Selecting Content Ontology Design Patterns for Ontology Quality Improvement

Birger Lantow[1], Kurt Sandkuhl[1,2], and Vladimir Tarasov[2]

[1] Institute of Computer Science, University of Rostock
Albert-Einstein-Str. 22, 18059 Rostock, Germany
{birger.lantow, kurt.sandkuhl} @uni-rostock.de

[2] School of Engineering at Jönköping University
P.O. Box 1026, 551 11 Jönköping, Sweden
tavl@jth.hj.se

Abstract. While Ontology Design Patterns (ODPs) are intended to support the process of ontology engineering, they can also be used in order to improve existing ontologies. This paper describes strategies for the selection of candidate ODPs that qualify for the improvement of a given ontology. Starting point is the ExpertFinder ontology that allows for competency description of researchers. ODP selection strategies are performed on content ODPs from the ODP wiki-portal that was initiated by the NeOn-project. Lessons learned and problems faced are discussed, and possible future developments are mapped out. The contributions of this paper are (1) a strategy for selecting ODP suitable for improving a given ontology, (2) experiences from using this strategy for selecting ODP, (3) recommendations for a better support of ODP selection.

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

1 Introduction

Work presented in this paper combines techniques from ontology engineering with experiences from quality management. Quality is considered an essential factor for acceptance of technologies and solutions in many disciplines. Furthermore, quality also contributes to efficiency of work and operation processes and to robustness and usability of products. Due to the growing use of ontologies in knowledge-based systems for industrial and administrative applications, standards, procedures and practices for quality improvement of ontology construction processes and the artifacts produced during these processes gain of importance. Although considerable efforts have been spent on developing ontology assessment and evaluation approaches, including ways to measure quality and techniques to improve it (cf. Section 2.2), generally accepted practices for industrial are still missing.

The objective of this paper is to contribute to quality ontologies by focusing on the use of ontology design patterns for improving the quality of existing ontologies.

Ontology design patterns (ODP) have been proposed as encodings of best practices (cf. Section 2.1) supporting ontology construction by facilitating reuse of proven solution principles. In this paper, focus is specifically on Content ODP and on investigating feasibility and utility of incorporating them into ontologies. Our working hypothesis is that Content ODP which have proven to be useful in constructing quality ontologies also can be supportive in improving ontologies. The intention is to gather experience how to best apply Content ODP for this purpose.

The contributions of this paper are (1) a strategy for selecting ODP suitable for improving a given ontology, (2) experiences from using this strategy for selecting ODP, (3) recommendations for a better support of ODP selection.

The remaining part of the paper is structured as follows: Section 2 gives a brief overview to ontology design patterns and approaches for quality improvement. Section 3 describes a strategy for ODP selection in order to improve existing ontologies. The ExpertFinder ontology is introduced as a case study in section 4, while section 5 applies the suggested ODP selection strategy based on the ExpertFinder ontology. The final section 6 summarizes the experiences and gives recommendations for a better support of ODP selection and usability.

2 Background on ODPs

Relevant background for this paper includes ontology design patterns (section 2.1) and approaches for quality assurance of ontologies by use of ODPs (section 2.2).

2.1 Ontology Design Patterns

In a computer science context, ontologies usually are defined as explicit specifications of a shared conceptualization [10]. Due to the increasing use of ontologies in industrial applications at larger scale, ontology construction and ontology evaluation have become a major area of ontology engineering. The aim is to efficiently produce high quality ontologies as a basis for knowledge management, semantic web applications or enterprise systems. Despite quite a few well-defined ontology construction methods and a number of reusable ontologies offered on the Internet, efficient ontology development continues to be a challenge, since this still requires a lot of experience and knowledge of the underlying logical theory.

Ontology Design Patterns (ODP) are considered a promising contribution to this challenge. In 2005, the term ontology design pattern in its current interpretation was mentioned by Gangemi [2] and introduced by Blomqvist & Sandkuhl [3]. Blomqvist defines the term as "a set of ontological elements, structures or construction principles that solve a clearly defined particular modeling problem" [4]. Ontology design patterns are described as encodings of best practice, which reduce the need for extensive experience when developing ontologies. Using ODPs, less experienced engineers can apply the well-defined solutions provided in the patterns when creating ontologies.

[5] discusses different types of ODP under investigation with their differences and the terminology used. The two types of ODP probably receiving most attention are logical and content ODP. Logical ODP focus only on the logical structure of the representation, i.e. this pattern type is targeting aspects of language expressivity, common problems and misconceptions. Content ODP offer actual modeling solutions within an application domain and are often instantiations of logical ODP. Due to the fact that these solutions contain actual classes, properties, and axioms, content ODP are considered by many researchers as tailor-made for a specific domain, even though the domain might focus on general issues like 'events' or 'situations'. This paper has its focus on the use of content ODPs. Platforms offering ODP currently include the ODP wiki portal initiated by the NeOn-project[1] and the logical ODPs maintained by the University of Manchester.

2.2 Quality assurance of ontologies and ODP

Work in the area of quality assurance for ontologies includes different perspectives, such as the quality of the ontology as such, the quality of the process of ontology construction, and tools supporting the ontology engineer in achieving high quality. In the context of this paper, the focus is on quality of the ontology as such. Quality assessment of ontologies as such has been subject of many research activities [6], but the quality criteria vary considerably between different approaches and often address structural, logical, and computational aspects of ontologies. Furthermore, metrics originating from software quality evaluation have been investigated [7]. Many of the metrics proposed during last years lack an empirical validation in a large number of cases, i.e. what metrics value can be considered as „good“ or as „bad“ often has not been defined due to an insufficient number of reported applications.

Evaluation of the accuracy of ontology content, i.e. suitability and conformance with the domain to be represented, can be performed using a gold standard. In this context similarity metrics, as proposed for example by [8], are used to measure the deviation from the gold standard. These approaches are criticized for mainly using structural graph similarity and for not taking into account the semantics of class definitions or that different kinds of deviations should be weighted differently. Furthermore, a (single) gold standard often is difficult to develop due to a very limited number of experts available in the domain.

Furthermore, approaches were proposed for evaluating „ontologies in use“, i.e. to evaluate the fitness for a task to be performed with an ontology in a defined scenario. An ontology of high quality "helps the application in question produce good results on the given task" [9]. However, it is difficult to generalize the results from such approaches, since they can hardly capture all aspects potentially relevant.

The general consensus of our work is that ODPs as best practices have an inherent proven quality and that their use in ontologies increases for example readability and thus reusability. Additional support for reusability stems from the expectation that ODP can set quasi-standards.

[1] http://ontologydesignpatterns.org

3 Strategies for ODP Selection

Goal of the pattern selection strategies is to efficiently find appropriate patterns that are good candidates for ontology improvement. The final decision should be based on the expertise of the ontology engineer. Thus, the number of choices for the ontology engineer should be minimized/decreased by stepwise filtering the set of ODPs on the base of certain criteria. In the case of automated pre-selection, complexity should also be minimized. In general, this process may also lead to an empty set of ODPs

Prior to the selection of ODPs for ontology improvement, the scope of the improvement process needs to be defined. This influences the applicability of certain filter criteria, as we will see later. In our context, there is a difference between *ontology reengineering* and *ontology restructuring*.

Ontology reengineering covers the complete ontology engineering process. This includes the requirements definition. Thus, ODPs serve as best practices for domain specific or general requirement definition. For example, additional competency questions may be defined that result in additional ontology concepts and in additional information stored in the knowledge base.

Ontology Restructuring on the other hand just aims at the ontology quality by *refactoring* and does not change the informational requirements. As seen in section 2.3 ontology quality has many aspects. In the case of Ontology Restructuring we see for example computational aspects for ontologies in use (reduction of required storage) and benefits in readability and reusability of ontologies. The latter are important for the process of ontology engineering. In the context of this paper, the notion of “living” ontologies that need to be adapted to changes in the real world and to changes in requirements respectively, implies that these quality aspects are relevant. The measurement of the effects on quality themselves is out of focus of this paper.

By *Ontology Restructuring* new conceptualizations may be introduced, others may become obsolete or are going to be represented differently. However, there must exist a mapping that completely describes the newly structured ontology based on the old structure. With reference to Haslhofer and Klas [10], *Ontology Restructuring* includes at least one of the three activities that must be considered during the ontology transition:

1. Linking to ontologies that add conceptualizations and representation structures, e.g. linking to ODPs.
2. Transformation of ontology structure
3. Instance transformation

The tasks that need to be performed can be relevant for filtering, if effort is considered as a criteria for ODP selection. If restructuring steps 2 and/or 3 are performed the application logic around an application ontology needs to be changed-This can lead to considerable additional effort. Effort is not considered in the suggested approach but can be added as an additional step in the selection process.

The suggested approach for ODP selection includes several stages of filtering:

1. Filter by Domain: While ODPs of the “general” domain should always be considered for ontology improvement, also those of the same or closely related

domains compared to the ontology in focus are relevant.

2. Filter by requirements: This filter should not be applied for ontology reengineering tasks since the goal is to derive new requirements. Those should not be filtered out.
   A common method for ontology requirements definition are competency questions. A filter by requirements would check for similar competency questions in ODPs and in the ontology in focus. As stated in [11] by Noy and McGuiness, competency questions may on the one hand serve for testing the ontology but on the other hand they are a help to roughly describe the scope of a domain and do not need to be exhaustive.
   However, requirements specification may also be done in many different ways. Looking into the documentation provided with the ODPs on the ODP wiki, besides "Competency Questions" we will find "Intent", "Solution description", and "Scenarios" as documentation elements that are candidates to provide requirements that are fulfilled by the patterns. However, "Intent" and "Competency Questions" seem to the most appropriate fields for filtering by requirements. "Solution Description" and "Scenarios" in contrast provide help for understanding the used conceptualizations which is useful for the next 3rd step of filtering.

3. Filter by shared conceptualizations: The number of shared conceptualizations should be counted here. A first threshold would be 1. Thus, an ODPs remains in the set of candidates if it shares at least 1 conceptualization with the ontology in focus. The threshold may be increased if necessary. Shared conceptualizations can be identified automatically by comparing the IRI[2]s of the used conceptualizations. This is only possible, if the same representation has been used in ontology and ODP. Another way would be the comparison of used labels, maybe in addition with a synonym data base. However, a manual review may reveal additional conceptualizations that are identical or that overlap but are not represented identically. This filter step is close to the idea of selecting ODPs by their names as described by Hammar et al. in [12]. There ontology engineers selected patterns if the name corresponded to their modeling needs. This is assumed to generally happen on the level of conceptualizations. Therefore the filtering by name at an earlier step than this one does not seem appropriate. This also emphasizes on the need of ODPs to be small enough in order to clearly understand the respective conceptualizations behind them.

4. Filter by compatibility: It should be checked whether all conceptualizations of the remaining patterns are compatible with the ontology in focus. Since only restructuring is intended, there must a transformation rule that populates the classes of pattern based on the current ontology. Structural incompatibilities like abstraction level discrepancies (see for example [11]) may be an obstacle here.

[2] Internationalized Resource Identifier

After narrowing down the set of ODPs to compatible patterns that can be used in order to fulfill the ontology requirements, further selection can be done based on the evaluation of expected effort of restructuring and based on the expected ontology improvement. But this is outside the focus of this paper and will not be discussed.

4 Case Study: ExpertFinder

The ExpertFinder ontology is the result of an internal research project at Jönköping University. It has undergone several development steps and has also been investigated for the possibilities to foster information reuse and interoperability with ODPs and Linked Data. Results are in [13].

The ontology is the base for an ExpertFinder application that allows to find potential experts among all researchers and teachers of the university. The search is done on competence profiles that are represented in the ExpertFinder ontology. The ontology is implemented in the OWL language.

Reliable information about the researchers and teachers at Jönköping University therefore needs to be maintained and efficiently retrieved into the ontology. This process should be supported by a software system for gathering experts' competencies in different areas, and proposing suitable experts to the user.

Overall, the ontology is in the domains of *Research* and *Teaching*. Since organizational structures (position of an expert) are relevant, the Domain of *Management and Organization* may be added. Due to the scientific focus at Jönköping University, *Computer Science* and *Electrical Engineering* are further domains of the ontology.

Specification is available in the form of competency questions:

1. Finding experts in teaching
   - 1.1. Who can give a quest lecture about ontology applications in medicine in a master's course?
   - 1.2. Who can give lectures on knowledge management in a master's course?
   - 1.3. Who can supervise labs in web programming in a bachelor's course?
   - 1.4. Who can be the course coordinator for the Embedded Systems Architectures course?
   - 1.5. Who can supervise master's theses in information engineering?
2. Finding experts in research
   - 2.1. Who has been doing research projects about sensor networks?
   - 2.2. Who has participated in industrial projects in avionics engineering?
   - 2.3. Who has PhD in computer science and is involved in EU projects?
   - 2.4. Who are the authors of journal papers on distributed databases?
   - 2.5. Who is the expert in ontology engineering?
   - 2.6. What is the expertise of person X?

and in the form of requirement definitions that define the following information about a person as required:

1. Education of the expert including field(s) of education, level and year,
2. University/scientific degrees including area and year,
3. Research areas the expert works/worked in (these should be supported by publications/projects),
4. Research papers written by the expert including research field, publication type, year, length,
5. Research projects the expert participates/participated in including research field(s), project type and start/end date,
6. Courses taught by the expert including field(s) of education, course level, number of credits, type of involvement, how many times was involved, year of the last involvement,
7. Employment type, affiliations ,
8. Contact details,
9. Previous experience

Ontology engineering resulted in the ExpertFinder ontology as shown in figure 1. This is the starting point for the investigation of a possible restructuring based on ODPs.

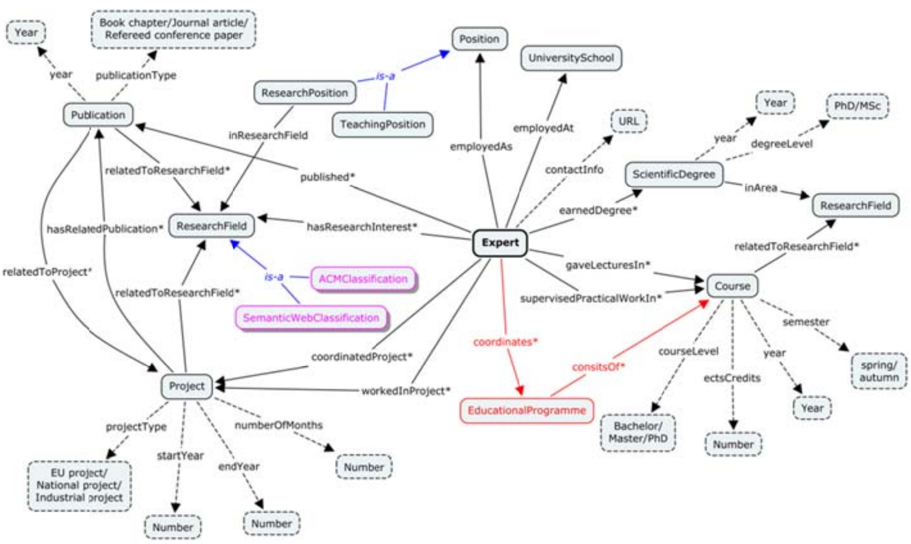

Fig 1. The ExpertFinder ontology

5 Application of the ODP Selection Strategy

In the following, we describe the filtering process of section 3 on the example of the ExpertFinder ontology that has been introduced in section 4. Occurring problems in the several steps are described. Furthermore, solutions and suggestions for future support of ODP selection are derived.

5.1. Filter by Domain

Domain filtering reduced the complete set of 97 Content-ODPs from the ODP wiki down to 72 candidate ODPs. Among the domains of the ExpertFinder ontology and synonymously labeled domains only the “Management” has been found. Thus, ODPs of the domains “Management”, “General”, “Parts and Collections” which could be a subset of general, and ODPs with no given domain remained in the set of candidates. Some of the ODPs with no given domain could have been singled out regarding the domain they actually represent. Therefore, adding the domains here would be an improvement for the possibility of pattern filtering. Furthermore, an automated would be possible if domains and their relations would be more formalized. A taxonomy could be an improvement.

Exemplary decisions (- = neglected, + = passed):

Pattern	Domain	Decision	Comment
ClimaticZone	Fishery	-	%
Co-Participation	General	+	%

5.2. Filter by requirements

It seemed reasonable to use the high abstraction level of the “intent” description of ODPs in order to do a separate filter step based on “intent”. Filtering was done by answering the question whether or not the pattern intent fits to the purpose of the ExpertFinder ontology. The set of candidates was narrowed down to 41 patterns. No intent was given for 14 of these patterns.

Exemplary decisions (- = neglected, + = passed):

Pattern	Intent	Decision	Comment
Communication Event	To model communication events, such as phone calls, e-mails and meetings,….	-	%
Agent Role	To represent agents and the roles they play.	+	Different Roles of Experts described

The next step was the comparison of the competency questions. This step took more effort per pattern. Each pattern needed to be opened separately and several competency questions needed to be compared. A problem arose when comparing pattern competency questions to the competency questions of ExpertFinder ontology. There was a different abstraction level. Only three patterns qualified - all of them because they are dealing with questions of participation which fits to competency question 2.2. The main reason may be that there were almost no patterns specific to the domains of ExpertFinder ontology, but only of the “General” domain. However, also considering the rest of the ExpertFinder specification in relation to the pattern competency questions a set of 35 candidate patterns remained. No competency questions were given for 18 of them. Again, it seems that better documentation of ODPs would foster their application.

Exemplary decisions (- = neglected, + = passed):

Pattern	Competency Questions	Decision	Comment
Types of Entities	What kind of entity is that? Is this an event or an object?	-	General types of ontology elements clear and not in question.
Participation	Which objects do participate in this event?	+	Competency Questions only: 2.2: Who (object) has participated in industrial projects (events) in avionics engineering?
SimpleOrAggregated	What elements are aggregated members of this object?	+	Complete specification: e.g. requirement 1.6: aggregation of courses taught by expert

5.3. Filter by shared conceptualizations

An automated matching has not been tested. However, there were no identical IRIs in the patterns and the ExpertFinder ontology. Even just looking into the patterns, the same conceptualizations where represented by different IRIs. Manual interpretation by reviewing the OWL representations of the patterns, the “Solution Description”, and the “Scenarios” did not lead to a reduction of candidate patterns because there were at least overlaps of the conceptualizations.

An exemption forms the “Template Instance” pattern, which does not introduce new conceptualizations in addition do RDF basics. It adds an annotation “Template” and describes how to model individuals with recurring property values as templates based on that annotation. In general, it is applicable if there are such individuals in the ontology either before or after restructuring based on other implemented ODPs. The goal of the “Template Instance” pattern is a reduction of ontology size. This is different from the other ODPs. The question of the purpose of ODP implementation pops up again.

Exemplary decisions (- = neglected, + = passed):

Pattern	Conceptualization(s)	Decision	Comment
Collection	Collection	+	“EducationalProgramme” is a specialization of “Collection”

5.4. Filter by compatibility

Manual evaluation of conceptualizations left 14 candidates. Four of them have no competency questions given. Compared to the 18 patterns without competency questions in step 2, it seems that a lot of effort could have been saved if there were competency questions for filtering.

Exemplary decisions (- = neglected, + = passed):

Pattern	Conceptualization(s)	Decision	Comment
Criterion	Description	-	Contains specialization of "Description" which does not fit to the purpose of the ontology
Collection	Collection	+	"EducationalProgramme" is a specialization of "Collection". There are no further conceptualizations in the pattern.

5.5. Final pattern selection and implementation

A closer look into the remaining patterns revealed that some were incompatible with each other. Some are specializations or inclusions of other patterns. Also there is for example a basic pattern building block "TimeIndex" that is recurring in several patterns but is not listed as a pattern itself.

Information about the relationships between the patterns can save a lot of effort at this step. A specialization of an incompatible pattern or an inclusion of it will generally be also incompatible. At the end, five patterns remained. Only one of them, namely "Nary participation", was among the patterns, found in step 2 considering solely a comparison of competency questions. Therefore, the restriction of requirements filtering to competency questions has to be seen critically.

The "Time indexed Participation" pattern for example describes basically the same as the "Nary Participation" pattern but uses a different conceptualization and structure which is due to abstraction level discrepancies. Thus, only one of the patterns can be implemented. The "Nary Participation" pattern seemed to be less complex and had been chosen. The "Agent Role" pattern is a specialization of conceptualizations in the "Object Role" pattern. Finally, the following patterns have been selected:

- "Nary Participation": It can be used to describe the participation of experts in projects and courses. It contains the patterns or pattern building blocks "Participation", "Situation", "Time interval".
- "Collection": It can be used to describe the courses within an educational programme.
- "Classification": It can be used to classify "Course", "Project", and "Publication" by "ResearchField"
- "Persons": It can be used to express the relation of the "Expert" to "Position" and "UniversitySchool". It contains the patterns or pattern building blocks "AgentRole", "Classification", "Description".Altkough, "Description" is used to provide a definition to a "SocialPerson", such information is not part of the original ExpertFinder ontology. Therefore, this conceptualization will be left out and an adaptation of "Persons" is used.
- "Topic": It can be used to model the relations between "ResearchField" instances more comprehensive.

Steps for the actual implementation of patterns in ontologies are described for example in [14] and [13]. These steps would follow the pattern selection, but they are not in our focus.

6 Conclusions

In general, it has been proven that appropriate patterns for ontology quality improvement can be found in a structured way by the application of the proposed strategy. However, some problems were evident and there is a lot of space for improvements .

A first step on side of the provision of ODPs would be a better documentation of patterns. Several patterns missed information like domain, intent, and competency questions. Additionally, some standard for domain description would be helpful. Since there are close dependencies between ODPs like specialization and inclusion, an overview or a formal description of these dependencies is of interest. This idea has also been proposed quite similarly in [12]. Regarding the incompatibilities of the ODPs it should be investigated whether a harmonization is possible or some patterns may be neglected because they base on similar conceptualizations or incompatibilities should be part of pattern descriptions in conjunction with recommendations which pattern to use in what scenario. Maybe a restriction to a smaller set of basic patterns would be helpful too. This basic patterns would be generally smaller in size than patterns that combine several sub-patterns as they are present now. In consequence, less dependencies have to be considered and understandability of patterns increases. Another problem were unclear conceptualizations. For example, what exactly is an "object" in the context of a pattern, what a "concept"? A definition of such terms as part of the pattern documentation is suggested.

Looking at the process of pattern selection there is little potential for automatization. A semi-automatic process may include a search for domains and conceptualizations by IRIs or labels in conjunction with information about synonyms and related terms.

For example, Sabou et al. discussed such approaches in [15]. An approach to formalize competency questions and use them for automated pattern selection is seen critical. As shown earlier, competency questions do not need to cover the ontology scope completely. Furthermore, the level of abstraction of formulated competency questions varies depending on the ontology engineer. This phenomenon can also be observed looking at the competency questions of ODPs. Some steps of the pattern selection will always have to be done manually.

A question that is still open and cannot be answered in this paper is, what actual quality improvement is achieved by use of patterns. Some patterns like "Template instance" aim at computational aspects, namely space requirements, but what do the others aim for? What if ODPs are incompatible with standard ontologies as described in [13]. Is the ODP a bad pattern in this case or not? What are the preferences of the ontology engineer – being compatible with standards or achieve whatever quality improvement by the pattern?

References

1. Gruber, T. (1993) A translation approach to portable ontology specifications. In Knowledge Acquisition, vol. 5, pages 199–220, 1993.
2. Gangemi, A. (2005) Ontology design patterns for semantic web content. In The Semantic Web ISWC 2005, Vol. 3729, Lecture Notes in Computer Science. Springer, 2005.
3. Blomqvist E., Sandkuhl K. (2005) Patterns in Ontology Engineering – Classification of Ontology Patterns. Proc. 7th International Conference on Enterprise Information Systems, Miami, USA, May 2005.
4. Blomqvist, E. (2009) Semi-automatic Ontology Construction based on Patterns. PhD thesis, Linköping University, Department of Computer and Information Science, 2009.
5. Gangemi, A. and V. Presutti (2009) Ontology design patterns. In Handbook on Ontologies, 2nd Ed., International Handbooks on Information Systems. Springer, 2009.
6. Vrandečić, D., Sure, Y., 2007. How to Design Better Ontology Metrics, in: Franconi, E., Kifer, M., May, W. Eds., The Semantic Web: Research and Applications. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 311–325.
7. Duque-Ramos, A., Fernandez-Breis, J.T., Stevens, R., Aussenac-Gilles, N., 2011. OQuaRE: A SQuaRE-based approach for evaluating the quality of ontologies. Journal of Research and Practice in Information Technology 43, 159.
8. Maedche, A., Staab, S., 2002. Measuring Similarity between Ontologies, in: Gómez-Pérez, A., Benjamins, V.R. Eds., Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 251–263.
9. Brank, J., Grobelnik, M., Mladenić, D., 2005. A survey of ontology evaluation techniques, in: Proceedings of the Conference on Data Mining and Data Warehouses (SIGKDD 2005). Ljubljana, Slovenia.
10. Haslhofer, Bernhard; Klas, Wolfgang (2010): A survey of techniques for achieving metadata interoperability. In *ACM Comput. Surv* 42 (2).
11. Natalya F. Noy and Deborah L. McGuinness (2001): Ontology Development 101: A Guide to Creating Your First Ontology. In *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880*.2001
12. Hammar, Karl (2012): Ontology Design Patterns in Use - Lessons Learnt from an Ontology Engineering Case. In Eva Blomqvist, Aldo Gangemi, Karl Hammar, Mar\'ıa del Carmen Suárez-Figueroa (Eds.): Proceedings of the 3rd Workshop on Ontology Patterns, Boston, USA, November 12, 2012: CEUR-WS.org (CEUR Workshop Proceedings, 929).
13. Hammar, K., Tarasov, V. & Lin, F. (2010). Information Reuse and Interoperability with Ontology Patterns and Linked Data: First Experiences from the ExpertFinder Project. In: Witold Abramowicz, Robert Tolksdorf, Krzysztof Wecel (Ed.), Business information system workshops: BIS 2010 International Workshops, 3rd Workshop on Information Logistics and Knowledge Supply in conjunction with 13th International Conference on Business Information Systems (pp. 168-179). Berlin: Springer, LNBIP 57
14. Presutti, Valentina; Gangemi, Aldo (2008): Content Ontology Design Patterns as Practical Building Blocks for Web Ontologies. In Qing Li, Stefano Spaccapietra, Eric S. K. Yu, Antoni Olivé (Eds.): Conceptual Modeling - ER 2008, 27th International Conference on Conceptual Modeling, Barcelona, Spain, October 20-24, 2008. Proceedings: Springer (Lecture Notes in Computer Science, 5231), pp. 128–141.
15. Sabou, Marta; Lopez, Vanessa; Motta, Enrico (2006): Ontology Selection for the Real Semantic Web: How to Cover the Queen’s Birthday Dinner? In Steffen Staab, Vojtech Svátek (Eds.): Managing Knowledge in a World of Networks, 15th International Conference, EKAW 2006, Podebrady, Czech Republic, October 2-6, 2006, Proceedings: Springer (Lecture Notes in Computer Science, 4248), pp. 96–111.