SWiM – A Semantic Wiki for Mathematical Knowledge Management

Christoph Lange

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

Abstract. 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 being used for authoring content dictionaries, i. e. collections of uniquely identified mathematical symbols, and prepared for managing a large-scale proof formalisation effort.

1 Research Background and Application Context: Mathematical Knowledge Management

A great deal of scientific work consists of collaboratively authoring *documents*—taking down first hypotheses, commenting on results of experiments, circulating informal drafts inside a working group, and structuring, annotating, or re-organising existing items of knowledge, finally leading to the publication of a well-structured article or book. Here, we particularly focus on the domain of mathematics and on tools that support collaborative authoring by utilizing the knowledge contained in the documents. In recent years, several *semantic markup* languages have been developed to represent the clearly defined and hierarchical structures of mathematics. The XML languages MathML [9], OpenMath [11], and OMDoc [3] particularly aim at exchanging mathematical knowledge on the web. OMDoc, employing Content MathML or OpenMath representing the functional structure of mathematical *formulæ*—as opposed to their visual appearance—and adding support for mathematical *statements* (like symbol declarations or axioms) and *theories*, has many applications in publishing, education, research, and data exchange [3, chap. 26]. The main challenge is *acquiring* a large collection of OMDoc-formalised knowledge that can power such added-value services. In an open, collaborative environment, the workload can be distributed among many authors, but as semantic markup make fine-grained structures explicit, it is tedious to author. As the community can only benefit from added-value services after a substantial initial investment (writing, annotating and linking) on the author's part, we sought for motivating the author into action by offering "elaborate [. . .] services for the concrete situation" they are in [2].

2   Key Technology: Semantic Wiki and Ontologies

Our research is motivated by the assumption that 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 (cf. [13]). User motivation in semantic wikis by instant gratification has been investigated in earlier works [1]. The ultimate goal of our work is to achieve a feedback loop where users are supported to contribute well-structured knowledge, which is then exploited to offer services, which in turn facilitate editing and motivate new contributions [5].
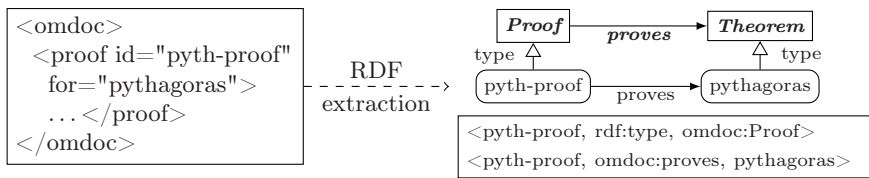

Fig. 1. RDF extraction from OMDoc markup in a wiki page

Semantic markup has deep structures: an OMDoc document can contain theories containing statements that contain formulæ referring to symbols defined in other theories. This is uncommon for most semantic wikis, where the structures are rather flat and one aims at small pages to prevent editing conflicts and to facilitate search and navigation. So to adapt OMDoc's model of knowledge to a semantic wiki, we had to choose an appropriate granularity of wiki pages and arrived at one page holding one mathematical statement or one theory. To make knowledge from OMDoc documents usable on the semantic web, information about the resources represented by pages and their interrelations (e.g. "a *proof for* the Pythagorean theorem") are extracted to RDF. As a vocabulary for this, we modeled OMDoc's structures explicitly in a *document ontology* [5] in OWL-DL. This 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 (Fig. 1). 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.

3   The SWiM 0.2 Prototype: IkeWiki + OMDoc

As a base system for the implementation, we chose IkeWiki [12]. Among the systems evaluated, it offered the richest XML infrastructure—a key requirement

Fig. 2. A mathematical document in SWiM

for adding OMDoc support—and was found to be most extensible [4]. Its backend consists of a PostgreSQL database for the page contents, a Jena RDF store for the RDF graph and the ontologies. Additional ontologies can easily be imported. The frontend heavily relies on the Dojo Ajax toolkit.

Technically, the extension of IkeWiki to SWiM required supporting OMDoc in addition to the HTML-like wiki page format. To foster stepwise formalisation of informal text, we chose to mix OMDoc fragments with wiki markup. Thus we could still rely on IkeWiki's WYSIWYG HTML editor, which just had to be enhanced by support for OMDoc XML elements. Moreover, this choice allowed for an easier maintenance of the OMDoc-related enhancements to the SWiM code base and avoided changes to the underlying database schema. The document ontology is preloaded into the RDF store. RDF triples are extracted from the OMDoc markup upon saving a page or importing an OMDoc file. Additional XSLT template rules care for rendering embedded OMDoc fragments. In order to render mathematical formulæ, there is a *notation definition* for every semantic symbol. These notation definitions can be imported and edited right in the wiki, as parts of OMDoc documents [6]. An efficient, specialised renderer supporting the upcoming MathML 3 standard [10,9] applies them to the symbols in the formulæ. In the editing view, statement- and theory-level structures of OMDoc are made accessible as special HTML tables, whereas mathematical formulæ given in semantic markup are made accessible in a simplified ASCII notation of OpenMath. OMDoc 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 (cf. Fig. 1), as well as triples inferred by a reasoner[1].

SWiM also relies on the ontology for reacting on changes to notation definitions. When an author changes a notation definition n for a symbol s, exactly those wiki pages that contain a formula using s or that include other pages containing such formulæ need to be re-rendered. Looking up the symbol s rendered by n, the formulæ f_i using s, or pages (transitively) including the f_i would be clumsy in the OMDoc XML sources, but is easy in the RDF graph, as this information is extracted from the documents and represented using ontology properties such as *NotationDefinition–renders–Symbol* and *Statement–contains–Formula; Formula–uses–Symbol*. This service allows for instant visual debugging of notation definitions [6]. For upcoming releases, more ontology-powered services are planned, including more general change management, learning assistance, and editing facilitations like editing of subsections and auto-completion of link targets [7]. There is some evidence that many services can be based on the most generic relations of dependency and (physical or logical) containment [5]. With scientists and knowledge engineers in mind, we envisage SWiM as a development environment that conveniently supports refactorings of knowledge[2].

4   Use Cases and Applications

Now that viewing, browsing, editing, importing and exporting mathematical documents basically 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 [8]. The main challenge is giving an interested visitor an impression of the extent of the project and, using appropriate SPARQL queries, showing him where work needs to be done. Currently we are investigating how the original LATEX sources can be utilised by automatically converting them to HTML with MathML, then to informal OMDoc, breaking that into wiki pages, and letting the users formalise them stepwise. For the upcoming **OpenMath 3** standard, SWiM is currently being extended to an editor for OpenMath Content Dictionaries [6], which could be regarded as flat OMDoc theories that just define symbols and do not import anything. There, mainly editing Dublin Core metadata and notation definitions is of interest.

---

[1] The ontology is prepared for DL reasoning, but currently only the RDFS reasoner built into Jena is used.

[2] This is common in mathematics, e. g. in algebra: If one just needs groups, they can be defined by a theory with the four well-known axioms. For explicitly modeling related structures as well, one would break this into smaller theories—*semigroup* just defining an associative operation on a set, *monoid* importing this and extending it by an identity element, and finally the refactored *group*, adding inverse elements.

5   Conclusion and Related Work

SWiM makes mathematical documents editable collaboratively and particularly facilitates browsing them by exploiting the knowledge they contain. Domain-specific services are powered by an ontology that models structures of documents—an advantage over generic semantic wikis, which would not be able to offer additional services for mathematical knowledge. Competing non-semantic approaches like the math encyclopædia *PlanetMath* (evaluated in [4]) 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 [6] and is likely to be portable to other domains as well, e. g. for the chemical markup language CML.

References

1. Aumüller, D., Auer, S.: Towards a semantic wiki experience – desktop integration and interactivity in WikSAR. In: 1st Workshop on The Semantic Desktop (2005)
2. Kohlhase, A., Müller, N.: Added-Value: Getting People into Semantic Work Environments. In: Rech, J., Decker, B., Ras, E. (eds.) Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications, Idea Group (in press, 2008)
3. Kohlhase, M.: OMDoc – An Open Markup Format for Mathematical Documents [version 1.2]. LNCS (LNAI), vol. 4180. Springer, Heidelberg (2006)
4. Lange, C.: SWiM – a semantic wiki for mathematical knowledge management. Technical Report 5, Jacobs University (2007), `http://kwarc.info/projects/swim/pubs/tr-swim.pdf`
5. Lange, C.: Towards scientific collaboration in a semantic wiki. In: Hotho, A., Hoser, B. (eds.) Bridging the Gap between Semantic Web and Web 2.0 (2007)
6. Lange, C.: Mathematical Semantic Markup in a Wiki: The Roles of Symbols and Notations. In: The 3rd Semantic Wiki Workshop at ESWC (submitted, 2008), `http://kwarc.info/projects/swim/pubs/semwiki08-notation-semantics.pdf`
7. Lange, C.: SWIM development roadmap (2008), `https://trac.kwarc.info/swim/roadmap/`
8. Lange, C., McLaughlin, S., Rabe, F.: Flyspeck in a semantic wiki, 2008. In: The 3rd Semantic Wiki Workshop at ESWC 2008 (submitted, 2008), `http://kwarc.info/projects/swim/pubs/flyspeck-wiki-eswc08.pdf`
9. Mathematical Markup Language (MathML) version 3.0. W3C working draft, World Wide Web Consortium (2007), `http://www.w3.org/TR/MathML3`
10. Müller, C., Müller, N., Kohlhase, M.: A library for transforming Content MathML/OpenMath into Presentation MathML (2008), `http://kwarc.info/projects/mmlkit/`

11. The Open Math standard, version 2.0. Technical report, The Open Math Society (2004), `http://www.openmath.org/standard/om20`
12. Schaffert, S.: IkeWiki: A semantic wiki for collaborative knowledge management. In: 1st International Workshop on Semantic Technologies in Collaborative Applications (STICA) (2006)
13. Schaffert, S.: Semantic social software. In: Sure, Y., Schaffert, S. (eds.) Semantics (2006)