

# DC Proposal: Towards an ODP Quality Model

Karl Hammar

Jönköping University, Jönköping, Sweden  
karl.hammar@jth.hj.se

**Abstract.** The study of ontology design patterns (ODPs) is a fairly recent development. Such patterns simplify ontology development by codifying and reusing known best practices, thus lowering the barrier to entry of ontology engineering. However, while ODPs appear to be a promising addition to research and while such patterns are being presented and used, work on patterns as artifacts of their own, i.e. methods of developing, identifying and evaluating them, is still uncommon. Consequently, little is known about what ODP features or characteristics are beneficial or harmful in different ontology engineering situations. The presented PhD project aims to remedy this by studying ODP quality characteristics and what impact these characteristics have on the usability of ODPs themselves and on the suitability of the resulting ontologies.

## 1 Research Problem

In spite of the development of several well-defined ontology construction methods, efficient ontology development continues to be a challenge. One reason for this is that such work requires both extensive knowledge of the domain being modeled, and a sufficient understanding of ontology languages and logical theory. Ontology Design Patterns (ODPs) are considered a promising contribution to this challenge [2]. Such patterns encode best practices, which helps reduce the need for extensive experience when developing ontologies. Additionally, by promoting a limited set of best practice outlooks on common problems, patterns help harmonize how these problems are viewed, supporting interoperability of ontologies developed using them.

Previous study into the state of ODP research indicates that while patterns are being used as tools in the ontology research community, and while there are patterns being developed and presented, patterns are rarely studied as artifacts of their own. The amount of work done on evaluation of patterns and pattern development and usage methods is limited [10]. There exists no established theory or model of ODP quality, and consequently there is no way of telling with certainty what is a good pattern for solving a particular type of problem based on pattern features. We will within the presented PhD project attempt to remedy this situation, by studying what quality characteristics or features of Ontology Design Patterns that are beneficial in different types of ontology engineering situations, and inversely, what such features or characteristics that could be considered harmful.

To this end, three research questions have been established:

1. Which quality characteristics of ODPs can be differentiated, and how can these be measured?
2. Which quality characteristics of ODPs affect the suitability of the resulting ontologies for different uses?
3. How do the quality characteristics of ODPs affect one another?

The remainder of this paper is structured as follows: Section 2 covers some other work that has been published on the topic of ontology evaluation, conceptual model quality, and design patterns. Section 3 presents what we hope this PhD project will contribute to the research community. The state of the work so far is presented in Sect. 4, hypotheses based on this tentative research is presented in Sect. 5, and the road ahead is mapped out in Sect. 6.

## 2 Related Work

### 2.1 Ontology Design Patterns

The use and understanding of ontology design patterns has been heavily influenced by the work taking place in the FP6 NeOn Project, the results of which include a pattern typology [14] and the eXtreme Design collaborative ontology development methods, based on pattern use [3]. While this view is influential and the NeOn typology is referenced frequently, it is not the only perspective on patterns - for instance, Blomqvist [1] presents a different typology based on the level of abstraction and granularity of the reusable solution.

eXtreme Design (XD) is defined as "*a family of methods and associated tools, based on the application, exploitation, and definition of ontology design patterns (ODPs) for solving ontology development issues*" [13]. The XD approach to selecting patterns is based on the pattern containing a written description of the type of problem for which the original pattern developer considers it appropriate (the *Generic Use Case*), that the ontology developer can match against his/her modelling problem (in XD parlance the *Local Use Case*). Additionally, to find candidate patterns for a given problem, search against pattern keywords can be performed.

### 2.2 Ontology Evaluation

While the amount of work on ODP evaluation is limited, there are quite a few methods and frameworks proposed for the evaluation of ontologies that may be relevant for ODPs also. The semiotic metaontology  $O^2$  is used as a basis for instantiating *oQual*, a proposed ontology for the evaluation and selection of ontologies for a given task [6]. These two ontologies are complemented by and used in the *QoodGrid* methodology of ontology selection [7]. The *QoodGrid* framework is detailed and proposes some useful measures and formalizations of properties, but it has not been tested extensively. Also, it is unknown how well suited this complex framework would be for small reusable solutions.

One of the most cited works in ontology evaluation is [9], introducing *Onto-Clean*. This method tests whether an ontology is consistent with the real world it is supposed to model by applying a number of formal metaproperties (*essence, identity, unity*, etc.) to the concepts in the ontology and then checking for any inconsistencies in how these metaproperties are instantiated.

### 2.3 Conceptual Model Quality

It seems reasonable that we in this PhD project also consider established knowledge regarding the quality of other types of conceptual models, such as UML or ER models. While the such models may differ from ODPs in terms of computational functionality and logic formalisms used, research performed on them may still give guidance on how humans interpret and understand graphical and text representation of ODPs, i.e. the semiotics of conceptual computer models.

Genero et al. [8] present a simple experiment on the effect of structural complexity in ER models to the understandability and modifiability of such models. While the complexity measures used are not directly transferrable, the experimental method and test measures used are very applicable to our case. The difference between objective/quantitative metrics and subjective/qualitative ones are emphasized by Moody & Shanks [12], who also work with ER models. In [12] we also find an important discussion of the effects of various metrics/qualities on one another (for instance, the negative correlation between completeness of a model and implementability of that same model).

Lindblad et al. [11] present a discussion on the quality of more logically abstract models. They emphasize that models should be minimal in order not to overconstrain systems and users, while at the same time being *feasible complete*, i.e. that they should contain all the statements describing the domain such that the cost of adding the statement does not outweigh the benefit it brings to the model (a kind of scoping of the problem). It should be interesting to consider how this view of appropriate model size is impacted by the open world assumption of the semantic web. Lindblad et al. also stress the importance of comprehensibility of models requiring support for visualization and filtering.

## 3 Contribution

The ODP selection method proposed in XD [13] is appropriate for finding patterns that has been described as satisfactorily solving a particular problem from a larger set of patterns. It may also be possible to automate, provided that an appropriate logical vocabulary for describing *Local Use Cases* and *Global Use Cases* is developed. It does however not guide the user in selecting, from a given set of functionally appropriate patterns, the one that is best suited for use in their situation. The right choice then could depend on non-functional requirements on the ontology as a whole (expandability, performance, testability, etc), or it could depend on quality attributes of the pattern itself (how easy is it to apply, how it is documented, is there an example ontology using it, etc). An ODP quality

model should in this scenario guide the developer in selecting patterns to use that, apart from solving the functional requirements of their modeling problem, also has features and qualities that are appropriate and helpful to them.

We also anticipate that the quality model resulting from this PhD work will be helpful in creating or extracting new high quality ODPs for a variety of purposes. While there exists already some work on these topics (primarily in [14]), this work is described in terms of specific technical issues, dealing with how to go about specializing patterns, generalizing existing models, and reengineering from other types of logic models. The evaluation which features that ought to be present in an ODP for a particular purpose is left with the ontology engineer. Providing this ontology engineer with a defined quality model may help them in developing better ODPs.

It can be argued that ontologies are similar enough to OOP inheritance hierarchies that an understanding of object-oriented design pattern quality should be sufficient also when constructing ontologies, and that the specific study of ODPs is therefore not required. We find this comparison and argument to be flawed, for two reasons. To begin with, OOP design patterns are used in the modeling and production of information systems only, whereas ontologies have a much wider applicability as shared conceptualizations for various purposes, and therefore are often more general in nature. Secondly, OOP languages/designs are not based in description logic and do not make use of classification and inferencing as ontologies do. It is our opinion that such capability is of key importance in many ontology usage scenarios, such as document classification, situation recognition, constraint modeling, etc. Consequently, ontologies are different enough from object-oriented designs that developing an understanding of quality as it applies to ontology design patterns specifically is a worthy contribution.

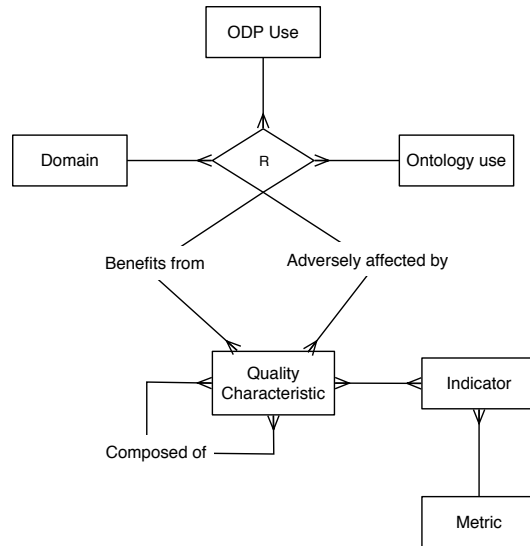
## 4 Current State

At the time of writing, the PhD project is just over one year into its four year runtime. The time elapsed so far has been spent on familiarizing the author with semantic technologies and their applications, establishing that there is a need for this research (see [10]), initiating some small-scale initial experiments, and developing a metamodel for representing ODP qualities.

### 4.1 Quality Metamodel

The ODP quality metamodel is illustrated in Fig. 1. The purpose of this metamodel is to systematize how we discuss and think about ODP quality, to establish a vocabulary and theory that experiments can build upon and observations relate to. It is obviously subject to change as the PhD project proceeds.

The topmost half of the figure displays the relation  $R(D, ODP_U, OU)$  where  $D$  denotes a domain,  $ODP_U$  denotes an ODP Use (*ontology engineering, ontology matching, transformation*, etc), and  $OU$  denotes a use to which the deliverable result of the ODP usage is put.



**Fig. 1.** Proposed Quality Metamodel

An instantiation of this relation  $R$  could be said to form a usage example of ODPs. For each such instantiation  $R$  a number of quality characteristics affect the result, for better or for worse. Some quality characteristics are abstract, consisting of compositions of other quality characteristics. Quality characteristics are mapped to concrete indicators that are measurable using some metric.

## 4.2 Initial data gathering

For the purpose of eliciting hypotheses, some small scale experiments have been performed. The XD methodology has been used in the scope of an internal project dealing with expert finding in the academic domain. The problem was decomposed according to XD methodology and a number of patterns from the ontologydesignpatterns.org repository were used to iteratively develop solutions for the problems, integrating the solutions at the end of each iteration. The end result was a rather complex ontology, highly heterogeneous in terms of annotation quality and density.

Several interviews have been staged with master student groups using ODPs in the development of ontologies used within master thesis projects. From these interviews it was learned that when selecting patterns students prioritized high quality documentation, both in terms of annotations on the example pattern instantiation (if one is provided), and in terms of written usage examples and instructions. They also preferred smaller patterns as they were easier to "wrap one's head around", however, the patterns should not be as small as to be trivial.

## 5 Hypotheses

Based on the initial data gathering efforts discussed in Sect. 4.2, a number of hypotheses regarding the influence of certain ODP characteristics on a resulting ontology have been developed, as detailed in Table 1. For formal definitions of the structural quality characteristics mentioned, see [5]. These attributes are not yet aligned to the previously presented quality metamodel. The belief is that it would be wiser to collect data proving or disproving their usability before instantiating a quality model based on them, since this data collection process may result in a changed understanding of ODP quality that necessitates changing the underlying metamodel also.

**Table 1.** ODP quality characteristics influences on resulting ontology.

<b>ODP quality characteristic</b>	<b>Resulting ontology feature/characteristic</b>
Small size	Low average depth, high breadth
Small size	High MO count & high cycle count
Small size	Variable density
Small size	Varying annotation quality
Small size	Complex result
Small size	Poor computational performance

All of these first hypotheses deal with the issue of size, i.e. the number of classes and properties in an ODP. Essentially, the intuition and experience from the initial small experiments is that if one uses smaller patterns, one has to use more of them to cover a particular problem or domain. By using more patterns, the total complexity of the end result will increase, as each pattern-based module will be integrated via subsumption links to many of the other small pattern-based modules that make up the ontology. If one instead used larger patterns, each such module would in itself hold a certain set of functionality, and fewer links would need be added to integrate the modules. This complexity in integration is also believed to lead to a higher subsumption hierarchy cycle count, which is detrimental to computational performance.

Some further hypotheses presented in Table 2 deal with the impact of particular features on the comprehensibility and reusability of the ODPs themselves. As indicated by the performed interviews with students, smaller patterns are easier to understand. We also believe that using a simpler logical language with less advanced constructs (i.e. RDFS rather than OWL) will help in understanding the ODP. Further, we say that a highly abstract pattern likely will be more general in nature and thus more reusable in many different scenarios. At the same time, it will be more difficult to understand and apply than a more concrete and well defined pattern. Finally, minimalism (defined as fulfilling the requirements of one problem and containing no extraneous statements) is believed to make a pattern easier to understand and applicable in a greater set of scenarios.

**Table 2.** ODP quality characteristics impact on ODP comprehensibility/reusability.

ODP quality characteristic	Impact on ODP
Small size	High comprehensibility
Simple language	High comprehensibility
High abstraction level	High reusability
High abstraction level	Low comprehensibility
Minimalism	High comprehensibility
Minimalism	High reusability

## 6 The Road Ahead

Two more experiments using bachelor and master students have been initiated and will be completed during the fall. In each of these experiments, two groups of students have been provided with two sets of patterns, with which to solve a set of ontology engineering tasks. The first set of patterns are representative of the patterns available in the [ontologydesignpatterns.org](http://ontologydesignpatterns.org) repository. The second set have been constructed based on solutions in [4]. The latter are generally larger and more complex, but also more thoroughly described. An analysis of the resultant ontologies will be performed to study the consequences of these differences. Also, interviews with the student groups will be performed to gauge the perceived comprehensibility of the two sets of patterns.

Later in the fall we will work on a project that deals with situation recognition using input from a number of sensor subsystems. We have been granted access to a reference implementation of a particular system for this purpose (not using ontologies), as well as the design documentation and specifications for the system. The goal is build an ontology-based equivalent using ODPs in the construction of the system ontology. Key considerations in the ontology development process are computational performance of the system, high documentation quality, and the ability to reconfigure the ontology depending on available sensor subsystems or situation recognition contexts.

Testing how well OntoClean works on ODPs looks to be an interesting experiment - on the one hand, ODPs are intended to be general solutions, and as such, OntoClean metaproperties might not apply cleanly to the more general concepts they contain. On the other hand, ODPs are small enough that it in terms of effort expended would be quite simple to apply the OntoClean metaproperties to them (applying OntoClean to a full ontology is generally considered to be a rather time-consuming process). We hope to be able to test such use of OntoClean on ODPs during the spring and if the results of this work are positive, will attempt to integrate OntoClean metaproperties and constraints into our quality model.

Finally, discussions are underway with an industry partner where we hope to be able to test the usability of ODPs in an information logistics context. The scenario involves a large set of heterogeneous production data that needs to be integrated and made easily searchable. This is a type of scenario where it is common to see ontologies used as shared vocabularies. Such ontologies need not

be very complex, as they are not subject of much reasoning and classification. However, they need to be flexible and general, in order to accommodate the rapid growth of input data. For these reasons, this project should be an interesting contrast to the previously mentioned situation recognition project.

## References

1. Blomqvist, E.: Semi-automatic ontology construction based on patterns. Ph.D. thesis, Department of Computer and Information Science, Linköpings universitet (2009)
2. Blomqvist, E., Gangemi, A., Presutti, V.: Experiments on pattern-based ontology design. In: Proceedings of the fifth international conference on Knowledge capture. pp. 41–48. ACM (2009)
3. Daga, E., Blomqvist, E., Gangemi, A., Montiel, E., Nikitina, N., Presutti, V., Villazon-Terrazas, B.: NeOn Deliverable D2.5.2 Pattern based ontology design: methodology and software support. NeOn Project. <http://www.neon-project.org> (2008)
4. Fowler, M.: Analysis patterns: reusable object models. Addison-Wesley (1997)
5. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Ontology evaluation and validation: an integrated formal model for the quality diagnostic task. Tech. rep., Laboratory for Applied Ontology, ISTC-CNR, Roma/Trento (Italy) (2005)
6. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Modelling ontology evaluation and validation. *The Semantic Web: Research and Applications* pp. 140–154 (2006)
7. Gangemi, A., Catenacci, C., Ciaramita, M., Lehmann, J.: Qood grid: A metaontology-based framework for ontology evaluation and selection. In: Proceedings of the EON'2006 Workshop (2006)
8. Genero, M., Poels, G., Piattini, M.: Defining and validating measures for conceptual data model quality. In: *Advanced Information Systems Engineering*. pp. 724–727. Springer (2006)
9. Guarino, N., Welty, C.: Evaluating ontological decisions with OntoClean. *Communications of the ACM* 45(2), 61–65 (2002)
10. Hammar, K., Sandkuhl, K.: The State of Ontology Pattern Research: A Systematic Review of ISWC, ESWC and ASWC 2005–2009. In: *Workshop on Ontology Patterns: Papers and Patterns from the ISWC workshop* (2010)
11. Lindland, O., Sindre, G., Sølvberg, A.: Understanding quality in conceptual modeling. *IEEE software* pp. 42–49 (1994)
12. Moody, D., Shanks, G.: What makes a good data model? Evaluating the quality of entity relationship models. *Entity-Relationship Approach—ER'94 Business Modelling and Re-Engineering* pp. 94–111 (1994)
13. Presutti, V., Daga, E., Gangemi, A., Blomqvist, E.: eXtreme Design with Content Ontology Design Patterns. In: *Proceedings of the Workshop on Ontology Patterns (WOP 2009)*, collocated with ISWC (2009)
14. Presutti, V., Gangemi, A., David, S., de Cea, G., Surez-Figueroa, M., Montiel-Ponsoda, E., Poveda, M.: NeOn Deliverable D2. 5.1. A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies. NeOn Project. <http://www.neon-project.org> (2008)