A SEMANTIC WIKI FOR COLLABORATIVE KNOWLEDGE FORMATION

Sebastian Schaffert, Andreas Gruber, Rupert Westenthaler

Knowledge-based Information Systems Group, Salzburg Research, Austria
sschaffe,agruber,rwesten@salzburgresearch.at

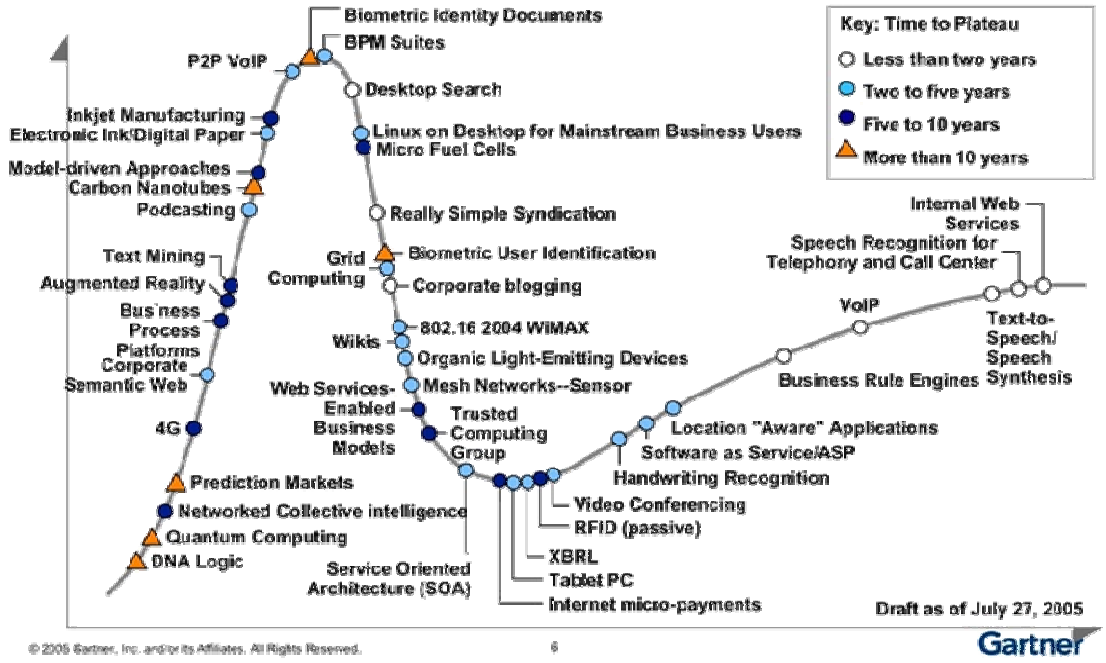
Abstract. *In this paper, we explore some of the apparent problems in adopting semantic technologies for content management and investigate a methodology and a tool to overcome these barriers. Our approach tries to benefit from both the networked collective intelligence of social software and the obvious need for methodologies and well-defined explicit semantics in our information and knowledge systems in order to support business processes.*

1. Introduction

The Semantic Web. In recent years, various approaches have been investigated in the so-called "Semantic Web" initiative, with a view to making the meaning of Web data – normally only amenable to human readers – accessible to machines. Indeed, Semantic Web research has become one of the major research endeavours in information technology world wide, covering a wide range of technologies. As a recent Gartner report shows (Figure 1), some of these technologies are now close to the "plateau of productivity" (e.g. "Business Rule Engines"), while some others are still in their infancy and just emerging (e.g. "Corporate Semantic Web"). It is particularly interesting that the term "Semantic Web" itself has disappeared in the 2005 report while it was at the peak of the hype cycle in 2004. This reflects the recent perception of the Semantic Web as a collection of technologies (some of which are more mature than others), rather than a single technology.

Although the Semantic Web has recently been gaining significant attention from both academia and industry, the amount of knowledge available in formal representations that are accessible by machines is still small compared to the "traditional" Web. A major reason for this is that Semantic Web technologies and tools require considerable technical expertise, and are thus not well suited for users outside the field of computer science. Also, knowledge engineering tools for the Semantic Web are currently mostly single user and do not provide good support for the collaborative creation of formal knowledge. This makes it hard for domain experts and knowledge engineers to work together on knowledge engineering tasks.

Figure 1 Gartner's Hype Cycle Special Report for 2005, http://www.gartner.com



Social Software. On the other hand, *social software*, like wikis or weblogs, has in the last few years significantly simplified the creation of content on the traditional Web (Wikipedia being the most well-known representative). Also, social software is heavily based on collaboration between users. In a sense, social software helps to realise the original vision of the Web as a space where anyone can participate. With the dynamic nature and the growing amount of the content, however, there is also a growing need to make the semantics of this space at least partly machine accessible, so that efficient searching and navigation of the content becomes feasible.

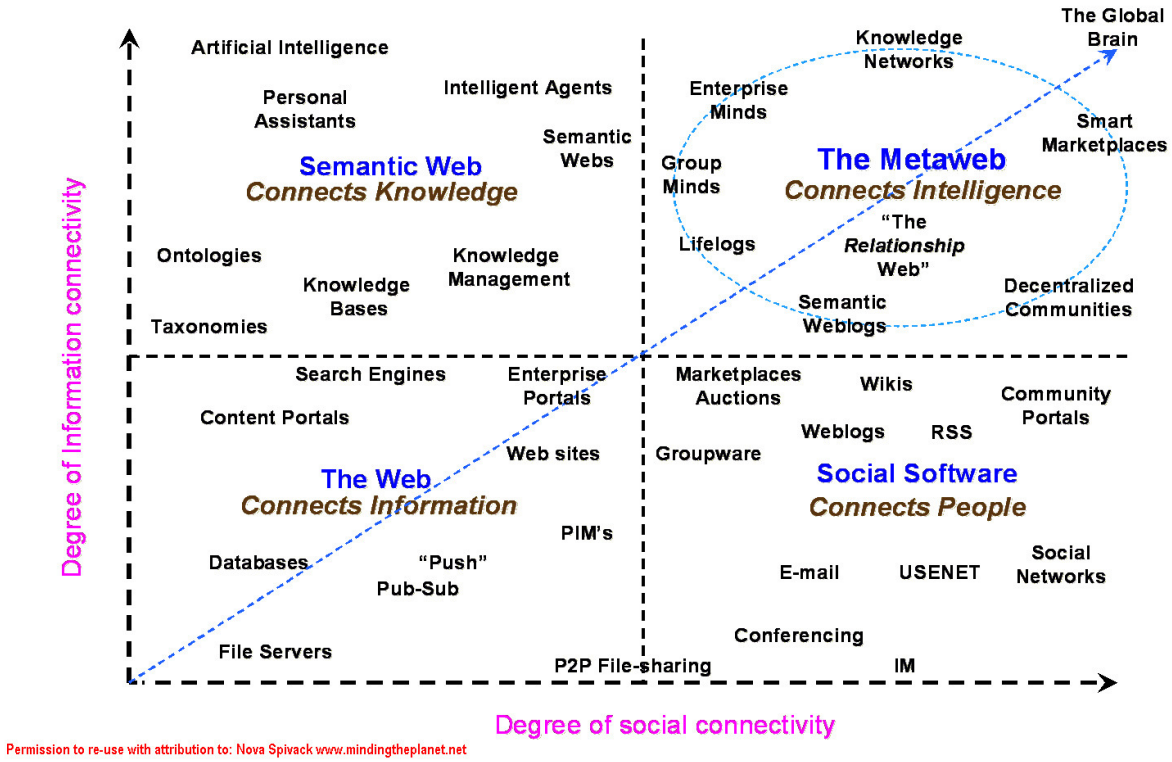
Beyond the Semantic Web and Social Software. Given these developments, combining the Semantic Web with Social Software appears natural. On the one hand, social software can support the creation of semantically enriched content by lowering technical barriers and by allowing domain and knowledge experts to collaborate. On the other hand, social software can itself benefit from semantic annotations that allow easier searching, navigation, and integration of content.

Indeed, various sources see this as the next big step in the evolution of the Web. For example, Nova Spivack proclaimed in 2003/2004 the “Metaweb” as a convergence of the Web, Social Software, and the Semantic Web (Figure 2).[1]

---

1 http://novaspivack.typepad.com/nova_spivacks_weblog/2004/03/the_metaweb_is_.html

Figure 2 Nova Spivak: The Metaweb



In this article, we aim to investigate how social software can help the Semantic Web to "take off". We begin with a short summary of different knowledge representation formalisms (Section 2) and then present preliminary results of work done towards a methodology that helps users to decide which kind of knowledge representation and which technology to choose (Sections 3 and 4). We then introduce a semantic wiki system called IkeWiki that we developed as a tool for collaborative creation of formal knowledge (Section 5).

2. Knowledge Models

"Knowledge" can be represented on many different levels of formality and in a wide variety of formalisms. Although languages like OWL or RDF are considered state-of-the-art for the Semantic Web, other formalisms (such as XML, XML Schema, Topic Maps, and even relational databases) are in many situations better suited for knowledge representation.

Also, although the term "Semantic Web" is commonly associated with very complex and formal ontologies, the most successful knowledge models tend to be very simple and specific. For example, the well-known Dublin Core and FOAF models (as well as LOM and News-ML) are good examples of simple, yet successful knowledge models in their domains. Thesauri like WordNet and

DMOZ are also successfully used in state of the art software applications. As Jim Hendler proclaims[2], "a little semantics goes a long way".

On the other hand, complex and abstract knowledge models like DOLCE [9], SUMO [10], or OpenCyC [11] are not widely accepted in commercial settings, but are used in research projects and prototypes, and will possibly gain more acceptance in the near future.

Knowledge models are known under many different names. Foundational, domain, sector, group, or application ontology are used to refer to knowledge models based on their level of generality, granularity and scope. Term list, thesaurus, taxonomy, and ontology segment the range of knowledge models according to the richness of their formal semantics .

3. Ontology classification with a 3D Matrix

The first step in designing a knowledge-based system is to identify what kind of knowledge is actually being modelled. Based on this insight, a knowledge engineer can then make the decision on what modelling languages and tools to use. In general, we distinguish two different types of knowledge models, based on their intended usage:

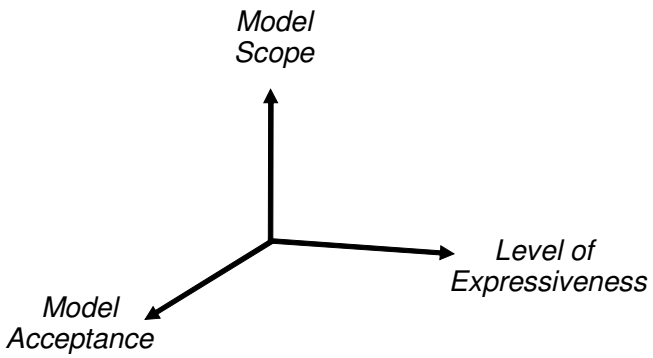
- *ontologies* are semantic models describing a thematic part of the world. Top-level ontologies concentrate on the semantic principles, whereas domain ontologies concentrate on features specific to a certain domain. To avoid redundancy and the need for ontology alignment, the scopes of different domain ontologies should not overlap.
- *application/service profiles* aim to capture all semantics needed to provide some functionality. Typically, they combine semantic statements from different domains and therefore, from different ontologies. The scopes of different application profiles typically overlap, especially if the applications are in the same business sector.

In the next section, we propose a scheme for classifying ontologies using three properties: scope, expressiveness, and acceptance. Our model is based on Nicola Guarino's traditional and well known scheme for classifying ontologies [1], which is widely accepted and used as a two-dimensional matrix (scope and formalisation). Our extension splits the "scope"-dimension into two aspects, "model scope" and "model acceptance".

---

[2] as conference chair in the opening speech of the 2003 International Semantic Web Conference; Sanibel Island, Florida, USA, October 2003

Figure 3 3D Matrix

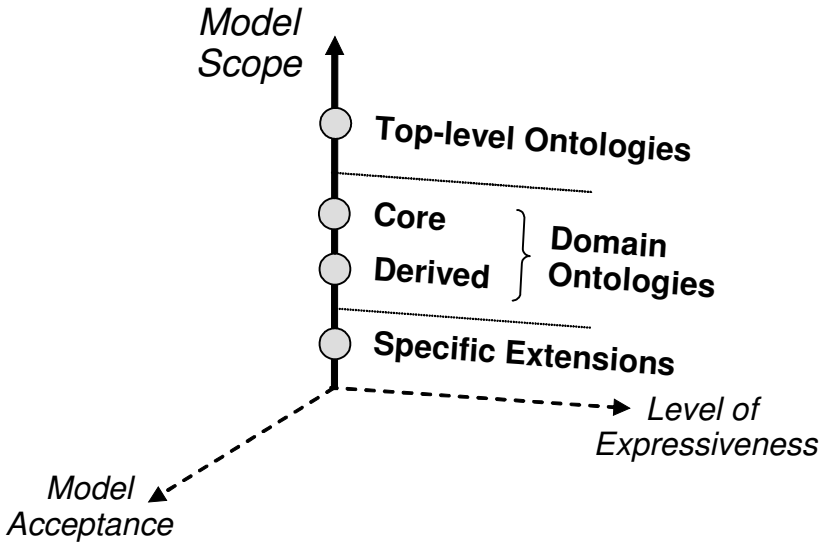


The following sections explain the three dimensions in more detail. The final section provides some example classifications of well known knowledge models.

3.1. Scope

The model scope refers to the areas of semantics that is of interest to a model. It is important to understand that the scope does not restrict the observable individuals, but only the observable features of such individuals. Individuals with no features inside the model scope cannot be described and are therefore invisible for the model.

Figure 4 Model Scope Dimension



The classification scheme of the scope dimension is intended to be used for ontologies. Application profiles will typically include semantics of all of the different classifications defined in the scope dimension.

The scope of *top-level ontologies* includes axiomatic concepts and relations of global relevance, such as space, time, matter, event, action, etc. Such ontolo-

gies are often called foundational ontologies, but this term usually indicates a higher level of expressiveness and thus does not really convey the intended meaning. For example, a simple list with axiomatic terms can also serve as a top-level ontology. As there may be several top-level ontologies with the same scope, it means that the different top-level ontologies have to be integrated. In practice, one would opt for a single top-level ontology when building application profiles.

The scope of *domain-level ontologies* includes generic as well as specific concepts, as well as relations of a specific thematic area. Tasks, weather, date/time zones and food are examples of rather generic domains; wine, beer, and pizza are also domains, but more specific ones. The scope of domain ontologies with different domains (e.g. wine and beer) should not overlap with each other. Different domain ontologies about the same domain will have similar scopes and therefore describe the same knowledge with different semantics. This means that alignment is typically only necessary between domain ontologies describing the same domain (typical example: an OO-domain model and a relational data model might require alignment via O/R mapping frameworks). Within domain ontologies, we distinguish two sub types:

- core domains, defining semantics which are not dependent on semantics of other domains;
- derived domains, combining and extending definitions of other "parent" domains; the scope of derived domains should be more specific than the union of the scopes of all "parent" domains.

For the definition of an application profile, it is important that the domain ontologies participating in defining the profile should not have any overlapping scopes. This implies that all domain ontologies should use the same (or no) top-level ontology and that only one ontology can be used for each domain.

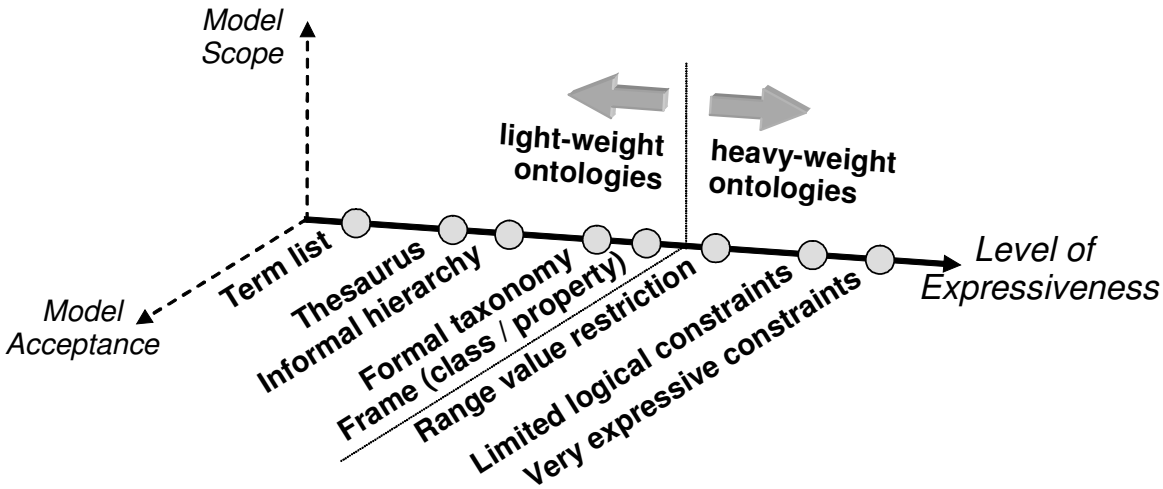
Specific extensions to a domain ontology build the last scope classification category. Such extensions are usually defined in the context of a distinct usage of a more general knowledge model. Typically, this includes additional constraints to existing resources or new concepts and properties which are only valid in the context of the actual application.

Independent of possible application profiles, ontologies as described above constitute an interdependent framework which we call "ontology stack". Although such stacks are not common at present, we strongly believe that the definition of such frameworks will become more necessary in the future to improve granularity, interoperability, and re-usability of ontologies. Such ontology stacks would also improve semantic interoperability between application profiles which are based on the same stack.

3.2. Level of Expressiveness

The expressiveness dimension (dubbed "level of formalism" in [1]) is already well defined. In our 3D Matrix, we use an existing classification scheme for the expressiveness dimension. We add the aspect of purpose to this category, providing a further criterion for the required expressiveness. This scheme by O. Corcho [2] is shown in the following figure.

Figure 5 Model Expressiveness Dimension



Corcho distinguishes between the two main groups – light-weight ontologies and heavy-weight ontologies – and defines eight sub categories based on their level of expressiveness:

1. a term list or controlled vocabulary contains a list of keywords. Such lists are typically used to restrict possible values for properties of some kind of instance data in the domain;
2. a thesaurus also defines relations between terms, e.g. proximity of terms;
3. an informal taxonomy defines an explicit hierarchy (generalisation and specialisation), but there is no strict inheritance, i.e. an instance of a subclass is not necessarily also an instance of the superclass. Most available dictionaries like DMOZ are members of this category;
4. a formal taxonomy, in contrast, defines a strict inheritance hierarchy;
5. a frame or class/property based ontology is similar to object-oriented models. A class is defined by its position in the subclass hierarchy and its properties. Properties are inherited by subclasses and realised in instances;
6. a range value restriction defines, in addition, restrictions for the defined properties. Possible restrictions are data type or domain;
7. by using logic constraints, property values may be further restricted;
8. very expressive ontology languages often use first-order logic constraints. These constraints may include disjoint classes, disjoint coverings, inverse relationships, part-whole relationships, etc.

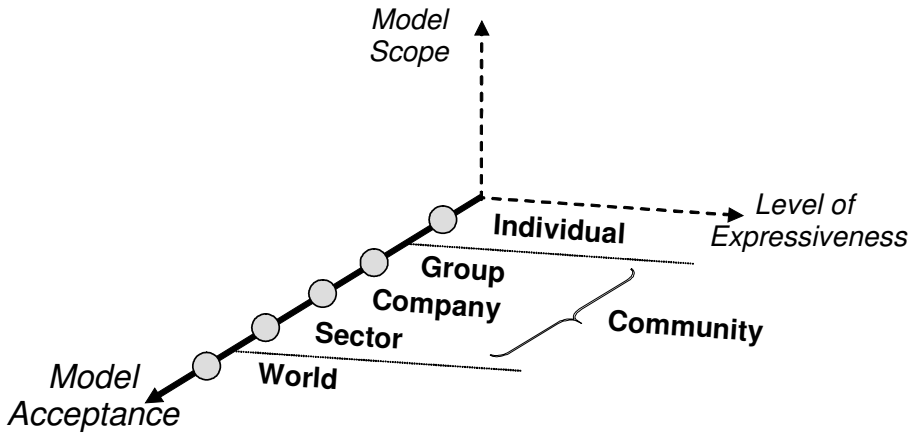
The adequate level of expressiveness for a knowledge model is driven by the requirements of its usage. According to John Harris [3], there are five layers in his "anatomy of an ontology". Apart from his layers zero (ontology definition language) and four (operational data), there are three layers which define possible purposes for a semantic model:

1. a knowledge model (or a part of such a model) can be used as reference data. Typically, there are no instances for this use case: the semantics defined in the model is only used to annotate instances built by another part of the model. This is e.g. common for term lists, thesauri, and taxonomies. Knowledge models used for such purposes are often referred to as "controlled vocabularies".
2. a knowledge model can be used as a data structure. Here, the model is mainly used to work with operational data of the application, which means that instances are created based on the semantics defined in the model. Such knowledge models are often referred to as "data models" and are typically implemented by database management systems. The expressiveness needed for a knowledge model used as data structure is approximately the border between light- and heavy-weight ontologies. In practice, relational, object-oriented, and hierarchical models are used to define such models, but RDFS and OWL-DL are used more and more frequently.
3. a knowledge model can be used to make assertions or to define constraints. In this case, knowledge models and particular ontologies are used by the application to gain information about individuals by analysing constraints defined in the model e.g. by integrity checking and inferencing. This imposes the highest requirements on the expressiveness of the model. Models requiring these additional checks are usually referred to as domain models and are typically hard coded in the business logic of applications. The corresponding realisation, based on Semantic Web technologies, is to encode all semantics using a formal ontology language like first-order logic and to use a generic rule/inference engine to make the semantics operational.

3.3. Model Acceptance

The acceptance dimension is useful for classification of knowledge models w.r.t. two processes that are of importance for ontology engineering. The first aspect is to be clear about target communities of the application and its knowledge model. The second aspect deals with various methods of building consensus within a specific community. Involved communities include not only the intended users, but also the developers of a system. As far as web-based, distributed systems are concerned, wide acceptance is an important criterion of successful applications.

Figure 6 Model Acceptance Dimension



The acceptance dimension ranges from the individual, over different levels of communities, to a worldwide acceptance of the model. It is important to notice that every transition towards a higher level requires different methods and tools.

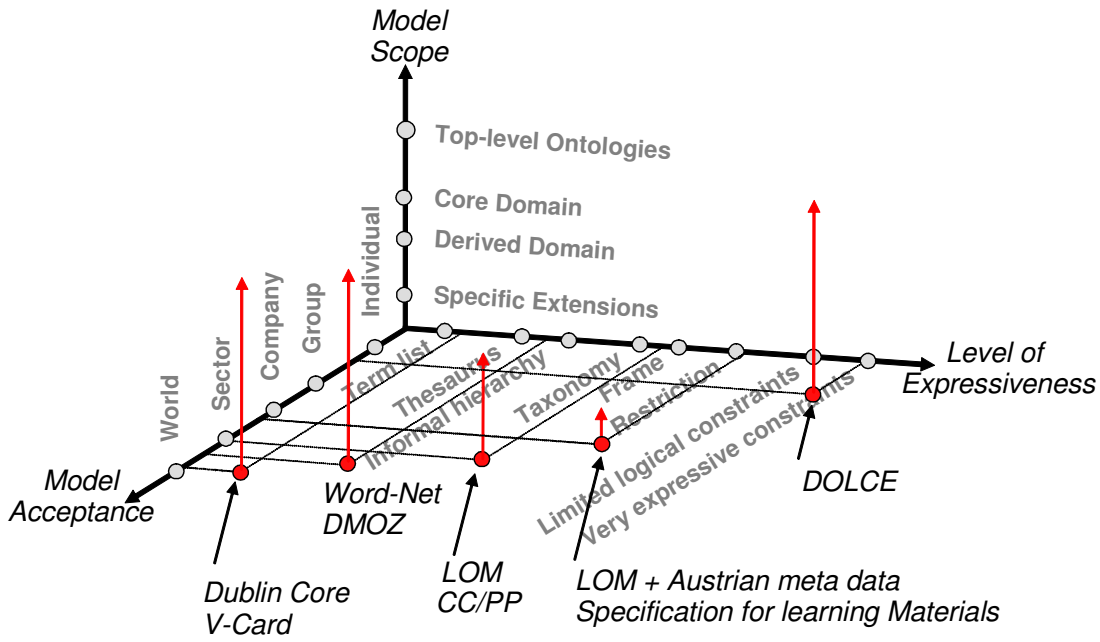
- An individual is seen as the starting point of the knowledge model creation process and as its first intended user. Knowledge model development does not always start from scratch, but first attempts to solve a modeling problem are often done by an individual.
- The group level requires collaboration between the members of a group of developers and/or domain experts; it is essential for quality assurance of any knowledge model. The deployment of the application and/or knowledge model at company level requires economic measurements of the models. New kinds of collaboration for different stakeholders – and therefore, various methods – must be integrated. Different stages of development have to be clearly defined in a life-cycle model. Sector level applications are needed, where the user scenario includes the whole sector. Consequently, standardisation is an important condition to reach this level.
- The development of knowledge models on a world level is controlled by various criteria that are out of the scope of our methodology (diffusion through market mechanisms, as well as community needs). In our perception, it seems almost impossible to describe explicit constructive processes that will be needed to reach this level.

3.4. Classification Examples

In the following, we use the 3D Matrix to classify some of the knowledge models presented in Section 2. Classification is mainly based on the expressiveness and acceptance dimensions. The scope dimension would be of importance to analyse the internal structure (the ontology stack), but this is not in the

main focus of these examples. Nonetheless, the height of the vertical lines mark the core category in the scope dimension of the associated knowledge model.

Figure 7 Classification examples



- Dublin Core and V-Card are both classified as term lists, as they define attributes without using value constraints (V-Card only uses very few). The acceptance of Dublin Core and V-Card is near to "world" because both are used in different sectors by various applications.
- Both WordNet and the DMOZ Directory define informal hierarchies. Their acceptance is not sector specific, but they are not as widely used as Dublin Core or v-card.
- IEEE LOM (Learning Object Metadata)[3] is a standard for the e-learning sector; it defines a hierarchy of elements together with some value types and value ranges. The CC/PP standard defines a frame-structure in RDFS. It is primarily used to create instances (profiles) for multimedia devices such as smart phones or set top boxes.
- DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) is a highly formalised ontology of particulars; it is intended as a reference system for top-level ontologies.

As a specific extension of the LOM (and Dublin Core) standard, the Austrian Metadata Specification [4] represents an application profile. Using the LOM standard as basis and as classification model, it adds a country specific taxonomy that acts as controlled vocabulary to fill attributes defined in LOM.

---

3 http://ieeeltsc.org/wg12LOM/

In addition, it defines cardinality constraints to identify optional and required fields. Such a knowledge model must be classified as a heavy-weight ontology. Within the acceptance dimension, it can be classified as accepted on company level – in this specific case a sector within an country. Furthermore, it is important to notice that, based on the different parts of the application profile, it is possible to create different external views (export models).

By omitting the cardinality constraints and the taxonomy, it is still possible to define LOM-compliant export models. The categorisation of such an export model would be equal to the LOM standard. That means it would be accepted by the e-learning sector and therefore applicable for interoperability with other applications in this sector.

To conclude: the plane spanned by the expressiveness and acceptance dimensions is an important measure of the complexity one can expect for implementing semantic applications. It also explains why relatively simple models such as Dublin Core or V-Card have such high levels of acceptance. It is furthermore observable that newer standards such as CC/PP or News-ML come with a higher degree of expressiveness, indicating an increasing need to formalise semantics of business sectors in order to increase effectiveness, and reduce IT costs.

4. Towards a Methodology and Adequate Tools

After the "gold rush" phase of the Semantic Web and – from the cost perspective – unsuccessful developments of intelligent applications and ontologies, we are now returning to methodological questions.

There is sort of a déjà vu [5] in ontology engineering: software engineering and its methods (architectures, requirements analysis, object-oriented analysis and design, design patterns) [6] are also useful to ontological engineering.

The previous sections showed typical properties of different knowledge models. By contrast, a methodology must first identify the necessary characteristics of a knowledge model before it can answer questions on how to specify a knowledge model.

As ontology engineering is still in its infancy, only a few methodologies for building ontologies can serve as a good basis for more integrated approaches, because some important parts of these methodologies are still missing. In particular, the "acceptance" dimension has so far received little attention, although it is one of the main distinguishing features of ontologies, as compared to data-schemas and domain models. Communities that are supposed to accept a

knowledge model must have the possibility to contribute their own knowledge during the design process.

Almost none of the well-known knowledge engineering methodologies or tools provide support for collaborative work. Only the DILIGENT methodology provides a process for distributed engineering of knowledge structure from the building, local adaption, analysis, revision, and local update of locally-defined ontologies.

Cristiani and Cuel [7] summarise that "these methodology and tools should allow both the creation of a schemata from scratch (analyzing documents, repeated occurrences within databases, etc.) and the management of sense-making processes on concepts", and noticed some emerging problems, e.g. "most of the tools give support for designing and implementing the ontologies, but they do not support all the activities of the ontological life-cycle, as defined by several software engineering methods."

More than 50% of all 96 ontology editors evaluated by Michael Denny [8] are described as having no multi-user support at all. Most of the others have only simple support for collaboration, like CVS or simple user management. Only a few of the ontology editors provide basic collaborative features beyond simple user management (e.g. communication features like notification of changes made or instant messaging, annotations, or views). But none of these tools provide advanced capabilities, that allow collaboration of subject experts and engineers based on different views that can be defined on the models.

5. Semantic Wikis

In the following, we introduce the idea of "semantic wikis" as a means to support knowledge engineers in their task of formalising knowledge. We conclude this section with a presentation of our own semantic wiki called *IkeWiki*.

5.1. The Wiki Idea

For the "traditional" Web, so-called "wiki" systems have been very successful in enabling non-technical users to create Web content. A wiki (*Hawaiian*: "quick", "fast") is essentially a collection of Web pages that allows users to add content via a browser interface. Content is usually expressed in a simplified hypertext format ("wiki syntax") that is much easier than HTML for humans to grasp. Anyone can change anything in a wiki – often, edits are completely unrestricted (but usually all edits can be undone using a rollback mechanism). Collaborative knowledge creation is thus a central aspect of a wiki system. Wiki pages are accessible and usable at any time, and the content constantly evolves.

Unlike other groupware or content/knowledge management tools, a wiki system gives users almost complete freedom over the content development process without rigid workflow, access restrictions, or predefined structures. Users need not adapt their practice to the "dictate of the system", but can allow their own practice to define the structure. This is important, because different domains often have – or even require – different kinds of workflow.

Wiki systems are currently used for a wide variety of purposes, including:

- *encyclopaedia systems:* collect knowledge in a certain area (e.g. Wikitravel[4]) or unrestricted (e.g. Wikipedia[5]) in a community effort with contributions from a wide range of users
- *software development:* collaboratively create documentation, collect ideas, track bugs; most of today's high-profile Open Source projects (e.g. Apache, Mozilla, OpenOffice) use wikis for coordination
- *project knowledge management:* brainstorming and exchange of ideas, coordination of activities, coordination and records of meetings, notepad for common information items
- *personal knowledge management:* sketchpad to collect and elaborate personal ideas, addresses, dates, tasks, bookmarks, etc.

5.2. Semantic Wikis

Arguably, wikis have changed the way content is authored on the Web. In a sense, they have helped to realise the original vision of the "traditional Web" by allowing everyone to participate and freely share information. This leads to the question of whether "semantic wikis" could help realise the "Semantic Web" in the same way. Possible advantages of semantic wikis include:

- lowering the technical barrier for non-technical users by hiding (to some extent) the complexity of Semantic Web technologies such as RDF or OWL
- supporting the evolution of knowledge along the "expressiveness axis" (cf. Section 3.2) from informal text to formal ontologies or similar representations
- allowing instant access to and usability of knowledge, even if it is not yet completely formalised
- allowing for collaborative creation of knowledge ("model acceptance axis", Section 3.3) such that domain experts and ontology experts can work together
- giving freedom over the knowledge creation process to users

---

4 http://wikitravel.org/en/Main_Page

5 http://www.wikipedia.org/

A number of systems that integrate Semantic Web technologies with traditional wiki systems are currently under development.[6]

5.3. IkeWiki

IkeWiki (*ike:* "knowledge", *wiki:* "fast") is a prototype wiki system currently being developed at Salzburg Research. IkeWiki serves several purposes: (1) it can be used to annotate existing data with semantic terms (e.g. typing relationships between pages) to improve searching and navigation; (2) it can be used to create instance data, based on an existing ontology; and (3) it can be used as a (limited) tool for creating and editing ontologies themselves. All three purposes can be followed at the same time, possibly by users with different roles and different levels of experience in knowledge engineering. Indeed, many more complex knowledge engineering tasks would probably require this kind of collaboration.

Besides this, IkeWiki has the following design goals:

- *compatibility in syntax and look and feel with existing systems* (currently Wikipedia); this allows users to take existing knowledge (e.g. from Wikipedia), import it into IkeWiki, and begin formalising the knowledge straight away
- *compatibility with existing Semantic Web technologies*; currently, IkeWiki uses RDF and OWL for storing and reasoning with formal knowledge
- *immediate exploitation of existing formal knowledge for enhanced navigation and editing*; users need to get an instant reward for the additional effort they put into formalising their knowledge
- *easy access to frequent tasks*; but still give users the full capabilities and complexity if they so desire
- *feeling of an application, not a Web site*; the user interface should support the user beyond "wiki syntax" by providing modern, graphical interaction with the system (e.g., WYSIWIG editing)

Knowledge creation in IkeWiki is supposed to be an open community process where experts from different fields can collaborate. A domain expert could begin by describing his domain knowledge in IkeWiki, up to the point where his expertise in knowledge technologies is not sufficient to do more. If necessary, a knowledge engineering expert could then join in, and help to create more formal representations.

---

[6] A constantly updated overview is given on http://wiki.ontoworld.org/index.php/Semantic_Wiki_State_Of_The_Art

5.4. Implementation

IkeWiki is implemented as a Java web application running on the Tomcat server and using the Jena RDF library for storing metadata. It makes use of up-to-date technologies like AJAX for user interaction. The system is freely available under Lesser GNU General Public License (LGPL) from http://ikewiki.salzburgresearch.at.

6. Acknowledgments

Research work for this paper was partly funded by the following projects: Dynamont, BMVIT FIT-IT Semantic Systems (2005-2007); Metokis, EU IST FP6 Semantic-based Knowledge Systems (2004-2005). The authors thank Rich Morin and the other cross-readers for suggesting important improvements to the article.

7. References

[1] Guarino, N., Formal Ontology and Information Systems, pp. 3-15, 1998, http://www.loa-cnr.it/Papers/FOIS98.pdf

[2] O. Corcho, M. Fernández-López and A. Gómez-Pérez: Methodologies, tools and languages for building ontologies. Where is their meeting point? Data & Knowledge Engineering 46(1), pp. 41–64, 2003.

[3] John Harris: Judging the likely Success of an Ontology, http://www.virtualtravelog.net/entries/2004/01/judging_the_likely_success_of_an_ontology.html

[4] Österreichische Metadatenspezifikation für elektronische Lernressourcen. Version 1.32, Stand: 2004-01-12. Im Auftrag des Bundesministeriums für Bildung, Wissenschaft und Kultur, MR Dr. Robert Kristöfl, http://www.bildung.at.

[5] Vladan Devedzic (2002): Understanding ontological development. *Communications of the ACM*. April 2002/Vol. 45, pp. 136-144.

[6] Aldo Gangemi (2005): Ontology Design Patterns for Semantic Web Content. *Proceedings ISWC 2005,* LNCS 3729, 2005, pp. 262-276.

[7] Mateo Cristani und Roberta Cuel (2005): A survey on ontology creation methodologies. *International Journal on Semantic Web & Information Systems.* Idea Group Publishing, 2005.

[8] Michael Denny (2004) Ontology Tools Survey, Revisited. http://www.xml.com/lpt/a/2004/07/14/onto.html, [Last visited: 09.10.2005]

[9] http://www.loa-cnr.it/DOLCE.html

[10] http://suo.ieee.org/SUO/Evaluations/

[11] http://www.cyc.com/