Towards Scientific Collaboration in a Semantic Wiki

Christoph Lange

Computer Science, Jacobs University Bremen*, ch.lange@jacobs-university.de

Abstract. On the Web 2.0, there are numerous projects for collaboratively creating and using scientific knowledge in a wiki—think of the scientific sections of *Wikipedia* or domain-specific platforms like *PlanetMath*. They do, however, not yet offer semantic services that could promote collaboration both of scientific knowledge engineers and of scholars or that take semantics emerged from such communities or acquired from page contents into account.
On the other hand, there are several semantic wikis—wikis enhanced with Semantic Web technologies. Current semantic wikis, however, only offer rather generic semantic services, such as semantic navigation, semantic-based editing assistance, and semantic search. Semantic services tailored to scientific knowledge and its specific structures (e. g. theories depending upon each other) are not yet provided.
Based on the argument that current semantic wikis lack scientific services because domain-specific ontologies are not properly integrated, this article proposes the basic architecture of a semantic wiki centered around an ontology of scientific markup languages. Two services to be designed on top of this ontology abstraction layer are outlined, and suggestions on how to improve them by making them community-aware are discussed.

1 State of the Art and Problem Statement

Current wiki projects for scientific knowledge range from comprehensive encyclopediae like *Wikipedia*, which covers all domains, even non-scientific ones, to projects specialised to a particular domain, such as *PlanetMath*, a wiki for mathematics[1]. As new content can quickly and easily be created and linked, wikis are also suitable for corporate knowledge management [9]—and for teams of scientists in a similar way. Neither *Wikipedia* nor *PlanetMath* offer certain services desirable for scientific communities, though.

A non-semantic wiki lacks a deeper understanding of the network of links between its pages. Semantic wikis [20] address this problem by typing pages and links with terms from ontologies [14]; usually, one page describes one real-world concept (e. g. a scientific theory). Page and link types can serve as the foundation for semantic services. Two services that are desirable in a scientific

* formerly International University Bremen

[1] See `http://www.wikipedia.org` and `http://www.planetmath.org`, respectively.

community will be discussed in section 4: one that suggests topics to learners, and another one that manages dependencies among concepts, which is useful for theories in development. Solutions for both of these problems have been available on the Semantic Web for years (see [2] or [10], resp.), but not yet integrated with (semantic) wikis on a large scale. A wiki extension for learning has been proposed with the *WikiTrails* system [16], which augments wiki content with navigation trails. These trails are either generated automatically by tracking user interaction, or they can be created manually (e. g. by a teacher), but the *knowledge contained in the wiki pages* is used in neither case. Integrating services that exploit this knowledge has been hampered by the fact that domain-specific ontologies are considered optional at best in most semantic wikis: They usually allow for ad-hoc modeling new ontologies or importing available ones [21], but as there is no uniform ontology layer at the *core* of these wikis, they cannot exploit characteristic traits of domain-specific knowledge.

Structural semantic markup is a common way to represent scientific knowledge. Available markup languages include OMDoc, a mathematical markup language that comprises and extends Content MathML and OpenMath [5], which only allow for representing formulæ, PhysML [1], an OMDoc variant adapted to physics, and the Chemical Markup Language CML[2] for chemical concepts like molecules and reactions. Semantic services for mathematical knowledge are the most advanced ones so far; for OMDoc, for example, services for learning assistance, semantic search, publishing (including community-specific notations of mathematical symbols), theory management, as well as proof verification have been developed [5, chapter 26]. Our work group, in collaboration with experts from scientific domains other than mathematics, is currently concerned with designing a unified "scientific markup language" and transferring these techologies to other domains, including physics, chemistry, geosciences, environmental sciences, and software engineering.

2 SWiM, a Semantic Wiki Prototype for Mathematics

Semantic wikis are appropriate for building "community-authored knowledge models" where informal natural language descriptions created by domain experts are formalised in collaboration with knowledge engineers [17]; the stepwise refining process of formalising human-readable texts they support is a common task for scientists [5]. Before scientific services can be implemented in a wiki, a base system supporting scientific documents must exist. So far, I have developed the SWiM prototype, a semantic wiki for mathematics [7], which is a modified *IkeWiki* [17] with OMDoc as its page format. It is capable of rendering OMDoc in a human-readable way using XSL transformations and extracting RDF triples used as typed navigation links from the markup. Other markup languages are not supported, and further semantic services are not yet offered. SWiM will serve as the base for implementing a wiki with services for scientific communities, tentatively named SWiM$^+$.

[2] `http://cml.sourceforge.net/`

As of March 2007, there is only one more semantic wiki dedicated to mathematics: se(ma)^{2}wi [23] is an experiment with a *Semantic MediaWiki* [19] fed with OMDoc-formatted mathematical knowledge from the *ActiveMath* learning environment [11]. Most of the structural semantics explicitly given in OMDoc is, however, lost during this import: The formulæ are converted to presentational-only LaTeX, and the links between wiki pages that represent mathematical statements, for example a link from a theorem to its proof, are not typed.

3 Representing Scientific Knowledge

To represent scientific knowledge, I follow the three-layered structure model of mathematics and science that M. Kohlhase has successfully applied to mathematics with OMDoc (see section 1): *Objects* (symbols, numbers, equations, molecules, etc.), *statements* (axioms, hypotheses, measurement results, examples, with relations like "proves", "defines", or "explains") and *theories*—collections of interrelated statements, which set symbols into their context[3]. This model has already been extended towards physics (PhysML) with just a few additions [1], and the PhysML creators anticipate that it also holds for other sciences.

For use in Semantic Web software, this model of scientific knowledge needs to be formally, explicitly specified in an *ontology*. The ontology behind the OMDoc markup format defines which knowledge can be represented in OMDoc and thereby approximates the general way of knowledge representation in mathematics. For the SWiM prototype, a subset of that ontology, which is given in a merely human-readable way in [5], has been explicitly modeled in OWL-DL: most statement types and their interrelations, theories and their "import" relation, with the addition of a generic transitive dependency relation. To make SWiM$^+$ support multiple scientific domains, ontologies of multiple markup languages will have to be formalised. Building on the work of the researchers working on a unification of scientific markup languages (cf. section 1), who will identify common traits of knowledge in all sciences covered—most likely including the three-layer stack of objects/statements/theories as well as generic containment and dependency relations—, one generic ontology, to be called "*upper* document ontology"[4] here, will be formalised in an appropriate language, such as OWL-DL[5].

One SWiM$^+$ page will most likely hold one statement or one small theory, which leads to small pages that are suitable for reuse by linking. As Semantic Web tools are not ready to use knowledge represented as markup in documents, relevant parts of it must first be *extracted* to a more formal representation like

[3] e. g. the glyph h as the height of a triangle in a theory of elementary geometry or Planck's quantum of action in a theory from quantum mechanics.

[4] A variation on the term "upper ontology", which the IEEE Standard Upper Ontology Working Group defines as an ontology "limited to concepts that are meta, generic, abstract and philosophical, and therefore are general enough to address (at a high level) a broad range of domain areas"; see `http://suo.ieee.org/`.

[5] A more formal definition of generic document ontologies is currently being developed by N. Müller and A. Mahnke, members of our group.

RDF (cf. [13]), using terms from the ontology behind the respective markup language. For example, a mathematical proof, marked up in OMDoc as `<proof xml:id="py-proof" for="pythagoras">`, would be represented by the two RDF triples `<py-proof, rdf:Type, om:Proof>` and `<py-proof, om:proves, pythagoras>`, terms from OMDoc's ontology being prefixed with `om:`. To make this extraction scale to multiple markup languages, solutions that use mappings from XML schemata to ontologies and back, such as *WEESA* [15], will be evaluated.

4 Semantic Services for Collaboration

Two key services envisaged for implementation within SWiM$^+$ are an interactive learning assistant for scholars and a dependency management assistant for scientists. In a work environment where scientists collaboratively formalise their ideas into theories, dependency management is an important feature: For example, if scientist E. In. Stein decides to base his newest ideas about "relativity" on assumptions about "gravitation" currently being developed by his colleague N. Ew. Ton and then Ton *changes* one of them, Stein's considerations might become invalid.

In an educational semantic wiki, course modules would be connected by links typed as "prerequisite". In terms of scientific markup, course modules can be realised as theories whose transitive "import" relation is interpreted as "prerequisite". If, for example, one member of a community interested in mathematics and its applications knows that, to fully understand MP3 encoding (let this be covered on a wiki page named *MP3Encoding*), one must know what a discrete cosine transform (DCT) is, he may connect these two pages with a "prerequisite" link. Imagine a second user who knows that orthogonal matrices are one basis for discrete cosine transforms and connect the pages DCT and *OrthogonalMatrix* accordingly. A Semantic Web reasoner can densify the network of knowledge by inferring additional knowledge not explicitly contained in the wiki pages, namely that reading (and understanding!) the wiki page *OrthogonalMatrix* is a prerequisite for fully understanding *MP3Encoding*. On the user interface, the direct and indirect prerequisites could then be recommended for reading.

The two services introduced here only rely on generic relations like dependency; therefore, they can be implemented on top of the above-mentioned "upper document ontology" and thus work across scientific domains. The same holds for two projected services that facilitate editing—ontology-based auto-completion of link targets and section-wise editing [8], but not for all envisioned services: Integrating the OMDoc-based formula search engine *MathWebSearch* [6], for example, is specific to the the domain of mathematics and requires access to full structural markup of formulæ instead of just extracted RDF triples.

5 Added Value from and for the Community

To design and improve services for SWiM$^+$ in a user-centered way, the method of "added-value analysis" [3] will be employed: First, specify a core problem, propose a solution for it, and establish the benefits and sacrifices of the solution, as perceived from a user's "micro-perspective". Benefits and sacrifices given, evaluate the core solution with regard to the core problem. Both benefits and sacrifices may spawn new core problems and thus ideas for further services—services that provide *added value* to the user [3].

5.1 Learning from the Community

Applied to the problem of (1) helping a user, named S. Ch. Olar here, to understand topics, an added-value analysis could lead to the following results: First we might propose the display of direct links from the current page on a navigation bar, grouped by types like "prerequisite" or "example",—a service offered by most semantic wikis, including *IkeWiki*, for free. The benefit of that solution is a concise overview of direct prerequisites and examples, at the expense of other direct links (e. g. from a topic to its type) also being shown and indirect prerequisites not being accessible. Taking up the latter sacrifice, we arrive at the new problem of (2) exploring direct *and* indirect prerequisites, which can be solved by computing all prerequisites beforehand and displaying links to all of them. The benefit is that all of them are now accessible within one click, at the expense that the list may contain too many links irrelevant to our user Olar.

Now, Olar needs to (3) distinguish relevant from irrelevant prerequisites—a ranking or pre-selection would be helpful. The social way, one could record how many of the prerequisites offered *other* readers of the same page actually clicked and rank or restrict the generated list based on that information. The benefit is that prerequisites most users considered irrelevant will not be included in the pre-selection computed for Olar and hence not distract him. A severe sacrifice is that this solution does not satisfy Olar's needs if they greatly differ from the needs of those who visited the respective page before. The new problem is to (4) give a better estimate of which preferences the user really has. In a single-user context, a *user model* containing the user's previous knowledge (e. g. obtained from his history of interaction with the system) and a user profile containing his learning goal and other preferences, such as notational ones, as in *ActiveMath* [11] would solve this problem.

A community-powered environment allows for giving improved estimates, though: If we assume that many users, divided into different sub-communities, are collaborating on one SWiM$^+$ site, problem (4) can be solved by finding out to which sub-community the user belongs. The lectora project [12] for enhancing the community-awareness of collaborative mathematical software, ran in our group, aims at improving search rankings and offering other services based on community models. While lectora is conceptually an independent system, the communication interface to SWiM$^+$ is currently being designed

in close cooperation[6]. LECTORA, connected to SWiM$^+$, would steadily be fed with information about all users' actions (reading, writing, annotating, setting preferences, ...). If LECTORA then finds out that OLAR belongs to the same subcommunity as another user, named L. E. ARNER, she henceforth recommends documents ARNER has rated as relevant or useful to OLAR—or, getting back to our added-value analysis, use this rating to rank a list of prerequisites computed by SWiM$^+$ for OLAR.

5.2 Better Collaboration for the Community

An added-value analysis starting from the problem of (1) managing dependencies between theories or course modules across changes could first lead to the following simple solution—applied to the two physicists from section 4: If R ("relativity") depends on G ("gravitation") and the user TON has changed G, thereby maybe breaking the current version of R, the next user to edit R—here: STEIN—could be notified that a page depended upon has been changed. This situation-dependent notification is a benefit over the usual list of recent changes in a wiki, which does not consider dependencies at all and which STEIN would have to visit on his own. Still, he has to make sacrifices: He needs to figure out whether TON has changed the semantics of G—instead of just fixing a typo, for example, —, and if so, whether that affects the consistency of R. If R is affected, STEIN must first figure out and then apply the appropriate change he has to make to R on his own. The problem that STEIN (2) does not know whether TON has changed the semantics of G could be solved the wiki way: Many wikis would allow TON to mark his change as "major" or "minor" [22]. This distinction is entirely subjective and thus not helpful, but we could offer the alternatives "semantics changed/not changed" instead. STEIN would benefit from that additional knowledge, but TON would have to make the sacrifice to correctly classify his change to G.

locutor [13], an ontology-driven management of change system developed in our group, will be a possible solution to the problem (3) of finding out whether one change actually affects other documents by computing "long-range effects" of changes. With its more detailed "taxonomy of change relations" it will provide a better solution to problem (2), too. If R is affected by the change of G, *locutor* will either be able to automatically make the required adaptations to R to restore consistency, or it will at least be able to pinpoint all effects of changes, to that STEIN would exactly know what to fix manually[7]. *locutor* is being implemented as an extension of the version management system *Subversion*[8]; thus, it can be integrated into SWiM$^+$ as a backend, replacing part of the SQL database used by the SWiM prototype.

[6] personal communication with CH. MÜLLER
[7] personal communication with N. MÜLLER
[8] See `http://kwarc.info/projects/locutor/` for a prototype.

6 Conclusion and Outlook

Mission ... Among others, the two services introduced in this article will be implemented in SWiM$^+$ and evaluated in long-term case studies in 2008, from which I expect feedback leading to further improvements. A scientific case study, focusing on dependency management and other services that support scientists in developing new knowledge, will be conducted in the cross-domain setting of the unified scientific markup language developed in collaboration with our group. An educational case study, focusing on the prerequisite learning assistant and on search facilities not presented here, will be conducted in an introductory course to computer science, the lecture notes of which are available in sTEX [4], an OMDoc-related format.

... and Vision: SWiM$^+$ will demonstrate how services on top of a semantic social software can make users benefit both from semantics extracted from formal documents and from semantics that emerged from communities of users. Following an idea from S. Schaffert [17], the achievements made in the "testbed" of a SWiM$^+$-driven site may then, thanks to the ontology abstraction layer, be transferred to the "large" web.

Acknowledgments The author would like to thank his colleagues Christine Müller and Normen Müller, as well as his advisor Michael Kohlhase, for productive discussions and for supporting his work, and the anonymous reviewers for broadening his horizon with their valuable suggestions.

References

1. E. Hilf, M. Kohlhase, and H. Stamerjohanns. Capturing the content of physics: Systems, observables, and experiments. In J. Borwein and W. M. Farmer, editors, *Mathematical Knowledge Management 2006*, number 4108 in Lecture Notes in Artificial Intelligence. Springer, 2006.
2. M. C. A. Klein, D. Fensel, A. Kiryakov, and D. Ognyanov. Ontology versioning and change detection on the web. In A. Gómez-Pérez and V. R. Benjamins, editors, *EKAW*, number 2473 in Lecture Notes in Computer Science, pages 197–212. Springer, 2002.
3. A. Kohlhase and N. Müller. Added-Value: Getting People into Semantic Work Environments. In *Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications*. Idea Group, 2007. To appear; chapters under review.
4. M. Kohlhase. sTEX: A LaTeX-based workflow for OMDoc. In OMDoc [5], chapter 26.15.
5. M. Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in Lecture Notes in Artificial Intelligence. Springer Verlag, 2006.
6. M. Kohlhase and I. Şucan. A search engine for mathematical formulae. In T. Ida, J. Calmet, and D. Wang, editors, *Proceedings of Artificial Intelligence and Symbolic Computation (AISC)*, number 4120 in LNAI, pages 241–253. Springer Verlag, 2006.

7. C. Lange. SWiM – a semantic wiki for mathematical knowledge management. Technical Report 5, Jacobs University Bremen, 2007. http://kwarc.info/projects/swim/pubs/tr-swim.pdf.
8. C. Lange. Towards a Semantic Wiki for Science. http://kwarc.info/projects/swim/pubs/swimplus-resprop.pdf, 2007. Research proposal for a Ph. D. thesis.
9. B. Leuf and W. Cunningham. *The Wiki Way: Collaboration and Sharing on the Internet*. Addison-Wesley Professional, 2001.
10. M. d. O. Lidia Silva Muñoz, José Palazzo. Applying semantic web technologies to improve personalization and achieve interoperability between educational adaptive hypermedia systems. In *Proceedings of the International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL)*, 2004.
11. E. Melis, G. Goguadze, A. G. Palomo, A. Frischauf, M. Homik, P. Libbrecht, and C. Ullrich. OMDoc in ActiveMath. In OMDoc [5], chapter 26.8.
12. C. Müller. Lectora – towards an interactive, collaborative reader for mathematical documents. http://kwarc.info/cmueller/papers/Mueller_ResearchProposal_2007-03-14.pdf, 2007. Research proposal for a Ph. D. thesis.
13. N. Müller. An Ontology-Driven Management of Change. In *LWA (Lernen, Wissensentdeckung und Adaptivität) conference proceedings*, 2006.
14. E. Oren, R. Delbru, K. Möller, M. Völkel, and S. Handschuh. Annotation and navigation in semantic wikis. In Völkel et al. [20].
15. G. Reif, H. Gall, and M. Jazayeri. WEESA: Web engineering for semantic web applications. In A. Ellis and T. Hagino, editors, *Proc. of the 14th WWW conference*, pages 722–729. ACM, 2005.
16. S. Reinhold. WikiTrails: Augmenting wiki structure for collaborative, interdisciplinary learning. In D. Riehle and J. Noble, editors, *Proceedings of the 2006 International Symposium on Wikis*, ACM Press, Aug. 2006.
17. S. Schaffert. Semantic social software – semantically enabled social software or socially enabled semantic web? In Sure and Schaffert [18].
18. Y. Sure and S. Schaffert, editors. *Semantics 2006: From Visions to Applications*, 2006.
19. M. Völkel, M. Krötzsch, D. Vrandečić, H. Haller, and R. Studer. Semantic Wikipedia. In *Proc. of the 15th WWW conference*, 2006.
20. M. Völkel, S. Schaffert, and S. Decker, editors. *1st Workshop on Semantic Wikis*, volume 206 of *CEUR Workshop Proceedings*, Budva, Montenegro, 2006.
21. D. Vrandečić and M. Krötzsch. Reusing ontological background knowledge in semantic wikis. In Völkel et al. [20].
22. Minor edit (from Wikimedia meta-wiki). http://meta.wikimedia.org/w/index.php?title=Help:Minor_edit&oldid=398318, 2006.
23. C. Zinn. Bootstrapping a semantic wiki application for learning mathematics. In Sure and Schaffert [18].