Classification of Web-Based Ontology Building Method Guidelines: a Case Study

Sari Hakkarainen, Darijus Strasunskas, Lillian Hella, Stine Tuxen

Norwegian University of Science and Technology,
Sem Sælands vei 7-9, NO-7491 Trondheim, Norway
{sari, dstrasun, hella, stinemt}@idi.ntnu.no

Abstract. Ontology is the core component in semantic Web applications. The employment of an ontology building method affects the quality of ontology and the applicability of ontology language. A weighted evaluation approach for ontology building guidelines is presented in this paper. The evaluation criteria are based on an existing classification scheme of a semiotic framework for evaluating the quality of conceptual models. A sample of Web-based ontology building method guidelines is evaluated in general and experimented with when using data from a case study in particular. Directions for further refinement of ontology building methods are discussed.

Keywords: semantic web, ontology building methods, quality evaluation

1 Introduction

The vision for the next generation web is the *semantic Web* [1], where information is accompanied by metadata about its interpretation, so that more intelligent and more accessible information-based services can be provided. The core components in the semantic Web and its applications will be the ontologies. An ontology can be seen as an explicit representation of a shared conceptualization [9] that is formal [25], and will encode the semantic knowledge and enable sophisticated information services. The quality of a semantic Web application will be highly dependent on the quality of its underlying ontology. The quality of the underlying ontology will in turn depend on factors such as 1) the appropriateness of the language used to represent the ontology and 2) the quality of the engineering environment, including tool support and method guidelines, as provided for creating the ontology by means of that language.

There are also situated factors, such as the complexity of the specific task at hand and the expertise of the persons involved. With a small number of developers the need for rigid method guidelines may be smaller than for larger projects. Similarly, with highly skilled modelling experts, the need for method guidelines may be smaller than for less experienced people. Method guidelines can thus be seen as an important means to make ontology creation possible for a wider range of developers, e.g., not only a few expert researchers in the ontology field but also companies wanting to develop semantic Web applications for internal or external use.

Method guidelines can provide homogeneous instructions for creation of ontologies in a federated ontology engineering environment. However, the current situation is that while many ontology representation languages have been proposed, there is much less to find in terms of method guidelines for how to use these languages – especially for the newer Web-based ontology specification languages. Similarly, if there is little about method guidelines for Web ontology building, there is even less about evaluating the appropriateness of these method guidelines.

The objective is to inspect available method guidelines for semantic Web-based ontology specification languages. The approach is to adapt the method classification part of a model quality framework [14], define a computational framework for the analytic evaluation of method guidelines and conduct an experiment in a case study.

The outline is as follows. Section 2 describes related work. Section 3 describes weighting method for seven categories in the classification framework. Section 4 classifies the selected method guidelines. Section 5 analyses their means to achieve quality goals in general and compared to the industrial case in particular. Finally, Section 6 concludes the paper and suggests directions for future work and for further refinement of ontology building methods.

2 Ontology Building Support and Evaluation Methods

During the last decade, a number of ontology representation languages have been proposed. The so-called traditional ontology specification languages include: CycL, Ontolingua, F-logic, CML, OCML, Telos, and LOOM. There are Web standards that are relevant for ontology descriptions for semantic Web applications, such as HTML, XML and RDF. Finally, there are the newer Web ontology specification languages such as XOL and SHOE, and those that are based on the layered architecture for the semantic Web, such as OIL, DAML+OIL, and OWL. The latter group of the so-called semantic Web enabling languages for ontology building is in the focus of this study.

There exist several methodologies to guide the process of Web ontology building that vary both in their level of generality and granularity. Some of the methodologies describe an overall ontology development process yet not the ontology creation itself. Such methodologies are primarily intended to support the knowledge elicitation and management of the ontologies in a basically centralised environment. [7] proposes an evolving prototype methodology with six states as ontology life-cycle and includes activities related to project management and ontology management. [22] proposes an application driven ontology development process in five steps emphasizing the organisational value, integration possibilities and the cyclic nature of the development process. [23] proposes a top-down approach for deriving domain specific ontologies from common upper level ontologies and includes steps for requirements elicitation and for implementing the derived ontologies. [24] proposes a general framework for the ontology building process consisting of four steps including quality criteria for ontology formalisation.

The above methodologies provide a life cycle in an overall ontology development process as analysed in [2, 6, and 24] but only a few user guidelines for carrying out the steps and for actually creating the ontology. In order to increase the number and the scale of practical applications of the semantic Web technologies, the developers need to be provided detailed instructions and general guidelines for the actual ontology creation. A limited selection of method guidelines were found for the Web ontology specification languages [4, 13, 17], which are at the foci of this study. [4] presents a user guide with method guidelines for making ontologies in the representation language DAML+OIL, again by means of Protégé. [13] presents a tutorial containing method guidelines for making ontologies in the representation language OWL by means of the open source ontology editor Protégé. [17] presents method guidelines for making ontologies, called “Ontology Development 101”. Unlike the other two, this method is independent of any specific representation language.

As for evaluation of ontology specification approaches, a comprehensive evaluation of representation languages was done in [21], covering all the languages mentioned above except OWL. The paper also evaluates some tools for ontology building: Ontolingua, WebOnto, WebODE, Protégé 2000, OntoEdit, and OilEd. Similarly, [3, 8, 19] evaluate various ontology languages. These studies concentrate on evaluating the representation languages (and partly tools), not hands-on instructions or ontology building guidelines. Given the argumentation above, such studies are targeting the audience of highly skilled modelling experts rather than the wide spectrum of potential developers of semantic Web applications.

The semiotic quality framework proposed in [14] builds on an earlier framework [15]. This early version distinguished between three *quality categories* for conceptual models (syntactic, semantic, and pragmatic) according to steps on the semiotic ladder [5]. The *quality goals* corresponding to those categories were syntactic correctness, semantic validity and completeness, and comprehension (pragmatic). The framework also took care to distinguish between goals and *means to reach the goals* (where, e.g., various types of method guidelines would be an example of the latter). In later extensions by Krogstie, more quality categories have been added, so that the entire semiotic ladder is included, e.g., *physical*, *empirical*, *syntactic*, *semantic*, *pragmatic*, *social*, and *organizational quality*.

Since our evaluation is based on the method classification part of the framework of [14], it is most closely related to previous work using that same framework [10, 11], and especially the evaluation of ontology languages and tools in [21]. In this paper the framework is used for evaluating something different, namely method guidelines for ontology building. Moreover, an interesting question is to which extent it is suitable for this new evaluation task, so customizations to the framework are suggested in order to improve its relevance for evaluating method guidelines in general, and method guidelines for ontology building in particular.

3 Criteria Weight Computation for Seven Semiotic Categories

As argued in the introduction above, the developers typically need instructions and guidelines for ontology creation in order to support the learning and co-operative deployment of the semantic Web enabling languages in practice. [14] describes a methodology classification framework consisting of seven semiotic categories of modelling methodologies. We adapt the categories for classification of the ontology building method guidelines [10] and suggest selection criteria and coverage weight function for them. The principle modification here is that the concept of application system (as the end product of the development process) is consequently replaced by ontology (as the end product of applying the method guidelines). The experiences from the case study [11] suggested that numerical values could be used for the classification and thus qualify weighted selection techniques such as the [16] PORE methodology. Therefore, we adapt PORE methodology here and define the coverage weights -1, 1 and 2 for each category. The method guidelines are classified accordingly in the next section.

Let *CF* be a classification framework such that *CF* has a fixed set *Ç* of categories *ç*, where *Ç* = {$ç_1$, $ç_2$, $ç_3$, $ç_4$, $ç_5$, $ç_6$, $ç_7$} and $ç_i \in Ç$. Each *ç* is a quadruple <*id*, *descriptor*, *C*, *cw*>, where *id* is the name of the category, *descriptor* is a natural language description, *C* is a set of selection criteria *c*, and *cw* defines a function of *S* that return -1, 1, or 2 as coverage weight, where *S* is a set of satisfied elements *c* in the selection criteria *C* of each category in *Ç*. Intuitively, we define a number of selection criteria alongside an associated coverage weight function for each category in the classification framework. The categories are as follows.

$ç_1$ - Weltanschauung describes the underlying philosophy or view to the world. For a method guideline we examine why the ontology construction is addressed in a particular way in a specific methodology. In accordance with the FRISCO report [5], three views can be identified: the *objectivistic view*, i.e. reality exists independently of any observer, where the relation between reality and the model is trivial or obvious, the *constructivistic view*, i.e. the reality exists independently of any observer, where observer possesses only a restricted mental model and the relationship between reality and models of this reality are subject to negotiations among the community of observers and evolve, and the *mentalistic view*, i.e. reality and the relationship to any model is totally dependent on the observer we can only form mental constructions of our perceptions. Weltanschauung can be $ç_1c_1$ – *explicit*, i.e. stated in the document, $ç_1c_2$ – *implicit*, i.e. derivable from the documentation, or $ç_1c_3$ – *undefined*, i.e. non derivable.

$$cw_1(S_1) = \begin{cases} 2, & if \quad ç_1c_1 \in S_1. \\ 1, & if \quad ç_1c_2 \in S_1. \\ -1, & if \quad ç_1c_3 \in S_1. \end{cases} \qquad \textbf{(1)}$$

$ç_2$ - Coverage in process concerns the method's ability to address $ç_2c_1$ – *planning for changes*, $ç_2c_2$ – *single and co-operative development* of ontology or aligned ontologies, which includes analysis, requirements specification, design, implementation and testing, $ç_2c_3$ – *use and operations* of ontologies, $ç_2c_4$ – *maintaining and evolution* of ontologies, and $ç_2c_5$ – *management of planning, development, operations and maintenance* of ontologies.

$$\mathrm{cw}_2(S_2) = \begin{cases} -1, & if \quad |S_2| = 0. \\ 1, & if \quad 0 < |S_2| \le 2. \\ 2, & if \quad 2 < |S_2| \le 5. \end{cases} \qquad \textbf{(2)}$$

$ç_3$ - Coverage in product is described as the method concerns planning, development, usage and maintenance of and operate on $ç_3c1$ – *one single ontology*, $ç_3c_2$ – *a family of related ontologies*, $ç_3c_3$ – *a whole portfolio of ontologies* in an organization, and $ç_3c_4$ – *a totality of the goals, business process, people and technology* used within the organization.

$$cw_3(S_3) = \begin{cases} -1, & if \quad |S_3| = 0. \\ 1, & if \ 0 < |S_3| \le 2. \\ 2, & if \ 2 < |S_3| \le 4. \end{cases} \quad (3)$$

ç$_4$ - Reuse of product and process support reuse of ontologies as products or reuse of method as processes in order to avoid re-learning and recreation. There are six dimensions of reuse: *ç$_4$c$_1$ – Reuse by motivation*. For example productivity, timeliness, flexibility, quality, and risk management goals. *ç$_4$c$_2$ – Reuse by substance.* A product is the set of deliverables that are produced during a project, such as models, documentation and test cases. Reusing a development or maintenance method is process reuse. *ç$_4$c$_3$ – Reuse by development scope*. Scope may be either external or internal to a project or organization. *ç$_4$c$_4$ – Reuse by management mode*. Reuse may be planned in advance with existing guidelines and procedures, or ad-hoc. *ç$_4$c$_5$ – Reuse by technique*. Reuse may be compositional and/or generative. *ç$_4$c$_6$ – Reuse by intentions*. The elements may be used as they are, slightly modified, used as a template or just used as an idea.

$$cw_4(S_4) = \begin{cases} -1, & if \ 0 < |S_4| \le 2. \\ 1, & if \ 2 < |S_4| \le 4. \\ 2, & if \ 4 < |S_4| \le 6. \end{cases} \quad (4)$$

ç$_5$ - Stakeholder participation reflects the interests of different actors in the ontology building activity. The stakeholders may be categorized into those *ç$_5$c$_1$ – responsible for developing* the method, those with *ç$_5$c$_2$ – financial interest* and those who have *ç$_5$c$_3$ – interest in its use*. Further, there are different forms of participation. *Direct* participation means every stakeholder has the opportunity to participate. *Indirect* participation uses representatives, thus every stakeholder is represented through other representatives that are supposed to look after their interests.

$$cw_5(S_5) = \begin{cases} -1, & if \quad |S_5| = 0. \\ 1, & if \ 0 < |S_5| \le 1. \\ 2, & if \ 1 < |S_5| \le 3. \end{cases} \quad (5)$$

ç$_6$ - Representation of product and process can be based on linguistic and non-linguistic data such as audio and video. Representation languages for both product and process can be *ç$_6$c$_1$ – informal*, *ç$_6$c$_2$ – semi-formal* or *ç$_6$c$_3$ – formal*, having a logical or executional semantics.

$$cw_6(S_6) = \begin{cases} -1, & if \ |S_6| = 1. \\ 1, & if \ |S_6| = 2. \\ 2, & if \ |S_6| = 3. \end{cases} \quad (6)$$

ç$_7$ - Maturity is characterized on different levels of completion. Some methodologies have been used for a long time; others are only described in theory and never tried out in practice. Several conditions influence maturity of a method, namely if the method is $ç_7c_1$ – *fully described*, if the method lends itself for $ç_7c_2$ – *adaptation, navigation and development*, if the method is $ç_7c_3$ – used and updated through *practical applications*, if it is $ç_7c_4$ – *used by many organizations*, and if the method is $ç_7c_5$ – *altered* based on experience and scientific study of its use.

$$cw_7(S_7) = \begin{cases} -1, & if \quad |S_7| = 0. \\ 1, & if \ 0 < |S_7| \leq 3. \\ 2, & if \ 3 < |S_7| \leq 5. \end{cases} \quad (7)$$

The selection criteria are exhaustive and mutually exclusive in the categories $ç_1$, and $ç_6$, exhaustive in $ç_5$, whereas the set of satisfied criteria S of the remaining categories may also be the empty list {}. The coverage weight *cw* is independent of any category-wise prioritisation. Since the intervals are decisive for the coverage weight they can be adjusted depending on preferences of the evaluator. However, when analysing different evaluation occurrences the intervals need to be fixed in comparison, but may be used as dependent variable.

4 Method Guidelines for Ontology Building - General Coverage

Three method guidelines among the semantic Web-based ontology specification languages are categorized, namely *Denker, 2003* [4] which is based on DAML+OIL and Protégé, *Knublauch et al., 2003* [13], which is based on OWL and Protégé, and *Noy and McGuinness, 2001* [17] which is language independent yet uses Protégé in the examples. Protégé2000[1] is an open-source ontology editor developed at Stanford University and uses Java technology. All the method guidelines meet the selection criteria as supporting semantic Web applications and assume RDF/XML notation rather than HTML or plain XML as the underlying Web standard. The studied method guidelines are shortly described and characterised in the sequel.

Denker, 2003 is a user's guide of the DAML+OIL plug-in for Protégé2000. The ontology building method is based on DAML+OIL language and Protégé as the ontology development tool. The ontology building process consists of three basic steps; create a new ontology, load existing ontologies, save ontology. The creation of new ontology consists of five types of instructions; define classes, properties (slots), instances, restrictions, and Boolean combinations.

Comment: The method does not contain any explicit description of the development process. However, the sequence of the sections in the documentation gives an indication of how to create an ontology.

Knublauch et al., 2003 is a tutorial that was originally created for the 2nd International Semantic Web Conference. The ontology building method is based on OWL language and assumes Protégé as the ontology development tool. The ontology building process consists of seven iterative steps, namely determine scope, consider re-

[1] Hereafter abbreviated Protégé as in http://protege.stanford.edu/

use, enumerate terms, define classes, define properties, create instances, and classify ontology.

Comment: The development activity requires some experience and foresight, communication between domain experts and developers, and a tool that is both comprehensible and powerful, including support for ontology evolution.

Noy and McGuinness, 2001 is a guide to building ontologies, called Ontology Development 101. The ontology building method is language and ontology development tool independent yet it uses Protégé in the examples. The ontology building process consists of seven iterative steps, namely determine the domain and scope of the ontology, consider reusing existing ontologies, enumerate important terms in the ontology, define the classes and the class hierarchy, define the properties of classes – slots, define the facets of the slots, and create instances.

Comment: The methodology provides three fundamental rules, for making development decisions, namely that 1) there is no single correct way to model a domain, that 2) ontology development is necessarily an iterative process, and that 3) concepts in the ontology should be close to objects, physical or logical, and to relationships in the domain of interest.

The classification of the selected method guidelines into the semiotic categories of [14] is summarized in Table 1. The rows of the table summarize the above classification criteria as they are met by the studied method guidelines. The cells in the second column of the table describe how the method meets the criteria in numeric terms. The table here indicate only an intermediate result, however. Later on, in Section 5 we will describe how the categorization can be applied in practice. The aim is to support the decision making process when looking for the most appropriate method guideline in a particular organization, rather than ranking the methods.

In summary, the discriminating classification criteria between the studied method guidelines are in 1) *weltanschauung*, where the world view is explicit (constructivistic) in two of the method guidelines and undefined for DAML+OIL-Tutorial, in 2) *coverage in process*, where none of the guidelines are fully complete, but DAML+OIL-Tutorial is the least complete, in 3) *reuse of product and process*, where DAML+OIL-Tutorial only mentions the support of import functionality, whereas the other two include reuse as one step in the building process, where Ontology development 101 provides more description and functionality, and in 4) *maturity*, where Ontology development 101 is the most mature. None of the method guidelines are even close to complete concerning *coverage in product* whereas all of them cover *representation of product and process* on a good or medium level.

Table 1. Classification of method guidelines

<table>
<tr><th>Evaluation criteria</th><th>Category name</th><th>Coverage weight</th></tr>
<tr><th colspan="3">Explanation</th></tr>
<tr><td colspan="3">Tutorial: DAML+OIL with Protégé (DAML+OIL-Tutorial)</td></tr>
<tr><td>ς_1</td><td>Weltanschauung</td><td>-1</td></tr>
<tr><td colspan="3">Undefined. The method does not explicitly state its worldview and it is not possible to implicitly deduce the worldview.</td></tr>
<tr><td>ς_2</td><td>Coverage in process</td><td>1</td></tr>
<tr><td colspan="3">The method contains no explicit description of the development process yet the sequence of the sections in the documentation indicates how to proceed in order to create an ontology. The importance of reuse is not covered and it does not describe how to plan for changes. The evolution and use of Protégé are</td></tr>
</table>

Evaluation criteria	Category name	Coverage weight
Explanation		
described. The coherence between the development tool and the ontology language is considered.		
ς_3	**Coverage in product**	**1**
A single ontology. However, it describes situations where the user would like to import concepts created in another ontology. The method does not allow references to resources located in another ontology except for four explicitly stated URIs[2].		
ς_4	**Reuse of product and process**	**-1**
Considers only technical aspect of reuse and describes only import of DAML+OIL files.		
ς_5	**Stakeholder participation**	**-1**
The tutorial is available through the Artificial Intelligence Center at SRI International, and is linked through the DAML homepage. The physical editor(s) author(s) are unknown other than the contact person regarding the plug-in and the user guide.		
ς_6	**Representation of product and process**	**-1**
The document is basically written in natural language on top of screenshots that explain the ontology building method with Protégé. The user participant does not need to be aware of the underlying syntax of the ontology language.		
ς_7	**Maturity**	**2**
The tutorial is based on DAML+OIL as ontology language, released in December 2000. It has been subject for evaluation. Protégé is used by a large community and is a well-examined system. The method is not complete. The method guideline describes the uncovered or unimplemented functionalities.		
OWL- Protégé tutorial (OWL-Tutorial)		
ς_1	**Weltanschauung**	**2**
Constructivistic. The first step in the development method is to determine the scope. By doing that, the domain that is to be covered in the ontology will be explicitly stated. The method states that communication between domain experts and developers is necessary.		
ς_2	**Coverage in process**	**2**
Defines seven iterative steps. It has a detailed yet unstructured and incomplete description of ontology development. The first three steps: determine scope, consider reuse and enumerate terms, are just mentioned. The tool guidance does not follow the steps in the building process, but is presented rather ad hoc. There are no explicit procedures to prepare for changes.		
ς_3	**Coverage in product**	**1**
Protégé is described as a toolset for constructing ontologies that is scalable to very large knowledge bases and enables embedding of standalone applications in the Protégé knowledge environment. It does not describe the relationship between heterogeneous ontologies, nor the requirements the tool should fulfill prior to use in larger context.		
ς_4	**Reuse of product and process**	**1**
The tutorial considers reuse partially in the ontology building activity. The *development scope* and *technical* prerequisite of reuse are covered, but not why, when or how to consider reuse. It does not provide examples of how reuse is carried out in practice. It describes how to import existing OWL files that are developed with another tool or developed with some previous version of Protégé. It lists formats from which ontologies may be read (imported), written to (exported) or inter-converted between.		
ς_5	**Stakeholder participation**	**2**
The tutorial is comprehensible for inexperienced stakeholders with development or financial interests and supports the interests of novice user participants. Since it is written by those *responsible for developing* the tool, the guide has a deep and detailed description of practical use. Several members of the user community, i.e. those who have *interest in its use*, have contributed to the method *indirectly* through material such as visualization systems, inference engines, means of accessing external data sources and user-interface features.		
ς_6	**Representation of product and process**	**1**
It is mostly *informal*, written in natural language yet presents a narrow description of the Semantic Web and ontologies. On the *visual* part it has a multitude of screenshots that explain and make the semi-		

[2] http://www.daml.org/2001/03/daml+oil#, http://www.w3.org/1999/02/22-rdf-syntax-ns#, http://www.w3.org/2000/01/rdf-schema#, and http://www.w3.org/2000/10/ XMLSchema#

<table>
<tr><th>Evaluation criteria</th><th>Category name</th><th>Coverage weight</th></tr>
<tr><th colspan="3">Explanation</th></tr>
<tr><td colspan="3">structured tool concepts and the formal language elements comprehensible. The development process is covered in a graphical representation yet not explained. In overall, the method is mostly informal and provides feasible graphical representation.</td></tr>
<tr><td>ς_7</td><td>Maturity</td><td>1</td></tr>
<tr><td colspan="3">The tutorial is based on OWL, the newest contribution in this field. The language itself has hardly been examined yet. However, guidance for OWL modeling benefits from experiences with guidelines for Protégé, RDF and OIL. The plug-in that is used in Protégé is also new, but the core Protégé is well-examined. The method covers the latest release, and is up-to-date in both regarding the language and the tool. The method is not complete, since not all the steps in the development process are fully described.</td></tr>
<tr><td colspan="3">Ontology development 101</td></tr>
<tr><td>ς_1</td><td>Weltanschauung</td><td>2</td></tr>
<tr><td colspan="3">Constructivistic. It presents a list of different reasons for creating an ontology, e.g. to make domain assumptions explicit. The method argues that an explicit specification is useful for new users.</td></tr>
<tr><td>ς_2</td><td>Coverage in process</td><td>2</td></tr>
<tr><td colspan="3">It covers seven iterative steps, each of which is described in detail. For example, there are several guidelines for developing a class hierarchy. This feature provides participants a checklist to avoid mistakes such as creating cycles in a class hierarchy. It has good coverage in process. Reuse is considered, but there is no plan for changes. The actual implementation of an ontology is not covered.</td></tr>
<tr><td>ς_3</td><td>Coverage in product</td><td>1</td></tr>
<tr><td colspan="3">The method is an initial guide to help creating a single new ontology. There is awareness of the possible integration to other ontologies and applications. Further, translating an ontology from one formalism to another is not considered a difficult task. However, instructions for this are not provided.</td></tr>
<tr><td>ς_4</td><td>Reuse of product and process</td><td>2</td></tr>
<tr><td colspan="3">It covers reuse in step 2. Reusing existing ontologies is a requirement if the system needs to interact with applications that have already committed to some ontologies. Reuse is not fully covered yet references to available libraries of ontologies are given.</td></tr>
<tr><td>ς_5</td><td>Stakeholder participation</td><td>1</td></tr>
<tr><td colspan="3">The method guideline provides introduction to ontologies and describes why they are necessary. The method is suitable for experienced as well as novice participants since it mainly uses informal languages, yet provides comprehensive descriptions.</td></tr>
<tr><td>ς_6</td><td>Representation of product and process</td><td>1</td></tr>
<tr><td colspan="3">It makes no explicit reference to any specific ontology language. It is written in natural language, with only a few logical or executable statements. The language is informal and the method offers adequate description of each concept. There are illustrations based on screenshots from Protégé to support comprehensibility. A semi-structured scenario is given and used as a reference throughout the guideline.</td></tr>
<tr><td>ς_7</td><td>Maturity</td><td>2</td></tr>
<tr><td colspan="3">Published in 2001. Many researchers in the field reference the method guideline, many readers examine it, and acknowledged Web sites such as the Protégé Web site provide hyperlinks to it. The method does not claim it has been tried out in practice, but by searching on the Web, several projects that use the method can be located. However, it has not been updated in response to such experiences.</td></tr>
</table>

5 Method Guidelines for Ontology Building – the edi Case

The case study is based on edi (engaging, dynamic innovation) which is a system developed by a student project group. edi is intended to support exchange of business ideas between the employees within a large Norwegian company, which is an integrated oil and gas company with business operations in 25 countries. At the end of 2002, there were over 15 000 employees in the company. Consequently, the amount of information and knowledge provided by the employees is rapidly increasing. There is an increasing need for more effective information retrieval and efficient sharing of knowledge. edi intended as idea management tool and a motivator for elicitation and generation of ideas, as well as for enabling the employees to focus on the relevant aspects of their activities.

The overall approach for the edi system is to create a connection for communication and knowledge sharing between employees from different business areas as well as domain experts and department managers. The current plan is to utilize semantic Web and Web service technology for that purpose. Ontologies will play a crucial part in the edi system, both in supporting common access to information and enabling implementation of Web and ontology-based search. The users will be experts on ontology building, on enterprise processes, on creativity and on processes that support creativity, all of which possess different qualities, modeling skills and domain knowledge.

edi requirements The current status is that the functional requirements for edi have been gathered systematically. However, before the system can be developed a thorough analysis need to be conducted, and a decision about the purpose of the ontology has to be made. Information about the domain plays an important role in this process and it can be gathered in various ways. Unavoidably, there will be many different participants involved in such a process; for instance end users such as possible idea contributors and people in the edi network for evaluating proposed ideas. This is analogous to software development in general [3], hence starting with ontology requirements analysis. The requirements specification should describe what the ontology must support, sketching the scope of the ontology application and identifying valuable knowledge sources. Oil industry is a business in constant change, and the large international coverage of the company makes the changes even more complex. edi needs to have high durability, be adaptable to changes in the environment, be maintainable and have high reliability in order to secure the investment. Thus, a careful analysis needs to be made early in the process, which places elaborate requirements on the ontology development environment.

Quality-based edi requirements An ontology should be built in a way that supports automatic reasoning and provides a basis for high quality Web-based information services. The underlying assumption is that a high quality engineering process assures high quality end product. The quality of ontology building process depends on the environmental circumstances under which the ontology is used. Further, a model is expected to have high degree of quality if it is developed according to its specification. Similarly, a method guideline is expected to have high quality degree if it describes a complete set of steps and instructions for how to arrive at a model, which is valid with respect to the language(s), it supports.

In the following, the requirements are categorized according to the categories of the classification framework [14] and the importance weights are calculated according to Eq. 8 as follows. Let R be a set of weighted requirements such that R has a fixed set $RÇ$ of categories $rç$, where categories in $RÇ$ are the same as in the fixed set $Ç$ of categories $ç$ of the classification framework CF, i.e. $RÇ = Ç$, and $ç \in Ç$, $rç \in RÇ$. $rç$ is a triple $<id, req\_descriptor, iw>$, id is the name of the category, $req\_descriptor$ is a natural language description of requirement, and $iw_{rç}$ defines a function of I that returns 1, 3, or 5 as importance weight based on priorities and policy of the company.

$$iw_{r\varsigma}(I) = \begin{cases} 1, & if\ r\varsigma \text{ may be satisfied, is optional,} \\ 3, & if\ r\varsigma \text{ should be satisfied, is recommended,} \\ 5, & if\ r\varsigma \text{ must be satisfied. is essential,} \end{cases} \quad \textbf{(8)}$$

We adopt PORE methodology [16] to prioritise the classification criteria based on edi requirements [11] in order to evaluate the ontology building guidelines in this particular situation. Table 2 shows weighted importance criteria based on edi requirements priority.

Table 2. Classification of edi requirements

Category of requirements	Category name	Importance weight
Description of requirements		
rç1	**Weltanschauung**	**3**
Constructivistic world view – however this is not a crucial requirement. The end users may have different models of the reality depending on for example, their geographical location or the business area in which they are involved.		
rç2	**Coverage in process**	**5**
Ontology building method for *edi* must be extensively covered to support large development teams and heavily illustrated to support inexperienced project participants.		
rç3	**Coverage in product**	**1**
Development of a single ontology in a stand-alone application may be supported.		
rç4	**Reuse of product and process**	**5**
Important, must be integrated in the process. Feasible guidance including illustrative examples should be provided. ontology building method for edi should provide feasible guidance including illustrative examples, and the procedures should be integrated into steps in the development process.		
Rç5	**Stakeholder participation**	**3**
Ontology building method for *edi* should cover the participants' development and financial interests of the involved creators of the method as well as the low experience of its user group participants.		
rç6	**Representation of product and process**	**3**
Informal (natural language) representation and rich illustration are important. Independent of the method, the language should cover the required level of formality in the product to support automated reasoning.		
rç7	**Maturity**	**3**
Ontology building method for *edi* should be widely adopted and well-examined in order to support evolution, co-operation and management of the ontology.		

In summary, table 2 shows that the key criteria for meeting edi requirements with high utility are *coverage in process*, *reuse of product and process*, and *representation of product and process*. The discriminating criteria are *coverage in process*, and *reuse of product and process* with the assigned importance weight equal to 5. The least discreminating criteria is coverage in product where the weight is equal to 1.

Finally, total coverage weights Tw_i for each ontology building guideline i are calculated. In table 3, we have summarized which of the studied ontology building methods that meet the situated, quality-based requirements for the edi system. Here, the importance weights from table 2 are multiplied by the coverage weights from table 1 and total weights calculated using equation (9) are used as overall feasibility rate for supporting the choice of ontology building guidelines.

$$Tw_i = \sum_{\varsigma \in Ç} (cw_{\varsigma} \times iw_{\varsigma}) \quad \textbf{(9)}$$

Table 3. Final evaluation of method guidelines according to importance of edi requirements

Evaluation criteria	Importance weight (*iw*)	DAML+OIL-Tutorial		OWL-Tutorial		Ontology development 101	
		Coverage weight (*cw*)	**Total**	Coverage weight (*cw*)	**Total**	Coverage weight (*cw*)	**Total**
ς_1	3	-1	**-3**	2	**6**	2	**6**
ς_2	5	1	**5**	2	**10**	2	**10**
ς_3	1	1	**1**	1	**1**	1	**1**
ς_4	5	-1	**-5**	1	**5**	2	**10**
ς_5	3	-1	**-3**	2	**6**	1	**3**
ς_6	3	-1	**-3**	1	**3**	1	**3**
ς_7	3	2	**6**	1	**3**	2	**6**
		$Tw_{DAML+OIL\ Tutorial}$:	**-2**	**$Tw_{OWL\ Tutorial}$:**	**34**	**$Tw_{OntDev101}$:**	**39**

In summary, table 3 colligates the situated evaluation in favor of Ontology development 101 with the total coverage weight $Tw_{OntDev101} = 39$. Next most relevant is OWL Tutorial with the score $Tw_{OWL\ Tutorial} = 34$. Moreover, out of the key requirements for edi, the discriminating criteria are *coverage in process*, and *reuse of product and process*. The Ontology development 101 tutorial meets the both criteria completely, and OWL Tutorial partially, whereas DAML+OIL-Tutorial has shortages in both cases. All the guidelines support *coverage in product* on the level as required for edi (*iw=1*) and support the *representation of product and process* in a range, where Ontology development 101 and OWL Tutorial meet the requirements completely, and DAML+OIL-Tutorial only partially. Out of the remaining categories of edi requirements DAML+OIL-Tutorial fails to meet any of them, OWL Tutorial meets two completely and fails in one, whereas Ontology development 101 meets two completely and one partially. Thus, according to our metrics Ontology development 101 seems most suitable to guide the edi ontology creation.

6 Concluding Remarks

An evaluation of three method guidelines for semantic Web ontology building was conducted using the [10, 14] framework. Evaluation of method guidelines was performed in two steps, one general evaluation, i.e. their applicability for building ontologies in general, and one particular, i.e. how appropriate are they for ontology development in a real world project - how applicable is the framework in practice. The main results are as follows.

— The method classification part of the framework [14] has potential for evaluating method guidelines. Use of the numerical values for the weights and adoption of the PORE methodology [16] produce the more explicit evaluation results.

- The categorization according to Weltanschauung, i.e. the applied modelling worldview, was expected to be the same for all the method guidelines, but turned out to be discriminating as selection criteria in the case study. However, the Weltanschauung most probably is the same for the studied guidelines, since they support languages, which all are constructivistic; it was merely not derivable for one of the guidelines.
- In both steps, in the general classification and in the evaluation against the situated requirements, the method "Ontology Development 101" [17] came out on top, since meeting most of the evaluation criteria. This was also the only method guideline, which is independent of any specific representation language and has the longest history.
- Major weaknesses were identified for all the methods, as expected because of the current immaturity of the field of Web-based ontology construction. None of the method guidelines are complete concerning *coverage in product* whereas all of them cover *representation of product and process* fairly well.

The contribution of this paper is twofold: First, an existing evaluation framework was tried out with other evaluation-objects than it has been used for previously; Second, numerical values and metrics were incorporated to the classification framework for the classification and thus supporting qualification of weighted selection. The experimental case study suggests that, given the small adjustments, the framework intended for model classification is applicable in evaluation of method guidelines regardless if the classification is used for their selection, quality assurance, or engineering.

The concrete ranking of methods may be of limited use, as new ontology languages and method guidelines are developed and the existing languages evolve and some of them became more mature. Nevertheless, it can be useful in terms of guiding the current and future creators of such languages and their method guidelines. Drawing attention to the weakness of current proposals, these can be mended in future proposals, so that there will be higher quality languages and method guidelines to choose from in the future. The underlying assumption for our work is that high quality method guidelines may increase and widen the range and scalability of the semantic Web ontologies and applications.

There are several interesting topics for future work, such as supplementing the theoretical evaluations with empirical ones as larger scale semantic Web applications arise utilizing the empirical nature of [14], as well as evaluating more methods as they emerge, e.g. [12, 18, 20]. Further possibilities are in investigating the appropriateness of the formalisation quality criteria in the [24] Unified methodology as a complement to the semiotic quality framework [15] in order to conduct evaluation of the process oriented methodological frameworks that were out of scope of this study.

References

1. Berners-Lee T., Handler J., Lassila O.: The Semantic Web. Scientific American, May (2001).
2. Corcho O., Fernández-López M., Gómez-Pérez A.: Methodologies, tools and languages for building ontologies: where is their meeting point?, Data & Knowledge Engineering, 46:1, pp 41 – 64, (2003).

3. Davies, I., Green, P., Milton, S., Rosemann, M.: Using Meta-Models for the Comparison of Ontologies. In: Proc. of EMMSAD'03, Velden, Austria, (2003).
4. Denker G.: DAML+OIL Plug-in for Protége 2000 – User's Guide. SRI International AI Center Report 7/8/03, (2003).
5. Falkenberg, E.D., Hesse, W., Lindgreen, P., Nilsson, B.E., Oei, H., Rolland, C., Stamper, R.K., van Asssche, F.J.M., Verrjin-Stuart, A., Voss, K.: FRISCO - A Framework of Information Systems Concepts. IFIP WG 8.1 Technical Report, (1997).
6. Fernández-López M.: Overview Of Methodologies For Building Ontologies. In: Benjamins, V.R., Chandrasekaran, B., Gómez-Pérez, A., Guarino, N., Uschold, M. (Eds.): Proc. of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5) Stockholm, Sweden, (1999).
7. Fernándes-López M., Gómez-Péres A., Juriso N.: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. Proc. of AAAI-97 Spring Symposium on Ontological Engineering. Stanford University, (1997).
8. Gómez-Péres, A., Corcho, O.: Ontology Languages for the Semantic Web. IEEE Intelligent Systems, 54-60 (2002).
9. Gruber T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition. 5(2), (1993).
10. Hakkarainen, S., Hella, L., Tuxen, S.M., Sindre, G.: Evaluating the quality of web-based ontology building methods: a framework and a case study. Proc. of 6th Intl. Baltic Conf. on Databases and Information Systems (Baltic DBIS'04), Riga, Latvia, (2004).
11. Hella, L., Tuxen, S.: An Evaluation of Ontology Building Methodologies - An analysis and a case study, TDT4730 Information Systems Specialization, Study Report, NTNU, November, 2003.
12. Knublauch, H.: Protégé OWL Tutorial, 7th Intl. Protégé Conf., Maryland, July 6-9, 2004
13. Knublauch H., Musen M.A., Noy N.F.: Creating Semantic Web (OWL) Ontologies with Protégé. Tutorial at 2nd Intl. Semantic Web Conf. Sanibel Island Florida USA, (2003).
14. Krogstie J.: Conceptual Modeling for Computerized Information System Support in Organizations, PhD Thesis 1995:87 NTH, Trondheim, (1995).
15. Lindland, O.I., Sindre, G., Sølvberg, A.: Understanding Quality in Conceptual Modeling. IEEE Software, 11(2): 42-49 (1994).
16. Maiden, N.A.M., Ncube, C.: Acquiring COTS Software Selection Requirements. IEEE Software March/April, pp 46-56 (1998).
17. Noy N.F., McGuinness D.L.: Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, (2001).
18. Pepper, S.: The TAO of Topic Maps - Finding the Way in the Age of Infoglut, Ontopia AS, Norway, 2004. URL: http://www.ontopia.net/topicmaps/materials/tao.html
19. Schuette, R.: Architectures for evaluating the quality of information models – a meta and an object level comparison. In: Proc. of ER'99, Paris, France (1999).
20. Smith, M. K., Welty, C., McGuinness, D. L.: OWL Web Ontology Language Guide, W3C Recommendation 10 February, World Wide Web Consortium, 2004
21. Su, X., Ilebrekke L.: A comparative study of ontology languages and tools. Halpin, T., Siau, K., Krogstie, J. (Eds.): Proc. of EMMSAD'02, Toronto, Canada, May (2002)
22. Sure Y., Studer R.: On-To-Knowledge Methodology – Final Version. Institute AIFB, University of Karlsruhe. September 26, (2002).
23. Swartout, B., Ramesh P., Knight, K., Russ, T.: Toward Distributed Use of Large-Scale Ontologies. Symposium on Ontological Engineering of AAAI. Stanford (California), (1997).
24. Uschold M.: Building Ontologies: Towards a Unified methodology. In: Proc. of the 16th Annual Conf. of the British Computer Society Specialist Group on Expert Systems. Cambridge. United Kingdom, (1996).
25. Uschold M., Gruninger M.: Ontologies: Principles, methods and applications. Knowledge Engineering Review, 11(2), (1996).