ONTOLOGY CONSTRUCTION IN AN ENTERPRISE CONTEXT: COMPARING AND EVALUATING TWO APPROACHES

Eva Blomqvist, Annika Öhgren and Kurt Sandkuhl
Department of Electronic and Computer Engineering, School of Engineering, Jönköping University
P.O.Box 1126, SE-551 11 Jönköping, Sweden

Keywords: Ontology engineering methods, Enterprise ontologies, Ontology evaluation.

Abstract: Structuring enterprise information and supporting knowledge management is a growing application field for enterprise ontologies. Research work presented in this paper focuses on construction of enterprise ontologies. In an experiment, two methods were used in parallel when developing an ontology for a company in automotive supplier industries. One method is based on automatic ontology construction, the other method is a manual approach based on cookbook-like instructions. The paper compares and evaluates the methods and their results. For ontology evaluation, selected approaches were combined including both evaluation by ontology engineers and evaluation by domain experts. The main conclusion is that the compared methods have different strengths and an integration of both developed ontologies and used methods should be investigated.

1 INTRODUCTION

The area of ontology engineering is developing fast, new methods and tools are introduced continuously. When considering small-scale application cases the need for reducing effort and expert requirements is obvious. One way of reducing the effort is by further facilitating semi-automatic construction of ontologies. Other ways are to further detail the existing manual methods, and to facilitate reuse in the construction process.

Our earlier research has resulted in two different methods for constructing enterprise ontologies, a manual method described in (Öhgren and Sandkuhl, 2005) and an automatic method, exploiting ontology patterns for the construction process, described in (Blomqvist, 2005). These methods have been used in parallel during a project with industrial partners and now an evaluation and comparison of the results have been conducted. Enterprise ontologies in this case denotes domain and application ontologies within enterprises, for structuring of enterprise information.

Section 2 presents definitions and related work. Experiences from the project are presented in section 3 and the results of the evaluation in section 4. Finally in section 5 a discussion about the results is presented and some conclusions are drawn.

2 BACKGROUND

This section presents background and definitions together with an overview of existing evaluation approaches.

2.1 Ontologies

In this paper ontology is defined as:

An ontology is a hierarchically structured set of concepts describing a specific domain of knowledge, that can be used to create a knowledge base. An ontology contains concepts, a subsumption hierarchy, arbitrary relations between concepts, and axioms. It may also contain other constraints and functions.

Even using this definition, ontologies can be used for different purposes, and can be constructed in many different ways. One of the most common ways to describe the level of generality of an ontology is by using the structure suggested by (Guarino, 1998), where a general top-level ontology can be specialised into a domain ontology or a task ontology. Domain and task ontologies can in turn be specialised, and combined, into application ontologies.

Another categorisation is to classify ontologies by their intended use, as in (van Heijst et al., 1997).

Blomqvist E., Öhgren A. and Sandkuhl K. (2006).
ONTOLOGY CONSTRUCTION IN AN ENTERPRISE CONTEXT: COMPARING AND EVALUATING TWO APPROACHES.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - ISAS*, pages 86-93
DOI: 10.5220/0002446700860093

There are three main levels, terminological ontologies, information ontologies, and knowledge modelling ontologies, where each level adds further complexity. This work is concerned with enterprise ontologies on the domain or application level, intended for structuring of enterprise information.

2.2 Ontology Evaluation

Ontology evaluation is not a very mature research field but since ontologies are becoming more and more common, there is an urgent need for well-defined evaluation methods. The approaches that do exist differ in their aims, some are used to determine how to choose between several ontologies, while others aim at validating a single ontology.

A recent deliverable by the Knowledge Web Consortium, see (Hartmann et al., 2005), tries to give an overview of the current state-of-the-art in ontology evaluation. They identify three different stages of evaluation, namely evaluating an ontology in its pre-modelling stage, its modelling stage, or after its release. The first stage involves evaluating the material the ontology will be based on, the second stage checks the ontology correctness while building it, and the final stage involves comparing existing ontologies and monitoring ontologies in use.

2.2.1 Evaluation During Construction

When evaluating single ontologies during (or right after) construction, guidelines exist for manually evaluating correctness. One approach is described in (Gómez-Pérez, 1999), where the focus is on evaluating a taxonomy. The guidelines are quite brief, so some expert knowledge is definitely needed. The idea is to spot and correct common development errors.

Another approach is the OntoClean methodology, presented in (Guarino and Welty, 2002). The methodology aims at exposing inappropriate or inconsistent modelling choices by using metaproperties to characterise the modelled knowledge. Three properties are discussed; rigidity, identity, and unity, and these can be used to evaluate if subsumption has been misused.

2.2.2 Evaluation After Construction

A quite mature method dealing with comparing ontologies is the OntoMetric framework described in (Lozano-Tello and Gómez-Pérez, 2004). The method uses a multilevel framework of characteristics as a template for information on existing ontologies. Five dimensions are used; content, language, methodology, cost and tools. Each dimension has a set of factors which are in turn defined through a set of characteristics. The evaluation results in an overall score of the suitability of the ontology in a specific case. To ease the evaluation of ontology concepts glosses (natural language explanations) could be generated, as in (Navigli et al., 2004), to let domain experts evaluate concepts without the aid of ontology experts.

A similar approach, using quality factors and an ontology of knowledge quality, is described in (Supekar et al., 2004). Here the focus is more on ”objective quality” while in OntoMetric the focus is on subjective usefulness. Yet another similar approach is presented in (Davies et al., 2003), where the authors suggest that the meta-models of the ontologies can be used to compare them.

A very natural way of comparing and evaluating ontologies is of course to test how well they perform on certain tasks. Such an approach is suggested in (Porzel and Malaka, 2004) but it is based on a ”gold-standard”, which can be very hard to decide on. As noted in (Brewster et al., 2004) there are, as of now, no standard tools for evaluating ontologies in specified task environments.

When it comes to ontology content there exist different ways to compare the content similarity of two ontologies. Such approaches have been implemented to match and integrate ontologies, like Chimaera described in (McGuinness et al., 2000) and PROMPT described in (Noy and Musen, 2000). There are also others which for example measure cohesion of ontology concepts and modules, as in (Yao et al., 2005). This approach is simple but gives a good and intuitive idea of how the ontology is organised.

2.2.3 IR-related Approaches

In (Brewster et al., 2004) the authors describe why classical Information Retrieval methods and measures, like precision and recall, cannot be used to evaluate ontologies or ontology construction methodologies in general (although other authors do use this for special cases, like in (Navigli et al., 2004)). Instead (Brewster et al., 2004) suggests an architecture for evaluating the fit of an ontology to a certain corpus of texts. This is done by extracting information, expanding the information and then mapping it against the ontology.

3 EXPERIMENT

This section describes the experiment performed to develop an ontology for the same purpose and with the same scope but using two different methods, a manual and an automatic method. The experiment was part of the research project SEMCO. SEMCO aims at introducing semantic technologies into the development process of software-intensive electronic systems in order to improve efficiency when managing variants and versions of software artifacts.

The scope of the experiment was to construct a selected part of the enterprise ontology for one of the SEMCO project partners. The purpose of the ontology is to support capturing of relations between development processes, organisation structures, product structures, and artifacts within the software development process. The ontologies are so far limited to describing the requirements engineering process, requirements and specifications with connections to products and parts, organisational concepts and project artifacts.

The two methods for constructing ontologies are quite new and have previous to this scenario only been used in smaller research test-cases. Both construction processes used the same set of project documents as starting point and major knowledge source. Furthermore, for the evaluation the same methods, tools, and evaluation teams were used.

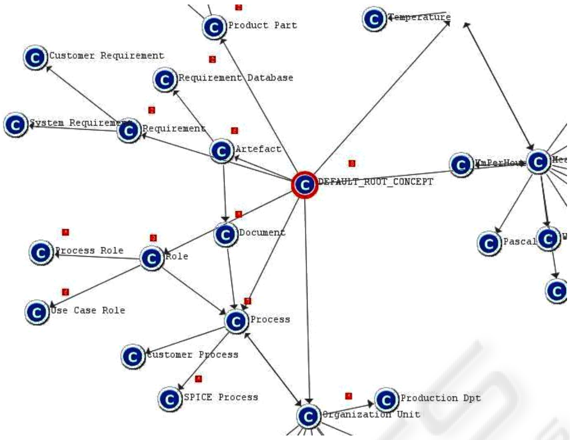

Figure 1: Part of the ontology resulting from the manual construction.

3.1 Manual Construction

In a previous paper we have described the development of a methodology to fit the requirements in small-scale application contexts, see (Öhgren and Sandkuhl, 2005). Below we give a short description of the proposed methodology consisting of four different phases; requirements analysis, building, implementation, and evaluation and maintenance.

In the requirements analysis phase formalities of the ontology are specified, e.g. the the purpose and scope, intended users and uses etc. Usage scenarios of how the ontology can be applied should be developed. In order to shorten the development time, one step is to check whether there are any ontologies that can be integrated with the one being built.

The building phase is iterative and uses a middle-out approach. The implementation phase primarily consists of implementing the ontology in an appropriate ontology editor tool. The implemented ontology finally needs to be evaluated to check that it fulfils the requirements. It should also be evaluated according to criteria such as clarity, consistency and reusability.

The manual construction in our project followed these four phases. First of all a user requirements document was produced. Information was mainly given by the SEMCO project leader, for example on intended users and uses of the ontology, purpose and scope. Different knowledge sources were identified, and we also looked for other ontologies to integrate but found none which was considered relevant.

In the building phase the starting point was to use the available project documents as a basis for building a concept hierarchy. It was decided that natural language descriptions for each concept were not necessary at this point. It was quite hard to derive relations, constraints, and axioms from the documents so after document analysis focus was switched to interviews with selected employees at the company.

The interviews were performed in two sessions. At the first session the interviewees discussed the top-level concepts, then went further in the hierarchy discussing each concept and its subconcepts. Feedback was given as suggestions, like ”Restructure this” or ”This concept is really not that important”. After the first interview session the ontology was changed according to the suggestions. The second interview session was carried out in the same way, resulting in minor corrections to the ontology.

The evaluation and maintenance phase was partly integrated with the building phase, where the interviewees reviewed the ontology. The other parts of the evaluation are described in the following sections. The maintenance part has not yet been performed.

The resulting ontology has 8 concepts directly beneath the root and 224 concepts in total. In Fig. 1 a small part of this ontology is illustrated.

3.2 Automatic Construction

There are a number of existing semi-automatic approaches for ontology construction. In most cases there are existing knowledge sources, the question is how to extract the knowledge from these sources automatically and reformulate it into an ontology. Some of the parts present in most systems are term extraction by linguistical analysis and relations extraction by co-occurrence theory. Some systems also try to automatically build a concept taxonomy by using concept clustering, these are the ones that come closest to being fully automatic. Our method on the other hand aims at being completely automatic.

The idea of using patterns is a commonly accepted way to reduce development effort and increase reuse in for example Software Engineering, but the ontology community has not yet adopted the idea on a broader scale. There exist a few patterns for ontologies, but none of the manual methods uses any design patterns for ontologies today, nor does any semi-automatic approach to the best of our knowledge.

Our approach, as presented in (Blomqvist, 2005), uses design patterns as buildings blocks for ontology construction. Our approach also uses existing tools to extract concepts and relations. The general idea is then to take the extracted terms and relations, match them against design patterns, and depending on the result use parts of the patterns to build the ontology.

Prior to this experiment 25 patterns were developed (examples can be found in (Blomqvist, 2005)). The text corpus used consisted of software development plans and software development process descriptions. The matching of the patterns against the extracted concepts was done using a lexical matching tool.

The score representing matched concepts then was weighted together with a score of matched relations into a total score. This resulted in 14 patterns above the predefined threshold. The accepted patterns were compiled into an ontology using some heuristics and other rules.

The resulting ontology contains 35 concepts directly beneath the root concept and in total 85 concepts. In Fig. 2 a small part of this ontology is illustrated.

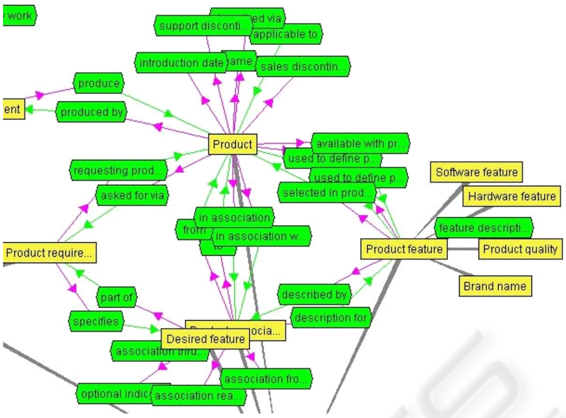

Figure 2: Part of the ontology resulting from the automatic construction.

4 EVALUATION

This section presents the choice of evaluation methods and a description of the evaluation and its results.

4.1 Evaluation Setup

A decision was made to use several evaluation approaches, both intended for ontology expert and domain expert review, to get a broader view of the ontologies and indirectly also the construction methods.

First, a general comparison of the ontologies was needed to get an idea of differences and similarities. This comparison was done based on some intuitive metrics, like number of concepts, average number of attributes per concept, average number of subclasses per concept and average number of association per concept. Also, the cohesion metrics from (Yao et al., 2005) were used, since we feel that they complement the other measures well. These metrics are: number of root classes, number of leaf classes and average depth of inheritance tree.

Second, an evaluation was performed by internal ontology experts using the two most well-known approaches for taxonomic evaluation, presented in (Gómez-Pérez, 1999) and (Guarino and Welty, 2002). Internal ontology experts were used for these evaluations, mainly because of their previous knowledge of the evaluation methods. Since we are evaluating both the ontologies and (indirectly) the methodologies for creating them, the errors discovered can give valuable indications on advantages and disadvantages of each construction method.

Finally, to evaluate the content of the ontologies a subset of the OntoMetric framework in (Lozano-Tello and Gómez-Pérez, 2004) was used. For our purpose only the dimension content was deemed interesting, and only one level of characteristics for each factor. Some characteristics were not applicable to both ontologies and since this is mainly a comparison, these were taken out of the framework. The computation of the final score was not performed, since the number of factors and characteristics were low enough to give a general impression. Domain experts from the company in question formed the evaluation team, but internal ontology experts prepared the material, assisted through the evaluation and collected the results.

The most desirable method of evaluation would of course be to apply the ontologies in their intended application context. This is not yet possible though, since the resulting application of the SEMCO-project is still in its planning stage. Automatic gloss-generation could be a future improvement of the OntoMetric method, but at this time no such tool was available. Finally, the reason for not using any IR-related approach was mainly that this would give an unfair advantage to the automatically created ontology since this was constructed using similar methods and already completely based on the available company documents.

4.2 General Comparison

First some general characteristics of the ontologies were collected. In Table 1 these are presented for the two ontologies (the ontology created automatically is denoted ”Aut” and the other one ”Man”). The results show that the automatically created ontology has a large number of root concepts, it lacks some abstract general notions to keep the concepts together in groups. It is also quite shallow and many concepts lack subconcepts altogether. Despite this, the concepts are much more strongly related through non-taxonomic relations and has more attributes than in the other ontology.

The manually created ontology on the other hand contains a larger number of concepts. It also contains a top-level abstraction dividing the ontology into subject areas. There are few attributes and relations, this might be due to that many attributes are actually represented by other specific concepts, they are just not connected by an appropriate relation. Relations seem to be hard to elicit from interviews.

4.3 Evaluation by Ontology Engineers

Two evaluation methods were used in the expert evaluation, first the general taxonomic evaluation criteria and then the OntoClean framework.

4.3.1 Taxonomy Evaluation

The ontologies were evaluated by ontology engineers according to the criteria presented in (Gómez-Pérez, 1999). The criteria discussed are the following:

- Inconsistency: circularity, partition and semantic errors
- Incompleteness: incomplete concept classification and partition errors
- Redundancy: grammatical redundancy, identical formal definitions of concepts or instances

There exist no circularity errors in the automatically created ontology since there is no multiple inheritance present, this also prevents most errors belonging to the inconsistency partition errors group. Multiple inheritance in the manually created ontology occurs only in a few cases, and no circularity errors were discovered among these. This also reduces the possibilities for partition errors, as mentioned previously. There are no exhaustive decompositions or partitions specified in either ontology so this eliminates the possibility of finding any other kind of partition errors.

Semantic inconsistency errors are more subtle to discover. This is a question of identifying wrong classifications. In the automatically created ontology there exist two concepts which could be thought of as wrongly classified since they make no sense in the context of this ontology, they are simply ”junk” which happened to enter the ontology due to the immaturity of the ontology construction process. Semantic inconsistencies could also occur when two overlapping patterns are both included in the ontology, but this seems not to be the case in the ontology at hand. Concerning the manually created ontology these errors can only be assumed to have been discovered in the interview sessions with the domain experts.

Table 1: Comparison of general characteristics.

Characteristic	Man	Aut
Number of concepts	224	85
Number of root concepts	8	35
Number of leaf concepts	180	64
Avg depth of inheritance	2,52	1,95
Avg number of rel. concepts	0,13	0,79
Avg number of attributes	0,01	0,46
Avg number of subclasses	1,00	0,57

Next, the incompleteness criteria was examined. Incomplete concept classifications might exist in the automatically created ontology due to concepts missing in the patterns or in the text corpus used to develop the ontology. Since this will be an application ontology, not the whole domain needs to be modelled but only the parts needed for this specific application. When comparing the two ontologies though, the automatically created ontology seems to lack more specific concepts, such as names and company specific terms. But even for the manually constructed ontology it is difficult to determine the incompleteness criteria until the ontology is used in its intended context.

Several occurrences of partition errors were found in the automatically created ontology, especially lack of disjointness definitions. This should be included in the patterns in order for it to propagate into the created ontologies. Also some cases of exhaustive knowledge omission were found, but on the other hand the knowledge might not be needed for this specific application. In the manual construction process disjointness and exhaustive partitions were not discussed before building the ontology, so it is at this point not certain that there is a need for it. Deciding this ought to be part of the construction methodology.

Finally, there are no concepts with identical formal definitions but different names or redundant subclass-of relations in either ontology. Redundant subclass-of relations are not present in the patterns used in the automatic approach and no overlapping patterns have introduced it in this case. It is worth studying when considering overlap between patterns though.

4.3.2 OntoClean

The second expert evaluation was performed by using the OntoClean methodology. Every concept in the ontology was annotated with the properties rigidity, identity and unity. This resulted in a backbone taxonomy containing 25 concepts in the automatically constructed ontology. Here two violations of the unity and anti-unity rule were found and one violation of the incompatible identity rule. When analysed the unity problems arise because in this company ”work” is seen as a ”product”, but ”work” is generally not a whole. The identity conflict has the same cause since it is a question of ”work” being defined as subsumed by the concept ”product”, but products in general are identified by a id-number while work is not. This is a quite serious problem which requires some consideration to solve in a good way, so that the solution still reflects the reality of the company in question.

For the manually constructed ontology, the backbone taxonomy contains 178 concepts. One violation of the unity and anti-unity rule was found, and none of the other kinds. The violation exists between the concepts ”function” and ”code”, while a function is a clearly defined unit the concept of code is more abstract and cannot generally be seen as a homogeneous unit. This violation exists mainly due to that the abstraction level differs too much among the concepts on the same level of the ontology hierarchy. The fact that no other violations were found is perhaps due to the simple structure of the ontology, it is very much like a simple taxonomy of terms. A summary of the results is presented in Table 2, where the manually created ontology is denoted ”Man” and the automatically created ontology ”Aut”.

4.4 Evaluation by Domain Experts

As a third step an evaluation was performed by domain experts from the company in question. The evaluation was done based on a part of the OntoMetric framework in (Lozano-Tello and Gómez-Pérez, 2004). Only the dimension ”content” was considered and also no final score was computed, since the assessed characteristics are quite few and can tell us much about the nature of the ontologies.

The dimension ”content” contains four factors: concepts, relations, taxonomy and axioms. For each of these factors characteristics applicable in this case were chosen. The scale suggested in (Lozano-Tello and Gómez-Pérez, 2004) ranging from ”very low” to ”very high” in five steps, was used as scoring. The characteristics used and the resulting scores for each ontology are presented in Table 3, where ”Man” denotes the manually created ontology and ”Aut” the automatically created one.

Table 2: Result of the OntoClean evaluation.

OntoClean rule	Man	Aut
Incompatible identity	No	1
Incompatible unity criteria	No	No
Unity/anti-unity conflict	1	2
Rigidity/anti-rigidity conflict	No	No

The table shows that both ontologies contain an appropriate number of concepts, but the concepts in the manually constructed ontology are deemed more essential. This is most likely due to that the concepts are more specific. The automatically created ontology also lacks some general abstract concepts to give it a comprehensible structure. On the other hand, the automatically created ontology contains more attributes which help to describe and define the concepts and make the need for natural language descriptions less.

The automatically created ontology contains many more non-taxonomic relations than the manually created one, even such relations that the company might not have thought of itself but which are still valid. The manually created ontology mostly contains relations explicitly stated by the company. It is the non-taxonomic relations that gives structure and the automatically created ontology while the manual ontology relies on specificity of concepts and precise naming.

The automatically created ontology of course also has a taxonomic structure even though it lacks both some abstract top-level and the most specific levels, compared to the manually created one. Despite this, it is perceived as having quite a large depth due to detailed division of the intermediate levels. The manually created ontology has a larger number of subclasses per concept since a high number of very specific concepts exist.

The number of axioms is low in both ontologies, and the ones present are very simple. More advanced ”business rules” is something that the company might need if the implemented application using the ontology is to function efficiently. In the manual method the question is how to elicit such rules using interviews, in the automatic method these should be included in the patterns but then needs to be appropriately matched to the knowledge extracted.

In addition to the evaluation of the characteristics an interview was conducted in order to see what parts might be completely missing. Natural language descriptions of concepts was one such item of discussion. For the task to be performed by the implemented ontology the interviewed domain experts thought this was not needed, since it was quite clear from the naming and context how a certain concept would be used. In a longer perspective, for evolution and maintenance of the ontology, this would still be desirable though.

Table 3: Result of the domain expert evaluation.

CHARACTERISTIC	SCORE ”Very low”	”Low”	”Medium”	”High”	”Very high”
CONCEPTS					
Essential concepts in superior levels	Aut			Man	
Essential concepts		Aut			Man
Formal spec. coincides with naming			Aut	Man	
Attributes describe concepts	Man			Aut	
Number of concepts				Man Aut	
RELATIONS					
Essential relations		Man		Aut	
Relations relate appropriate concepts			Man	Aut	
Formal spec. of rel. coincides with naming			Aut	Man	
Formal properties of relations	Man			Aut	
Number of relations		Man			Aut
TAXONOMY					
Several perspectives			Man Aut		
Maximum depth			Man	Aut	
Average number of subclasses			Aut	Man	
AXIOMS					
Axioms solve queries		Man	Aut		
Number of axioms	Man	Aut			

5 CONCLUSIONS AND FUTURE WORK

With respect to the evaluation methods for performing this experiment some impressions can be noted: The general characteristics used give a good idea of the overall structure of each ontology, but they cannot be used directly to evaluate the ontologies. The intended context of the ontology needs to be taken into account. If the ontology is intended for use by an automatic system there might not be a need for an intuitive top-level structure for example.

The same problem applies to the taxonomic evaluation method. This method might not be appropriate when the exact usage of the application of the ontology is still not specified in great detail. This is an explanation for the somewhat inconclusive results produced by that part of the evaluation.

The OntoClean evaluation on the other hand produces more conclusive and exact results, which do not only expose faults in the ontology design but can point out difficulties and ambiguities of the real-world case at hand. This evaluation produced valuable results although it was not very easy to perform and required a deep understanding of the metaproperties involved.

The OntoMetric-framework produced some good results, but still has some disadvantages. Here the problem was to present the ontologies to the domain experts in an understandable way, without at the same time influencing the reviewers. This difficulty will always remain when using domain experts without ontology expertise, but perhaps it could be reduced in the future by using something like the gloss generation discussed in section 2.2.2.

Finally, the most interesting evaluation approach, where the ontologies are tested against their goals and application scenarios, is still not performed. This was not possible to include in this study since development of a pilot application in the SEMCO-project is still a future task. Also no general task-oriented application environments exist for testing ontologies, as noted in section 2.2.2. To further explore the advantages and disadvantages of the two methods, some additional measures, like construction time, could be interesting to evaluate. This has not yet been possible, since the systems of the automatic method are not yet fully integrated and thereby for example requires some file conversions and manual procedures which are not really part of the method.

To summarise the performed evaluations, some strengths and weaknesses can be noted in both the manual and the automatic approach. The automatic approach will probably never capture company specific concepts, since these will not be part of general patterns. Also, the method can only capture what is in the patterns on the upper levels, so a choice has to be made whether to include some abstract concepts at the top level of most patterns or to be satisfied with a less coherent ontology. Finally, the correctness of the resulting ontology is very much dependent on the correctness of the patterns. The automatic approach has its strengths in relying on well-proven solutions and easily including complex relations and axioms.

The manual approach gives a result with less complex relations and axioms. Furthermore, the extent to which the application domain is covered by the ontology depends significantly on the interviewed experts, different domain experts might often present different views. On the other hand, the manual approach has one big advantage, since it also captures the most specific concepts that the enterprise actually uses. Also, the more abstract concepts at the upper level give an intuitive idea of the scope of the ontology.

Neither of the approaches produce many errors in the ontology, according to the evaluations, but some improvements can be made in both methods. Improvements of the automatic method could be to evaluate the patterns more thoroughly. The patterns could also be enriched with more axioms and natural language descriptions. Improvements of the manual approach could be to use a larger set of knowledge acquisition methods to elicit more complex structures from the document sources and domain experts.

The main conclusion, which can be drawn is that the ontology engineering approaches each have both strengths and weaknesses and complement each other well. This might suggest that a combination of the approaches could give the best results, but it is too early to state this firmly, since the methods have only been tested in parallel for one single case. The next step is to repeat this experiment in other cases in order to be able to generalise these results and perhaps arrive at some solution for combining the approaches. Also the resulting ontologies complement each other quite well. In this particular project a possible combination of the two ontologies for generating the application ontology needed will be investigated.

ACKNOWLEDGEMENTS

This work is part of the research project Semantic Structuring of Components for Model-based Software Engineering of Dependable Systems (SEMCO) based on a grant from the Swedish KK-Foundation (grant 2003/0241). We thank all evaluation teams for contributing to the results of this study. Also special thanks to three anonymous reviewers for valuable comments on how to improve this paper.

REFERENCES

Blomqvist, E. (2005). Fully Automatic Construction of Enterprise Ontologies Using Design Patterns: Initial Method and First Experiences. In *Proc. of The 4th Intl Conf. on Ontologies, DataBases, and Applications of Semantics (ODBASE)*, Cyprus.

Brewster, C., Alani, H., Dasmahapatra, S., and Wilks, Y. (2004). Data Driven Ontology Evaluation. In *Proc. of Intl Conf. on Language Resources and Evaluation*, Portugal.

Davies, I., Green, P., Milton, S., and Rosemann, M. (2003). Using Meta Models for the Comparison of Ontologies. In *Proc. of Eval. of Modeling Methods in Systems Analysis and Design Workshop-EMMSAD'03*.

Gómez-Pérez, A. (1999). Evaluation of Taxonomic Knowledge in Ontologies and Knowledge Bases. In *Banff Knowledge Acquisition for Knowledge-Based Systems, KAW'99*, volume 2.

Guarino, N. (1998). Formal Ontology and Information Systems. In *Proceedings of FOIS'98*, pages 3–15.

Guarino, N. and Welty, C. (2002). Evaluating Ontological Decisions with OntoClean. *Communications of the ACM*, 45(2):61–65.

Hartmann, J., Spyns, P., Giboin, A., Maynard, D., Cuel, R., Suarez-Figueroa, M. C., and Sure, Y. (2005). D1.2.3 methods for ontology evaluation. Version 1.3.1, Available at: http://knowledgeweb.semanticweb.org/, Downloaded 2005-05-10.

Lozano-Tello, A. and Gómez-Pérez, A. (2004). ONTOMETRIC: A Method to Choose the Appropriate Ontology. *Journal of Database Management*, 15(2).

McGuinness, D., Fikes, R., Rice, J., and Wilder, S. (2000). An Environment for Merging and Testing Large Ontologies. In *Proc. of the 7th Intl Conf. on Principles of Knowledge Representation and Reasoning (KR2000)*, Colorado, USA.

Navigli, R., Velardi, P., Cucchiarelli, A., and Neri, F. (2004). Automatic Ontology Learning: Supporting a Per-Concept Evaluation by Domain Experts. In *Workshop on Ontology Learning and Population (ECAI 2004)*, Valencia, Spain.

Noy, N. F. and Musen, M. A. (2000). PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *17th National Conf. on AI (AAAI-2000)*, Austin, Texas.

Öhgren, A. and Sandkuhl, K. (2005). Towards a Methodology for Ontology Development in Small and Medium-Sized Enterprises. In *IADIS Conference on Applied Computing*, Algarve, Portugal.

Porzel, R. and Malaka, R. (2004). A Task-based Approach for ontology Evaluation. In *Workshop on Ontology Learning and Population (ECAI 2004)*.

Supekar, K., Patel, C., and Lee, Y. (2004). Characterizing Quality of Knowledge on Semantic Web. In *Proceedings of AAAI Florida AI Research Symposium (FLAIRS-2004)*, Miami Beach, Florida.

van Heijst, G., Schreiber, A. T., and Wielinga, B. J. (1997). Using explicit ontologies for KBS development. *Intl Journal of Human-Computer Studies*, 46(2-3).

Yao, H., Orme, A. M., and Etzkorn, L. (2005). Cohesion Metrics for Ontology Design and Application. *Journal of Computer Science*, 1(1):107–113.