

3.2 Pattern Use

Having aligned to some common ontologies, the next task of the second iteration of development was to try to implement ontology design patterns. While the motivation of using such patterns often are to speed up initial development of ontologies from scratch, in this case the motivation was rather to find bad design choices made in the first iteration.

In order to find applicable patterns the OntologyDesignPatterns.org repository was studied. A number of patterns matching needs in the ExpertFinder domain were identified, and out of these a subset were implemented in the ontology, solving design issues that existed. The patterns that were selected, based on ease of applicability and description quality, were:

- **Information Realization**[1]
  One key concept in the ExpertFinder domain is courses. Courses can be held multiple times, each time having different teachers, participants, schedules, etc. but still conceptually being the same course, giving the same number of study credits, covering the same topics, and having same course description. In the initial version of the ExpertFinder ontology, the differentiation between a course as such a static concept and a course as a recurring event was not modeled. This led to an unnecessary duplication of information, as each time the course was run was modeled as a whole new course. This design also makes it unnecessarily difficult to query for information that involves multiple runs of the same course. Implementing the Information Realization pattern solved this problem. While Information Realization is documented as being intended to model information objects and their physical realizations (i.e. a written work and the actual printed book), it works equally well when describing a concept and the performance or implementation of that concept, in this case a course and the actual instance of that course.
- **Partition**[2]
  The partition pattern is a logical pattern published in [13] that models a concept that is composed of several disjoint subconcepts. In terms of OWL and classes, it can be useful as a way of dividing a classification into subclassifications. In the ExpertFinder ontology, two instances of this pattern were applied, dividing the Project class into various types of projects (EU Project, National project, Industry Project, etc.), and dividing the ScientificDegree class into fitting subclassifications (BachelorDegree, MasterDegree, PhDDegree, etc.). Previously this same information about the classification of projects and scientific degrees were present in the ontology, but only as simple data properties that could not be as easily reasoned about.
- **Nary Participation**[3]
  The nary participation pattern models time indexed participation of objects in events. In the ExpertFinder domain, the participation of teachers

[1] http://ontologydesignpatterns.org/wiki/Submissions:Information_realization
[2] http://ontologydesignpatterns.org/wiki/Submissions:Partition
[3] http://ontologydesignpatterns.org/wiki/Submissions:Nary_Participation

in courses or experts in projects match this pattern very well. However, the available reusable pattern building block makes extensive use of other design pattern building blocks from the OntologyDesignPatterns.org repository (events, situations, time intervals, etc) that are not compatible with the previously aligned semantic web ontologies. The general idea of the Nary Participation pattern was thus deemed appropriate for ExpertFinder, but the implementation details of the pattern as it was available made it less well suited. For this reason, the pattern implementation was partially recreated based on concepts already existing within or imported through the aligned ontologies, as displayed in figure 1. This was a successful design and provides the application with the possibility of modeling time indexed expert participation in projects, which is suitable for projects with extensive run times or different phases.

3.3 Content Reuse

One key requirement developed in the first iteration was the need to model fields of education and research areas. Three existing classifications were found that could potentially be reused: the international standard classification of occupations (ISCO-08)[14], the Eurostat classification of fields of education and training (1999) [15], and the ACM computing classification system (1998) [16]. Since the ISCO and Eurostat classifications are not detailed enough, the ACM classification was selected. [17] is a topic hierarchy in rdfs based on the ACM computing classification.

However, the ACM computing classification system has not been updated since 1998 and thereby lacks some important concepts, such as the semantic web domain. Therefore, we introduced an extended taxonomy of semantic web topics [18] (available as a SKOS vocabulary and also available in OWL as an extension of the SWRC ontology) into the ExpertFinder ontology.

The knowledge content of the ontology, the actual facts used to fill the knowledgebase, are imported from the data sources available at the university and in this sense reused from personal web pages (experts' profiles), DiVA (the database for storage of research publications and student theses used at many Swedish universities), Neverlost (the software used for class scheduling), course syllabi, and project description spreadsheets. Since these data sources are all published in different formats, various tools for importing the information needed to be constructed. Since common keys were available to correlate the data in many cases (login names, email addresses, course codes, etc) little integration efforts were necessary, and since the common format of an OWL ontology was already defined, no specific information integration metalanguage needed to be implemented, leading these tools to be relatively simple.

4 Experiences

This section reports the experiences learned from the ExpertFinder project with regard to information modeling and reuse in ontologies.

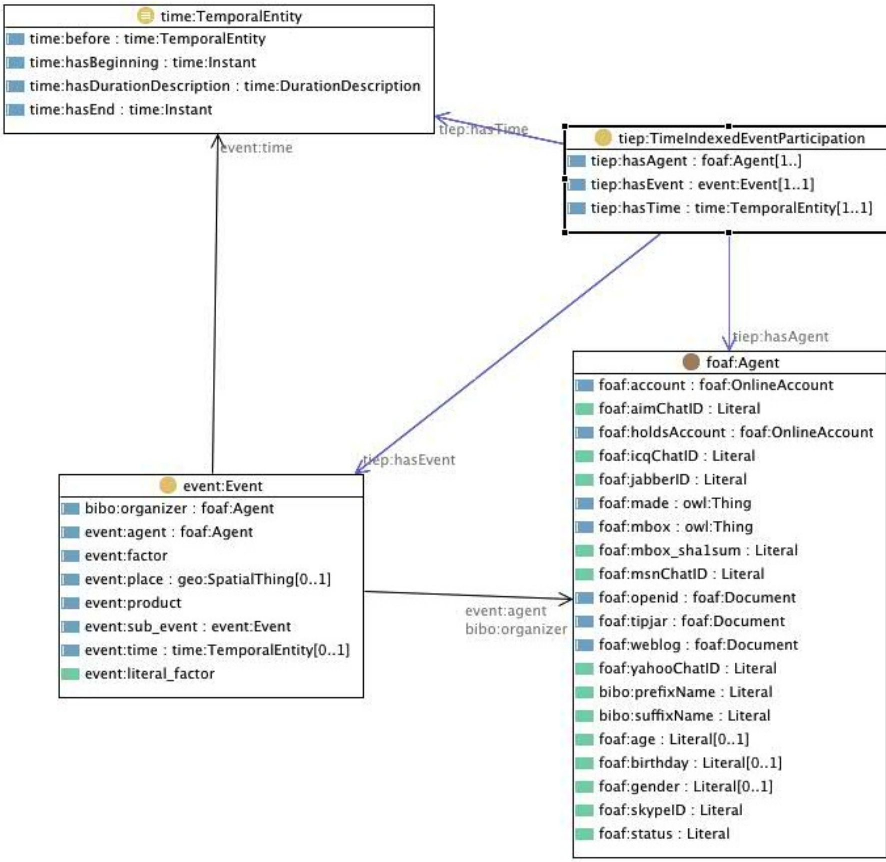

Fig. 1. Alternate implementation of Nary Participation pattern

The modifications resulted in changes to six former top level classes within the class subsumption hierarchy, and the addition of 13 new classes at various levels of the hierarchy. Compared to the 190 classes in the default ontology namespace this is rather few (3,5 %) and the impact of the performed changes may therefore be interpreted as being very small. However, the changes performed concern key concepts within the ontology, making this low number somewhat misleading - many of the 190 classes are in fact research classifications imported as described in section 3.3, having less importance for the overall structure and reasoning capacity of the ontology. The changes made to the properties hierarchy are likely more indicative of the impact of changes performed: 42 % of all properties within the ontology have been changed either in range, domain, or inheritance, indicating a significantly larger change.

4.1 Ontology Design Patten (ODP) Experience

The way of documenting and providing the patterns in the ODP portal is sound, although a lot of patterns currently listed there have rather poor documentation. Our perception is that we actually could save time and increase quality of our ontology by using ODPs. However, patterns related to our area of work, in particular content patterns in the Academy or University domains, do not exist in the ODP portal. Instead we partly reused the ACM topic and semantic web topics hierarchies, which are not patterns in themselves, but provide commonly agreed upon vocabularies within the domain. This improved the quality of our ontology and saved time but to substantiate this statement we need to do more experiments and consider more patterns. We also experienced a lack of tool support for integrating ODPs into an ontology.

One fault in the design of the ontology that has been experienced when applying the partition pattern is the frequent occurrences of cases when strings are used to describe what class a certain individual belongs to, when in fact it would make more sense to model this using the subsumption hierarchy. This was common enough in the ExpertFinder ontology to warrant further study, perhaps being indicative of an antipattern.

Another problem that was experienced in applying the patterns from OntologyDesignPatterns.org is the situation that arose with the Nary Participation pattern, that is, that while the basic idea of a certain pattern is sound, the implementation provided as a reusable OWL building block is not compatible with the ontology, leading one to have to reconstruct the implementation. This has in our case only been a minor problem. If more mapping patterns were developed to connect the concepts within the pattern repository to other well used semantic web ontologies, the occurrences of this type of problems could be minimized.

4.2 Experiences in Aligning Ontologies

The experience of aligning ontologies reinforces our opinion that aligning at the outset is probably a more efficient solution than doing so at a later time, since doing the latter requires both finding the appropriate ontologies and appropriate ways of connecting them, which can necessitate a lot of refactoring of the design. In the ExpertFinder case only a certain subset of the features were aligned in this iteration, due to the complexity of the task.

On the other hand, one important experience is the truth that lies in the commonly used expression that even a little semantics goes a long way - even though the ExpertFinder is not perfectly compatible with every imaginable web ontology, it does at this time allow for querying using BIBO and FOAF syntax, allowing automatic integration of bibliographic and personal information via linked data. This is quite a feat compared to traditional relational database-backed applications or SOA web service type solutions that tend to necessitate quite a lot of forethought and design in order to achieve good interoperability.

Finding the right semantic web ontologies to use was also experienced to be a somewhat tricky task. Since the question of objective quality metrics for ontologies is still an open issue, there are no simple easily quantifiable ways of measuring what is a good ontology and what is not, when picking possible ontologies to work with. Instead, one has to look at factors like how well the ontology in question covers the problem space, how well it is used, and how much community backing it has.

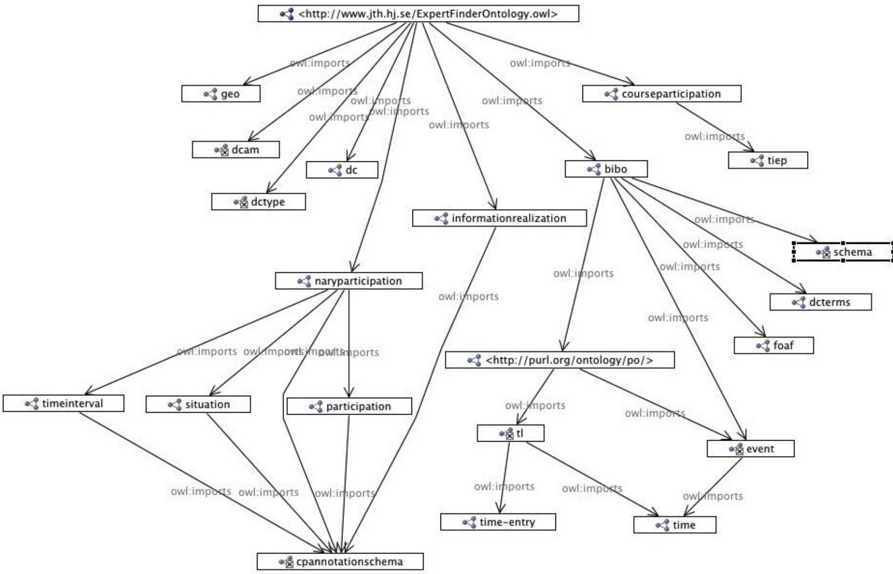

Fig. 2. ExpertFinder ontology import graph. Note that this figure is intended to illustrate the size and complexity of the import closure graph, not what individual ontologies are imported and labelled.

One problem that was experienced in performing alignment was that a lot of semantic web ontologies import and reuse other base ontologies. Since the OWL language has no feature for performing partial imports, and since imports are transitive, this gives the result that the sum total of concepts and properties within the import closure of one's ontology rapidly becomes very large, as displayed in figure 2. This growth in complexity makes alignment more difficult as the possible choices of how to align concepts also grows very rapidly, and understanding the pros and cons of each choice requires extensive knowledge of all of the imported ontologies and how they are designed, not just the first imported one. Another problem with this situation is that the ontology development tools available are not well suited for dealing with this situation when very large numbers of classes and properties are imported. One possible solution to this problem is the RDFex solution proposed by [19], whereby an import proxy service can be used to filter out only the necessary classes and properties from a certain namespace.

5 Summary and Future Work

In this paper, we have discussed our experiences of reusing knowledge and information in the ExpertFinder project, by way of ontology design patterns and the reuse of existing university data sources. We have also looked at designing for reuse, by aligning to ontologies already available on the semantic web.

Our experiences indicate that while there is still work to be done in developing both ontology design patterns and tools for using such patterns, the general idea of reusing design knowledge in this manner seems to be sound. As for ontology alignment, the experiences have also been positive, though we note that there is as of yet no clear consensus on or objective measurement of what ontologies are the most useful to align to, leaving room for future studies.

The ExpertFinder project is still ongoing and several future challenges were discovered while implementing these technologies. To begin with, the way that research areas and research topics are modeled within the ontology will need an overhaul in order to provide better support for reasoning and retrieval with SPARQL queries. Since no documented design patterns covering this area have been found, it is likely that we will have to design our own solution, possibly publishing it as a pattern if found to be general enough.

Another future research task deals with version management of ontologies - a lot of now redundant properties could be pruned from the ontology after the second development iteration, but it must first be studied what the consequences of such pruning might be. Research on this problem becomes even more important when one considers that other ontologies might import the ExpertFinder ontology, making any changes performed cascade throughout the semantic web.

Acknowledgements

Andreas Billig, Michael Ricklefs, and Annika Öhgren have provided valuable contributions to the ExpertFinder project.

This work is partly funded by the project DEON (Development and Evolution of Ontologies in Networked Organizations) based on a grant from STINT (The Swedish Foundation for International Cooperation in Research and Higher Education); grant IG 2008-2011.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284(5), 34–43 (2001)
2. Shadbolt, N., Hall, W., Berners-Lee, T.: The Semantic Web Revisited. IEEE Intelligent Systems 21(3), 96–101 (2006)
3. Hendler, J., Berners-Lee, T., Miller, E.: Integrating Applications on the Semantic Web. Journal of the Institute of Electrical Engineers of Japan 122(10), 676–680 (2002)

4. Svatek, V.: Design Patterns for Semantic Web Ontologies: Motivation and Discussion. In: Proceedings of the 7th International Conference on Business Information Systems, Poznan, Poland (2004)
5. Blomqvist, E.: Semi-automatic Ontology Construction based on Patterns. PhD thesis, Linköping University, Department of Computer and Information Science (2009)
6. Presutti, V., Gangemi, A.: Content Ontology Design Patterns as Practical Building Blocks for Web Ontologies. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 128–141. Springer, Heidelberg (2008)
7. Presutti, V., Gangemi, A., David, S., de Cea, G., Suárez-Figueroa, M., Montiel-Ponsoda, E., Poveda, M.: D2.5.1: A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies. NeOn Project Deliverable (EU-IST-2005-027595) (2008)
8. Blomqvist, E., Gangemi, A., Presutti, V.: Experiments on Pattern-based Ontology Design. In: Proceedings of the Fifth International Conference on Knowledge Capture, Redondo Beach, CA, USA, pp. 41–48 (2009)
9. Öhgren, A.: Towards an Ontology Development Methodology for Small and Medium-sized Enterprises. Licentiate Dissertation 1401, Linköping University, Department of Computer and Information Science (2009)
10. D'Arcus, B., Giasson, F.: Bibliographic Ontology Specification, `http://bibliontology.com/specification` (accessed: 2010-03-01)
11. Brickley, D., Miller, L.: FOAF Vocabulary Specification 0.97, `http://xmlns.com/foaf/spec/20100101.html` (accessed: 2010-03-01)
12. Hobbs, J., Pan, F.: An Ontology of Time for the Semantic Web. ACM Transactions on Asian Language Information Processing (TALIP) 3(1), 66–85 (2004)
13. Noppens, O.: Concept Partition Pattern. In: Proceedings of the Workshop on Ontology Patterns, Washington D.C., USA (2009)
14. International Labour Organization: ISCO - International Standard Classification of Occupations, `http://www.ilo.org/public/english/bureau/stat/isco/isco08/index.htm` (accessed: 2010-03-02)
15. Eurostat: Classification of fields of education and training (1999), `http://ec.europa.eu/eurostat/ramon/nomenclatures/index.cfm?TargetUrl=LS%T_NOM_DTL&StrNom=EDU_TRAINI` (accessed: 2010-03-03)
16. Association for Computing Machinery: The ACM Computing Classification System (1998 Version), `http://www.acm.org/about/class/1998` (accessed: 2010-03-02)
17. Universität Karlsruhe: Acm topic hierarchy, `http://www.aifb.uni-karlsruhe.de/WBS/pha/bib/acmtopics.rdf` (accessed: 2010-03-04)
18. Diederich, J.: Semantic Web Topic Hierarchy, `http://semanticweb.org/wiki/Semantic_Web_Topic_Hierarchy` (accessed: 2010-03-04)
19. Knublauch, H.: Rdfex - partial imports of common rdf namespaces, `http://sparqlpedia.org/rdfex/` (accessed: 2010-03-04)