# A Generic Corporate Ontology Lifecycle

Markus Luczak-Rösch and Ralf Heese

Freie Universität Berlin, Berlin 14195, Germany,
{luczak,heese}@inf.fu-berlin.de

**Abstract.** Weaving the Semantic Web the research community is working on publishing publicly available data sources as RDF data on the Web. To facilitate the adoption of Semantic Web technologies in corporate environments some issues on ontology engineering have to be addressed, e.g., support unexperienced employees to work collaboratively on ontologies. Although, existing methodologies structure well the process of ontology engineering, we miss an adequate tool support. We describe the Lekapidia case study and derive requirements for ontology engineering in a corporate environment. Furthermore, we present an extended ontology lifecycle integrating ontology engineering and ontology usage.

## 1 Introduction

Within the past years the Semantic Web community has developed a comprehensive set of standards and data formats to annotate semantically all kinds of resources, e.g., documents and images. Currently, a main focus lies on integrating publicly available data sources and publishing them as RDF on the Web, e.g., linking open data [1]. In contrast, many corporate IT areas are just starting to engage in Semantic Web technologies. Early adopters are in the areas of enterprise information integration, content management, life sciences and government [2]. Applying Semantic Web technologies to corporate content is known as *Corporate Semantic Web*.

To facilitate the adoption of Semantic Web technologies in a corporate environment some issues have to be addressed. Although ontology engineering methodologies might be well thought out, we miss an adequate tool support for unexperienced participants and the economic-driven needs of companies.

In Section 2 we present the results of the Lekapidia case study. A team of six people modeled collaboratively an ontology on desert recipes. Afterwards, this process was examined under the theoretic foundations of the DILIGENT methodology and simulated using a wiki-based tool for ontology engineering. We analyze the results of the case study in Section 3 and derive requirements on tools for modeling ontologies in a corporate environment. As a result we present an innovative two parts ontology lifecycle. Furthermore, we describe a corporate Semantic Web scenario, developed in cooperation with the Projektron GmbH: a semantic ticket system.

## 2 Lekapidia Case Study

In this section we describe the Lekapidia case study which we use to evaluate wiki-based ontology engineering empirically and to derive requirements for ontology engineering in a corporate environment. First we outline the setting of the case study and describe how the DILIGENT methodology was applied to this scenario. Afterwards, we present our conclusions drawn from the case study which refers to the special needs of corporate environments. In the Lekapidia case study a team of six students were asked to develop collaboratively a semantic wiki for desert recipes including a desert recipes ontology, while other four people were working as the software engineers. The teams were free in choosing the tools to build the ontology, to develop the application, to control the collaborative work, and to produce a documentation. They used Protégé for modeling tasks and a conventional MediaWiki for discussions.We used the case study for a valuable proof-of-concept of the DILIGENT ontology engineering methodology [3]. DILIGENT [4] assumes that ontology engineering scenarios are characterized by unexperienced and unequally skilled participants working in a distributed environment having individual needs on the ontologies. It permits local adoptions of ontologies and also defines a structured and iterative process for user argumentation to discuss changes of the central ontology. After reaching a consensus a central board decides on the integration of these adoptions into the central ontology. We used a wiki-based tool, coefficientMakna, as an integrative support to facilitate DILIGENT. For that reason a semantic wiki system was extended to support two semantic models. One model for statements about normal wiki pages and on model for discussion pages and pages which are marked as development issues or ideas. The structured argumentations follow the DILIGENT argumentation ontology. Thus, it is possible to hold discussions related to design issues as well as ontology primitives. When a decision is made the ontology consensus is build automaticly by processing the arguments. It is possible to build multiple ontologies for multiple groups in that way.

The participants of the Lekapidia project had no experience in ontology engineering. Examining the activities of the working groups we discovered a lack of communication. Considering DILIGENT, we discovered that the argumentation-based approach has enabled the unexperienced users to discuss their design decisions in an intuitive way. However, it does not support application-dependent or scenario-oriented ontology engineering. Empirical studies such as [5] state that the adoption of wikis in enterprises fails due to missing participation and underestimated entrance barriers of wikis.

Lekapidia does not allow any proposition about a long-running ontology engineering lifecycle, because the developed ontology was not deployed in a productive system. Even though, the structure of the project is close to real-world ontology development processes. The simulation with coefficientMakna allows to draw conclusions from a concrete methodological approach. We come to the conclusion that wiki-based approaches do not adequately support ontology engineering tasks. We identify a strong gap between the currently accepted ontology engineering approaches, e.g. DILIGENT, which suggest the applicability of wiki-based ontology engineering, and the needs of ontology engineering in corporate contexts. Ontologies are commonly seen as an artifact without any application-dependence. We suggest it as the outcome of a process which is concurrent while the ontology is in use and which is not finished after a decisive iteration step. Thus, appropriate methodologies and tools are needed, which respect this perspective.

## 3 Requirements of Corporate Ontology Engineering

The Lekapidia case study results a lack of adequate methodologies and tools respecting the agile character of ontology engineering. In the following we present new requirements for ontology lifecycles in a corporate environment. A key feature of our lifecycle is that it includes a cycle feeding back requirements on the ontology derived from its usage.

### Corporate Ontology Lifecycle

Corporate Semantic Web refers to the usage of Semantic Web technology in a corporate environment. In a project of the same name we focus besides others on ontology engineering in a collaborative environment to increase the effectiveness of this process. A main advantage over realizing the Semantic Web is the controlled environment in a company. That allows us to name the boundaries of the setting as follows:

- Central allowance and control of the conceptualization
- Existing rules and workflows for employees
- Limited domain complexity
- Trust in semantic annotations

A main part of ontology engineering is the evolution of an ontology over life time. Figure 1 depicts our approach towards a corporate ontology lifecycle. The outer circle describes the engineering process by ontology engineers and domain experts while the inner one describes the adaption of the ontologies driven by usage requirements.

The ontology engineering process (outer circle) starts with the *creation/selection* phase by collecting and model knowledge fragments which results in a prototype ontology. This ontology is *validated* against the objectives. At the intersection point between the engineering and the usage cycles the engineers decide if the ontology reached a state to be used (*populated*) in the production system. If it does not meet the requirements or change requests arise from its usage the ontology engineers have to *evaluate* the current ontology. The *evolution/forward engineering* phase describes the task of changing the ontology to meet the new requirements. If an ontology has been populated to the production system then instances of concepts are generated by processing data and documents. The ontology is *deployed*. The *feedback tracking* phase is essential for adapting the

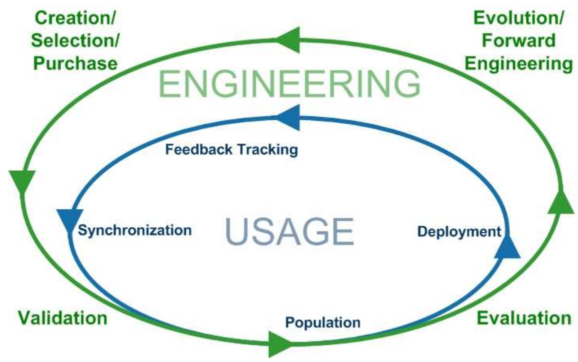


**Fig. 1.** The Corporate Ontology Lifecycle

ontologies to new requirements arising from its usage. A new requirement arises if a user gives explicit feedback, e.g., by arguing about concepts and relationships, or a system monitor generates conclusions by tracking the user behavior. The collected feedback and requirements are analyzed (*synchronization*) and if inconsistencies are recognized between the user's viewpoint and the ontology then ontology engineers start to adapt the ontology entering the outer cycle.

### Use Case Semantic Ticket System

We transfer our lifecycle model into practice to support a feasible evaluation. In cooperation with the Projektron GmbH we evaluate the proposed ontology lifecycle in a real-world scenario: a semantic ticket system. Projektron uses and sells a ticket system which can be used to collect requests of customers, e.g., bug reports for a software. Although tickets are annotated with keywords and categories, similar tickets cannot be detected automatically. A main reason is the usage of synonym terms, e.g., differences in the terminology of the customer and the operating company of the ticket system. The difference in terminology may originate from the adaption of a software product to the terminology of the customer. For example, the customer uses "job" or "issue" instead of "task". Having the information about similar tickets a software engineer could solve these tickets in a single run and, thus, save time.

Establishing an ontology lifecycle as described above helps to keep track of customer-specific changes in the terminology of a software product. Furthermore the ontology has to be adapted to the terminology of ticket submitters to be able to detect similar tickets in the system.

We name the following requirements for a semantic ticket system aiming at an integrative support for our lifecycle:

1. Expert design tools enable the operating company to develop valid and consistent ontologies.

2. (Semi-)Automatic knowledge acquisition performed by machine learning algorithms, amongst others, lessens the effort for the ontology engineers to develop valuable ontology prototypes, e.g., based on the common terminology of a customer.
3. (Semi-)Automatic knowledge retrieval lessens the additional work for the user to annotate relevant data at the run-time.
4. Lightweight extended communication platforms, e.g., forums or feedback forms, and the automatic recovery of user behavior feature the adaption of new requirements arising from ontology usage.
5. Alternative intuitive visualization, e.g., graph visualizer, provide an intuitive navigation for users of any level of experience and enable easy detection of similar concepts.
6. Interfaces for applications are necessary to allow a number of applications to integrate as much consistent ontologies as needed.
7. Ontology storage and versioning enable centrally administration and configuration of the interdependence, matching and alignment of the various coexisting ontologies.

We will extend and use these requirements in progress towards an architecture for a holistic corporate semantic web and implement a practical proof of concept which respects them. Thus, we do not just transfer Semantic Web technologies from web-scale to corporate-scale, but improve the foundations by innovative research results which start from another point of view, e.g. ontology engineering as a usage-oriented lifecycle.

## 4 Related Work

Current lifecycle models for ontologies [6–8] consider only one cycle consisting of the phases design, validation, population, deployment, maintenance, and evolution. To our best knowledge there exists only one approach dividing the process of ontology engineering into two orthogonal cycles [9]. In contrast, our approach assumes a spiral model. The NeOn project [10] also researches the development of ontologies and focuses on standardizing the interchange of knowledge between world-wide operating enterprises. We assume a corporate environment, e.g., the ontologies are developed to process data and documents in a company effectively.

## 5 Conclusion

In order to find an applicable set of functional requirements for an integrative tool-support for ontology engineering in corporate environments, we used the results of the Lekapidia case study to discard wiki-based tools for this task. Based on ideas of the DILIGENT methodology and assumed characteristics of corporate settings, we constructed an innovative ontology lifecycle. The semantic ticketing use-case provides the basis for functional requirements which comply with the lifecycle integrative.

We expect the corporate ontology lifecycle to evolve towards a generic model, which enables companies to estimate the complexity and the chances of a transition from conventional information systems to ontology-based information systems. The approach will suite intra-corporate as well as inter-corporate settings.

**Acknowledgement:** This work has been partially supported by the "Inno-Profile-Corporate Semantic Web" project funded by the German Federal Ministry of Education and Research (BMBF).

## References

1. W3C SWEO Community Project: Linking open data. http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData (2008)
2. Gartner, Inc.: Hype cycle for emerging technologies. http://www.gartner.com/it/page.jsp?id=495475 (2006)
3. Luczak, M.: Design and implementation of a wiki-based tool for collaborative ontology engineering. Master's thesis, Freie Universität Berlin (2007)
4. Pinto, S., Tempich, C., Staab, S., Sure, Y.: Distributed Engineering of Ontologies (DILIGENT). In: Semantic Web and Peer-to-Peer. Springer Verlag (2006) 301–320
5. Department of Personnel Economics and Human Resource Management of the University of Cologne: Wikis in enterprises. http://wikipedistik.de/survey/results.html (2008)
6. Gruninger, M., J., L.: Introduction. Commun. ACM **45**(2) (2002) 39–41
7. Novacek, V., Handschuh, S., Maynard, D., Laera, L., Kruk, S., Voelkel, M., Groza, T., Tamma, V.: Report and prototype of dynamics in the ontology lifecycle. Technical report, Galway, Ireland : Knowledge Web (2006)
8. Buitelaar, P.: NLP in the ontology life-cycle. http://www.lt4el.eu/content/files/ws_prague/eLearning-Prague.final.pdf (2007) Invited talk at the international workshop of the LT4eL project.
9. Staab, S., Studer, R., Schnurr, H.P., Sure, Y.: Knowledge processes and ontologies. IEEE Intelligent Systems **16**(1) (2001) 26–34
10. Tran, D.T., Haase, P., Lewen, H., Munoz-Garcia, O., Gómez-Pérez, A., Studer, R.: Lifecycle-support in architectures for ontology-based information systems. In: Proc. of the 6th Int. Semantic Web Conference (ISWC'07). (2007) 508–522