The Methodology, Methods and Tools for Agile Ontology Maintenance – A Status Report

Markus Luczak-Rösch

Supervisor: Robert Tolksdorf, Co-Supervisor: Natasha Noy, PhD Research Phase 2
Freie Universität Berlin, Institute of Computer Science,
Networked Information Systems Workgroup, Berlin D-14195, Germany,
markus.luczak-roesch@fu-berlin.de

Abstract. Ontologies are an appropriate means to represent knowledge on the Web. Research on ontology engineering reached practices for an integrative lifecycle support. However, a broader success of ontologies in Web-based information systems remains unreached while the more lightweight semantic approaches are rather successful. The linked data initiative for example became a huge success during the last few years. Relying on the technologies of RDF and on ontologies as the appropriate means for the underlying vocabularies, linked datasets are one of the biggest and most actively used application areas of ontologies on the Web. We assume, paired with the emerging trend of services and microservices on the Web, new dynamic scenarios gain momentum in which a shared knowledge base is made available to several dynamically changing services and applications with disparate requirements. Our work is a step towards such a dynamic scenario in which an ontology adapts to the requirements of the accessing services and applications as well as the user's needs in an agile way based on ontology usage. Thus, our approach reduces the experts' involvement in ontology maintenance processes.

1 Introduction and Problem Statement

Ontologies are an appropriate means to represent knowledge on the Web. Research on ontology engineering methodologies has come from describing the scratch development of ontologies and reached practices for an integrative lifecycle support. The ontology engineering discipline has changed from an individual art towards a collaborative and distributed process with disparate skilled users develop consensual models and distributed networks of ontologies[1, 10, 12, 14, 21, 22]. However, a broader success of ontologies in Web-based information systems remains unreached. They gained momentum in some characteristic and closed domains, such as health care and life sciences. On the every-day Web the more lightweight semantic approaches are rather successful which are based upon small vocabularies, e.g. the emerging linked data initiative[5, 4, 11]. But also this lightweight semantic cannot deploy its full potential. The Web 2.0 resulted huge so called data silos. By use of wrappers or crawlers huge RDF datasets are derived from the relational databases of such silos. Consolidating and integrating the whole data of a specific application-dependent purpose or a specific individual remains a cumbersome task. Not to mention the control of the evolving knowledge in the silos.

As a next logical step one should await that, against the trend of the data silos, the user holds and controls her data on her own. Paired with the emerging trend of services and micro-services on the Web [9] this results in a dynamic scenario in which a shared knowledge base is made available to several dynamically changing services with disparate requirements. This work envisions a step towards such a dynamic scenario in which an ontology adapts to the requirements of the accessing services and applications in an agile way.

The general and personal motivation for this work consists of three core parts. The first part is based upon our studies of the existing ontology engineering methodologies. It represents the fundamental direction of this work. As a second part, we derive from personal interviews with small and mid-sized enterprise (SME) partners of the project Corporate Semantic Web, that they look for a lightweight and dynamic process for ontology maintenance which minimizes the need for ontology experts to be present. We explicitly focus this scenario, however, we respect that there are enterprise settings as well which need and deal with heavyweight ontology engineering processes. On the whole, our idea meets the gap, which we identified as the result of the study of ontology engineering approaches and which has been also identified by others, such as [16]. That means concretely that research regards human-centered feedback as elementary part of the ontology lifecycle and ontology maintenance is more or less treated as the loop back to the beginning of the development process. Thus, ontology maintenance results as the specific direction of this work. The third part of the motivation is our personal vision of the next logical step of the Web from a social Web 2.0/3.0 towards a Web of services and alternative access devices. That means, that the next generation of the Web will be less driven by direct human access to contents and services by use of conventional client tools (e.g. Web browsers) but more by mobile devices and services. As a result of that the concepts of human-centered ontology engineering, such as argumentation to concepts and relations to reach the ontology consensus will loose impact.

Agile ontology maintenance or in other words dynamic ontology evolution, is an open problem in the research community. It is embedded in the hot research field of ontology dynamics in general, which gained momentum since the classical ontology engineering methodologies reached a mature state. Dynamic ontology evolution is currently addressed under different scopes – the domain-oriented scope and the application- or usage-oriented scope. The most important representative for research on the first scope is the work of Zablith[23], who concentrates on the usage of background knowledge for ontology evolution purposes. Regarding the two factors sufficiency of an ontology and conciseness of an ontology one could state that Zabliths work supports the former factor while ours supports the latter one. Related work with focus on usage-oriented ontology evolution is mentioned in the work of Stojanovic[19, 20].

1.1 An Exemplary Application Area

To clarify this motivation we will briefly come up with a simple running example for the problem which we want to solve. Consider a company's knowledge base which includes information about the employees. In the beginning only personal information have been collected conforming the friend of a friend (FOAF) vocabulary. One service uses the knowledge base which generates lists of employees with certain interests. Each time a new service is bound to the knowledge base, such as a service for displaying absent employees or information about the income (e.g. for the accounting), the maintainer of the knowledge base has to find out which facts, in the sense of the T-box of the ontology, are missing and how she can easily adopt the current T-box and possibly the A-box as well to these new application requirements. It is also possible that separate services require the same information represented in different vocabularies (e.g. foaf:name vs. myvocabulary:name) which yields the conflict for the maintainer whether to replace the present representation or to model a mapping between both. The decision for either the first or the latter depends on several criteria, such as the computability of the ontology for complex reasoning or obsolete and unused information.

1.2 A Real Life Use Case

Regarding the application of linked data or rather applications that use linked datasets, an interesting research gap appears. The information which were collected by classical Web usage mining techniques to improve Web pages in a user-oriented way do not work out. The well-known principles of sessions, paths or click-through do not exist in the same fashion for Web data as they exist for Web pages. However, the improvement of the quality of a dataset with reference to the user's needs has to be in focus of each single dataset host. This problem can be observed very well in the case of the DBpedia[2] dataset. The maintainers of the dataset regularly performed updates on the shared data and the underlying ontology. The changes were documented in a changelog[1] and let us reason that the version step from DBpedia 3.4 to DBpedia 3.5 contains several design decisions which should reflect the users needs, e.g. the consistent usage of centimeters instead of meters for the property height in the special case of the class Person. Our studies will use this data to check if and how the changes on the DBpedia dataset effectively conform to the user needs.

By now the DBpedia maintainers started to apply features which allow the user community to perform changes on the ontology and the mechanisms for instance population directly. However, it is an interesting question as well how our approach could help the user community in maintaining the DBpedia ontology in the future.

[1] http://wiki.dbpedia.org/ChangeLog

1.3 Research Questions

This motivating examples and the general problem description yield the central research questions of our work:

1. *How does a methodology for ontology maintenance in an agile environment look like?* We search for a process which puts less emphasize on the initial development of an ontology but more on the ontology usage and evolution.
2. *Can we reduce the necessary influence of human experts in the ontology maintenance process by tracking feedback about ontology usage?* In this case our work searches for a formal model that allows the analysis of ontology usage for ontology evolution purposes.

2 General Approach and Research Methodology

Our work is following the principles of design science research. Initially we started by a comprehensive analysis of the state of the art in ontology engineering. To our best knowledge we achieved an integrative overview of the different research directions and methodologies in this field. Thus, we identified one open problem to solve – the problem of agility and dynamics in the ontology engineering process. After this our work concentrated on defining the related context of our proposed solution explicitly. That includes a characterization of the type of ontologies we are addressing and an exemplary application scenario. The single parts of our solution were and are designed and developed stepwise in the following, before we finally will evaluate the theoretical saturation of our technologies in a multi-perspective way so it is possible to border its applicability from other approaches in a pragmatic way.

2.1 Aimed Contributions

Altogether, this work aims at a multi-layered contribution as it is depicted in Figure 1. From the underlying theories we derive a methodology for agile ontology life cycles. Then we design and implement the necessary methods and tools for the core phases of the methodology and finally evaluate both layers, the methodology as well as the methods and tools, multi-perspectively.

The thesis will provide a proper understanding of the problems of agile ontology maintenance and propose the methodology which helps tackling these problems. Since the methodology differs from state of the art approaches in ontology engineering it proposes several innovative process steps which will be supported by the above mentioned methods and tools. Another and important contribution of our work is that the Semantic Web usage mining approach closes the gap between classical Web usage mining techniques and the new technical and organizational issues when Web data instead of Web pages is regarded. The work will not only elaborate on these issues but it will also present a practical solution to tackle them.

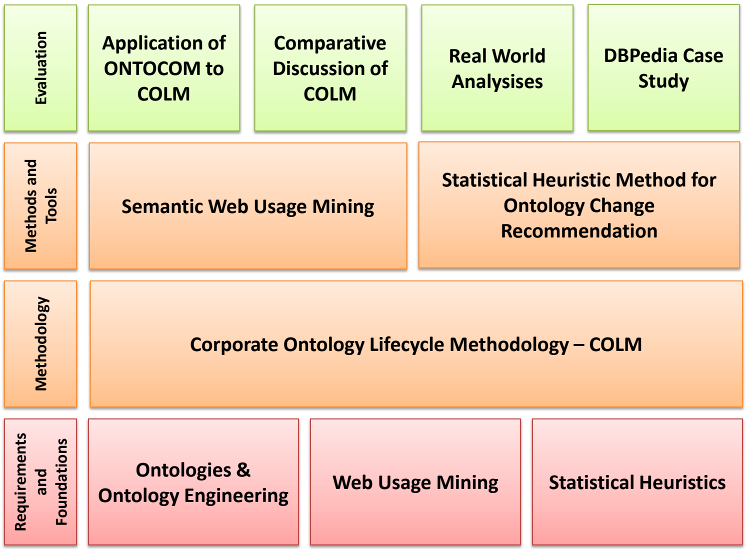
Evaluation
Application of ONTOCOM to COLM
Comparative Discussion of COLM
Real World Analysises
DBPedia Case Study
Methods and Tools
Semantic Web Usage Mining
Statistical Heuristic Method for Ontology Change Recommendation
Methodology
Corporate Ontology Lifecycle Methodology – COLM
Requirements and Foundations
Ontologies & Ontology Engineering
Web Usage Mining
Statistical Heuristics

Fig. 1. Architecture of the general approach

3 Proposed Solution

Three parts are the building blocks of our proposed solution for the problem stated so far: (1) A methodology for agile ontology engineering, (2) a method for Web usage mining in the context of the Semantic Web, and (3) a method for using statistical heuristics for ontology change recommendation. The latter part is in a preliminary design state, while the methodology and the usage mining method have already been developed and matured.

3.1 The Methodology for an Agile Ontology Life Cycle

The Corporate Ontology Lifecycle Methodology (COLM) reflects the agility of knowledge engineering processes and brings in application dependency. We define it as an agile ontology maintenance methodology since it is focused on continuously evolving ontologies in an application-dependent context. To clarify which process steps are more expert-oriented and thus need higher human involvement and those which need less, COLM consists of two different cycles, namely the engineering cycle (high involvement) and the usage cycle (less involvement). The overall goal is to use an intuitive reporting of tracked usage information which indicates the necessity of change.

As depicted in Figure 2, the process starts at the *selection / development / integration* phase. The result of this phase is an ontology, which is *validated* within an application-dependent context. If it is approved that the ontology

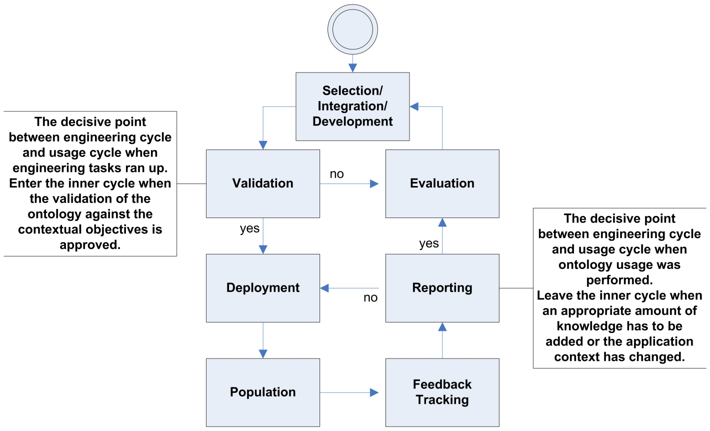
Selection/
Integration/
Development
The decisive point between engineering cycle and usage cycle when engineering tasks ran up. Enter the inner cycle when the validation of the ontology against the contextual objectives is approved.
Validation
no
Evaluation
yes
yes
The decisive point between engineering cycle and usage cycle when ontology usage was performed.
Leave the inner cycle when an appropriate amount of knowledge has to be added or the application context has changed.
Deployment
no
Reporting
Population
Feedback Tracking

Fig. 2. The Corporate Ontology Lifecycle Methodology COLM

suites the requirements it is *deployed* to be in use and it is *populated*. Throughout the whole *feedback tracking* phase, formal statements about users' feedback and behavior are recorded and finally a *reporting* of this information is performed. The usage cycle is left if any necessary change has been detected and the knowledge engineers *evaluate* the weaknesses of the current ontology.

3.2 Semantic Web Usage Mining

In this section we describe our approach to analyze server log files of (linked) data endpoints with the goal to retrieve information about the usage of the dataset and its underlying ontology. The log files contain information in the extended common log format[8, 7] about the access to single RDF resources and about SPARQL queries. Listing 1.1 shows the two relevant types of accesses to the DBpedia dataset – (1) the access to single resources ("GET /resource/..." or "GET /page/...") and (2) the performed SPARQL queries ("GET /sparql/?query=SELECT..."). Each log file contains information about one single day.

```
xxx.xxx.xxx.xxx - - [21/Sep/2009:00:00:01 -0600] "GET /resource/
    Bakemonogatari HTTP/1.1" 303 0 "http://www.google.com/search?as_q=%E5%82%
    B7%E7%89%A9%E8%AA%9E&hl=ja&num=50&btnG=Google+%E6%A4%9C%E7%B4%A2&as_epq=&
    as_oq=&as_eq=&lr=&cr=countryUS&as_ft=i&as_filetype=&as_qdr=all&as_occt=
    any&as_dt=i&as_sitesearch=&as_rights=&safe=images" "Mozilla/4.0 (
    compatible; MSIE 7.0; Windows NT 6.0; SLCC1; .NET CLR 2.0.50727; Media
    Center PC 5.0; .NET CLR 3.0.30618; .NET CLR 3.5.30729; Sleipnir/2.8.5)"
xxx.xxx.xxx.xxx - - [21/Sep/2009:00:00:01 -0600] "GET /sparql/?query=SELECT
    +%3Fabstract+WHERE+{+%3Chttp%3A%2F%2Fdbpedia.org%2Fresource%2FGao_Heng%3E
    +%3Chttp%3A%2F%2Fdbpedia.org%2Fproperty%2Fabstract%3E+%3Fabstract.+FILTER
    +langMatches(lang(%3Fabstract)%2C+%27en%27)+}&format=json HTTP/1.1" 200
    994 "" "PEAR HTTP_Request class ( http://pear.php.net/ )"
```

```
xxx.xxx.xxx.xxx - - [21/Sep/2009:00:00:01 -0600] "GET /page/F.C.
    _Copenhagen_season_2008%25E2%2580%259309/fb_cm3_match3/rep/Fb_report_2t
    HTTP/1.0" 200 7379 "" "msnbot/2.0b (+http://search.msn.com/msnbot.htm)"
```

Listing 1.1. Examplary log entries of the DBpedia dataset

Even though our log file analysis collects general statistical information about the number of requests, the number of different host, the peak access time slots, and the user agents amongst others, it is primarily intended to provide information about the returned results of the endpoints with reference to the users requests. It is easy to see that the primitive observation of HTTP error codes and response sizes does not yield the appropriate information about the size and quality of the result sets or any conclusions about ontology usage on a concept level. This aspect is why our approach differs from the work done in this field by Möller et al.[13].

The analysis is performed on demand and not at the real runtime of the requests. That means we re-run each single call from the log files against a mirror of the DBpedia dataset, which shares the appropriate dataset version with reference to the date of the currently analyzed log file. SPARQL queries are performed against the mirror server as they have been performed against the real endpoint. The requests for single resources are reorganized as SPARQL queries following the simple schema noted in Listing 1.2.

```
SELECT * WHERE { <http://dbpedia.org/resource/[resourceidentifier]> ?property
    ?hasValue }
```

Listing 1.2. Generated SPARQL query related to a simple resource request from the log file

We partition each SPARQL query into its atomic parts – namely (1) patterns, (2) filters and (3) triples – and check each of these parts individually by sepcificly generated SPARQL queries. So finally we collect the following information about the requests: (1) Which queries are executed? (2) Which queries contain errors? (3) Which queries return a non-empty result set? (4) Which query patterns exist? (5) Which query patterns return a non-empty result set? (6) Which filters are used? (7) How do filters effect on the size of result sets? (8) Which triples are requested in queries? (9) Which triples do return a non-empty result set? (10) Which entities appear as subject, predicate or object?

3.3 Statistical Heuristics for Ontology Change Recommendation

Heuristics are guesses about efficient rules that can be used for problem solving. They rely on practical experience in certain types of problem-solving activities. In our case the problem is to evaluate the report which is generated by the usage mining method and recommending changes to be performed to the ontology and the dataset. A central question this heuristic approach should answer is whether and when the usage of an ontology primitive as a subject, predicate or object for example is significant. In this special case that has to respect the factor that the total number of properties which may be used as predicates is much smaller than the total number of classes and instances which may be used as subjects and objects in queries.

We are currently working on a combination of an algorithmic solution that relies on the retrieved statistics and, in the optimal case, on ontology engineering best practices and patterns, to recommend necessary ontology changes. Ontology changes can be recommended on the structural level of the T-box or on the instance level of the A-box, the dataset.

4 Evaluation

Evaluating design science artifacts is a complex thing. That is why we decided to evaluate our approach multi-perspectively with focus on qualitative evaluation methodologies because these are less strict methods with emphasis on stressing new approaches and ideas influenced by subjectivity and diverse research. To some extend the evaluations will focus the individual parts of our work to prove their applicability and pragmatic validity. But, we will also perform an evaluation that puts the things together and provides an integrative view to our solution.

The applicability of the developed methodology COLM will be evaluated in a comparative discussion with other ontology engineering methodologies. The most feasible model seems to create a goal-free and a goal-based evaluation as the basis for this discussion. The central ambition of the goal-free evaluation methodology is to compare different solutions for the same use case against certain criteria. A goal-based evaluation gives a qualified view on the achievement of a single solution, in reference to the recommendations and requirements of the implemented artifact, raised by the developers. In the end this evaluation should proof whether we succeeded in developing a methodology for ontology maintenance in an agile environment (research question 1).

Since the ONTOCOM[15] cost model for ontology engineering processes can serve as a state of the art approach to quantify the effectiveness of ontology engineering processes, we will apply it to COLM in the same fashion as it has been done for other methodologies. By that it will be possible to evaluate the second research question which we raised in the beginning. The question was if we can reduce the necessary influence of human experts in the ontology maintenance process by tracking feedback about ontology usage.

To prove the validity and applicability of our analysis method and the associated heuristics for change recommendation we will perform and document several exemplary executions of them on well-known and widely-used live datasets, such as the Semantic Web Dog Food Corpus or DBTune[2].

As it was mentioned before, we also want to evaluate our approach integrative. This will be done by a case study. The case study evaluation is aimed at the appraisal of how a definite process can be enforced referring to a given process description. In this special case dos that mean, that we want to convince practitioners of the applicability of the COLM methodology, the Semantic Web usage

[2] http://dbtune.org/

mining, and the statistical heuristic change recommendation method to the task of ontology maintenance. The case study is set up as a kind of laboratory experiment which means that we observe the actions by the DBpedia development team when they perform maintenance activities on the DBpedia ontology and the DBpedia dataset.

5 Conclusions and Future Work

In this paper we reported on our ongoing work towards agile ontology maintenance. A presentation of the general approach and research methodology was followed by a detailed description of the concrete contributions which we will achieve when this work is finished. The paper closes with an overview of the aimed evaluations.

The fundamental work for the definition of the research problem and its related context has already been done as well as the development of the first two of three major contributions – the methodology for agile ontology engineering and the methods for Semantic Web usage mining. At the moment we are working on the heuristic model for ontology change recommendation. In parallel we are performing analysis by use of our method on a massive amount of data from the DBpedia dataset and the Semantic Web Dog Food Corpus[3]. That data forms the basis for some of our evaluations which will be completed afterwards.

During the work on this thesis we detected several other related interesting research problems. The Semantic Web usage mining approach for example, which we presented in this report shortly, has a broad range of possibilities to be extended by the application of graph analysis methods. That could provide an interesting insight on the usage of Web data in form of so called heat maps of the RDF graphs or other visualizations that base on our metrics and statistics. An intersting discussion could also be why the approach of query observation and analysis has never been done in the world of databases. To our best knowledge, we were not able to find a comparable work in that area.

References

1. Auer, S., Herre, H.: RapidOWL - An Agile Knowledge En- gineering Methodology. In Proceedings of the Ershov Memorial Conference, volume 4378 of LNCS, pages 424-430. Springer, 2006.
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: A Nucleus for a Web of Open Data. 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, pages 722-735, 2007.
3. Berendt, B., Mobasher, B., Spiliopoulou, M., Wiltshire, J.: Measuring the accuracy of sessionizers for web usage analysis. In Workshop on Web Mining at the First SIAM International Conference on Data Mining, pages 7-14, April 2001.
4. Bizer, C.: The Emerging Web of Linked Data. IEEE Intelligent Systems, 24(5):87-92, 2009.

[3] http://data.semanticweb.org/

5. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. International Journal on Semantic Web and Information Systems (IJSWIS), 2009.
6. Cooley, R., Mobasher, B., Srivastava, J.: Web Mining: Information and Pattern Discovery on the World Wide Web. Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97), November 1997.
7. W3C Common Logfile Format: http://www.w3.org/Daemon/User/Config/Logging.htmlcommon, visited on June 9th 2010.
8. W3C Extended Common Logfile Format: http://www.w3.org/TR/WD-logfile.html, visited on June 9th 2010.
9. Davies, M.: Towards a Semantic Infrastructure for User Generated Mobile Services. In Proceedings of the European Semantic Web Conference 2009, volume 4825 of LNCS, pages 924-928. Springer, 2009.
10. Fernndez-Lpez, M., Gmez-Prez, A., Juristo, N.: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. AAAI-97 Spring Symposium on Ontological Engineering: Stanford, AAAI Press, 1997.
11. Hausenblas, M.: : Exploiting Linked Data For Building Web Applications. IEEE Internet Computing, 13(4):68-73, 2009.
12. Kotis, K., Vouros, A.: Human-centered ontology engineering: The HCOME methodology. Knowl. Inf. Syst.10(1), pages 109-131, 2006.
13. Mller, K., Hausenblas, M., Cyganiak, R., Grimnes, G. A.: Learning from Linked Open Data Usage: Patterns Metrics. Proceedings of the WebSci10: Extending the Frontiers of Society On-Line, Web Science Overlay Journal (United Kingdom). 2010, United Kingdom.
14. Pinto, H. S., Tempich, C., Staab, S., Sure, Y.: Distributed Engineering of Ontologies (DILIGENT). Semantic Web and Peer-to-Peer, Springer, 2005.
15. Simperl, E., Tempich, C.: How Much Does It Cost? Applying ONTOCOM to DILIGENT. Technical Report, FU Berlin, 2005.
16. Simperl, E., Tempich, C.: Ontology Engineering: A Reality Check. In Proceedings of the OTM Conferences volume 4825 of LNCS, pages 836-854. Springer, 2006.
17. Spiliopoulou, M.: Web Usage Mining for Web Site Evaluation. Comm. ACM 43 (8), 127-134, 2000.
18. Srivastava, J., Cooley, R., Deshpande, M., Tan, P.: Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. SIGKDD Explorations 1(2), 12-23, 2000.
19. Stojanovic, N., Stojanovic, L.: Usage-Oriented Evolution of Ontology-Based Knowledge Management Systems. In On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002. Lecture Notes In Computer Science, Vol. 2519. Springer, 2002.
20. Stojanovic, L., Stojanovic, N., Gonzalez, J., Studer, R.: OntoManager - A System for the Usage-Based Ontology Management. In CoopIS/DOA/ODBASE volume 2888. Springer, 2003.
21. Sure, Y., Studer, R.: On-To-Knowledge Methodology — Expanded Version. On-To-Knowledge deliverable, 17. Institute AIFB, University of Karlsruhe, 2002.
22. Tran, T. et al.: Lifecycle-Support in Architectures for Ontology-Based Information Systems. ISWC/ASWC, volume 4825 of LNCS, pages 508-522. Springer, 2007.
23. Zablith, F.: Dynamic Ontology Evolution. 7th International Semantic Web Conference (ISWC) Doctoral Consortium. 2008, Karlsruhe, Germany.