A Semantic Wiki for Mathematical Knowledge Management

Christoph Lange[1] and Michael Kohlhase[2]

[1] Informatik, Universität Trier, lange@castor.uni-trier.de
[2] Computer Science, International University Bremen, m.kohlhase@iu-bremen.de

Abstract. We propose the architecture of a semantic wiki for collaboratively building, editing and browsing a mathematical knowledge base. Its hyperlinked pages, containing mathematical theories, are stored as OMDoc, a markup format for mathematical knowledge representation. Our long-term objective is to develop a software that, on the one hand, facilitates the creation of a shared, public collection of mathematical knowledge (e.g. for education). On the other hand the software shall serve work groups of mathematicians as a tool for collaborative development of new theories.

1 A Semantic Web for Science and Technology via Mathematical Knowledge Management (MKM)

The Internet plays an ever-increasing role in our everyday life, and science is no exception. It is plausible to expect that the way we do (conceive, develop, communicate about, and publish) mathematics will change considerably in the next ten years. In particular, most of the mathematical activities will be supported by mathematical software systems (we will call them *mathematical services*) connected by a commonly accepted distribution architecture. It is a crucial but obvious insight that true cooperation of mathematical services is only feasible if they have access to a joint corpus of mathematical knowledge[3]. Therefore, a central prerequisite for this is the creation of a technology that is capable to create, maintain, and deploy *content-oriented* libraries of mathematics on the web. The world wide web is already now the largest single resource of mathematical knowledge, and its importance will be exponentiated by the emerging display technologies like MathML, which integrates LaTeX-quality presentation into the hypertext and multimedia capabilities of the WWW.

> The **Semantic Web** is a *Web* of *data* for *applications*, just as the WWW is a web of documents for humans.

If we extend this vision of Tim Berners-Lee's to mathematics on the web, many services come into mind:

[3] Be it one central knowledge base or many of them glued together through an exchange mechanism

1. cut and paste on the level of computation (take the output from a web search engine and paste it into a computer algebra system).
2. automatically checking published proofs, if they are sufficiently detailed and structured.
3. math explanation (e.g. specializing a proof to an example that simplifies the proof in this special case).
4. semantic search for mathematical concepts (rather than keywords): "Are there any objects with the group property out there?"
5. data mining for representation theorems: "Are there undiscovered groups out there?"
6. classification: given a concrete mathematical structure, is there a general theory for it?

All of these services can currently only be performed by humans, limiting the accessibility and thus the potential value of the information. On the other hand, the content-oriented mathematical libraries can only be generated by humans, as it has been proved by the successful *PlanetMath* project[4], which features free, collaboratively created entries on more than 8,000 mathematical concepts. *PlanetMath*, however, is not completely machine-understandable. There is a fixed set of metadata associated with each article, including its type (definition, theorem, etc.), parent topic, Mathematics Subject Classification, synonyms and keywords, but the content itself is written in LaTeX and can only be searched in full-text mode.

2 Semantic MK Markup with OMDoc

We will make use of the structural/semantic markup approaches using formats such as OpenMath [BCC+04], MathML [ABC+03], and OMDoc (Open Mathematical Documents [Koh06]), the latter of which embeds and extends the former ones. These formats, constituting the state of the art for representing mathematical knowledge, are now used in a large set of projects in automated theorem proving, eLearning, ePublishing, and in formal digital libraries. OMDoc builds on a semantic representation format for mathematical formulae (OpenMath objects or Content MathML representations) and extend this by an infrastructure for context and domain models from "formal methods". In contrast to those, these structural/semantic approaches do not require the full formalization of mathematical knowledge, but only the explicit markup of important structural properties. For instance, a statement will already be considered as "true" if there is a proof object that has certain structural properties, not only if there is a formally verifiable proof for it. Since the structural properties are logic-independent, a commitment to a particular logical system can be avoided without losing the automatic knowledge management which is missing for semantically unannotated documents. Work on the OMDoc format shows that most added-value services in knowledge management do not need tedious

[4] http://www.planetmath.org, see also [Kro03]

formalization, but can be based on the structural/semantic level. OMDoc assumes a three-layered structure model for semantic representation formalisms:

Object level: represents objects such as complex numbers, derivatives, equations etc. Semantic representation formats typically use functional characterizations that represent objects in terms of their logical structure, rather than specifying their presentation. This avoids ambiguities which would otherwise arise from domain specific representations.

Statement Level: (natural/social/technological) sciences are concerned with modeling our environment, more precisely with statements about the objects in it. We can distinguish different types of statements: model assumptions, their consequences, hypotheses, and measurement results. All of them have in common that they state relationships between scientific objects and have to be verified or falsified in theories or experiments. Moreover, all these statements have a conventionalized structure, such as Exercise, Definition, Theorem, Proof, and a standardized set of relations among each other. For instance, a model is fully determined by its assumptions (also called *axioms*); all consequences are deductively derived from them (via *theorems* and *proofs*), and therefore their experimental falsification uncovers false assumptions of the model.

Theory/Context Level: Representations always depend on the ontological context; even the meaning of a single symbol[5] is determined by its context, and depending on the current assumptions, a statement can be true or false. Therefore the sciences (with mathematics leading the way) have formed the habit to fix and describe the situation of a statement. Unfortunately, the structure of these situation descriptions remains totally implicit, and can therefore not be used for computer-supported management. Semantic representation formats make this structure explicit. In mathematical logic, a theory is the deductive closure of a set of axioms, i.e. the (in general infinite) set of logical consequences of the model assumptions. Even though this fully explains the phenomenon context in theory, important aspects like the re-use of theories, knowledge inheritance, and the management of theory changes are disregarded completely. Therefore, formalisms with context level use elaborate inheritance structures for theories, e.g. in form of ontologies in the Semantic Web or in form of "algebraic specifications" in program verification.

An important trait of the three-layer language architecture is the inherent dependency loop between the object and theory levels mediated by the statement level: the objects obtain their meaning from the theories their functional components are at home in, and the theories are constituted by special statements, and in particular the objects that are contained in them. Making these structures explicit enables the mechanization and automation of knowledge management and the unambiguous, flexible communication of mathematical objects

[5] e.g. the glyph h as the height of a triangle or Planck's quantum of action.

and knowledge that is needed for meaningful interoperability of software systems in science.

3 Cross-Fertilization of MKM and Wiki

Even though the work reported here was initially motivated by solving the MKM author's dilemma (see below), we contend that the new application area MKM can also contribute to the development of semantic wikis.

3.1 Benefits of a Wiki for MKM

As any semantic or traditional wiki, a wiki environment for MKM encourages users to collaborate: Non-mathematicians can collaborate in creating a "Wikipedia of mathematics" by compiling the knowledge available so far, while scientists can collaboratively develop new theories. However, to encourage users to contribute, wiki-like openness to anybody probably won't suffice. Unlike the text formats used by common semantic wikis, the OMDoc format makes the fine-grained semantic structure 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 the added-value services supported by the format — e.g. the creation of customized textbooks [MS04]. If author and beneficiary of such services were different persons, though, only few persons would be willing to contribute to a knowledge base. This "MKM author's dilemma" [KK04] can be overcome when the authors themselves are rewarded for their contributions by being offered added-value services, which improve immediately the more annotations and cross-references the users contribute, — for example a facility for navigation through the knowledge base along paths of semantic relations between the theories, which are computed from the OMDoc document collection.

Furthermore, mathematicians developing theories will be assisted to retain an overview of theory dependencies in order not to break them. Social software services will further utilize the semantic information available from the theories and from tracking the user interaction log ("Who did what on which page when?").

3.2 An Alternative 'Semantic Web'

Most semantic wikis are based on ideas and techniques from Berners-Lee's Semantic Web. In accordance with the general definition in the introduction, the Semantic Web uses RDF triples [LS99] to describe resources such as XML fragments in documents and the background knowledge in ontologies to draw inferences about their content. Note that the Semantic Web makes a conceptual *division between data* (arbitrary objects — called "resources" — that can be identified by URI references; usually XML fragments) and *context* (encoded in topic maps, or an ontology language like OWL [W3C04] or KIF [Gea92]). In contrast to this, content/context markup systems like OMDoc consider scientific *knowledge as the primary data* and take the *context* to be made up of *reified knowledge* (see the discussion in section 2). This makes collections of OMDoc documents into *referentially closed systems* (all the knowledge referred to can be expressed in the system itself), which in turn allows *ontological bootstrapping* (the ontologies needed to draw inferences can be built up as we build up the data). Note that only part of the mathematical knowledge embedded in mathematical documents can be exploited for ontological reasoning[6], as it cannot faithfully be expressed in first-order logic (much less so in description logics). Consider for instance the following fragment from a math book:

> **Definition**: $f \in \boxed{\mathcal{C}^0(\mathbb{R},\mathbb{R})}$, iff for all $x, y \in \mathbb{R}$ and $\epsilon > 0$, there is a $\delta > 0$, such that $|f(x) - f(y)| < \epsilon$ if $|x - y| < \delta$.
> **Definition**: $f \in \boxed{\mathcal{C}^0(\mathbb{R},\mathbb{R})}$, iff for all $x \in \mathbb{R}$ and $\epsilon > 0$, there are $f'(x)$ and $\delta > 0$, such $\left|\frac{|f(x)-f(x+h)|}{h} - f'(x)\right| < \epsilon$ for $h < \delta$.
> **Examples**: If $f(x) := |x|$ and $g(x) := 3x^2 + 2x - \pi$, then $\boxed{f \in \mathcal{C}^0(\mathbb{R},\mathbb{R})}$ and $\boxed{g \in \mathcal{C}^1(\mathbb{R},\mathbb{R})}$, but $\boxed{f \notin \mathcal{C}^1(\mathbb{R},\mathbb{R})}$.
> **Theorem**: $\boxed{\mathcal{C}^1(\mathbb{R},\mathbb{R}) \subseteq \mathcal{C}^0(\mathbb{R},\mathbb{R})}$
> **Proof**: Let $f \in \mathcal{C}^0(\mathbb{R},\mathbb{R})$, $x \in \mathbb{R}$ and $\delta = \epsilon > 0$, then $|f(x) - f(y)| \leq h \cdot |f(x)| \ldots$

Here, only the boxed fragments contain taxonomic information. Its justifications via ϵ/δ arguments cannot be (simultaneously) be expressed in description logics. Thus any web ontology that deals with objects such as the ones above will necessarily have to approximate the underlying mathematical knowledge.

Generally in science, knowledge comes in documents and constitutes the context, whereas description logic ontologies only reference and approximate the knowledge in a document. Therefore, with OMDoc we propose an *alternative vision for a 'semantic web for science and technology'* where the ontologies necessary for drawing inferences are views derived from normative documents. Where ontological fragments cannot be derived automatically (an interesting research problem in itself), they can be embedded into OMDoc-encoded documents as OWL, and later simply extracted. Thus OMDoc — as an document format with embedded taxonomic information — serves as its own ontology language.

3.3 Opportunities of MKM for Semantic Wikis

The enhancements of the data model semantic wikis bring along — compared to traditional wikis — are already present in the OMDoc format, so that an OMDoc-based wiki only needs to operationalize their underlying meaning. For

[6] For the sake of this argument we will use the term web ontology language synonymously with "description logic", as in OWL-DL; if we pass to more expressive logics like KIF, we lose decidability and thus the raison d'être for web ontologies.

example, typed links, which are implemented via an extension to the wiki syntax in *Semantic MediaWiki* [VKVH06] or editable through a separate editor in *IkeWiki* [Sch06], are implemented by means of the `for` attribute to OMDoc's elements (e.g. `<example for="#id-of-assertion">`). It remains left to the wiki to make them editable easily and to visualize them adequately.

More than a general semantic wiki, one targeted at mathematics must ensure that dependencies between concepts are preserved (see section 4.3). Results in this area will be interesting for non-mathematical semantic wikis as well, especially when they support higher levels of formalization such as ontologies.

4 Design of the OMDoc Wiki

Before we can go into the design of the OMDoc wiki system and the user interaction — including Web-2.0-like added-value services, we will concern ourselves with its information model: what a wiki page should comprise, what semantic information can be inferred from the OMDoc documents and the user interaction logs, and finally how this can be utilized.

4.1 What Should a Page Contain?

The smallest unit in a wiki that can be displayed, edited, linked to, or archived is a page. While in a non-semantic wiki, one page can have arbitrary contents, in a semantic wiki it usually describes one *concept*, including its properties and its relations to other concepts.

OMDoc groups closely related concepts into 'theories' and advises to follow a 'little theories approach' [FGT92], where theories introduce as few new concepts as possible[7]. We follow this intuition and restrict the pages of the OMDoc wiki to single (little) theories to keep them manageable. Moreover, OMDoc distinguishes the knowledge elements in theories into constitutive ones like symbols, definitions, and axioms (these are indispensable for the meaning of the theory) and non-constitutive ones, such as assertions, their proofs, alternative definitions of concepts already defined, and examples. We insist that the latter are rolled out into separate theories (and therefore wiki pages). Small pages also improve the effectivity of wiki usage, as they facilitate re-use by linking and allow for a better overview through lists of recent changes and other automatically generated index pages.

Each theory page has an associated discussion page, which provides an adequate space for questions, answers, and discussions about this theory. OMDoc will be used for discussion pages as well, with some proposed extensions for discussion posts: New elements for questions, explanations, opinions, etc. will be added.

[7] A theory may introduce more than one concept, if they are interdependent, e.g. to 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.

4.2 Utilizable Semantic Information

From the OMDOC wiki we can gain several kinds of semantic information, formally expressed as relations between concepts: First, there are *basic* relations provided by the individual theories. Then, there are basic relations given by the user interaction logs. Further, *inferable* relations can be defined as closures of the former and as unions of theory relations and interaction relations. Finally, there are other useful relations that the authors have to provide by manual annotations. All these definitions of relations are part of the system ontology[8] of the wiki, which will not be editable by the user.

Relations Provided by the OMDoc Theories Theories are related by theory imports (see section 2, "Theory Level") and by relations between their statements ("Statement Level"). For example, if theory t states an assertion using symbols defined in the theories t' and t'' or proves an assertion made in t' using a theorem from t'' as a premise, t is related to t' and t'', but the individual statements are also related to each other in a more fine-grained view.

Semantic information will only be extracted from the theory and statement levels of OMDOC — directly or through reasoning in the case of transitive closures —, not from the object level[9]. The most important relation our application utilizes is the dependency relation between theories, defined by theory import declarations, and the acyclic graph formed by this relation.

Relations Given by User Interaction The basic relation given by user interaction is, "Who *edited* which theory page when?". This information is available for free in a wiki; it can be logged when a page is saved. Accordingly, a relation could be defined which states that a user *read* a theory. This is, however, hard to determine because of HTTP caching. Further relations are defined by user *feedback* to navigation choices proposed by the wiki (see section 4.3).

Inferable Relations Further relations can be inferred from those introduced so far, for example a metric estimating the *degree of difficulty* of a page, calculated by counting the questions on the discussion page. From the user interaction log, sets of related pages can be identified, which are not already related through dependency. For this purpose, a notion of *transaction* must be introduced, i.e. edits carried out by the same user in short succession. Similarly to products bought in online shops, two theories are considered "related" when many users edited them in the same transactions.

Even more sophisticated relations can be inferred from both OMDOC and interaction relations. The software could, for example, track how many examples

[8] not to be confused with the *domain ontology* (for mathematics) embedded in the OMDOC theories.

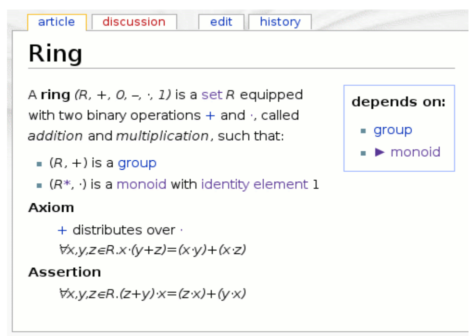
[9] The latter would be suitable for a future integration of computer algebra systems or automated theorem provers.

to a theory users read and improve the difficulty estimation by including those statistics.

4.3 User Interface and Interaction Model

Rendering Theory pages are presented to the user in a human-readable form (XHTML plus presentation MathML) generated by a style sheet. The XHTML contains inline hyperlinks to other theories where appropriate, for instance, from an example to the concept or assertion it explains. As OMDoc documents, however, need not contain any human-readable sections or comments — after all, the knowledge base might be used to support a theorem prover, not to create a textbook! — there is also a source code view with with lines indented, keywords highlighted and URIs displayed as hyperlinks. An intermediate view mode renders mathematical objects in the source code as formulae using MathML or TeX-generated images.

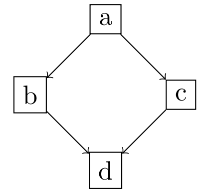
Dynamic Navigation Links Navigation bars with fixed links, such as links to global special pages like the recent changes list, as well as dynamic links to theories depending on the theory t being displayed or related otherwise are provided. Links anchored to particular statements are rendered inline, but links anchored to whole theories — as, for example, imports — must be displayed on a navigation bar. If morphisms from the imported theory to the importing theory are used, as is the case with the import from *monoid* to *ring*, which is used to define that a ring is a monoid w.r.t. multiplication[10], they are also displayed on request. The triangle next to the link to *monoid* in the figure points out that a morphism has been specified.



Dynamic navigation links improve usability by answering the questions "Where am I?" and "Where can I go?" [Nie99]. If dynamic linking directly depends on the page contents editable by the user, as is the case with theory dependency, users are instantaneously gratified for contributing to the structure of dependency by creating connections between theories [Aum05, sec. 3.2].

Navigating the Dependency Graph Not only will the user be able to navigate along the dependency graph, she will also be able to *interact* with the system: she will be asked whether she wants to explore the theories required as dependencies in further detail.

[10] This morphism basically maps the monoid's $\circ$ operator to the ring's multiplication operator $\cdot$ and renames the identity element from e to 1.



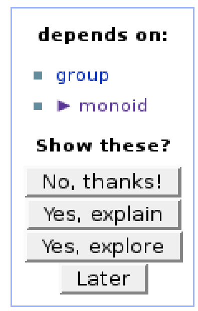
Suppose that the user is currently reading theory a, which depends on b and c, which in turn depend on theory d[11]. In this case the wiki will not only display navigation links to the direct dependencies b and c, but it will also provide unobtrusive buttons that allow the user to give one of the following acknowledgments:

No, thanks! "I already know b and c, please let me just read about a."

Explain "Please show me b and c so that I can learn about a's prerequisites." — b and c will be displayed.

Explore "Please show me *all* prerequisites for a." — In our example, these are b, c, and d, which could be opened in separate windows or serialized into one page.

Suspend "I want to know about b and c, but only later." — The system keeps a notice in the user's profile that she wants to read b and c sometime. Reminder links to suspended theories are shown on a separate navigation bar.



Not only the last case will be recorded — the others are interesting as well for the purpose of *social bookmarking*. For example, if many users requested a theory t to be explained, the system could default to display not only the direct dependencies but also the level-two dependencies, for it seems that t is too difficult for only being explained shallowly.

Furthermore, the system does not only keep track of which theories the user wants to be explained, but also which theories the user has already learned. For each theory, a button will be offered for telling the system "I have learned this". Links to theories learned can then be disabled or displayed in a more unobtrusive color than links to theories that are new to the user.

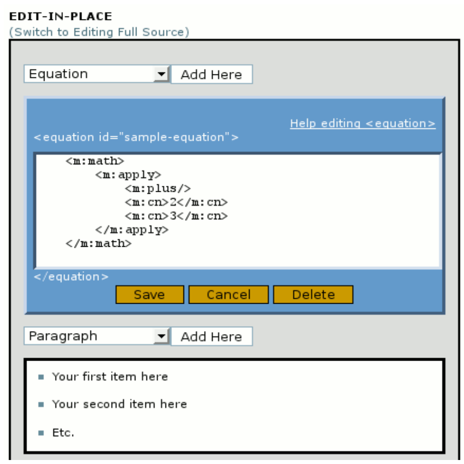
Preserving Dependencies on Editing So far, there has not been any approach to preserving dependencies between pages in a semantic wiki. Tracking dependencies and reasoning about them is an integral part of mathematical practice. Known results are often generalized to find the "real" dependencies, mathematical theories and books are rewritten to make dependencies minimal. Therefore this problem cannot be neglected in a mathematical wiki. In the special case of OMDoc, where dependencies need not be formally verifiable when they have sufficient structural properties (see section 2), a dependency could formally be broken but seem intact to the system anyway. Therefore, we propose a first, simple approach to this problem; a more sophisticated "management of change" process could be integrated later on the basis of work in formal methods [Hut04,AHMS02].

If a theory t depends on a theory t', which can be edited independently from t, modifying t' could break t because some definition in t' required by t might

[11] See [Koh06, fig. 6.1] for a real-world example of such a diamond graph.

have been changed fundamentally[12]. The OMDoc wiki keeps the knowledge base consistent by making hyperlinks not to theories in general, but to certain versions of them. When an author enters a link to `group-theory`, for example, this reference will be stored as `group-theory/latest`. On the other hand, the author of t depending on t' should be notified about updates to t' so that he can benefit from improvements made there. Such notifications can appear statically on the author's watch list[13], but also dynamically in an area near the editing box, while t is being edited. The author then can decide whether to adjust his references from `t'/old` to `t'/improved` — depending on whether t' has really been improved (e.g. with corrections or additional documentation) rather than changed in a dependency-breaking way. The other way round, a user editing t' will be notified that there is a theory t depending on the one he is editing and can decide whether to upgrade t's references to t' or to leave it.

User-friendly Editing The simplest user interface for editing a wiki page is a text area showing the whole contents of the page. As editing OMDoc theories this way is tedious, our wiki will provide alternatives.



The Ajax-based Edit-in-place interface from *Rhaptos* (the software run by *Connexions* [CNX06,The06b], a community-driven collection of open educational content) will be tailored to editing OMDoc. Edit-in-place [The06a] can insert or edit several types of page sections: paragraphs, equations, lists, and more. All sections are displayed in a near-WYSIWYG view, but clicking one of them replaces its view by a text area containing its XML source. Three buttons are displayed below the text area: "Cancel", "Save", and "Delete", the latter two of which commit the editing transaction by sending an asynchronous request to the server.

While Edit-in-place facilitates editing OMDoc on theory and statement level, it is not helpful on the object level because...

1. Mathematical formulae are deeply nested in most cases, while "Edit in place" has been designed with flat XML structures in mind.
2. There are shorter and more intuitive notations for formulae than OpenMath or Content MathML.

Therefore the OMDoc wiki allows for entering mathematical objects in the simpler syntax of *QMath* [Pal06], a batch processor that transforms plain text

12 It would be good style to copy t' to a new theory with a different name, anyway.
13 Watch lists are, for example, known from *MediaWiki*

to OMDoc. *QMath* uses tables mapping text symbols to their OMDoc or OpenMath representation; these tables are also made editable in the wiki. The wiki will keep mathematical objects entered in *QMath* in this format for usability reasons, only converting them to OMDoc when pages are exported to another application.

The same way as *QMath* facilitates the creation of mathematical formulae, wiki text syntax will be offered as a simple way to enter OMDoc's rich text [Koh06, sec. 14.5].

5 Implementation Notes

The OMDoc wiki presented in this paper is currently in a prototype stage under active development. Once completed, it will be released under an open source license; for earlier versions, please contact the authors. We have based our system on *IkeWiki*[14] system as a development platform because of its

Modular Design of Backend and GUI: There are separate stores for page contents and the knowledge base. After the XML-encoded contents of a page have been read from the database, small modules — so-called "wiklets" — perform tasks like enriching the DOM tree of the page with navigation side bars created from semantic annotations, and then the enriched page is rendered for presentation using customizable XSLT style sheets.

Rich Semantic Web Infrastructure: IkeWiki supports many standards of the Semantic Web. The knowledge base is stored as RDF; OWL-RDFS reasoning and SPARQL queries are supported.

User Assistance for Annotation: Editors for page metadata and link types, which can likely be utilized for editing OMDoc, are available.

Orientation Towards Learning: One objective in IkeWiki's ongoing development is its expansion towards a learning environment [SBB+06]. Upcoming versions will likely qualify as a base for an OMDoc wiki with learning features (see section 4.3).

Some parts of the OMDoc wiki will, however, be very different from *IkeWiki*'s operating principles and hence require substantial amounts of refactoring and rewriting — for example:

- The presentation view of a page, for example, cannot be generated by a single-pass XSL transformation from OMDoc to XHTML+MathML; instead, the multi-level OMDoc presentation workflow [Koh06, sec. 25] has to be adopted.
- The semantic relations between OMDoc theories are not exclusively stored in the RDF knowledge base, as is the case with semantic relations between *IkeWiki* pages; instead, the OMDoc wiki has to keep the annotations in OMDoc synchronized with the knowledge base, which will still be used for reasoning.

[14] http://ikewiki.salzburgresearch.at, see also [Sch06]

6 Conclusion and Outlook

Mission . . . The upcoming release of the OMDoc wiki presented in this paper will offer a user friendly editor for OMDoc's XML source code (section 4.3). Pages are viewable as XHTML+MathML as well as hyperlinked source code (4.3). Semantic relations, to be displayed on a navigation bar, will be inferred from the dependency relation between theories (4.3). Learning will be supported through interactive navigation along the dependency graph (4.3). There will be a simple assistance helping users preserve dependencies (4.3).

Later we will improve display of semantic relations, also taking into account the more fine-grained relations inferable from OMDoc's statement level (4.2) and from user interaction alone (4.2). Once techniques for management of changes to OMDoc documents have been developed, they will be integrated into the wiki to offer a more sophisticated dependency preservation.

. . . and Vision With the OMDoc wiki we pursue an alternative vision of a 'Semantic Web'. Like Tim Berners-Lee's vision we aim to make the web (here mathematical knowledge) machine-understandable instead of merely machine-readable. However, instead of a top-down metadata-driven approach, which tries to approximate the content of documents by linking them to web ontologies (expressed in terminological logics), we explore a bottom-up approach and focus on making explicit the intrinsic structure of the underlying scientific knowledge. A connection of documents to web ontologies is still possible, but a secondary effect: In OMDoc we can have explicit taxonomic relations as in "all rings are groups" — where the taxonomy is given by definition — or even implicit ones as in "all differentiable functions are continuous" — where the taxonomy is expressed by a theorem. If these theorems and definitions are of a suitable form, or explicitly indicated to be taxonomic by the author, we can harvest this information and transform it into a web ontology format such as OWL [W3C04] and make it available to the Semantic Web.

References

[ABC+03] Ron Ausbrooks, Stephen Buswell, David Carlisle, Stphane Dalmas, Stan Devitt, Angel Diaz, Max Froumentin, Roger Hunter, Patrick Ion, Michael Kohlhase, Robert Miner, Nico Poppelier, Bruce Smith, Neil Soiffer, Robert Sutor, and Stephen Watt. Mathematical Markup Language (MathML) version 2.0 (second edition). W3C recommendation, World Wide Web Consortium, 2003. Available at http://www.w3.org/TR/MathML2.

[AHMS02] Serge Autexier, Dieter Hutter, Till Mossakowski, and Axel Schairer. The development graph manager MAYA (system description). In Hélène Kirchner and Christophe Ringeissen, editors, *Algebraic Methodology and Software Technology — 9th International Conference AMAST 2002*, number 2422 in LNCS, pages 495–500. Springer Verlag, 2002.

[Aum05] David Aumüller. SHAWN: Structure Helps a Wiki Navigate. In W. Müller and R. Schenkel, editors, *Proceedings of the BTW-Workshop "WebDB Meets IR"*, March 2005.

[BCC$^+$04] Stephen Buswell, Olga Caprotti, David P. Carlisle, Michael C. Dewar, Marc Gaetano, and Michael Kohlhase. The Open Math standard, version 2.0. Technical report, The Open Math Society, 2004. http://www.openmath.org/standard/om20.

[CNX06] Connexions. Project home page at http://cnx.org, seen March 2006.

[FGT92] William Farmer, Josuah Guttman, and Xavier Thayer. Little theories. In D. Kapur, editor, *Proceedings of the 11th Conference on Automated Deduction*, volume 607 of *LNCS*, pages 467–581, Saratoga Spings, NY, USA, 1992. Springer Verlag.

[Gea92] M. Genesereth and R. Fikes et al. Knowledge interchange format: Version 3.0 reference manual. Technical report, Computer Science Department, Stanford University, 1992.

[Hut04] Dieter Hutter. Towards a generic management of change. In Christoph Benzmüller and Wolfgang Windsteiger, editors, *Computer-Supported Mathematical Theory Development*, number 04-14 in RISC Report Series, pages 7–18. RISC Institute, University of Linz, 2004. Proceedings of the first "Workshop on Computer-Supported Mathematical Theory Development" held in the frame of IJCAR'04 in Cork, Ireland, July 5, 2004. ISBN 3-902276-04-5. Available at http://www.risc.uni-linz.ac.at/about/conferences/IJCAR-WS7/.

[KK04] Andrea Kohlhase and Michael Kohlhase. CPoint: Dissolving the author's dilemma. In Andrea Asperti, editor, *Mathematical Knowledge Management, MKM'04*, number 3119 in LNAI, pages 175–189. Springer Verlag, 2004.

[Koh06] Michael Kohlhase. OMDoc *An open markup format for mathematical documents (Version 1.2)*. LNAI. Springer Verlag, 2006. to appear, manuscript at http://www.mathweb.org/omdoc/pubs/omdoc1.2.pdf.

[Kro03] Aaron Krowne. An architecture for collaborative math and science digital libraries. Master's thesis, Virginia Tech, 2003.

[LS99] Ora Lassila and Ralph R. Swick. Resource description framework (RDF) model and syntax specification. W3C recommendation, World Wide Web Consortium (W3C), 1999. http://www.w3.org/TR/1999/REC-rdf-syntax.

[MS04] E. Melis and J. Siekmann. Activemath: An intelligent tutoring system for mathematics. volume 3070, pages 91–101. Springer-Verlag, 2004.

[Nie99] Jakob Nielsen. *Designing Web Usability : The Practice of Simplicity*. New Riders Press, 1999.

[Pal06] Alberto González Palomo. QMath: A human-oriented language and batch formatter for OMDoc. In OMDoc *An open markup format for mathematical documents (Version 1.2)* [Koh06], chapter 27.2. to appear, manuscript at http://www.mathweb.org/omdoc/pubs/omdoc1.2.pdf.

[SBB$^+$06] Sebastian Schaffert, Diana Bischof, Tobias Bürger, Andreas Gruber, Wolf Hilzensauer, and Sandra Schaffert. Learning with semantic wikis. http://www.wastl.net/download/paper/Schaffert06_SemWikiLearning.pdf, 2006.

[Sch06] Sebastian Schaffert. IkeWiki: A semantic wiki for collaborative knowledge management. Technical report, Salzburg Research Forschungsgesellschaft, 2006.

[The06a] The Connexions Team. Connexions: Help on editing modules. http://cnx.org/help/EditingModules, 2006.

[The06b] The Connexions Team. Connexions: Sharing knowledge and building communities. http://cnx.org/aboutus/publications/ConnexionsWhitePaper.pdf, 2006.

[VKVH06] Max Völkel, Markus Krötzsch, Denny Vrandečić, and Heiko Haller. Semantic Wikipedia. In *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, May 23-26, 2006*, May 2006.

[W3C04] OWL web ontology language overview. W3C Recommendation, World Wide Web Consortium, February 2004.