SWiM – A Semantic Wiki for Mathematical Knowledge Management

Christoph Lange <ch.lange@jacobs-university.de>
Computer Science, Jacobs University Bremen

June 21, 2007

SWiM is a semantic wiki for collaboratively building, editing and browsing mathematical knowledge represented in the structural markup language OMDoc. It has been designed to enable groups of scientists to develop new mathematical theories in OMDoc and to enable scholars to browse such a corpus. After a short introduction to semantic wikis and their usefulness for mathematical knowledge, this article presents the architecture and the user interface of the current SWiM prototype and outlines the plans for developing its successor, an ontology-based platform for semantic scientific services that exploit the knowledge and make it accessible to the user.

1 Motivation

Current collaborative projects for mathematical knowledge range from comprehensive encyclopediæ like the mathematical sections of the free wiki-based encyclopedia *Wikipedia* [32] or the courseware repository and content management system *Connexions* [3] to more special projects like *PlanetMath* [13]. As new content can quickly and easily be created and linked, wikis are also suitable for corporate knowledge management [17]—and for teams of scientists in a similar way. In neither of the systems mentioned initially, the mathematical knowledge is represented in a way that is suitable for re-use by services, though. Their pages are categorised and searchable in full-text, and, in the case of *Wikipedia* or *PlanetMath*, formulæ are given in presentation-oriented LaTeX, which is insufficient for mathematical knowledge management (MKM): Imagine a wiki page about the Pythagorean Theorem, given as $a^2 + b^2 = c^2$. A search for the equivalent formula $x^2 + y^2 = z^2$ (or even $c = \sqrt{a^2 + b^2}$!) would not yield the theorem, neither could "all theorems about triangles for which a proof exists" be searched for. *Connexions*, on the other hand, could in principle solve the problem of formula search, as its formulæ are given in the structural markup language Content MathML [2], but it is not suitable for developing *new* mathematical ideas: Its markup language CNXML [5] neither allows for defining additional symbols for Content MathML nor for grouping related symbols and axioms into formally structured theories instead of just informal course modules.

2 State of the Art

The problem of retrieving mathematical statements by their type and their relations to other statements is solved by *semantic wikis*—wikis that use semantic web technologies for knowledge representation [29]: Pages and links are usually typed with terms from ontologies [23]. A proof of the Pythagorean Theorem could be put on a page typed as "Theorem", for example, and its link to the theorem itself could be typed as "proves". The problem of searching formulæ can be solved by *structural semantic markup* of scientific knowledge. OMDOC, a structural semantic markup language for mathematics that extends Content MATHML and the similar, but more extensible OPENMATH by a formalism for representing mathematical statements and theories [11], has many applications in publishing, education, research, and data exchange [11, chap. 26]. The e-learning environment *ActiveMath* [20], for example, can adapt the presentation of OMDOC-encoded learning objects according to the user's preferences. The semantic search engine MathWebSearch [12] harvests the web for Content MATHML and OPENMATH formulæ and allows for searching them by their meaning, regardless of their presentation.

3 OMDoc in a Semantic Wiki—Meeting Users' Needs

Semantic wikis as "community-authored knowledge models" [25] particularly support a collaborative workflow of stepwise formalisation of knowledge, which KOHLHASE identified as essential for MKM [11, chap. 4]—from human-readable text to a representation suitable for semantic web services to a full formalisation that can be verified by a theorem prover. Having projected SWIM as a **S**emantic **Wi**ki for **MKM** based on the OMDOC format, the question arose: Who is willing to participate in creating a huge collection of OMDOC-formatted knowledge?

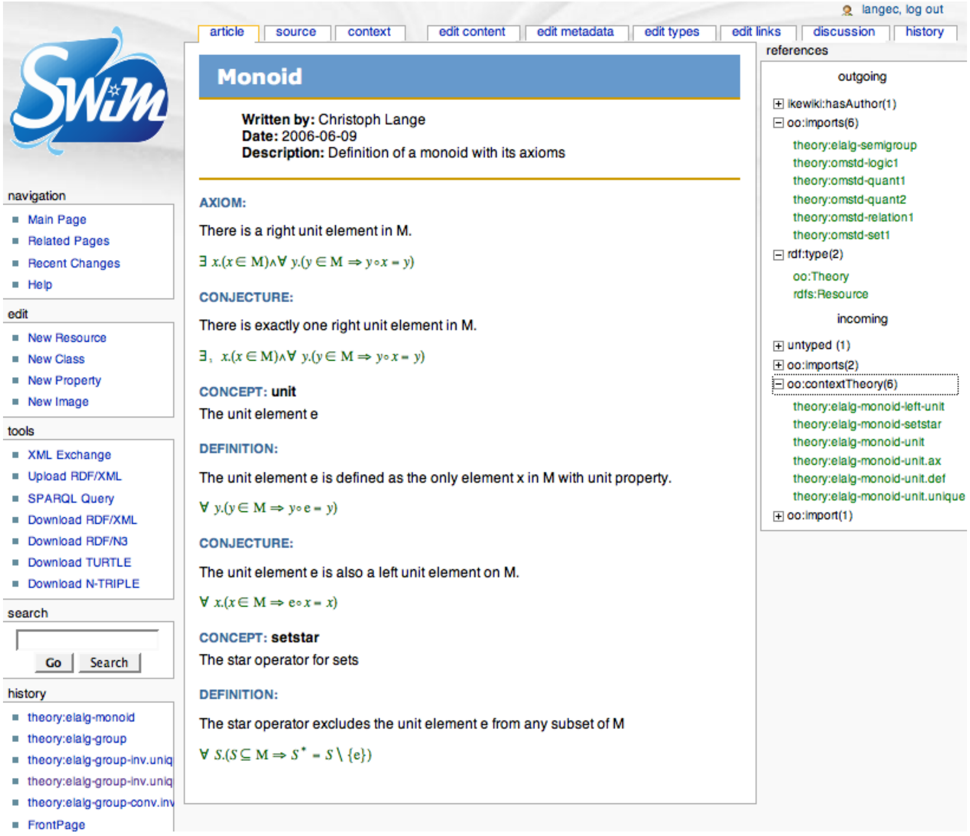
In an open, collaborative environment, the workload of creating knowledge can be distributed among many authors, but unlike the text formats used by common (semantic) wikis, OMDOC makes the fine-grained semantic structure that is implicit in the text explicit in the markup, making it tedious to author by hand. Moreover, only after a substantial initial investment (writing, annotating and linking) on the author's part, the community can benefit from added-value services like the above-mentioned OMDOC applications. If author and beneficiary of such services were different persons, though, only few persons would be willing to contribute to a knowledge base, as every rational user would wait for the others to take action. This "author's dilemma" can be overcome when the authors themselves are motivated into action by "elaborate [...] services for the concrete situation" they are in [9, 10]. The related approach of instantly gratifying users for their contributions inspired *WikSAR*, one of the first semantic wikis, which provides gratification by instantly improving navigation: Not only are incoming and outgoing links displayed for each page, grouped by their types, but also links to other semantically related pages are inferred [1].

4 User Interface and Interaction Model

After a survey of several semantic and non-semantic wikis with regard to their support of XML and semantic web technologies and their extensibility, *IkeWiki* [25], a semantic wiki implemented using Java Server Pages, was chosen as the base for SWIM [14]. The main classes for parsing a wiki page from its XML representation, for representing a page in memory, for extracting semantic relations from it into an RDF representation (see sect. 5), and for presenting a page to the human viewer have been forked each into a generic base class with two subclasses: one for the traditional wiki page format, which is still useful for creating text-only pages or link lists, and one new subclass supporting OMDOC.

Editing OMDoc. So far, the XML source code of an OMDOC document can be edited in SWIM; minimum feedback about its validity is given upon saving a page by displaying the exact error message from the XML parser. Planned improvements to the editor are discussed in section 6. The OMDOC syntax has been slightly adapted to the requirements of a wiki, which include easy linking and small pages. Small pages improve the effectivity of wiki usage, as they facilitate editing and re-use by linking and allow for a better overview through automatically generated index pages or the list of recently changed pages. Link targets need not be full URI references, but they can be abbreviated: A statement on a theory page can be referenced as `theory#statement`. While OMDoc only allows statements that are not constitutive for a theory[1] to live in their own documents, the SWiM-extended OMDoc document model allows any kind of statement to be rolled out to its own page.

Presenting OMDoc in SWiM. For presentation (see fig. 1), SWiM uses a slight adaptation of the multi-level XSL Transformation workflow to XHTML+MathML that has earlier been developed for OMDoc [11, chap. 25.1]. The links from symbols in formulæ to their definitions, which are generated by this transformation, improve the navigability of the wiki. Additionally, a hyperlinked source view is available. It is particularly useful for browsing complex OMDoc documents, as, so far, not all kinds of links between statements or theories have been mapped to RDF triples and thus semantic links navigable in the wiki (see sect. 5).



Monoid

Written by: Christoph Lange
Date: 2006-06-09
Description: Definition of a monoid with its axioms

AXIOM:

There is a right unit element in M.

$\exists\, x.(x \in \mathrm{M}) \wedge \forall\, y.(y \in \mathrm{M} \Rightarrow y \circ x = y)$

CONJECTURE:

There is exactly one right unit element in M.

$\exists_{1}\; x.(x \in \mathrm{M}) \wedge \forall\, y.(y \in \mathrm{M} \Rightarrow y \circ x = y)$

CONCEPT: unit
The unit element e

DEFINITION:

The unit element e is defined as the only element x in M with unit property.

$\forall\, y.(y \in \mathrm{M} \Rightarrow y \circ \mathrm{e} = y)$

CONJECTURE:

The unit element e is also a left unit element on M.

$\forall\, x.(x \in \mathrm{M} \Rightarrow \mathrm{e} \circ x = x)$

CONCEPT: setstar
The star operator for sets

DEFINITION:

The star operator excludes the unit element e from any subset of M

$\forall\, S.(S \subseteq \mathrm{M} \Rightarrow S^{*} = S \setminus \{\mathrm{e}\})$

Figure 1: A rendered page in the SWiM prototype

Exploiting Knowledge from OMDoc. *IkeWiki*, as most semantic wikis, considers each wiki page to represent one real-world concept. As for OMDoc, I considered small theories[2] and state-

---

[1] Symbols, definitions, and axioms are indispensable for the meaning of a theory ("constitutive"), but assertions, their proofs, alternative definitions of concepts already defined, and examples are not.

[2] According to M. Kohlhase, OMDoc advises to follow a "little theories approach", where theories introduce as few new concepts as possible. A theory may introduce more than one concept, if they are interdependent, e.g. to

ments appropriate page-level concepts, but not sub-statement structures like proof steps, as the latter would make it difficult to overlook complex statements[3].

Relations between theories and statements can be expressed in OMDOC either through containment of child elements within parent elements—a theory can have statements as its children, for example,—and via URI references—for instance, from an *example* element to the assertion or definition it explains. To enable a semantic web application to reason on these relations and to exploit them for navigation and interactive queries, the information about the concept instances and their interrelations must be made accessible as RDF [16] subject–predicate–object triples (see sect. 5). If such triples with the current page as subject (i. e. outgoing links) or object (i. e. incoming links) are available, they are displayed in a navigation box, grouped by type. This box is regenerated on reloading a page, e. g. after editing, and thus offers instant gratification (see sect. 3). The most recent version *IkeWiki* 1.99, which SWIM will soon be based on, can, moreover, render the neighbourhood of the current page in the RDF graph.

5 Knowledge Representation

To obtain a vocabulary of page and link types that can be used to express RDF triples, OMDOC's three-layered model of representing knowledge (objects, statements, and theories) had to be formally, explicitly specified in an *ontology*[4]. The ontology behind the OMDOC markup format, specified in natural language in [11], defines which knowledge can be represented in OMDOC and thereby approximates the general way of knowledge representation in mathematics. A subset of this ontology has been explicitly modeled in OWL-DL [18] (see fig. 2): theories, a hierarchy of several statement classes and generic transitive dependency and containment relations. The former subsumes the transitive import relation between theories and several relations between statements, where one statement further specifies another one—"symbol–has–definition", "proof–proves–assertion", and "example–exemplifies–statement(s)"—; the latter subsumes, among others, the containment relation between a statement and the home theory that fixes its context.

Note that this ontology does not directly represent *mathematical* concepts. Relations between the latter, such as "all differentiable functions are continuous", cannot be expressed directly in OMDOC; as OMDOC captures how scientists communicate *about* mathematics, they must be wrapped into mathematical statements, but could nevertheless be extracted if a DL representation is required[5].

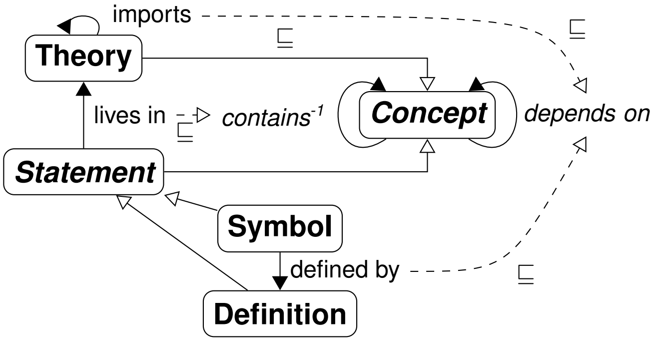


Figure 2: Subset of the document ontology

Given that, it remained to *extract* those parts of the knowledge that could be represented in terms of that ontology from OMDOC to a more explicit RDF representation; after all, the relevant knowledge is not available as sepa-

---

introduce the natural numbers via the Peano Axioms, we need to introduce the set of natural numbers, the number zero and the successor function at the same time.

[3] Note that figure 1 shows a theory page containing several statements. This is, however, accomplished using OMDOC's include mechanism. Actually, both the theory and the statements are stored in individual pages.

[4] Ontologies are formal, explicit specifications of a conceptualisation; they describe a specific domain and emerge as a result of a shared understanding among experts in that domain. On the semantic web, the most common formalism for ontologies is description logic (DL).

[5] The *proof* of this theorem about functions cannot be expressed in DL, though, as it requires higher-order logic. The latter is, however, disliked on the semantic web, as it is not decidable.

rate, handy annotation in OMDOC, but rather buried in the markup. 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:`. For the SWIM prototype, a simple RDF extraction procedure based on XPath expressions with a hard-coded mapping from XML elements to concepts of the ontology has been implemented; it extracts information about the types of theories and statements as well as the links between them. Currently, RDF is only extracted on page level, i. e., links whose either source or target is not the top-level concept of a page—e. g. if the authors have not manually broken a theory down into its statements—are not yet exploited for services like semantic navigation (see sect. 4).

6 Outlook: Integrating Services for Science

Over the next two years, several case studies about offering extended services for scientific collaboration on top of the SWIM platform will be conducted. That includes adding support for scientific markup languages other than OMDOC, as well as introducing an abstraction layer that serves as an "operating system" for implementing ontology-based services inside SWIM and integrating external ones [15].

Extension Towards Sciences. Our work group is concerned with a technology transfer of the applications that exist for MKM (cf. sect. 2) to general scientific knowledge management. OMDOC has already been successfully extended towards physics with only a few additions [6], and a collaborative effort of merging markup languages for different sciences using the three-layered knowledge representation of OMDOC is starting right now. Building on the work of these researchers, who will identify common traits of knowledge across sciences—most likely including the three-layer stack of objects/statements/theories as well as generic containment and dependency relations—, one generic ontology, tentatively named "*upper* document ontology"[6] here, will be formalised in an appropriate language, like OWL-DL[7].

Enticing researchers of multiple domains into SWIM requires making their preferred tools interoperable with SWIM. The most obvious step towards this is an external editor interface for wiki pages, as known from *MediaWiki* [31]. Concerning OMDOC, this enable SWIM users to benefit from the *Emacs* mode for OMDOC [7], the visual editor *Sentido* [24][8], or the MATHEMATICA to OMDOC converter [27], if the conversion back to MATHEMATICA is also implemented.

Service Interface and Planned Services. Current semantic wikis are not committed to certain domain-specific ontologies. They usually allow for ad-hoc modeling new ontologies or importing available ones [30], but there is no uniform ontology layer at their *core*. Therefore, they mostly offer generic services based on the knowledge contained in their pages, such as semantic navigation or search. Thanks to the document ontology and the knowledge extraction introduced in sect. 5, the services planned for SWIM will be enabled to exploit, for example, the dependencies among theories and statements.

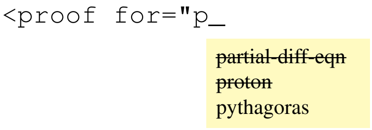
---

[6] 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/`.

[7] A more formal definition of generic document ontologies is currently being developed by N. MÜLLER and A. MAHNKE, members of our group.

[8] *Sentido* is implemented as an extension for the *Mozilla* browser, but as it is able to load or save local files, it can be considered a stand-alone editor independent of SWIM.

A use case for scientists is dependency maintenance: If a theory t builds on knowledge from another, imported theory u that is still in development and a basic assumption in u is changed, the author of t needs to be warned during editing. In the same setting, students can be supported: If a student is currently reading the page that contains t, a learning assistant could recommend him to study u first. Lightweight, straightforward solutions to these problems could be directly implemented in SWiM, but integrating—via a web service interface—external components specialised on certain tasks to the storage backend of SWiM and then providing an adequate user interface for them in the GUI of SWiM is an option for more powerful solutions. Candidates are the change management service `locutor` [22] for dependency maintenance, or an adaptation of the course generator service of the e-learning environment ActiveMath [19], where learning prerequisites could partly be inferred via the dependency relations in the ontology. A generic service for editing assistance that can be improved by ontological reasoning is auto-completion of link targets, as shown on the right. Instead of naïvely suggesting all names of pages starting with the letters typed so far, leading to overly long lists and semantically invalid links, the semantic relation expressed by the link currently edited should be looked up via the XML-to-ontology mapping, and the document ontology should be queried for the range of that relation (here: "*Proof–proves–?*"), leading to the answer *Assertion*, and then a list of all known instances (i. e. all pages) of that type starting with the respective letters should be displayed[9]. Thus, the user interface offers less choices to the user, but only relevant ones, and thus helps preventing mistakes *during* editing. *After* editing, better feedback about syntactical validitity of the OMDoc code is necessary, and a validation of the semantic structure against the ontology will also be investigated.



```
<proof for="p_
```

~~partial-diff-eqn~~
~~proton~~
pythagoras

Note that there are also certain relevant services that are domain-specific or that do not rely on the document ontology but on the page content: The formula search engine *MathWebSearch* [12] will be instructed to index SWiM's OpenMath formulae—as the full-text search engine *Lucene* currently does for mathematical vernacular—, and the search form and the search results page offered by *MathWebSearch* will be integrated into SWiM's user interface. Similarly, the LECTORA engine for community-aware reading and browsing of mathematical documents [21] will be connected to SWiM: It will steadily be fed with information about all users' interactions (reading, writing, annotating, setting preferences, . . .) and, based on that, discover communities of practice. According to a user's community, e. g. specific presentations of mathematical symbols could then be applied by SWiM. A long-term case study with LECTORA, SWiM, a course management system and a discussion forum in the context of an introduction to computer science at Jacobs University Bremen will be conducted from fall 2007 .

7 Related Work

SWiM was originally motivated by deficiencies in related collaborative systems like *Wikipedia*, *PlanetMath*, and *Connexions* (see sect. 1). Certain recent improvements to these systems are related to SWiM: For *MediaWiki*, a semantic web extension has been developed [28], which aims at being used in the *MediaWiki*-powered ***Wikipedia***. se(ma)^{2}wi [33] is a *Wikipedia*-independent experiment with a *Semantic MediaWiki* fed with OMDoc-formatted mathematical knowledge from *ActiveMath*. While the *ActiveMath* learning metadata are displayed in the wiki, 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 and therefore

---

[9] Auto-completion has first been investigated in the semantic wiki *Kaukolu* [8], but only on the RDF level in that case.

cannot be exploited for semantic navigation. The automatic linking algorithm of ***PlanetMath***, which uses natural language heuristics, is currently being generalised and modularised into an independent component. The plans for extending SWIM put higher emphasis on semi-automatically assisting manual linking (see sect. 6) instead, as the links currently supported by SWIM do not occur in mathematical vernacular but as formal annotations. The ***Connexions*** developers are working on lenses—customised views on the content according to user/community preferences, quality rankings, or trust considerations [4]—, a feature that has not been considered for SWIM so far.

8 Conclusion

The architecture and user interface of SWIM, a collaborative environment for managing mathematical knowledge, has been presented. The current prototype, based on *IkeWiki* but using the OMDOC format for pages, provides the basic editing and browsing features of a semantic wiki. Thanks to the underlying ontology, SWIM can consistently be extended to other sciences and serve as an integrated platform for various services. Selected services will be implemented and tested in case studies with scientists—including researchers and learners—to find out how to best support scientific working and thinking with software tools.

References

[1] D. Aumüller and S. Auer. Towards a semantic wiki experience – desktop integration and interactivity in WikSAR. In *Proc. of 1st Workshop on The Semantic Desktop*, Nov. 2005.

[2] R. Ausbrooks, S. Buswell, D. Carlisle, S. Dalmas, S. Devitt, A. Diaz, M. Froumentin, R. Hunter, P. Ion, M. Kohlhase, R. Miner, N. Poppelier, B. Smith, N. Soiffer, R. Sutor, and S. Watt. Mathematical Markup Language (MathML) version 2.0 (second edition). W3C recommendation, W3C, 2003.

[3] Connexions Team. Connexions: Sharing knowledge and building communities. `http://cnx.org/aboutus/publications/ConnexionsWhitePaper.pdf`, February 2006. seen August 2006.

[4] K. Fletcher. Lenses: Proposed functional description and high level design. Draft available at `http://rhaptos.org/architecture/lenses/`, June 2007.

[5] B. Hendricks and A. Galvan. The Connexions Markup Language (CNXML). `http://cnx.org/aboutus/technology/cnxml/`, 2007. Seen June 2007.

[6] 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*, number 4108 in LNAI. Springer, 2006.

[7] P. Jansen. An Emacs mode for editing OMDoc documents. In OMDOC – *An open markup format for mathematical documents [Version 1.2]* [11], chapter 26.16.

[8] M. Kiesel. Kaukolu: Hub of the semantic corporate intranet. In Völkel et al. [29].

[9] A. Kohlhase and M. Kohlhase. CPoint: Dissolving the author's dilemma. In A. Asperti, G. Bancerek, and A. Trybulec, editors, *Mathematical Knowledge Management*, number 3119 in LNAI. Springer, 2004.

[10] A. Kohlhase and N. Müller. Added-value: Getting people into semantic work environments. In *Emerging Technologies for Semantic Work Environments*. Idea Group, 2007. To appear; chapters under review.

[11] M. Kohlhase. OMDOC – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in LNAI. Springer, 2006.

[12] M. Kohlhase and I. Şucan. A search engine for mathematical formulae. In T. Ida, J. Calmet, and D. Wang, editors, *Proc. of Artificial Intelligence and Symbolic Computation*, number 4120 in LNAI. Springer, 2006.

[13] A. Krowne. An architecture for collaborative math and science digital libraries. Master's thesis, Virginia Tech, 2003. Available at `http://scholar.lib.vt.edu/theses/available/etd-09022003-150851/`.

[14] C. Lange. SWiM – a semantic wiki for mathematical knowledge management. Technical report, Jacobs University Bremen, 2007.

[15] C. Lange. Towards a Semantic Wiki for Science. `http://kwarc.info/projects/swim/pubs/swimplus-resprop.pdf`, Feb. 2007. Ph. D. research proposal.

[16] O. Lassila and R. R. Swick. Resource description framework (RDF) model and syntax specification. W3C recommendation, W3C, 1999. `http://www.w3.org/TR/1999/REC-rdf-syntax`.

[17] B. Leuf and W. Cunningham. *The Wiki Way: Collaboration and Sharing on the Internet*. Addison-Wesley Professional, 2001.

[18] D. L. McGuinness and F. van Harmelen. OWL web ontology language overview. W3C recommendation, W3C, Feb. 2004. Available at `http://www.w3.org/TR/2004/REC-owl-features-20040210/`.

[19] E. Melis, G. Goguadze, M. Homik, P. Libbrecht, C. Ullrich, and S. Winterstein. Semantic-aware components and services of activemath. *British Journal of Educational Technology*, 37(3), 2006.

[20] E. Melis and J. Siekmann. Activemath: An intelligent tutoring system for mathematics. In *Seventh International Conference 'Artificial Intelligence and Soft Computing' (ICAISC)*, volume 3070 of *LNAI*. Springer, 2004.

[21] C. Müller. Lectora – towards an interactive, collaborative reader for mathematical documents. `http://kwarc.info/cmueller/papers/Mueller_ResearchProposal_2007-03-14.pdf`, Mar. 2007. Ph. D. research proposal.

[22] N. Müller. An Ontology-Driven Management of Change. In *Wissens- und Erfahrungsmanagement LWA (Lernen, Wissensentdeckung und Adaptivität) conference proceedings*, 2006.

[23] E. Oren, R. Delbru, K. Möller, M. Völkel, and S. Handschuh. Annotation and navigation in semantic wikis. In Völkel et al. [29].

[24] A. G. Palomo. Sentido: an authoring environment for OMDoc. In OMDOC – *An open markup format for mathematical documents [Version 1.2]* [11], chapter 26.3.

[25] S. Schaffert. Semantic social software – semantically enabled social software or socially enabled semantic web? In Sure and Schaffert [26].

[26] Y. Sure and S. Schaffert, editors. *Semantics: From Visions to Applications*, 2006.

[27] K. Sutner. Converting MATHEMATICA notebooks to OMDoc. In OMDOC – *An open markup format for mathematical documents [Version 1.2]* [11], chapter 26.17.

[28] M. Völkel, M. Krötzsch, D. Vrandečić, H. Haller, and R. Studer. Semantic Wikipedia. In *Proc. WWW 2006*, May 2006.

[29] M. Völkel, S. Schaffert, and S. Decker, editors. *1st Workshop on Semantic Wikis*, volume 206 of *CEUR Workshop Proc.*, Budva, Montenegro, June 2006.

[30] D. Vrandečić and M. Krötzsch. Reusing ontological background knowledge in semantic wikis. In Völkel et al. [29].

[31] External editors (from Wikimedia meta-wiki). `http://meta.wikimedia.org/w/index.php?title=Help:External_editors&oldid=491214`, June 2006.

[32] Wikipedia, the free encyclopedia. `http://www.wikipedia.org`, 2001–2006.

[33] C. Zinn. Bootstrapping a semantic wiki application for learning mathematics. In Sure and Schaffert [26].