Flyspeck in a Semantic Wiki

Collaborating on a Large Scale Formalization of the Kepler Conjecture

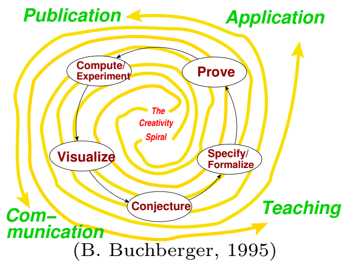
Christoph Lange[1], Sean McLaughlin[2], and Florian Rabe[1]

[1] Computer Science, Jacobs University Bremen,
`{ch.lange,f.rabe}@jacobs-university.de`
[2] School of Computer Science, Carnegie Mellon University, Pittsburgh,
`seanmcl@gmail.com`

Abstract. Semantic wikis have been successfully applied to many problems in knowledge management and collaborative authoring. They are particularly appropriate for scientific and mathematical collaboration. In previous work we described an ontology for mathematical knowledge based on the semantic markup language OMDoc and a semantic wiki using both. We are now evaluating these technologies in concrete application scenarios. In this paper we evaluate the applicability of our infrastructure to mathematical knowledge management by focusing on *the Flyspeck project*, a formalization of Thomas Hales' proof of the Kepler Conjecture. After describing the Flyspeck project and its requirements in detail, we evaluate the applicability of two wiki prototypes to Flyspeck, one based on Semantic MediaWiki and another on our mathematics-specific semantic wiki SWiM.

1 Scientific Communication and the Flyspeck Project

Scientific communication consists mainly of exchanging documents, and a great deal of scientific work consists of collaboratively authoring them. Common instances are writing down first hypotheses, commenting on results of experiments or project steps, and structuring, annotating, or re-organizing existing items of knowledge, as depicted in Buchberger's figure on the right.



(B. Buchberger, 1995)

Semantic markup languages for representing structures of scientific knowledge, and editing tools understanding them, are a promising approach to supporting this work.Besides generic approaches like SALT [6], the most extensive work in semantic markup has been in the domain of mathematics. Mathematical logic, depending on symbols and relationships between symbols, naturally lends itself well to formal exposition. Languages like MathML [24], OpenMath [29], and OMDoc [14] were developed to represent the clearly defined and hierarchical structures of mathematics in a way that preserves the intricate relationships. OMDoc employs Content MathML or OpenMath for structurally representing mathematical *objects* (symbols, numbers, equations, etc.) and adds two layers on top: Objects or informal text can be annotated as mathematical *statements* (symbol declarations, definitions, axioms, theorems, proofs, examples, etc.), and collections of interrelated statements can be grouped into *theories*.

With SWiM, a semantic wiki for mathematical knowledge management [22], we have investigated collaborative editing of OMDoc documents. Additionally, we host a public knowledge base and experimental ground about mathematical knowledge management on the web, powered by Semantic MediaWiki[3]. It has become evident that a wiki is a suitable tool for supporting the workflow of incremental formalization inherent to scientific writing. Wikis have not only shown to be appropriate for *writing*, but are also effective for project management, e. g. in corporate settings [23, 36]. We are therefore interested in applying our technologies to scientific knowledge engineering projects.

(http://tinyurl.com/3bxx2t)
Fig. 1. The face centered cubic packing

The target of our case study is the Flyspeck Project, which seeks to formally verify Thomas Hales' proof of the Kepler Conjecture [8, 9]. This conjecture asserts that the density of a packing of unit spheres in 3 dimensions is at most $\pi/(3\sqrt{2})$, the density of the face centered cubic and hexagonal close packings. Posed by Kepler in 1611, it formed part of Hilbert's 18th problem, and until its solution was recognized as one of the most famous unsolved problems of mathematics. Hales' proof, completed in 1995, was not accepted immediately by the mathematical community. Besides its considerable length, the proof relies essentially on computer calculations. The 300 pages of text and many thousands of lines of computer code made checking the proof for errors in the referee process unusually difficult, leading to a publication delay of nearly 10 years. In 2003, Hales proposed using computers to rigorously check the entire proof in detail, including the computer code. He dubbed this effort *Flyspeck*[4]. The software systems used in such formalizations are called *theorem provers* or *proof assistants*[5], examples being Isabelle [31], Coq [3], and Twelf [32]. With adequate human assistance they can verify that a purported proof follows from a given set of axioms and inference rules.

Modern proof assistants are still far from being able to check proofs at the level given in most journals and textbooks. A typical estimate is that it takes about a week to formalize a single page of mathematical text. Hales expects that it will take around 20 man-years to complete Flyspeck. Hales is compiling a LaTeX

[3] `http://mathweb.org/wiki/`

[4] The word "flyspeck" means, "to examine closely". It was found by Hales using a regular expression search of an English dictionary for the expression "F.*P.*K", for "Formal Proof of Kepler"

[5] The word "formalize" is used in many contexts in this field. In the remainder of this paper, we use "formal" and "formalize" loosely, possibly referring to any degree of colloquial or scientific formalization. We use "computerized" to mean that a theorem, proof or definition has been expressed in a proof assistant. Note that we consider computerized definitions and proofs formal "documents" as well.

book [10] of lemmas from different areas of mathematics that are needed in his proof. Its 450 pages contain a significant percentage of the mathematical results used in the proof, covering such disparate topics as plane, solid, and spherical geometry, graph theory and hypermaps, single and multivariable calculus, and plane and spherical trigonometry.

The first steps toward a computerized proof have already been taken. Nipkow and Bauer [27] proved the correctness of a fundamental algorithm in Isabelle. The other two main parts of the computer code, linear programming and global optimization, are currently being investigated in doctoral dissertations [39, 28]. A project page documents some of this progress and has a source repository containing the book of lemmas, as well as the formalized definitions of some important functions and inequalities [11]. Despite this considerable progress on the computer code, the bulk of the mathematical formalization remains to be done. This formalization will consist of two broad phases. First, a number of elementary mathematical theories (e.g. spherical geometry) need to be defined and the relevant lemmas proved. Then the specific aspects of the Kepler proof that relies on the elementary results need to be formalized. Given the content of the book mentioned above, we suspect that Flyspeck, in its final form, will consist of dozens of theories, with hundreds of definitions and thousands of lemmas.

Flyspeck is particularly appealing as a use case for a semantic wiki approach. While the ultimate result is to be a highly formal computerized proof, the current proof involves both highly formal and semi-formal mathematical knowledge. It contains descriptive and motivating yet informal text that should be preserved for human understanding. This quasi-formal information would be difficult to present in a strictly formal setting of a proof assistant. Secondly, the large number of lemmas, many independent or only loosely coupled, suggests a “crowdsourcing” approach will be beneficial. Both can be supported by a (semantic) wiki, as we will show in the following.

2 Supporting Flyspeck in a Semantic Wiki

Our focus in this work is on making the extent and structure of Flyspeck comprehensible, communicating where work needs to be done, and allowing collaborators to improve the structure and finally to contribute computerized proofs. For this the outline of the whole proof from the book [10] needs to be represented in the wiki, where the mathematical statements (including definitions, lemmas, and theorems) are available in a human-readable way (with formulae in LaTeX or presentational MathML) as well as a computerized presentation suitable for using in a theorem prover. In order to obtain a well-structured network of knowledge items, each mathematical statement should be presented on one wiki page, which shows its human-readable representation taken from the book, offers additional space for annotation, and allows for downloading a formal representation. Here, we are not yet considering formal proof checking *inside* the wiki, but rather using the wiki for communication about the projects and annotation of informal text.

2.1 Scenario

An example usage scenario is as follows (cf. fig. 2). A user wishes to contribute to Flyspeck. She looks at our wiki main page, which shows her what still needs to be done. Preferring trigonometry, she searches for open problems in that field. This returns a list of lemmas related to analysis from which she can choose one that seems possible given her time constraints. She reads the text of a paper proof culled from Hales' book and annotated by other wiki collaborators and downloads the relevant formal definitions and lemmas. She uses a proof assistant to begin formalizing the paper proof. At some point, she needs clarification on some definition and additionally has an idea on how to generalize this lemma. She thus asks for help, makes comments on the discussion pages of the wiki, and refines the annotations of the lemma. She completes her proof, and uploads the proof assistant file to the wiki. The wiki uses a theorem prover to check the proof for correctness and, if it is correct, adds it to the database.

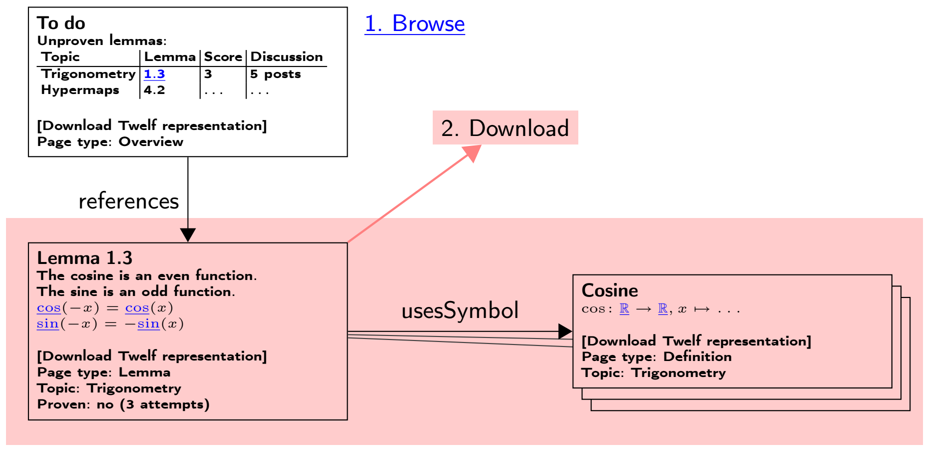

To do
Unproven lemmas:
Topic | Lemma | Score | Discussion
Trigonometry | 1.3 | 3 | 5 posts
Hypermaps | 4.2 | ... | ...
[Download Twelf representation]
Page type: Overview
1. Browse
2. Download
references
Lemma 1.3
The cosine is an even function.
The sine is an odd function.
$\cos(-x) = \cos(x)$
$\sin(-x) = -\sin(x)$
[Download Twelf representation]
Page type: Lemma
Topic: Trigonometry
Proven: no (3 attempts)
usesSymbol
Cosine
$\cos\colon \mathbb{R} \to \mathbb{R}, x \mapsto \ldots$
[Download Twelf representation]
Page type: Definition
Topic: Trigonometry

Fig. 2. Page Structure and Navigation

2.2 Requirements

With this scenario in mind, we propose that the wiki should minimally offer:

A knowledge base of the theory, constant, and lemma definitions.
A theory browser where a user can browse the knowledge by category, or search with keywords.
An editor to annotate and structure informal texts on their way to computerization.
A download area where one can download existing computerized definitions, lemmas, and proofs.
An upload area where one can upload new proofs.
Discussion pages to discuss issues involved in the formalizations.

The following set of annotations should support this minimal infrastructure:

Categorization by topic: In the beginning, one would mirror the narrative structure of the book (e. g. “sphere” being a subsection of “primitive volumes”, which in turn is a section of the chapter “volume calculations”). Standardized ways of classifying mathematical topics, such as the Mathematical Subject Classification (MSC) [1], could be added later.

Project-organization metadata such as whether the proof of a lemma has already been computerized, or if someone is currently attempting a proof. This is essential so that two people do not duplicate work.

Dependency links: These can be links from individual symbols in mathematical formulae to the place where they are declared, or from any page p to other pages containing knowledge that is required for understanding p: either pages in the same wiki, or external resources like PlanetMath or Wikipedia articles. Authors should be able to add them where they are missing.

Discussion posts should be strongly tied to the topic being discussed, and classified into categories like question, answer, explanation, etc.

An enticing page for visitors and potential collaborators should give an impression of the extent and structure of the project (e. g. its size and its specialization into diverse fields of mathematics). For the developer, there should be tools for browsing and querying the knowledge. Not only should it be possible to query knowledge items by their annotations, but important query results must also be available as dynamically generated lists. Examples for queries are:

1. “Which lemmas about composite regions need to be proved?”
2. “What lemmas are difficult to prove?”
   (a) . . . in the sense that many people have already attempted them, but given up
   (b) . . . in the sense that many people have asked questions in the related discussion
3. “Are there textual resources I can read in order to understand the Jordan Curve Theorem?”
4. “What other lemmas could help me to prove this one?” (e. g. because they prove a related statement)

A volunteer who is willing to work out and contribute a computerized proof for a lemma should be able to download a self-contained computerized representation of this lemma and everything it depends on. Different notions of “dependency” can be supported, the most straightforward being that a lemma depends on the declarations and definitions of all symbols it uses and on the transitive closure of all symbols used by the latter. Related lemmas could be downloaded and assumed as axioms, under the assumption that those will be proved later, perhaps by other collaborators. Finally, assuming that the Flyspeck book [10] is written in a linear order, *all* definitions and lemmas before the current one in the narrative order could be used.

During the formalization of the knowledge, we anticipate that the definitions will undergo refactoring in order to facilitate the actual development of the proofs. (Historically, this has been the case with many large computerized proofs, cf. [5].) Refactoring support by the wiki would thus be advantageous. In fact, as definitions rely so heavily on each other, and the lemma statements rely on the definitions, Hales needs to oversee the computerization of the definitions so that the mathematical constants are correct[6]. This could be done by allowing him and other experienced mathematicians to *rate* the contributions of the collaborators.

3 Case Studies and Evaluation

So far, the Flyspeck project has four core members who collaborate via GoogleCode [11]. While the services offered by GoogleCode (a Subversion repository, a mailing list, and others) were found to be sufficient for the core development team, we were not satisfied with the wiki integrated into the GoogleCode web interface. Lacking support for mathematical formulae, it would not even allow for presenting the theorems and lemmas to be computerized in a human-readable fashion. This is important, as we suspect people would prefer to look at traditional mathematics text than proof assistant scripts when browsing. Furthermore, GoogleCode offers very little *structuring* support, which we believe will be essential for browsing and querying Flyspeck's large knowledge collection.

In the following sections, we evaluate two semantic wiki prototypes for their applicability to Flyspeck with regard to their support for annotations, browsing, and querying, as specified in section 2.2. One is based on Semantic MediaWiki, the other one on our own semantic wiki SWiM. For the case study, we took a simplified view of Flyspeck, using only the TEX sources of the Flyspeck book [10] and a Twelf [32] computerization of the definitions and lemmas of the chapter dealing with the foundations of trigonometry. The goal was to present the trigonometry chapter in a compelling way that we believed would scale 2-3 orders of magnitude.

Both systems are semantic wikis, where one resource (e. g. one mathematical theorem) is represented by one wiki page and relations between resources by links between pages. Both pages and links can be typed with terms from ontologies [30], which are either preloaded into the wiki or modeled ad hoc [17]. This is the prevalent approach of adding semantics to wikis, although other ways have been investigated [37]. Note that we have developed an ontology for mathematical knowledge (see sec. 3.2), but as this only focuses on the most essential structures, keeping it extensible in the wiki may be beneficial. Semantic wikis offer enhanced navigation capabilities. For example, they can usually display a summary of all typed links, grouped by type, for each page. They support searching for pages by type or by a page being source or target of a typed link[7]. Such queries can either be executed interactively or automated as *inline* queries embedded into

[6] For example, one can represent a vector as a function from the integers to the reals, or as a tuple of reals. The operations of vector spaces will depend on this representation, etc.

[7] Both explicit and inferred links (RDF triples) can be considered [17]

the content of a page [17]. Both systems we consider support this basic set of semantic wiki features.

3.1 Semantic MediaWiki 1.0

Semantic MediaWiki [17] is a semantic extension to MediaWiki, the system driving Wikipedia. Plain MediaWiki supports mathematical formulae written in LaTeX and allows for categorizing pages. Semantic MediaWiki interprets category membership as an instance-of relationship and supports the creation and editing of typed links (called properties). External ontologies can be referenced from the wiki, but at most sites powered by Semantic MediaWiki, site-specific ontologies are developed in an ad hoc manner [34].

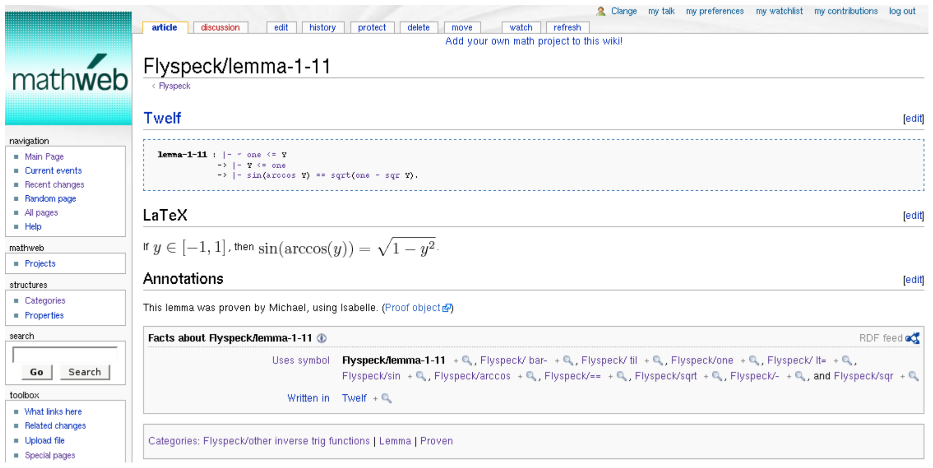
Prototype In Semantic MediaWiki, we imported the Twelf master source of Flyspeck via a custom upload page. The Twelf file was first enhanced by special comment lines marking the beginning and end of a declaration with information about topical categorization. The Twelf upload page handler breaks an uploaded file down into declarations and creates two wiki pages for each Twelf declaration: one page that just contains the Twelf listing, categorized in the OMDoc document ontology (e. g. *Lemma*; see section 3.2), and one container page that includes the Twelf page via MediaWiki's template inclusion mechanism, but also allows for including a LaTeX representation and leaves space for free-form annotations made by the contributors. Additionally, MediaWiki offers a discussion page for each page of mathematical content. The Twelf pages are overwritten on every import from the master source, whereas existing container pages remain untouched. This allows one to change the computerized version of a Twelf constant in the master source (e. g. if it is incorrectly specified) and re-importing it without losing the semantic markup and comments. During the import of a new symbol x, the upload extension recognizes all previously imported symbols y in the definition of the new symbol and creates links between x, y in the wiki.

The generated annotations can be used for browsing, either via the "fact box" (the summary of all typed links), or by the special "browse" page. For querying, Semantic MediaWiki offers a simple triple search, as well as inline queries. The query language corresponds to the small description logic $\mathcal{EL}^{++}$ [17], which, for example, does not support unrestricted negation. A query for unproven lemmas about a certain topic could only be performed if the "unprovenness" were explicitly annotated. The following queries additionally ask for lemmas available in a Twelf formalization:

```
<ask>[[Category:Unproven]] [[Category:Lemma]]
     [[Category:Trigonometry]] [[written in::Twelf]]</ask>
```

Exporting computerized representations of knowledge items is not yet supported conveniently. The Twelf listings can be viewed on their own pages, but due to the auto-generated symbol links in the source code, these are not suitable

[8] See http://mathweb.org/wiki/Flyspeck



Add your own math project to this wiki!

Flyspeck/lemma-1-11

< Flyspeck

Twelf

```
lemma-1-11 : |- ~ one <= Y
          -> |- Y <= one
          -> |- sin(arccos Y) == sqrt(one - sqr Y).
```

LaTeX

If $y \in [-1, 1]$, then $\sin(\arccos(y)) = \sqrt{1 - y^2}$.

Annotations

This lemma was proven by Michael, using Isabelle. (Proof object)

Facts about Flyspeck/lemma-1-11

Uses symbol	Flyspeck/lemma-1-11 +, Flyspeck/ bar- +, Flyspeck/ til +, Flyspeck/one +, Flyspeck/ lt= +, Flyspeck/sin +, Flyspeck/arccos +, Flyspeck/== +, Flyspeck/sqrt +, Flyspeck/- +, and Flyspeck/sqr +
Written in	Twelf +

Categories: Flyspeck/other inverse trig functions | Lemma | Proven

Fig. 3. A Flyspeck lemma in Semantic MediaWiki[8]

for download. One would either have to implement a special Twelf download page that cleans these sources again, or one would have to implement the symbol linking as an extension of the rendering process.

Evaluation We found the ad hoc ontology development useful while prototyping the annotations that might be required for Flyspeck, e. g. project-related metadata like the information whether a lemma has already been proven, or categorization by topic. Semantic MediaWiki did not meet the requirements in places where ontologies already existed. For example, in structures of mathematical documents, it was possible to reference *vocabulary* from the OMDoc document ontology (see below), but not to apply further inference rules given there to items of mathematical knowledge. This is because Semantic MediaWiki does not support a full *import* of external ontologies. Most annotations were modeled by categorization, i. e. instantiation of classes—certainly not the most formal way of structuring knowledge in view of many classes just corresponding to narrative sections of the book, but the one that is supported best by Semantic MediaWiki. The inline queries were intuitive to write but not as powerful as required. Complex reasoning tasks like inference of dependencies are not possible in Semantic MediaWiki; in the restricted domain-specific setting of Flyspeck one could realize them by hard-coded extension functions. Semantic MediaWiki does not understand the semantics of mathematical formulae, as the LaTeX formulae cannot be annotated. The Twelf listings could be annotated, but at the cost of making them harder to download.

3.2 SWiM 0.2

SWiM is a semantic wiki targeted at mathematical knowledge management. Based on the general-purpose semantic wiki IkeWiki [17], it adds support for browsing, editing, rendering, importing and exporting mathematical documents written in OMDoc. The semantics of mathematical knowledge is mainly captured in the OMDoc markup, and more explicitly in a *document ontology*; whenever a wiki page containing OMDoc fragments is saved, its type and its (typed) relations to other items of mathematical knowledge in the wiki are extracted from the OMDoc XML markup and explicitly represented as RDF triples using terms of the OMDoc document ontology [18]. This ontology models those aspects of the three layers of mathematical knowledge supported by OMDoc to the extent supported by the expressivity of OWL-DL [25], including a limited inference of dependencies. Modeling all modules of the OMDoc specification in this ontology is not totally complete, though most mathematical statements as well as key aspects of theories have been implemented. Relevant classes for Flyspeck would be *Lemma*/*Theorem*/*Corollary*/... (all being subclasses of *Assertion*), *Proof*, *Symbol* (a symbol declaration), *Definition*, and the properties *Proof–proves–Assertion* and *Symbol–hasDefinition–Definition*.

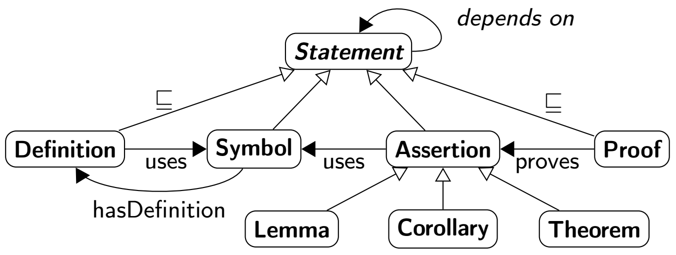


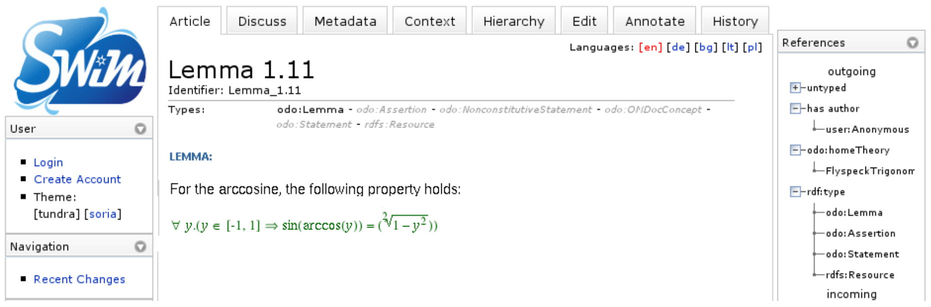
Fig. 4. A relevant subset of the OMDoc document ontology

In the current version 0.2 of SWiM, the browsing of mathematical documents is powered by the document ontology; whenever RDF triples having the current page as subject or object are available[9] the IkeWiki user interface can display them either as navigation links (see figure 5) or in a graph view. Documents are presented as XHTML+MathML, with mathematical symbols linked to their declarations.

Prototype We manually converted part of the trigonometry lemmas to OMDoc for SWiM. Additionally, we can auto-generate OMDoc documents from the Twelf source with a converter and import them into SWiM using the built-in import functionality.

As every SWiM page has an associated discussion page and discussion posts are semantically represented using the SIOC ontology [35], one can support the coordination of the project by queries like query 2b from section 2.2. Work on determining a relevant subset of OMDoc and its document ontology for discussions

[9] In a mathematical document such as those we consider, most of these triples use from the OMDoc document ontology.



Languages: [en] [de] [bg] [lt] [pl]

Lemma 1.11

Identifier: Lemma_1.11

Types: odo:Lemma - *odo:Assertion* - *odo:NonconstitutiveStatement* - *odo:OMDocConcept* - *odo:Statement* - *rdfs:Resource*

LEMMA:

For the arccosine, the following property holds:

$\forall\, y.(y \in [\text{-}1, 1] \Rightarrow \sin(\arccos(y)) = (\sqrt[2]{1-y^2}))$

Fig. 5. A Flyspeck lemma in SWiM

is currently in progress. Pages and non-OMDoc links can be annotated with types from ontologies loaded into the wiki[10].

Another powerful feature of SWiM is that authors can embed inline SPARQL queries into wiki pages. Query 1 can be posed without explicitly annotating “unprovenness”, making use of negation as failure [33]:

```
SELECT ?l WHERE { ?l rdf:type odo:Lemma .
                  ?l swrc:isAbout <Composite_Regions> .
                  OPTIONAL { ?p rdf:type odo:Proof .
                             ?p odo:proves ?l . }
                  FILTER ( ! bound(?p) ) }
```

As OMDoc supports all degrees of formalizing mathematical knowledge, computerized data can be downloaded in their OMDoc representation using SWiM’s export feature and then be converted to Twelf by client-side software [14, chap. 25.2].

Evaluation Annotating mathematical structures with SWiM is easy if the built-in OMDoc editor is used. Other annotations required for Flyspeck, such as categorizations or information about the progress of the project, can be made, but not in an ad hoc way, which we would have found useful in the prototyping phase. Instead, one would have to import an existing ontology into the wiki, or create it using the built-in ontology editor, and then one would be able to annotate documents using terms from that ontology.

Browsing is well supported, with incoming and outgoing navigation links being displayed. Additionally, the neighborhood of the current resource in the RDF graph can be browsed visually.

Queries are powerful, but not always short and intuitive (see above). Alternatively, one could enhance the ontology and make use of the integrated Pellet OWL-DL reasoner (see [17]), which supports a more powerful logic than Semantic MediaWiki, and get the same result with a simple query for instances

[10] Types of OMDoc links are automatically extracted from the markup; see above.

of a specially defined class. For unproven lemmas, the following axiom would suffice:

$$\text{LemmaWithoutProof} \equiv \text{Lemma} \sqcap \neg(\exists \text{proves}^{-1}.\text{Proof})$$

However, it remains to be evaluated how well the wiki scales with DL reasoning enabled. First experiments with Pellet let the system considerably slow down (an experience also made by the IkeWiki author [17]), so alternatives will have to be investigated as well.

System	Semantic MediaWiki	SWiM
Ontology availability	none built in	sufficient (OMDoc)
Ontology editing/ extensibility	easy, ad hoc in place	easy, but only via dedicated user interface
Page annotation	easy but not sufficiently expressive	easy and expressive
Inline queries	easy to write but not sufficiently powerful	harder to write but more powerful
Browsing	intuitive	intuitive, optional graph browser
Reasoning	not sufficient	powerful but slow
Semantics/annotation of formulæ	not supported	very powerful but harder to author
Annotation of computerized content	not directly supported by our extension	powerful (OMDoc markup)

Fig. 6. Summary of the evaluation of the features

4 Related Work

Outside of wikis, the combination of computerized proofs and human-readable text has been investigated in Isar [38], an alternative literate programming language for Isabelle, and in Mizar [26], whose language of Mizar is close to mathematical vernacular. In contrast to Isar, there is a large web-based library of Mizar proofs. It is browsable and searchable on the web but managed in a centralized and hierarchical way, which is not comparable to wiki collaboration.

Informal mathematical knowledge is currently managed in comprehensive encyclopediæ like the mathematical sections of Wikipedia[11] or in PlanetMath[11], which focuses on mathematics and is powered by a highly customized wiki-like system. The pages in these systems are categorized and searchable in full-text, with additional metadata records in PlanetMath. Neither of the systems is a *semantic* wiki, and for lacking typed links they fail to answer queries essential for

[11] See `http://www.wikipedia.org` or `http://www.planetmath.org`, respectively, and [19] for a more comprehensive evaluation.

Flyspeck, such as query 1 from section 2.2, and they do not link mathematical symbols to their declarations; instead, the author has to provide links he considers relevant in the text surrounding the formula.

Recently, there is a growing interest in integrating proof assistants with wikis. *Logiweb* is not a wiki but a distributed system for publishing machine checked mathematics in high-quality PDF that shares part of the key wiki principles [7]. Anybody can contribute to a Logiweb site and edit new pages in a simple text syntax. On the other hand, Logiweb does not offer other essential features. For example, browsing by traversing links is supported neither in the editor nor in the generated PDF, and a built-in search or query facility is not offered. Logiweb does not allow for *exchanging* knowledge as required for Flyspeck: Documents can only be exported in presentational formats like PDF or TEX, but their semantic structures cannot be exported in mathematical markup or theorem proving languages. The way Logiweb checks proofs is not compatible with other theorem provers, as all calculi and proof tactics need to be defined in the Logiweb system itself. *ProofWiki* is an integration of the ProofWeb Coq frontend into MediaWiki [4]. Coq's export tools are used to generate browsable HTML or LATEX with linked symbols from the proof scripts. Generating index pages, such as lists of all definitions or all theorems, is planned, but not yet in a way that could be customized by users. So far, there is just text search, and dependencies among knowledge items are only computed for exporting proofs but not used for browsing inside the system. Pages can either be formal proof scripts (with restricted possibilities to include informal comments) or informal wiki pages. Semi-formal documents or stepwise formalizing of knowledge are not supported. Importing and exporting Coq proof scripts to and from the wiki is possible. While the authors provide instructions on how to integrate other theorem provers, doing so would be a lot of work, as there is no abstraction layer or metalanguage for exchanging or converting data. Both Logiweb and ProofWiki are "semantic" in the sense that the integrated proof checker utilizes the mathematical knowledge in the wiki pages. But the semantics is not utilized for anything else, such as facilitating browsing or editing, or connecting to semantic web services. Developing and verifying formal proofs in the wiki is not yet the focus of Flyspeck in this early stage, but it may be required later.

5 Conclusion and Further Work

Our preliminary experiments lead us to believe that, due to its rich semantic web and OMDoc infrastructure, future work toward supporting Flyspeck should continue in the SWiM infrastructure. For the text-based page format of MediaWiki, features that rely on structures like the linking of symbols could only be realized in an ad hoc way using, say, regular expressions. Relying on the XML infrastructure of OMDoc, these features are either already available or easier to develop. However, rapidly *prototyping* our first ideas about the wiki support required for Flyspeck was easier in Semantic MediaWiki due to its ability to design ad hoc ontologies and its implementation in the interpreted language PHP.

Importing For this case study, we created OMDoc from Twelf. OMDoc also offers support for the alternative workflow of stepwise formalization as well. One could either start by converting the Flyspeck book from LaTeX to HTML with MathML formulæ and formalize the presentation markup into content markup step by step, or one could start the formalization on the TeX side. There, one would formalize the book to sTeX, a content-oriented TeX notation for OMDoc, which can then be converted to OMDoc [13]. Either way involves a TeX-to-XML transformation, which has been tested in large scale in our group [2].

Annotating The case study showed that the editing of ontologies in SWiM should become more flexible. While a fixed OMDoc document ontology can be preloaded, it should be possible to add other annotations ad hoc. We have not focused on document *editing* in detail here, but additional editing services relying on the document ontology are planned for SWiM 0.3 [20, 21]. Finally, using the module system of OMDoc and refactoring the knowledge into more smaller theories could help to simplify the structure of Flyspeck for browsing and to explicate the dependencies between components of the proof.

Browsing In the Semantic MediaWiki prototype we realized that the narrative structure of the book is not adequately represented by a simple hierarchy of categories. OMDoc has more powerful ways of putting content into narrative structures [15]. We are going to cover them with the document ontology and utilize them for browsing.

Querying Proof search will be greatly simplified if the semantic-aware search engine MathWebSearch [16] is used. It applies substitution tree indexing to mathematical formulae. That means, for example, that a query for $\int f(x\,?\,z)dx$ would also find $\int f(y+z)dy$. Equivalence up to α-renaming of bound variables is obviously essential for a serious query language.

Different Theorem Provers If several parts of the proof are done in different theorem provers, highly non-trivial and mostly novel translations become necessary to provide one single proof object. Here OMDoc could be used as an exchange format between theorem prover languages, and formal translations could be specified in OMDoc itself. While this line of research is interesting, it is difficult for us to foresee what kinds of translations, if any, will be needed.

Download Dependencies, which we need for bundling download packages, can partly be inferred by a DL reasoner using the document ontology, but for a complete support of OMDoc's notion of dependency, an OMDoc-specific calculus will have to be applied, which is currently in development.

Upload We have not implemented uploading a proof directly to the wiki to have it checked. This is easy in theory as we simply need to hook up the theorem prover, but requires some effort to get the theorem prover to run on the wiki server. This should be done soon, as it will relieve the maintainers.

Acknowledgments We would like to thank Stefan Decker, Michael Kohlhase, and Immanuel Normann for their feedback particularly during the case studies.

References

1. American Mathematical Society. 2000 mathematics subject classification. `http://www.ams.org/msc/`, 2000.
2. `arXMLiv`: Translating the arχiv to xml+mathml, 2007. `http://kwarc.info/projects/arXMLiv/`.
3. Y. Bertot and P. Castéran. *Interactive theorem proving and program development: Coq'Art: the Calculus of Inductive Constructions.* Texts in theoretical computer science. Springer, 2004.
4. P. Corbineau and C. Kaliszyk. Cooperative repositories for formal proofs. In Kauers et al. [12].
5. G. Gonthier. A computer-checked proof of the four colour theorem. Unpublished manuscript, 2005.
6. T. Groza, S. Handschuh, K. Möller, and S. Decker. SALT – Semantically Annotated LaTeX for scientific publications. In E. Franconi, M. Kifer, and W. May, editors, *ESWC*, volume 4519 of *Lecture Notes in Computer Science*. Springer, 2007.
7. K. Grue. The layers of Logiweb. In Kauers et al. [12].
8. T. Hales. A proof of the Kepler conjecture. *Annals of Mathematics*, 162:1065–1185, 2005.
9. T. Hales. The Kepler conjecture. *Discrete and Computational Geometry*, 36(1):1–269, 2006.
10. T. Hales. Flyspeck : A Blueprint of the Formal Proof of the Kepler Conjecture. Unpublished manuscript, 2008.
11. T. Hales and S. McLaughlin. The Flyspeck Project. `http://code.google.com/p/flyspeck`, 2007.
12. M. Kauers, M. Kerber, R. Miner, and W. Windsteiger, editors. *MKM/Calculemus 2007*, number 4573 in LNAI. Springer, 2007.
13. M. Kohlhase. sTeX: A LaTeX-based workflow for OMDoc. In OMDoc – *An open markup format for mathematical documents [Version 1.2]* [14], chapter 26.15.
14. M. Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2].* Number 4180 in LNAI. Springer, 2006.
15. M. Kohlhase, C. Müller, and N. Müller. Documents with flexible notation contexts as interfaces to mathematical knowledge. In P. Libbrecht, editor, *Mathematical User Interfaces Workshop*, 2007.
16. M. Kohlhase and I. Şucan. A search engine for mathematical formulae. In T. Ida, J. Calmet, and D. Wang, editors, *Artificial Intelligence and Symbolic Computation, AISC*, number 4120 in LNAI. Springer, 2006.
17. M. Krötzsch, S. Schaffert, and D. Vrandečić. Reasoning in semantic wikis. In G. Antoniou, U. Aßmann, C. Baroglio, S. Decker, N. Henze, P.-L. Pătrânjan, and R. Tolksdorf, editors, *3rd Reasoning Web Summer School*, volume 4636 of *LNCS*. Springer, 2007.

18. C. Lange. The OMDoc document ontology. `http://kwarc.info/projects/docOnto/omdoc.html`, 2007.
19. C. Lange. SWiM – a semantic wiki for mathematical knowledge management. Technical Report 5, Jacobs University Bremen, 2007.
20. C. Lange. SWiM development roadmap. `https://trac.kwarc.info/swim/roadmap/`, 2007.
21. C. Lange. Towards scientific collaboration in a semantic wiki. In A. Hotho and B. Hoser, editors, *Bridging the Gap between Semantic Web and Web 2.0*, 2007.
22. C. Lange. SWiM – a semantic wiki for mathematical knowledge management. In S. Bechhofer, M. Hauswirth, J. Hoffmann, and M. Koubarakis, editors, *ESWC*, volume 5021 of *Lecture Notes in Computer Science*, pages 832–837. Springer, 2008.
23. B. Leuf and W. Cunningham. *The Wiki Way: Collaboration and Sharing on the Internet*. Addison-Wesley Professional, 2001.
24. Mathematical Markup Language (MathML) version 3.0. W3C working draft, World Wide Web Consortium, 2007. `http://www.w3.org/TR/MathML3`.
25. D. L. McGuinness and F. van Harmelen. OWL web ontology language overview. W3C recommendation, W3C, 2004.
26. Mizar mathematical library. Web Page at `http://mizar.org/library/`.
27. T. Nipkow, G. Bauer, and P. Schultz. Flyspeck I: Tame Graphs. In U. Furbach and N. Shankar, editors, *International Joint Conference on Automated Reasoning*, volume 4130 of *LNCS*. Springer, 2006.
28. S. Obua. Proving bounds for real linear programs in isabelle/HOL. In J. Hurd and T. F. Melham, editors, *Theorem Proving in Higher Order Logics*, volume 3603 of *LNCS*. Springer, 2005.
29. The Open Math standard, version 2.0. Technical report, The Open Math Society, 2004. `http://www.openmath.org/standard/om20`.
30. E. Oren, R. Delbru, K. Möller, M. Völkel, and S. Handschuh. Annotation and navigation in semantic wikis. In Völkel et al. [37].
31. L. Paulson. *Isabelle: A Generic Theorem Prover*, volume 828 of *LNCS*. Springer, 1994.
32. F. Pfenning and C. Schürmann. System description: Twelf : A meta-logical framework for deductive systems. In H. Ganzinger, editor, *16th International Conference on Automated Deduction (CADE)*, volume 1632 of *LNAI*. Springer, 1999.
33. E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF. W3C Recommendation, World Wide Web Consortium, 2008. `http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/`.
34. Sites using Semantic MediaWiki. `http://www.semantic-mediawiki.org/w/index.php?title=Sites_using_Semanti%c_MediaWiki&oldid=781`, 2008.
35. SIOC – Semantically-Interlinked Online Communities, 2007. `http://sioc-project.org/`.
36. D. Tapscott and A. D. Williams. *Wikinomics – How Mass Collaboration Changes Everything*. Portfolio, 2006.
37. M. Völkel, S. Schaffert, and S. Decker, editors. *1st Workshop on Semantic Wikis*, volume 206 of *CEUR Workshop Proceedings*, 2006.
38. M. Wenzel. Isar — a generic interpretative approach to readable formal proof documents. In Y. Bertot, G. Dowek, A. Hirschowitz, C. Paulin, and L. Théry, editors, *Theorem Proving in Higher Order Logics: TPHOLs'99*, volume 1690 of *LNCS*, pages 167–184. Springer, 1999.
39. R. Zumkeller. Formal global optimisation with taylor models. In U. Furbach and N. Shankar, editors, *International Joint Conference on Automated Reasoning*, volume 4130 of *LNCS*. Springer, 2006.