

# Ontology Reuse

## *Experiences from Ontology Design Pattern Selection and Integration*

Birger Lantow<sup>1</sup>, Kurt Sandkuhl<sup>1,2</sup> and Vladimir Tarasov<sup>2</sup>

<sup>1</sup>University of Rostock, 18051 Rostock, Germany

<sup>2</sup>School of Engineering, Jönköping University, P.O. Box 1026, 551 11 Jönköping, Sweden

**Keywords:** Ontology Design Pattern (ODP), Ontology Alignment, ODP Selection, ODP Integration, Ontology Engineering, Ontology Reuse.

**Abstract:** While the main purpose of Ontology Design Patterns (ODPs) is to support the process of ontology engineering, they can also be used to improve existing ontologies. This paper has a focus on ODP selection and integration for ontology improvement. Based on the case of the ExpertFinder ontology, which allows for competency description of researchers, selection and integration of ODP is investigated with an explorative view. The current state of ODP selection strategies is discussed and problems arising during integration of ODP are shown. On this base, suggestions for improvements are made. Although this study deals with the integration into an existing ontology, most of the assumptions and suggestions are also valid for the general case of ODP usage.

## 1 INTRODUCTION

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 (2005) and introduced by Blomqvist and Sandkuhl (2005). Blomqvist defines the term as “a set of ontological elements, structures or construction principles that solve a clearly defined particular modelling problem“ (Blomqvist 2009). 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.

Gangemi and Presutti (2009) discuss the 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 modelling 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 (<http://ontologydesignpatterns.org>) and the logical ODPs maintained by the University of Manchester (<http://www.gong.manchester.ac.uk/odp/html/>).

The contributions of this paper are (1) a discussion of approaches for integrating ODP during ontology engineering, (2) experiences from the selection and integration of ODP in the ExpertFinder

case, and (3) recommendations regarding documentation and management of ODP. The paper extends previous work on ontology selection strategies (Lantow et al., 2013).

The remaining part of the paper is structured as follows: Section 2 discusses strategies for ODP integration. Section 3 describes the process of ODP selection and integration on the case of the ExpertFinder ontology (see Lantow et al., (2013)). The final section 4 summarizes the experiences and gives recommendations for a better documentation and management of ODP.

## 2 APPROACHES FOR ODP INTEGRATION

While the selection of ODP has already been intensively discussed in earlier work (Lantow et al., 2013), approaches to ODP integration are in focus of this work. This forms the base for a practical evaluation of selection and integration approaches in section 3 as well as a view on the influences between the steps of ODP usage.

After selecting an ODP, it has to be integrated into the ontology under construction in its current state. Generally, there are two approaches to ODP integration. First, the ODP can be considered as a guideline how to model certain aspects of a domain. This view on ODP is used by the ontology engineers in Hammar (2012). The selected ODP were only used to give an orientation for modelling. The OWL code provided with the ODP was not applied, but the given structures were remodelled manually. This approach has certain drawbacks:

1. Time savings by using ODP as predefined building blocks cannot be achieved.
2. Existing alignments of ODP cannot be used because they are bound to the specific OWL implementation (IRIs)
3. Considering ODP as tested, best practice components, the manual remodelling may introduce errors
4. Some semantic information of the patterns is not included in their documentation but in the OWL parts; there may be anonymous class definitions or additional axioms. Often, these are not fully considered, when doing manual remodelling.

Nevertheless, this approach fosters the knowledge about the consequences of ODP integration by the ontology engineers.

The second approach aims at the benefits of having ODPs as building blocks for ontology

creation. The steps for using the predefined OWL implementations for integration are first described in (Presutti and Gangemi, 2009) and included in the XD-Method for ontology engineering (Presutti et al. 2009). According to XD the integration consists of a first step of importing a selected ODP into the ontology under construction. Then four cases can be distinguished (Presutti and Gangemi, 2009):

1. Precise Matching: The ODP is directly usable.
2. Broader Matching: The ODP intent is more general than the local domain problem. In this case, there should be a search for more specialized ODP or a specialization step for the imported ODP is necessary.
3. Narrower Matching: The ODP intent is more specific than the local domain problem. In this case, there should be a query for more general ODP or a generalization step for the imported ODP is necessary.
4. Partial Matching: The ODP intent only partially matches the local domain problem. The ontology specification has to be decomposed in order to find parts that completely match ODP in the repository. This way several ODP have to be imported (specialized) and composed.

The task of composing ODP will commonly be a step after import/specialization because it is unlikely to have one ODP covering all requirements. Thus, either several ODP need to be imported or parts of the ontology stem from other sources. Composition can be done by the use of relations (Object Properties) and additional concepts in order to connect the parts of the ontology. Furthermore, specialization is seen as a task which commonly has to be performed on ODP integration. Since ODP should fit to a wide range of applications regarding their characteristics as patterns, they tend to be more general than local domain ontologies.

A last step seen in conjunction with ODP integration is test and fix. Presutti et al., (2009) suggest defining unit tests. First, the competency questions (CQ) are transformed into formal queries on the ontology (e.g. using SPARQL); second, some facts are added; and finally the queries are executed as a unit test. The unit tests are repeated with each integration step.

The XD toolset provides automated support for ODP integration. It contains an ODP browser, a wizard for ODP integration by specialization and a tool for quality assurance which helps to discover, among other issues, usability and reasoning problems by checking the existence of annotations and disjointness statements.

Another approach using predefined ontology constructs as building blocks for ontology creation stems from Iannone et al., (2009). However, this approach does not support the direct use of OWL-coded ODP. Instead, Ontology Pre-Processing Language (OPPL) is used to code the ODP. OPPL is a declarative manipulation language for ontologies. Thus, the necessary ontology changes to integrate an ODP can be described. The approach provides two benefits. First, OPPL code can be used to document the ODP usage in an ontology. Second, the consistent use of the ODP can be assured by using OPPL statements for instance creation. Like OWL-coded ODP, OPPL patterns are subject to specialization, generalization and composition but in contrast OPPL patterns themselves have to be changed by these operations. So far, no way has been described how to document these changes. This and the lack of tool support regarding the reuse of already existing OWL-coded patterns are the reasons for not considering OPPL further on.

### 3 EXPERIENCES IN ODP SELECTION AND INTEGRATION

This section reports on experiences in ODP selection (section 3.1) and ODP integration (section 3.2) and derives recommendations. The experiences are based on the ExpertFinder task as described in Lantow et al., (2013).

#### 3.1 ODP Selection

Selection of ODP for improving the ExpertFinder ontology has been subject of studies before. Hammar et al. (2010) suggest using the “Information Realization“ and the “N-ary Participation” pattern. Internal course material at Rostock University considers the construction of an ontology that fulfils the requirements of the ExpertFinder task based on “Information Realization”, “Participant Role”, “Time indexed Participation”, and “Topic” pattern. However, the process of ODP selection is not documented in the mentioned sources. Lantow et al. (2013) fill the gap and describe the selection process of ODP for improving the ExpertFinder ontology. The process followed the steps shown in Lantow et al., (2013):

1. Filter by Domain: No directly matching domain was discovered in the repository. The domains “General”, “Parts and Collections”, and

“Management” were considered as partly fitting. The domain filtering led to a reduction of ODP candidates from 97 to 72.

2. Filter by Requirements: ODP “Intent” and “Competency Questions” have been matched against the general requirements of the ExpertFinder ontology. Concentrating just on the CQ of the ExpertFinder ontology did not seem worthwhile because of the big difference between the abstraction levels. The number of candidate ODP was reduced to 35.
3. Filter by shared Conceptualizations: No direct matches were identified. However, considering ODP “Scenarios” and “Solution Description” revealed at least overlapping conceptualizations with the ExpertFinder ontology for all candidate ODPs. Thus, the number of candidate ODP remained 35.
4. Filter by compatibility: Manual comparison of candidate ODP and the ExpertFinder ontology reduced the number of candidates to 14. Checking the compatibility of the patterns themselves revealed that some of them were semantically incompatible and others were identified as integrated parts of other ODP. At the end of the selection process, five candidate patterns remained: “N-ary Participation”, “Collection”, “Classification”, “Persons”, and “Topic”.

The three independent attempts to select suitable ODP for use in the ExpertFinder task (Hammar et al., 2010, Rostock University course material, Lantow et al., 2013) obviously gave quite different results. However, the ODP catalogue and the ontology requirements were the same. This calls for further work on selection strategies. In general, there are patterns for roughly the same purpose that can be used alternatively. This is also mentioned by (Blomqvist et al., 2009) in a general study of pattern use. Hammar (2012) also emphasizes the importance of pattern dependencies for pattern selection. However, the author does not clarify what dependencies should be considered.

Based on the ExpertFinder case and the assumptions from the general work on ODP use, the following recommendations can be made in order to improve support for pattern selection:

1. Assure complete ODP description within the given ODP metadata schema (several ODP are not well documented (Hammar, 2012) (Lantow et al., 2013)). Although CQ are missing only in few cases, additional requirements may be derived by the use of other documentation items

- (step 2 of the selection process by Lantow et al., (2013)) and fitting ODP may be identified.
2. Provide a taxonomy of pattern domains. The result would be a controlled vocabulary for the domain description. Furthermore, sub-trees in the taxonomy may be excluded early in the ODP selection process. This would reduce the effort spent in step 1 of the pattern selection process proposed by Lantow et al., (2013).
  3. Provide specialization and composition relations between patterns. This can help to identify incompatibilities and helps to find specializations/generalizations. This would reduce effort spent in step 4 of the pattern selection process. Additionally, effort for ODP integration (see section 3.2) may be reduced since finding a more specialized ODP possibly avoids ODP specialization in the ODP integration step.
  4. Name alternative Patterns and provide information for decision between alternatives. There are several patterns that fit the same requirements regarding the knowledge to be covered. The diversity of patterns that have been proposed by the different sources for the ExpertFinder task proves this. For a purposeful selection, additional requirements have to be considered (world view, compatibility, query performance, etc.)
  5. Include descriptions of common mistakes in patterns use (Hammar, 2012)
  6. Structure the pattern catalogue by architecture tiers (Hammar, 2012)

The suggestions above are a starting point for an improved support for (semi-automatic) pattern selection. However, as stated in Lantow et al., (2013), additional aspects could also be taken into account. There are for example potential quality improvement and required effort for pattern integration. This in addition with the question for current support of ODP integration leads to the discussion of experiences from ODP integration.

### 3.2 ODP Integration

Previous work on ODP integration presented experiences only on a very coarse level. Blomqvist et al., (2009) report generally an increase in usability due to annotations, a better coverage of ontology requirements regarding satisfied CQ and a decrease of domain coverage regarding terminology. Hammar et al., (2010) describe an incompatibility of the “N-ary Participation” pattern with other ontologies that

have been integrated for reuse in the ExpertFinder case. Some of the concepts were overlapping and the pattern had to be modified for integration.

At a knowledge modelling tutorial at Rostock university “Time-indexed participation” and “Participant Role” ODP were chosen for integration in the ExpertFinder case. This revealed that both share some object properties with equivalent semantics but different names. Four “Equivalent to” statements had to be added:

“Event included in”  $\equiv$  “is event included in”  
 “Object included in”  $\equiv$  “is object included in”  
 “Object participating”  $\equiv$  “includes object”  
 “Participating in event”  $\equiv$  “includes event”

Furthermore, a specialization of both patterns in order to combine their characteristics had to be created. Specialization in this case required also the consideration of anonymous super classes. While the “Time-indexed participation” is an n-ary relation of at least 3 concepts it should be at least 4 concepts if a role is added.

After having these experiences from previous studies and practical application of ODP in the ExpertFinder task, a more detailed qualitative assessment of the integration task has been conducted. We separated the pattern selection and the pattern integration part. Furthermore, the goal was to gain some knowledge regarding the integration of ODP in existing ontologies. The patterns resulting from the proposed pattern selection strategy (see section 3.1) have been provided to an ontology engineer who had created an ontology for the ExpertFinder task without ODP before. This expert had to perform the integration task based on the methodology described in section 2. He was familiar with the methodology and the ontology he had created. Thus, we were able to combine the findings with those of previous studies and to avoid bias resulting from poor knowledge of domain and methodology.

The classes of the resulting ontology including the later ODP integration are shown in figure 1. A semi-structured interview has been performed afterwards in order to evaluate the support and possible problems in the integration step of ODP use. The interview contained the following questions and results:

*Q1: How did the ODP documentation support the integration-process of the ODP?*

So far, ODP documentation has only been considered for ODP selection methodology. However, documentation in software engineering is also important for maintenance and integration.

Thus, it is interesting whether this is also true for ODP use.

The expert generally mentioned the big differences between ODP regarding quality and completeness of documentation. Graphical visualizations of the ODP have been considered helpful for ODP integration. They usually help to grasp the intention of the pattern much faster. However, the diagrams did not contain enough information regarding concept identification. While concepts of different ODP had the same label in their visualization, they had different IRIs in their OWL implementation. This required additional attention in the integration process (see below).

*Q2: Was it possible to use the ODP-templates?*

This question aimed at the general applicability of the proposed XD approach of ODP integration (see section 2).

The expert was able to import all suggested ODP OWL implementations from the repository to the ontology.

*Q3: Were changes to the ODP necessary? What changes? Why? How many?*

This question aims at exceptions from the XD integration steps. Generally, changes to the ODP OWL implementations contradict the approach. ODP integration should be possible by just adding alignment information using specialization (generalization) and composition (see section 2).

The Persons ODP had to be changed for import. There were contradictions caused by the pattern based on the knowledge base. The problem was the following statement:

```
persons:SocialPerson : [persons:actsThrough
owl:cardinality 1 owl:Thing]
```

It was either inconsistent with the functional data properties or led to inferences as (same individuals):

```
expert_OhgA owl:sameAs expert_LiFe, expert_LunM
```

This does not reflect the real world. The following fix had to be implemented:

```
persons:SocialPerson : [persons:actsThrough
owl:minCardinality 1 owl:Thing]
```

Another problem was the use of different IRIs in ODP for the same concepts. The following statements had to be added:

```
description:Concept owl:equivalentClass
classification:Concept
```

```
topic:Concept owl:equivalentClass classification:Concept
```

A third problem that occurred was the

incompatibility of the Topic and the Classification patterns. Both patterns have been identified as suitable to reflect research fields. Making the ResearchField class a specialization of Topic:Topic and Classification:Concept, lead to an inconsistent ontology because both classes are defined as disjoint. Three case categories can be identified based on the described necessary changes:

1. ODP axioms are incompatible with the constructed ontology (case 1)
2. ODP are not aligned with each other. (case 2 and 3)
3. ODP are incompatible themselves. (case 4)

The integration of 5 ODP resulted in 4 problem cases that had to be solved by changes and additions to the original ODP OWL implementations. Figure 1 shows the inheritance cycles for the differently coded Concept classes. In this case the ODP use even results in a decrease of ontology quality.

*Q4: What new relations/object properties had to be introduced in the new Ontology besides the ones that were already in ODP and ExpertFinder? Why? How many?*

This question aims at the specialization (generalization) and composition steps of the XD methodology. These steps generally lead to additional relations besides those directly imported as ODP. In the case of an already existing ontology just additional inheritance may be necessary. Furthermore, in order to use the expressivity of ODP new object property assertions may have to be made for existing instances.

According to the Expert, no additional object properties had to be defined. Integration has been done by ODP specialization:

- Collection was added as a superclass of EducationalProgramme.
- classification:concept was used as a superclass of ResearchField. classification:classifies was used on the instances of ResearchField to relate them to the instances of Degree, Expert, and Projects.
- Expert was made a subclass of persons:NaturalPerson, Position and UniversitySchool - subclasses of persons:SocialPerson. persons:actsFor related experts to positions as well as positions to the school.
- Project and Course were made subclasses of participation:Event. Instances of projects/courses, experts and time intervals were related through participation:participationIncludes.

Overall, 7 new inheritance relations and 265 object

property assertions have been added.

*Q5: What new classes had to be introduced in the new Ontology besides the ones that were already in ODP and ExpertFinder? Why? How many?*

This question aims at the specialization (generalization) and composition steps of the XD methodology. It might be necessary to create new classes here. In the case of an already existing ontology just additional inheritance may be enough since existing classes may serve as specializations of ODP classes for example. New class definitions are only needed as long as appropriate classes are not yet defined. Seldom, the composition step requires new class definitions (see section 2).

In the ExpertFinder case no new classes had to be added.

*Q6: What new instances had to be introduced in the new Ontology besides the ones that were already in ODP and ExpertFinder? Why? How many?*

Considering the possibility to discover new requirements during ODP selection as described in Lantow et al., (2013), it might be necessary to add new instances to an existing knowledge base. This question tries to assess the validity of this assumption.

When the Classification ODP was added, 173 ResearchField instances had to be added as well. The taxonomy of research fields had been created by a class hierarchy but the ODP required instances in the place of the used classes.

When the Persons ODP was added, four Expert instances and two Position instances have been added too. These instances did not exist in the ontology before. By adding them the ontology semantics became more fine-grained.

When the n-ary Participation ODP was added, new instances of TimeInterval for projects and courses have been created. This was necessary to represent the time-related semantics aspects in a better way.

Two case categories for ODP use induced instance creation can be identified:

1. ODP lead to new requirements and thus new knowledge has to be collected in the field. (general case)
2. ODP lead to a different representation of already captured knowledge (classification case)

The number of additional instances highly depends on the domain. Little dependency to the existing ontology structure or the used ODP is assumed for the general case. Thus, the number of new instances is not considered in this qualitative study.

*Q7: What obstacles did you encounter?*

The earlier questions have been derived from the description of the existing methods for ODP selection and integration. This last question aims at collecting issues that might not be covered this way.

No new aspects revealed based on the expert's answers.

The interview only reflects the experiences of one ontology engineer. However, it already indicates potential areas for improving the process of ODP integration. Including previous studies on the area of ODP integration the following suggestions can be made:

1. In addition to the better documentation of patterns as proposed in section 3.1, ODP in the repository can be improved if equivalent classes and properties are coded equivalent or identically and if compatibility with certain "standard" ontologies is assured. The answer to Q3 of the interview showed that there were some equivalence relations missing in the ODP. The resulting effort for equivalence and alignment discovery for the ODP user can be avoided. Furthermore the goal of ontology quality improvement is better supported. Inheritance cycles or unnecessary complexity, respectively, can be avoided.
2. Documentation of ODP can be further improved if equivalent parts of several ODP are made explicit. This way the ODP user is aware of equivalent concepts and ODP parts. A better understanding of the ODP structure is assumed. This revealed to be problematic looking into the answers to Q1 and Q3.
3. The integration step includes in addition to specialization (generalization) and composition also ontology alignment and adaptation of ontology axioms. For example, cardinalities or anonymous classes need to be considered. It has been shown that the integration steps as described in the XD method are not enough. Axioms and anonymous classes have to be considered as well regarding reasoning of new facts and consistency. This problem showed in the knowledge modelling tutorial and in the answer to Q3 as well.
4. The test step should not only consider querying but also consistency. The case study has shown that ODP use may also lead to inconsistencies (see answer to Q3). These can hardly be discovered just by SPARQL queries.
5. Tools should allow for integration of ODP parts (see also Hammar (2014)) since sometimes parts of the ODP are already covered by existing classes.



Figure 1: ExpertFinder Ontology after ODP integration - Inheritance Structure.

6. More integration related aspects can be included into pattern selection. For example: necessity of additional instances in order to make ODP more usable or number of axioms that have to be evaluated during ODP integration.

## 4 CONCLUSION

Although this work is rather explorative, it covers recent approaches to ODP reuse and shows important challenges for a broad use of ODP in ontology engineering based on practical experiences. Furthermore, potential areas for improvement are discussed. With a rising number of ODP available in the repository, automatic or semi-automatic pattern selection becomes more important. So far existing approaches provide only little help.

Improvements regarding recall have been achieved (e.g. by Hammar (2014)) but precision is poor. However, if more and more ODP have a similar domain and similar requirements, the number of alternatively usable ODP increases. Thus, precision might not be that important in the future. Structured ODP selection processes might help to guide the ontology engineer. However, improved tool support and improved ODP documentation is needed. Improved ODP documentation includes using the current metadata schema more extensively and also the addition of new metadata, like pattern compatibility (cf. section 3.1 and 3.2). Also an

improvement of the patterns themselves regarding compatibility with existing ontologies, the treatment of equivalent classes and properties as well as the fitness of ODP axioms for reuse needs to be addressed.

The structure of the interview can be used for further investigations on ODP use and ontology reuse in general.

## REFERENCES

- 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.
- Blomqvist, Eva (2009) Semi-automatic Ontology Construction based on Patterns. PhD thesis, Linköping University, Department of Computer and Information Science, 2009.
- Blomqvist, Eva; Gangemi, Aldo; Presutti, Valentina (2009): Experiments on pattern-based ontology design. In: *K-CAP*, pp. 41–48.
- 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.
- Corcho, O., Roussey, C., Vilches-Blázquez, L. M., & Perez Dominguez, I. (2009). Pattern-based OWL ontology debugging guidelines. *CEUR Workshop Proceedings*.
- 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.
- 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.
- Gangemi, Aldo; Catenacci, Carola; Ciaramita, Massimiliano; Lehmann, Jos (2005): A theoretical framework for ontology evaluation and validation. In: *SWAP*, vol. 166..
- Gangemi, A. and V. Presutti (2009) Ontology design patterns. In *Handbook on Ontologies, 2nd Ed., International Handbooks on Information Systems*. Springer, 2009.
- 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
- Hammar, Karl (2012): Ontology Design Patterns in Use - Lessons Learnt from an Ontology Engineering Case. In Eva Blomqvist, Aldo Gangemi, Karl Hammar, Maria 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).
- Hammar, Karl (2014): Ontology Design Patterns: Adoption Challenges and Solutions Web Enterprise Adoption and Best Practice and *Second International Workshop on Finance and Economics on the Semantic Web Co-located with 11th European Semantic Web Conference, WaSABi-FEOSW@ESWC 2014*, Anissaras, Greece, May 26, 2014.
- Hong, Tzung-Pei; Chang, Wen-Chang; Lin, Jiann-Horng (2005): A Two-Phased Ontology Selection Approach for Semantic Web. In Rajiv Khosla, Robert J. Howlett, Lakhmi C. Jain (Eds.): *Knowledge-Based Intelligent Information and Engineering Systems*, vol. 3684: Springer Berlin Heidelberg (Lecture Notes in Computer Science), pp. 403–409. Available online at [http://dx.doi.org/10.1007/11554028\\_56](http://dx.doi.org/10.1007/11554028_56).
- Iannone, L., Rector, A. & Stevens, R. (2009). Embedding knowledge patterns into owl. In *The Semantic Web: Research and Applications* (pp. 218-232). Springer Berlin Heidelberg.
- Lantow, Birger; Sandkuhl, Kurt; Tarasov, Vladimir (2013): Selecting Content Ontology Design Patterns for Ontology Quality Improvement Knowledge Supply and Ontologies in Information Systems, Warsaw, Poland, September 23rd, 2013.
- 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.
- Mikroyannidi, E., Manaf, N. A. A., Iannone, L. & Stevens, R. (2012). Analysing Syntactic Regularities in Ontologies. In *OWLED* (Vol. 849).
- Poveda-Villalón, M., Suárez-Figueroa, M. C. & Gómez-Pérez, A. (2012). Validating ontologies with OOPS!. In *Knowledge Engineering and Knowledge Management* (pp. 267-281). Springer Berlin Heidelberg.
- 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.
- Presutti, Valentina; Daga, Enrico; Aldo Gangemi; Blomqvist, Eva (2009): eXtreme Design with Content Ontology Design Patterns with the *8th International Semantic Web Conference (ISWC-2009)*, Washington D.C., USA, 25 October, 2009.
- 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.
- 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.