The DILIGENT knowledge processes

Article *in* Journal of Knowledge Management · October 2005

DOI: 10.1108/13673270510622474 · Source: DBLP

4 authors:

Denny Vrandečić
Google Inc.
88 PUBLICATIONS **6,201** CITATIONS
SEE PROFILE

H. Sofia Pinto
Inesc-ID
87 PUBLICATIONS **2,360** CITATIONS
SEE PROFILE

Christoph Dr. Tempich
Karlsruhe Institute of Technology
61 PUBLICATIONS **2,001** CITATIONS
SEE PROFILE

York Sure-Vetter
Karlsruhe Institute of Technology
207 PUBLICATIONS **8,061** CITATIONS
SEE PROFILE

The DILIGENT Knowledge Processes

Denny Vrandečić[1], Sofia Pinto[2], York Sure[1], Christoph Tempich[1]

[1] Institute AIFB, University of Karlsruhe
Karlsruhe, Germany
{denny,sure,tempich}@aifb.uni-karlsruhe.de

[2] Dep. de Engenharia Informática, Instituto Superior Técnico
Lisboa, Portugal
sofia.pinto@dei.ist.utl.pt

May 6, 2005

Abstract

Case Study

Purpose: The ontology engineering methodology DILIGENT is presented, a methodology focussing on the evolution of ontologies instead of the initial design, thus recognizing that knowledge is a tangible and moving target.

Approach: First we describe the methodology as a whole, then detailing one of the five main steps of DILIGENT. The second part describes case studies, either already performed or planned, and what we learned (or expect to learn) from them.

Findings: With the case studies we were able to discover the strengths and weaknesses of DILIGENT. During the evolution of ontologies, arguments need to be exchanged about the suggested changes. We identified those kind of arguments which work best for the discussion of ontology changes.

Research implications: DILIGENT recognizes ontology engineering methodologies like OnToKnowledge or Methontology as proven useful for the initial design, but expands them with its strong focus on the user-centric further development of the ontology and the provided integration of automatic agents in the process of ontology evolution.

Practical implications: With DILIGENT we distil the experience from a number of case studies and offer the knowledge manager a methodology to work in an ever changing environment.

Originality: DILIGENT is the first methodology to put focus not on the initial development of the ontology, but on the user and his usage of the ontology, and on the changes introduced by the user. We take the users own view serious and enable the feedback towards the evolution of the ontology, stressing the ontologies role as a shared conceptualisation.

1 Introduction

De-centralised knowledge management systems are becoming increasingly important. Especially the evolving Semantic Web [Berners-Lee et al., 2001] will foster the development of numerous use cases for this new paradigm. Therefore, working based on traditional, centralized knowledge management systems becomes infeasible. There are some technical solutions toward Peer-to-Peer knowledge management systems (e.g., [Bonifacio et al., 2003]), like the technically sophisticated solutions of the EU IST project SWAP (Semantic Web and Peer-to-Peer [Ehrig et al., 2003]). Still, the traditional methodologies for creating and maintaining knowledge structures appear to become unusable in distributed and decentralised settings, and so the systems that depend on them will fail to cope with the dynamic requirements of big or open user groups. What we need is a methodology for the *DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies* – and DILIGENT.

The knowledge structures underlying today's knowledge management systems constitute a kind of ontology that may be built according to established methodologies *e.g.* [Schreiber et al., 1999]. These methodologies have a centralized approach towards engineering knowledge structures requiring *knowledge engineers*, *domain experts* and others to perform various tasks such as a *requirement analysis* and *interviews*. While the user group of such an ontology may be huge, the development itself is performed by a — comparatively — small group of domain experts who *provide the knowledge* that is to be modelled, and ontology engineers who *structure and formalize* it. Almost none of the actual users, if any at all, are involved.

In the SEKT British Telecom use case the modelling of the topic hierarchy will be one premium example of a knowledge structure, that neither can be created *a priori* by some domain experts nor may we regard it as being stable. Here we see that the ontology engineering will take place in a Distributed, evolvInG and Loosely-controlled setting. With DILIGENT we provide a process template suitable for distributed engineering of knowledge structures that we plan to extend towards a fully worked out and multiply tested methodology in the long run. In this article we highlight on a case study we have performed under the SWAP project, where DILIGENT was used in a virtual organization.

DILIGENT comprises five main activities of ontology engineering: **build**, **local adaptation**, **analysis**, **revision**, and **local update**. We will provide a sketch of this activities in the next section, and then take an exemplary, more detailed look at one of the steps (*local adaptation*).

Afterwards we take a look at the case studies, where DILIGENT was applied, and at the experiences we learned in them. We will end with a look at the next case studies we will use DILIGENT in, and the kind of refinement we hope to achieve there.

DILIGENT is not constrained to a certain ontology formalism or language. The methodology covers the whole range of possible ontologies, starting with simple taxonomies, vocabularies and topic hierarchies (represented as instances of an topic ontology) up to foundational ontologies with many axioms.

2 DILIGENT process

2.1 Process overview

Knowledge sharing in dynamic environments requires an ontology engineering process that can cope with the frequently changing user needs. [1] Therefore, we have drafted a template of such a process. The result, which we call DILIGENT, is described in the following.

The users: In DILIGENT there are several experts, with different and complementary skills, involved in collaboratively building the same ontology. In the context of the Semantic Web and other non-centralised environments they may even belong to competing organizations and be geographically dispersed. Typically the domain experts involved in building the ontology are also its users. However, most ontology users will typically not build or modify it, that is, the community of users is much larger than the community of active builders.

Birds-eye view: An initial ontology is built by a small group of builders. This ontology is made available and users are free to use and modify it locally for their own purposes. There is a central board that maintains and assures the quality of the shared core ontology. This central board is also responsible for updating the core ontology. However, updates are mostly based on changes re-occurring at and requests by *locally* working users. Therefore, the board only *loosely controls* the process. Due to the changes introduced by the users over time that entail the changes introduced by the board, the ontology *evolves*. Let us now survey the DILIGENT process at a finer level of granularity. DILIGENT comprises five main steps: (1) **build**, (2) **local adaptation**, (3) **analysis**, (4) **revision**, (5) **local update** (*cf.* Figure 1).

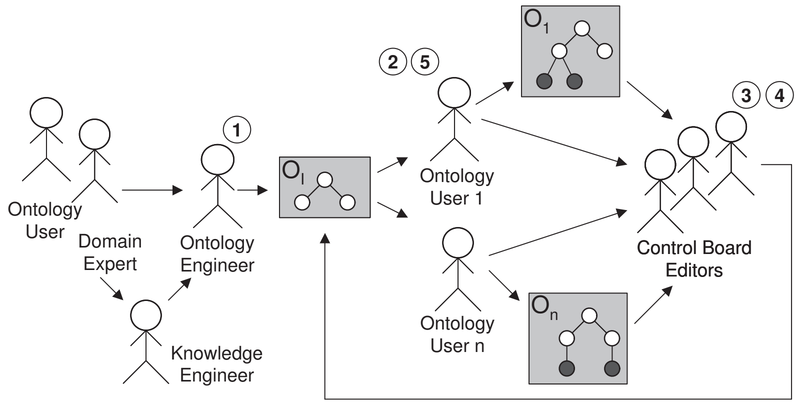


Figure 1: Roles and functions in distributed ontology engineering

Build. The process starts by having *domain experts*, *users*, *knowledge engineers* and *ontology engineers* **build** an initial ontology. In contrast to existing ontology engineer-

[1] In fact, we conjecture that the majority of knowledge sharing cases falls into this category.

ing methodologies (cf. [Staab et al., 2001, Gangemi et al., 1998, Gómez-Pérez et al., 2003, Pinto and Martins, 2001, Uschold and King, 1995]), we do not require completeness of the initial shared ontology with respect to the domain. The team involved in building the initial ontology should be relatively small, in order to find a small and consensual first version of the shared ontology in an easier way.

Local adaptation. Once the core ontology is available, users work with it and, in particular, adapt it locally to their own needs. Typically, they will have their own business requirements and correspondingly change their local ontologies [Noy and Klein, 2003, Stojanovic et al., 2002]. In their local environment, they are also free to change the reused core ontology. However, they are not allowed to directly change the core ontology. The control board collects change requests to the shared ontology and logs local adaptations (either continuously or at control points).

Analysis. The board **analyses** the local ontologies and the requests for changes and tries to identify similarities in users' ontologies. We expect tools that allow an efficient analysis of the change requests. Since not all of the changes introduced or requested by the users will be introduced to the shared core ontology,[2] a crucial activity of the board is deciding which changes are going to be introduced in the next version of the shared ontology. The input from users provides the necessary arguments to underline change requests. A balanced decision that takes into account the different needs of the users and meets their evolving requirements[3] has to be found. Moreover, this decision must follow good knowledge representation practices.

Revise. The board should regularly **revise** the shared ontology, so that local ontologies do not diverge too far from the shared ontology. The goal of the revision is to realign the ontology with the obvious user needs and thus gaining higher acceptance, 'sharedness' and less local differences. Therefore, the board should have a well-balanced and representative participation of the different kinds of participants involved in the process: knowledge engineers, domain experts, ontology engineers and users. In this case, users are involved in ontology development, at least through their requests and re-occurring improvements and by evaluating it, mostly from an usability point of view (understanding, actual advantage of use). Domain experts are responsible for evaluating it from a domain point of view (does it represent the domain, or does it contain factual errors?). Knowledge engineers in the board are responsible for evaluating the ontology, mostly from a domain and technical point of view (efficiency, standard conformance, logical properties like satisfiability). Ontology engineers too take responsibility for the technical point of views evaluation. They are also one of the major players in the analysis of arguments and in balancing them from a technical point of view. Another possible task for the controlling board, that may not always be a requirement, is to assure some compatibility with previous versions. Revision can be regarded as a kind of ontology development guided by a carefully balanced subset of evolving user driven requirements. Ontology engineers are responsible for updating the ontology, based on the decisions of the board. Revision of the shared ontology entails its evolution.

Local update. Once a new version of the shared ontology is released, users can **up-**

[2]The idea in this kind of development is not to merge all user ontologies.

[3]This is actually one of the trends in modern software engineering methodologies (see Rational Unified Process).

date their own **local** ontologies to better use the knowledge represented in the new version. Even if the differences are small, users may rather reuse *e.g.* the new terms instead of using their previously locally defined terms that correspond to the new terms represented in the new version.

3 Local Adaptation: Detailed view

In order to provide detailed guidance for possible participants in this kind of processes and to identify potential technical support for each process step, we have analysed them in detail. We exemplify the result of this analysis for the *Local adaptation* step, because here technology developed within the SEKT project will have its biggest impact.

The analysis includes the identification of (1) major roles, (2) input and (3) output information, (4) decisions and (5) actions within the process step.

Roles. The actors involved in the local adaptation step are users of the ontology. They use the ontology to retrieve knowledge or information, e.g. documents which are related to topics modelled in the ontology or more structured data like the list of projects an employee was involved in. Though information gathering is not their main objective, they do need the information to fulfil their individual tasks. **Input.** Besides the common shared ontology in this step the information available in the local information space is used. These can be existing databases, ontologies, folder structures and documents, as well as an ontology-based index used for classifying their information.

Output. The output of this step is a locally changed ontology which better reflects users needs. Each change is supported by arguments explaining the reasons for it. We emphasise that changes are not propagated to the shared ontology. Only in the *analysis step* does the board gather all ontology change requests and corresponding arguments for the common shared ontology to evolve (in the *revision step*).

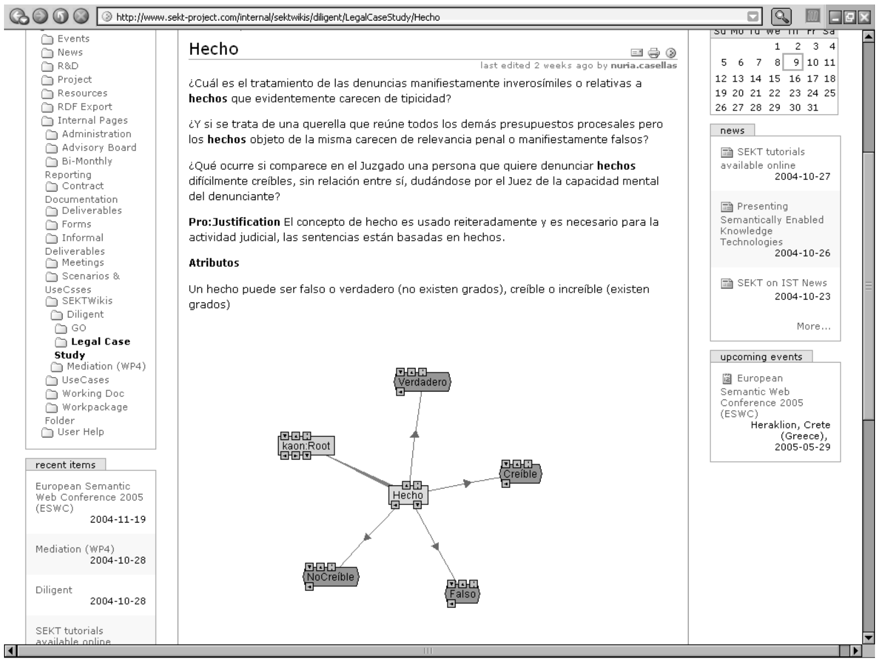
Decisions. The users decide which changes to introduce into their ontology: if and where in the hierarchy new concepts are needed, and which relations a concept should have. They must also provide reasons why they made those decisions.

Actions. To achieve the desired output the user takes different actions namely: *Understand shared ontology*, *Identify similarities between own and shared conceptualization*, *Identify missing conceptualizations*, *Change conceptualization* and finally *Organize local knowledge according to the conceptualization*.

The last three actions of the process are performed in a cyclic manner until a new common ontology is released by the board and the whole process starts anew.

The single actions performed manually would require a grounded understanding of ontologies and their underlying formal representation. We cannot expect such knowledge from all actors participating in the process. The process should rather be integrated seamlessly in the environment the user works in. Hence we now describe the available technology to support each action.

Understand shared ontology. An ontology should represent a shared conceptualization of the world. In fact a completely shared ontology can never be engineered, since different people have varying interpretations of the real world and the ontology to be



Hecho

last edited 2 weeks ago by **nuria.casellas**

¿Cuál es el tratamiento de las denuncias manifiestamente inverosímiles o relativas a **hechos** que evidentemente carecen de tipicidad?

¿Y si se trata de una querella que reúne todos los demás presupuestos procesales pero los **hechos** objeto de la misma carecen de relevancia penal o manifiestamente falsos?

¿Qué ocurre si comparece en el Juzgado una persona que quiere denunciar **hechos** difícilmente creíbles, sin relación entre sí, dudándose por el Juez de la capacidad mental del denunciante?

Pro:Justification El concepto de hecho es usado reiteradamente y es necesario para la actividad judicial, las sentencias están basadas en hechos.

Atributos

Un hecho puede ser falso o verdadero (no existen grados), creíble o increíble (existen grados)

Figure 2: Screenshot of the software used for discussing the legal ontology

built depends heavily on the use it is going to have, which may be different between different users. Therefore, as a first action one needs to relate one's own interpretation of the world to the shared conceptual model. Thus, the actor must learn where the different concepts are located in the ontology and how they are interrelated with other concepts. The ontology can be very complex, thus comprehension of the ontology depends mainly on its presentation. Different technologies can be used to provide the user with a context sensitive view on the ontology which does not overwhelm him. Relevant technology to support these actions are visualisation tools and ontology browsing facilities.

Identify similarities between own and shared conceptualization. After understanding the ontology, the user should try to identify the similarities between the own and the shared conceptualization.

To support this step technically we can use the formal conceptualization available on the local machines. To identify the degree of similarity we can use mapping methods [Ehrig and Sure, 2004] to find correspondences between locally available formal models and the shared ontology. Moreover, the documents can be used in part for ontology learning which then identifies concepts and relations based on the local text or the documents the user has browsed through.

Identify missing conceptualizations. Besides identifying similarities the same techniques can be applied in the subsequent step to support the user in identifying missing conceptualizations.

Depending on the scenario the user might have access to other user's ontologies and use their local adaptations as further input to identify missing concepts in his own conceptual model.

Add missing conceptualizations. After identifying missing conceptualizations the user must be enabled to introduce his or her changes. This is not so much a technical challenge than one of usability. The user should not be bothered with suggestions all the time and might not tolerate wrong suggestions. Since automated methods can not be 100% correct it depends on the user context when and how to apply the changes to the ontology.

At a subsequent stage, the board analyses the changes performed by the users. In order to do that and better understand change requests, the actor should provide reasons for each request. Again the rationales used within the automated methods can be used as input here. To further support the user in providing reasons, a part of the process model is an argumentation framework. This framework focuses the user on providing relevant arguments.

Organize local knowledge according to ontology. At this point the ontology should reflect the user's conceptualization. Now he can instantiate the ontology with the information available locally and hence contribute to collective knowledge. Text classification and natural language processing can be used to facilitate this action. The implementation of tools to support single actions must be done in close cooperation with the user and with a special regard towards usability. The steps are complex, but still it is crucial to find an easy way to enable the users to follow the process.

We have shown how different techniques developed in the course of the SEKT project can be used within the process to support the actors to follow the process. The output is a locally adapted ontology. Now the board can retrieve the changes and analyse the reasons underlying each change. The reasons can either be provided by tools evaluating the ontology and automatically suggesting changes or manually following the DILIGENT argumentation model [Tempich et al., 2005].

4 Experiences with DILIGENT

This section offers a short rationale for DILIGENT and explains its development. Moreover it presents some reasons why we believe that our ideas have actually been long proven practices.

4.1 Ontologies & Evolution

After having a fist version of the methodology, we noticed the close similarity of the suggested methodology to the practises as used in the biology domain. We used this to get a first impression of the applicability of DILIGENT. The taxonomy of living things is essential for those studying, classifying and understanding life. By analysing its evolution since 1735 one notes that it completely follows the 5-step DILIGENT process.

The taxonomy was first **built** by Linnaeus, based on observable features. Since the initial proposal, the taxonomy has changed a lot. Let us take the "highest" level: *kingdom*. Initially two kingdoms were identified: *animals* and *plants*. When microorganisms were discovered, the moving ones were classified in the *animals* kingdom and the colored (non moving) ones in the *plants* kingdom. A few of them were classified in both kingdoms. To more easily identify organisms in both classes, Haeckel (1894) proposed a new kingdom: the *Protista* kingdom. Users were **locally adapting** the taxonomy for their own purposes.

Given the advances in knowledge about an ever growing number of life forms, and the difficulty of sharing an up-to-date knowledge with all stakeholders about so many life forms, several problems in designing and managing this complex and dynamic taxonomy arose. For some time now, names of plants and animals have been controlled by different **boards**. They receive requests for changes, **analyse** them, balance pros and cons, decide upon the most adequate changes to introduce and **revise** the taxonomy accordingly. Once a new version is made available users should **locally update** and use it.

So, the evolution of the taxonomy is driven by a specialized set of users, so called taxonomists, and the revision is loosely controlled by appropriate boards, that make new versions available for all users.

4.2 Testing local adaptation

This case study following a DILIGENT approach a took place with seven peers. It was about the tourism domain of the Balearic Islands. The needs of the tourism industry there, which accounts for 80% of the islands' economy, are best described by the term 'coopetition'. On the one hand the different organizations *compete* for customers against each other. On the other hand, they must *cooperate* in order to provide high quality for regional issues like infrastructure, facilities, clean environment, or safety — that are critical for them to be able to compete against other tourism destinations.

To collaborate on regional issues a number of organizations now collect and share information about *indicators* reflecting the impact of growing population and tourist fluxes in the islands, their environment and their infrastructures. Moreover, these indicators can be used to make predictions and help planning. For instance, organizations that require "*Quality & Hospitality management*" use the information to better plan, e.g., their marketing campaigns. As another example, the governmental agency IBIT[4], the Balearic Government's co-ordination center of telematics, provides the local industry with information about *new technologies* that can help the tourism industry to better perform their tasks.

Due to the different working areas and objectives of the collaborating organizations, it proved impossible to set up a centralized knowledge management system or even a centralized ontology. They asked explicitly for a system without a central server, where

[4]`http://www.ibit.org`

knowledge sharing is integrated into the normal work, but where very different kinds of information could be shared with others.

Now let's see how the DILIGENT methodology was implemented.

Building. Two knowledge engineers and two ontology engineers were involved in building the first version of the shared ontology. The former also received additional training, so that, later on, they were able to act as ontology engineers on the board as well.

The ontology engineering process started by identifying the main concepts to be represented in the ontology through the analysis of competency questions and their answers [Grueninger and Fox, 1994]. From the competency questions we created a first ontology with 22 concepts and 7 relations. This took about three hours.

Local Adaptation. The developed ontology was distributed among the users and they were asked to extend it with their local structures. With the assistance of the developers they created on average 14 concepts. The users mainly created sub concepts of concepts in the initial ontology. In other cases they created their own concept hierarchy and aligned it with the initial ontology.

Analyzing. The members of the board gathered the evolving structures and analyzed them. Here are some of the observations we made:

The users matched and extended the provided concepts of the initial ontology with their own concepts. They also extended the ontology with a missing top level concept (*project*). We also observed that some concepts were readily extended, while others were not even used. Some of the concepts were relabeled, thus providing themselves with a more comfortable interface, as they could continue to use their own terminology.

From the discussions with the domain experts we have the impression that the local extensions indeed are a good indicator for the evolutionary direction of the ontology.

Revision. The board extended the core ontology where it was necessary and performed some renaming. More specifically the board introduced one further top level concept (*Project*) and four new subconcepts.

Local update. The extensions to the core ontology were distributed to the users. The feedback of the users was in general positive.

Further details on the case study can be found in [Sure et al., 2005].

4.3 Participants reach agreement by focused argumentation

The complexity of the maintenance tasks correlates with the level of acceptance the provided core ontology and its revisions receive. In order to increase acceptance, we want to provide easily accessible rationales for the decisions of the board and create a sense of community in building and evolving the ontology.

We formulated the hypothesis, that an appropriate argumentation framework can facilitate the ontology engineering and evolution process. We studied the arguments and practices used in the evolution of the taxonomy of living beings which seem to point in the same direction. In order to substantiate this notion, we pursued an experiment in a computer science department.

We performed the experiment in two steps: in the first, participants were not constrained in any way; in the second, participants were asked to use a subset of arguments, and were given stricter rules, to conduct their discussions. The task in both sessions was to build an ontology, which represents the knowledge available in the research group, can be used for internal knowledge management, and makes the research area comprehensible for outsiders. Both experiments lasted each for one and a half hours. Concepts were only added after argumentation and some consensus was achieved.

The participants met in a virtual chat room. A moderator was responsible to remind people to stay on the subject and to include the modeling decisions into the formal ontology which was visualized on a web page. At this stage very few procedural and methodological restrictions were a-priori imposed. They agreed only on a few concepts, while the discussion was very chaotic and hard to follow, but gave us a good idea on how to improve the setting.

By analyzing the discussion we identified the kind of arguments which had most impact on the creation of the ontology: *elaboration, evaluation/justification, examples, counter examples*, *alternatives*.

Further on, with respect to the experimental setup we identified the following problems:

- Participants started too many discussion threads and lost the overview,
- the discussion proceeded too fast, hence not everybody could follow the argumentation,
- it was difficult for the moderator to intervene, and
- there was no explicit possibility to vote or make decisions.

Even in this setting – where participants shared a very similar background knowledge – the creation of a shared conceptualization *without any guidance* was almost impossible, at least very time consuming. We concluded, that a more controlled approach is needed with respect to the process and moderation.

In the second experiment participants were asked to extend the ontology built in the first round. In this phase the formalism to represent the ontology was fixed. The most general concepts were also initially proposed, to avoid philosophical discussions. For the second round only the arguments *elaboration*, *examples*, *counter examples*, *alternatives*, *evaluation/justification* were allowed. The Rhetorical Structure Theory [Mann and Thompson, 1987], RST, allows for a far bigger number of argument classes, but we decided to restrict ourselves to those with the most impact. The participants in the second case study joined two virtual chat rooms. One was used for providing topics for discussion, hand raising and voting. The other one served to exchange arguments. When the participants - the same as in the first experiment - wanted to discuss a certain topic *e.g.* the introduction of a new concept, they had to introduce it in the first chat room. The topics to discuss were published on a web site, and were processed sequentially. Each topic could then be expatiated with the allowed arguments. Participants could provide arguments only after hand raising and waiting for their turn. The participants decided autonomously when a topic was sufficiently discussed, called for a vote and thus decided how to model a certain aspect of the domain. The evolving ontology was again published on a web site. The moderator had the same tasks as in the first experiment, but was more restrictive. Whenever needed, the moderator called for an example of an argument to enforce the participants to express their wishes clearly.

As expected the discussion was more focused, due to the stricter procedural rules. Agreement was reached quicker and a much wider consensus was reached. With the stack of topics which were to be discussed (not all due to time constraints), the focus of the group was kept.

The restricted set of arguments is easy to classify and thus the ontology engineer was able to build the ontology in a straightforward way. It is possible to explain to new attendees why a certain concept was introduced and modeled in such a way, by simply pointing to the archived focused discussion. It is even possible to state the argumentation line used to justify it. The participants truly shared the conceptualization and did understand it. In particular, in conflict situations, when opinions diverged, the restriction of arguments was helpful. In this way participants could either prove their view, or were convinced.

More details on the argumentation model of DILIGENT can be found in [Tempich et al., 2005].

5 Lessons learned

The analysis of the arguments and process driving the evolution of the taxonomy of living beings showed a high resemblance to the 5-step DILIGENT process and its accompanying argumentation framework. The two case studies tested the two parts of DILIGENT .

The tourism case study helped us to generally better comprehend the use of ontologies in a distributed environment. All users viewed the ontology mainly as a classification hierarchy for their documents. The ontology helped them share their own local view on documents with other users. Thus finding documents became easier.

Currently we doubt that our manual approach to analysing local structures will scale to cases with many more users. Therefore, we look into technical support to recognize similarities in user behaviour. Furthermore, the local update will be a problem when changes happen more often. Last, but not least, we have so far only addressed the ontology creation task itself – we have not yet measured explicitly if users get better and faster responses with the help of DILIGENT-engineered ontologies. All this remains work to be done in the future.

Despite the technical challenges, the users provided very positive feedback when asked, pointing to the integration into their daily workflow through the use of a tool built by us and embedded in their work environment, which could easily be used.

With our second case study, we have given strong indication – though not yet full-fledged evidence – that a restriction of possible arguments can enhance the ontology engineering effort in a distributed environment. The restricted set of arguments will allow for reasoning over the debates, and maximises the efficiency of the debate by providing only those kind of arguments which proved to be the strongest in the debate. In addition, the second experiment underlines the fact that appropriate social management procedures and tool support help to reach consensus in a smoother way.

The process could certainly be enhanced with better tool support. Besides the argumentation stack, a stack for householding possible alternatives would be helpful. Arguments, in particular *elaboration*, *evaluation & justification* and *alternatives*, were discussed heavily. However, the lack of appropriate evaluation measures made it difficult, at some times, for the contradicting opinions to achieve an agreement. In that case, the argumentation should be focused on the evaluation criteria.

6 Future work

In the SEKT Digital Library case study, led by BT, we will apply the DILIGENT methodology for the evolution of the topic ontology within the individual information spaces, used for the personalization of the users experience and usage of the Digital Library.

The Digital Library provides a common portal allowing an unifying view on query results from heterogeneous content repositories, not bothering the user with syntactical differences or representational issues but instead enhancing the ability to find and understand relevant content. In order to allow the semantical integration of the heterogeneous content repositories, the content is mapped to a central information space, which, in DILIGENT terms this correspondents to the common shared ontology. Users currently are overwhelmed by the search possibilities in classical knowledge retrieval systems. To personalize the systems behaviour towards the user, information spaces are created which capture particular views upon the Library's contents. Spaces can be either private, to an individual user, or can be shared between groups of people, where the space holds a summary of all information about, say, *knowledge management*.

The Digital Library User Interface will allow for the easy and simple querying of the knowledge base. Parts of the ontology identified as being relevant for the query will be presented to the user and can be used for query refinement. The queries are logged and used in order to identify concepts missing from the shared ontology. These missing concepts can be introduced to the users individual information space, together with reasons for these suggested changes. Tools may also gather information or evaluate the ontology and suggest changes automatically, providing generated arguments.

Compiling these changes, i.e. gathering them from the users and agents and analysing and rendering them representable to the control board, will allow the control board to decided upon changes to be introduced into the common shared ontology. The control board is able to estimate the grade of acceptance among the users by reviewing the usage data and the changes suggested by the users. As it consists of ontology users as well as ontology engineers, it is able both to understand the impact for the further usage of the ontology and for the domain knowledges correctness, as well as the consequences for the underlying ontology driven framework.

In order to apply the changes, the framework evOWLution [Haase et al., 2004] will be used. evOWLution allows us to define constraints on the suggested changes, thus ensuring e.g. that the ontology is within OWL DLP.

7 Conclusion

It is now widely agreed that ontologies are a core enabler for the Semantic Web vision. The development of ontologies in centralized settings is well studied and established methodologies exist. However, current experiences from projects suggest, that ontology engineering should be subject to continuous improvement rather than a one time action and that ontologies promise the most benefits in decentralized rather than centralized systems. Hence, a methodology for distributed, loosely-controlled and dynamic ontology engineering settings is needed. The current version of DILIGENT is a major step towards such a methodology.

Acknowledgements. Research reported in this paper has been partially financed by EU in the IST project SWAP (IST-2001-34103), the IST thematic network OntoWeb (IST-2000-29243), the IST project SEKT (IST-2003-506826) and Fundação Calouste Gulbenkian (21-63057-B). In particular we want to thank Immaculada Salamanca and Esteve Lladó Martí from IBIT for the fruitful discussions and the other people in the SWAP team for their collaboration towards SWAPSTER. We thank Steffen Staab from University of Koblenz for his ideas and valuable feedback.

References

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 2001(5). available at http://www.sciam.com/2001/0501issue/0501berners-lee.html.

[Bonifacio et al., 2003] Bonifacio, M. et al. (2003). Peer-mediated distributed knowldege management. In van Elst, L., Dignum, V., and Abecker, A., editors, *Proceedings of the AAAI Spring Symposium "Agent-Mediated Knowledge Management (AMKM-2003)"*, Lecture Notes in Artificial Intelligence (LNAI) 2926, Berlin. Springer.

[Ehrig et al., 2003] Ehrig, M., Haase, P., van Harmelen, F., Siebes, R., Staab, S., Stuckenschmidt, H., Studer, R., and Tempich, C. (2003). The SWAP data and metadata model for semantics-based peer-to-peer systems. In *Proceedings of MATES-2003. First German Conference on Multiagent Technologies*, LNAI, Erfurt, Germany. Springer.

[Ehrig and Sure, 2004] Ehrig, M. and Sure, Y. (2004). Ontology mapping - an integrated approach. In Bussler, C., Davis, J., Fensel, D., and Studer, R., editors, *Proceedings of the First European Semantic Web Symposium*, volume 3053 of *Lecture Notes in Computer Science*, pages 76–91, Heraklion, Greece. Springer Verlag.

[Gangemi et al., 1998] Gangemi, A., Pisanelli, D., and Steve, G. (1998). Ontology integration: Experiences with medical terminologies. In Guarino, N., editor, *Formal Ontology in Information Systems*, pages 163–178, Amsterdam. IOS Press.

[Gómez-Pérez et al., 2003] Gómez-Pérez, A., Fernández-López, M., and Corcho, O. (2003). *Ontological Engineering*. Advanced Information and Knowlege Processing. Springer.

[Grueninger and Fox, 1994] Grueninger, M. and Fox, M. (1994). The role of competency questions in enterprise engineering. In *IFIP WG 5.7, Workshop Benchmarking. Theory and Practice*, Trondheim/Norway.

[Haase et al., 2004] Haase, P., Sure, Y., and Vrandecic, D. (2004). Ontology management and evolution – survey, methods and prototype. SEKT formal deliverable D3.1.1, Institute AIFB, University of Karlsruhe.

[Mann and Thompson, 1987] Mann, W. C. and Thompson, S. A. (1987). Rhetorical structure theory: A theory of text organization. In Polanyi, L., editor, *The Structure of Discourse*. Ablex Publishing Corporation, Norwood, N.J.

[Noy and Klein, 2003] Noy, N. and Klein, M. (2003). Ontology evolution: Not the same as schema evolution. *Knowledge and Information Systems*.

[Pinto and Martins, 2001] Pinto, H. S. and Martins, J. (2001). A Methodology for Ontology Integration. In *Proc. of the First Int. Conf. on Knowledge Capture (K-CAP2001)*, pages 131–138, New York. ACM Press.

[Schreiber et al., 1999] Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., van de Velde, W., and Wielinga, B. (1999). *Knowledge Engineering and Management — The CommonKADS Methodology*. The MIT Press, Cambridge, Massachusetts; London, England.

[Staab et al., 2001] Staab, S., Schnurr, H.-P., Studer, R., and Sure, Y. (2001). Knowledge processes and ontologies. *IEEE Intelligent Systems, Special Issue on Knowledge Management*, 16(1):26–34.

[Stojanovic et al., 2002] Stojanovic, L., Maedche, A., Motik, B., and Stojanovic, N. (2002). User-driven ontology evolution management. In *Proceedings of the 13th European Conference on Knowledge Engineering and Knowledge Management EKAW*, Madrid, Spain.

[Sure et al., 2005] Sure, Y., Tempich, C., Pinto, H. S., and Staab, S. (2005). A case study in supporting distributed, loosely-controlled and evolving engineering of ontologies (diligent). In *Intelligent Learning Infrastructures for Knowledge Intensive Organisations*. Idea Group Publishing, Inc. to appear.

[Tempich et al., 2005] Tempich, C., Pinto, H. S., Sure, Y., and Staab, S. (2005). An argumentation ontology for distributed, loosely-controlled and evolving engineering processes of ontologies (diligent). In Bussler, C., Davies, J., Fensel, D., and Studer, R., editors, *Second European Semantic Web Conference, ESWC 2005*, LNCS, Heraklion, Crete, Greece. Springer. to appear.

[Uschold and King, 1995] Uschold, M. and King, M. (1995). Towards a methodology for building ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95*, Montreal, Canada.