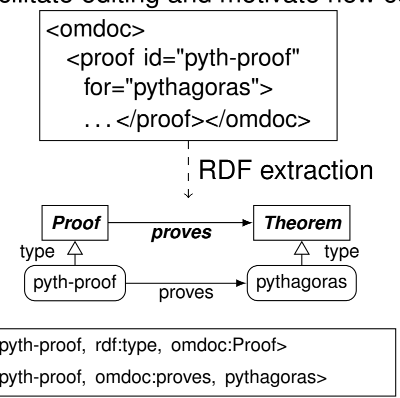
SWiM – A Semantic Wiki for Mathematical Knowledge Management

SWiM is a semantic wiki for collaboratively building, editing and browsing mathematical knowledge represented in the domain-specific structural semantic markup language OMDoc. It motivates users to contribute to collections of mathematical knowledge by instantly sharing the benefits of knowledge-powered services with them. SWiM is currently used for authoring content dictionaries (collections of uniquely identified mathematical symbols) and prepared for managing a large-scale proof formalisation effort.

Research Background and Application Context: Mathematical Knowledge Management Collaboratively authoring *documents* is essential in science—taking down first hypotheses, circulating informal drafts inside a small group, and structuring or reorganising existing knowledge items, finally leading to a consistent publication. We particularly focus on the domain of mathematics and on authoring tools that utilise the knowledge contained in the documents. Several *semantic markup* languages have recently been developed to represent the clear and hierarchical structures of mathematics. The XML languages MathML, OpenMath, and OMDoc[1] particularly target the web. OMDoc, employing MathML or OpenMath for representing the functional structure of *formulæ*—as opposed to their visual appearance—and adding support for *statements* (like symbol declarations or axioms) and *theories*, has many applications in publishing, education, research, and data exchange. *Acquiring* a large collection of formalised knowledge that can power such added-value services is challenging. On the web the workload can be distributed among many authors, but as semantic markup makes fine-grained structures explicit, it is tedious to author.

Key Technology: Semantic Wiki and Ontologies In this context a semantic wiki comes in handy. OMDoc supports all levels of formalisation, from human-readable texts to fully formal representations for automated theorem proving, and semantic wikis have been found appropriate for collaboratively refining knowledge models and motivating users by instant gratification. The ultimate goal of our work is to achieve a feedback loop where users are supported in contributing well-structured knowledge, which is then exploited by services, which in turn facilitate editing and motivate new contributions.



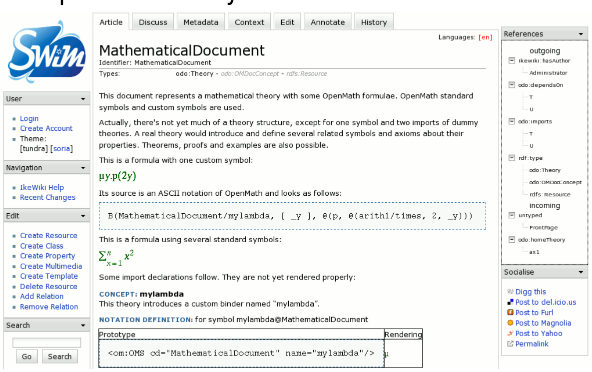
RDF extraction from OMDoc markup in a wiki page

Mathematical structures are deeper than the flat structures found in most semantic wikis: A document can contain theories containing statements that contain formulæ referring to symbols defined in other theories. But in a wiki aims at small pages to prevent editing conflicts and to facilitate search and navigation, so we adapted OMDoc's model of knowledge by putting one mathematical statement or one theory on one wiki page. To make this knowledge compatible to the semantic web, information about the resources represented by pages and their interrelations (e. g. "a *proof for* the Pythagorean theorem") is extracted to RDF, using an RDF vocabulary of mathematical structures. This *document ontology* contains e. g. the information that both theorems and proofs are specialisations of a general "mathematical statement", and that a proof can prove a theorem. Moreover, generic transitive dependency and containment relations have been modeled. For example, having one theory import another theory (and reusing symbols defined there) establishes a dependency. One theory logically contains its statements; similarly, statements can contain sub-statements, as in the case of a proof that consists of multiple steps.

The SWiM 0.2 Prototype: IkeWiki + OMDoc As a base system for the implementation, we chose IkeWiki[2] because it offered the richest XML infrastructure—a key requirement for adding OMDoc support—and was found to be most extensible. To foster stepwise formalisation of informal text, we

[1] See http://w3.org/Math/, http://www.openmath.org, and http://www.omdoc.org, resp.
[2] http://ikewiki.salzburgresearch.at

mixed OMDoc fragments with the HTML-like wiki markup. In the editing view, statement- and theory-level structures of OMDoc are made accessible as tables, whereas semantically structured mathematical formulæ are editable in an ASCII notation of OpenMath. Additional XSLT template rules care for rendering OMDoc fragments embedded into pages. *Notation definitions* for every semantic symbol specify how to render mathematical formulæ for the human reader. The OMDoc document ontology is preloaded into the RDF store. RDF triples are extracted from the OMDoc markup upon saving a page or importing a local file. Documents are browsable via inline links manually set in the informal parts, via links from occurrences of symbols in formulæ to the place of their declaration, set by the formula renderer, and via RDF links, displayed in a separate box by IkeWiki. The latter comprise those triples that are extracted from the markup, as well as triples inferred by a reasoner.



A mathematical document in SWiM

SWiM relies on the ontology for reacting on changes to notation definitions. When an author changes a notation definition for a symbol, exactly those wiki pages that contain a formula using the symbol or that include other pages containing such formulæ need to be re-rendered. The occurrences of the symbol can be looked up easily in the RDF graph. This service allows for instant visual debugging of notation definitions. For upcoming releases, more ontology-powered services are planned, including more general change management, learning assistance, and editing facilitations like auto-completion of link targets. There is some evidence that many services can be based on the most generic relations of dependency and (physical or logical) containment. We envisage SWiM as a development environment for scientists and knowledge engineers that conveniently supports refactorings of knowledge.

Use Cases and Applications Now that viewing, browsing, editing, importing and exporting mathematical documents works, we are evaluating SWiM in practical settings. The **Flyspeck** project is about large-scale formalisation of a proof of the Kepler conjecture. We are starting to support this effort by "crowdsourcing" the knowledge compiled so far (hundreds of proof sketches that are not yet machine-verifiable) on a SWiM site. The main challenge is giving an interested visitor an impression of the extent of the project and showing him where work needs to be done. We are investigating how to automatically convert the original LaTeX sources to HTML with MathML, then to informal OMDoc, to break that into wiki pages, and to let users formalise them stepwisely. For the upcoming **OpenMath 3** standard, SWiM is being extended to an editor for Content Dictionaries, which are similar to OMDoc theories but simpler. There, mainly editing metadata and notation definitions is of interest.

Conclusion SWiM makes mathematical documents editable collaboratively and facilitates browsing them by exploiting the knowledge they contain. Domain-specific services are powered by an ontology that models structures of the domain—an advantage over generic semantic wikis, which could not offer specific services for mathematical knowledge. Competing non-semantic approaches like the math encyclopædia *PlanetMath*[3] are less flexible, as they cannot exploit the structures of their presentation-oriented LaTeX formulæ and rely on a fixed set of metadata. Most services for editing and browsing need to be hard-coded, which potentially restricts the scale of knowledge managment tasks the systems can be applied to. The SWiM approach of integrating a semantic markup language into a wiki by choosing an appropriate page granularity, modeling a document ontology, and extracting relevant facts from the markup into RDF has successfully been applied to OMDoc and the closely related but syntactically different OpenMath and is likely to be portable to other domains as well, e. g. for the chemical markup language CML.

Contact	Prof. Dr. Michael Kohlhase	Christoph Lange	**Jacobs University Bremen**
Phone:	+49 (0)421 200–3140	+49 (0)421 200–3194	
www.kwarc.info	m.kohlhase@	ch.lange@	jacobs-university.de

[3] http://planetmath.org