Mathematical Semantic Markup in a Wiki: the Roles of Symbols and Notations

Christoph Lange

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

Abstract. We present semantic markup as a way to exploit the semantics of mathematics in a wiki. Semantic markup makes mathematical knowledge machine-processable and thus allows for a multitude of useful applications. But as it is hard to read and write for humans, an editor needs to understand its inherent semantics and allow for a human-readable presentation. The semantic wiki SWiM offers this support for the OpenMath markup language. Using OpenMath as an example, we present a way of integrating a semantic markup language into a semantic wiki using a document ontology and extracting RDF triples from XML markup. As a benefit gained from making semantics explicit, we show how SWiM supports the collaborative editing of definitions of mathematical symbols and their visual appearance.

1 Making Mathematical Wikis More Semantic

What does a wiki need in order to support mathematics in a semantic way? First, there needs to be a way to edit mathematical formulæ. Many wikis offer a LaTeX-like syntax for that, and they have been used to build large mathematical knowledge collections, such as the mathematical sections of Wikipedia [30] or the mathematics-only encyclopædia PlanetMath [16]. But LaTeX, which is mostly presentation-oriented, despite certain macros like `\frac{num}{denom}` or `\binom{n}{k}`, is not sufficient to capture the *semantics* of mathematics. One could write $\mathcal{O}(n^2 + n)$, which could mean "$\mathcal{O}$ times $n^2 + n$" (with redundant brackets), or "$\mathcal{O}$ (being a function) applied to $n^2 + n$", or the set of all integer functions not growing faster than $n^2 + n$, and just by common notational convention we know that the latter is most likely to hold.

For being able to express the semantics of $\mathcal{O}(n^2+n)$, we need to make explicit that the Landau symbol $\mathcal{O}$ is a set construction operator and n is a variable. The meaning of $\mathcal{O}$ has to be defined in a vocabulary shared among mathematical applications such as our wiki. This is analogous to RDF, where a vocabulary—also called ontology—has to be defined before one can use it to create machine-processable and exchangeable RDF statements. In a mathematical context, these vocabularies are called *content dictionaries* (CDs). As with ontology languages, one can usually do more than just listing symbols and their descriptions in a CD: defining symbols formally in terms of other symbols, declaring their types formally, and specifying their visual appearance. Thus, CDs themselves are special mathematical documents that could again be made available in a mathematical wiki. Then it would be possible to create an unambiguous link from any occurrence of $\mathcal{O}$ in a formula to its definition in the wiki, and knowledge from the wiki could be shared with any other mathematical application supporting this CD. As a practical solution, we present the OpenMath CD language in sect. 2 and its integration into the semantic wiki SWiM in sect. 4.

2 Semantic Markup for Mathematics with OpenMath

Semantic markup languages for mathematics address the problems introduced in sect. 1 by offering an appropriate expressivity and semantics for defining symbols and other structures of mathematical knowledge. This is a common approach to knowledge representation not only in mathematics, but generally in science[1].

OpenMath [7] is a markup language for expressing the logical structure of mathematical formulæ. It provides its own sublanguage for defining CDs—collections of symbol definitions with formal and informal semantics. One symbol definition consists of a mandatory symbol name and a normative textual description of the symbol, as well as other metadata[2]. Formal mathematical properties (FMPs) of the symbol, such as the definition of the sine function, or the commutativity axiom that holds for the multiplication operator, can be added, written in OpenMath and possibly using other symbols (see fig. 1). Type signatures (such as $\sin\colon \mathbb{R} \to \mathbb{R}$) and human-readable notations (see sect. 3) of symbols are defined separately from the CD in a similar fashion.

As semantic markup makes mathematical formulæ machine-understandable, it has leveraged many applications. For OpenMath, it started with data exchange between computer algebra systems, then automated theorem provers, and more recently dynamic geometry systems. OpenMath is also used in multilingual publishing, adaptive learning applications, and web search [10]. OpenMath CDs foster exchange by their modularity. Usually, a CD contains a set of related symbols, e. g. basic operations on matrices (CD `linalg1`) or eigenvalues and related concepts (CD `linalg4`), and a CD group contains a set of related CDs, e. g. all standard CDs about linear algebra (CD group `linalg`). In this setting, agents exchanging mathematical knowledge need not agree upon one large, monolithic mathematical ontology, but can flexibly refer to a specific set of CDs or CD groups they understand[3].

---

[1] Consider e. g. the chemical markup language CML [23]

[2] OpenMath 2 uses an idiosyncratic schema for metadata, but Dublin Core is likely to be adopted for OpenMath 3.

[3] A communication protocol for such agreements is specified in [7, sect. 5.3].

```
<CDDefinition>
<Name>sin</Name>
<Description>The sine function on real numbers</Description>
<CMP>The sine function is defined in terms of the exponential function as
 $\sin x = \frac{1}{2}(e^{ix} - e^{-ix})$.</CMP>
<FMP>
  <OMOBJ xmlns="http://www.openmath.org/OpenMath"
   version="2.0" cdbase="http://www.openmath.org/cd">
    <OMA><OMS cd="relation1" name="eq"/>
      <OMA><OMS name="sin" cd="transc1"/>
        <OMV name="x"/></OMA>
      <OMA><OMS name="divide" cd="arith1"/>
        <OMA><OMS name="minus" cd="arith1"/>
          <OMA><OMS name="exp" cd="transc1"/>
            <OMA><OMS name="times" cd="arith1"/>
              <OMS name="i" cd="nums1"/>
              <OMV name="x"/></OMA></OMA>
          <OMA><OMS name="exp" cd="transc1"/>
            <OMA><OMS name="times" cd="arith1"/>
              <OMA><OMS name="unary_minus" cd="arith1"/>
                <OMS name="i" cd="nums1"/></OMA>
              <OMV name="x"/></OMA></OMA></OMA>
        <OMA><OMS name="times" cd="arith1"/>
          <OMI>2</OMI>
          <OMS name="i" cd="nums1"/></OMA></OMA></OMA></OMOBJ></FMP>
</CDDefinition>
```

Fig. 1. A definition of the sine function in OpenMath. FMP stands for "formal mathematical property", whereas the "C" in CMP stands for "comment". OMA is an application of a symbol (OMS) to some arguments, OMV denotes a variable, and OMI denotes an integer. For any symbol, its CD (resolved relatively to a base URI) and name have to be given. The CD used here are standard ones officially approved by the OpenMath Society.

3 Authoring, Navigating and Presenting Semantic Markup

As semantic markup is obviously hard to read and write for humans (see fig. 1), proper authoring software is desirable for writing it. For reading, it should be transformed to presentation markup. LaTeX or presentational MathML [2] are suitable output formats. With "authoring", we refer both to authoring instances, i. e. formulæ that can use symbols from any CD, but also to authoring ontologies, i. e. CDs. In fact, as with editors for semantic web ontologies, a sophisticated editor should cover both levels, as existing vocabulary (here: mathematical symbols) is not just used to create instance data, but also to define new vocabulary in terms of old one.

Semantic markup is commonly presented by defining the *notation* of every symbol as a mapping from a single semantic symbol—or a pattern matching a set of semantic markup structures in which the symbol can occur—to fragments of presentation markup, where arguments to symbols are presented by recursive application of the rules [14, 19]. One such pattern-based language for notation definitions has been proposed as a part of the CD language of the MathML 3 standard (see sect. 5) and, within the current process of aligning both languages, is likely to be adopted for OpenMath 3 as well. For XML languages, semantics-to-presentation-mappings are commonly given in XSLT [12], either directly, or generated from a more concise representation [14, 19], but there are also non-XSLT implementations (see sect. 5).

In the course of opening up new mathematical areas, definitions of new symbols and their axiomatization are not fixed initially but subject to continuous evolution and refactoring—a workflow that a semantic wiki should support. In this paper, we assume that the semantics of symbols is fixed, but then it is still the *notation* of the symbol that can evolve. On the one hand there is evolution in the course of time. Before the 16^{th} century, a prefixed letter R or r (= radix) had mainly been used for square roots, but then the more abstract symbol $\sqrt{\ }$ was developed [8]. On the other hand the notation of a symbol depends on the *context* the symbol is used in. The context can be determined by the language of the author or the reader, by previous knowledge, by the area of application, and other criteria [14, 27]. For example, a binomial coefficient is written as $\binom{n}{k}$ in German or English, but as $\mathcal{C}_n^k$ in French. A mathematician uses the symbol i for the imaginary unit, whereas an electrical engineer would write j. In a strict style, one would express asymptotic growth as $f \in \mathcal{O}(g)$, but the sloppy style of $f = \mathcal{O}(g)$ is far more common. For applications this means that reusing existing mathematical content in a new context requires adapting the notation [14].

The fact that there is no single, definitive notation for a mathematical symbol leads to the requirement that an integrated editor for mathematical documents and CDs should instantly reflect changes of notation definitions in the places where they are used for presentation. That means that whenever the notation of a symbol σ has changed, all the presentation markup generated from formulæ containing σ has to be invalidated and re-rendered upon the next request. In a single-author environment this frees the author from recompiling all the affected presentation markup and gives instant visual feedback about whether the new notation works, and in a collaborative environment it relieves *other* authors from worrying whether they are looking at an up-to-date presentation of a document.

Moreover, an authoring tool should make the semantic relations between definitions of symbols, their notations, and their type signatures, as well as the relations between instances (here: symbols used in formulæ) and classes explicit in the user interface, as this facilitates orientation both for authors and for readers.

4 OpenMath CDs in the Semantic Wiki SWiM

SWiM aims at satisfying the above requirements for authoring, navigating, and presenting mathematical knowledge in a semantic wiki [18]. It is based on the general-purpose semantic wiki IkeWiki [26] and enhances it by support for OpenMath and subsets of MathML and the more comprehensive OMDoc [13]. In this paper, we focus on OpenMath.

SWiM follows the approach of representing one subject of interest by one wiki page and modeling it as one resource in an RDF graph; this shall henceforth be called "page-level annotation". This way of knowledge representation is the most common in semantic wikis, but nowhere near the only one [24, 6]. Our experience with developing SWiM and working in other semantic wikis following the same approach, such as Semantic MediaWiki [29], shows that this works best if pages are small. Otherwise, too many subjects of interest would be described in subsections of pages and thus could not be represented in RDF. But small pages are advisable in a wiki anyway, as they reduce the potential of editing conflicts. With small pages, page-level annotation is at its most impressive in terms of easy-to-use authoring, navigation, and search from a user's point of view, and easy maintenance within the system[4].

An important task in implementing the OpenMath support in SWiM was the choice of the granularity of pages. The OpenMath Society, the body standardizing OpenMath, considers whole CDs as units that are subject to review or change [7, sect. 4.5], and the OpenMath CD language reflects that by only providing "date" and "author" metadata fields on CD level. But quite a lot of CDs contain more than 10 symbol definitions, covering several screen pages in a browser. With SWiM we are not only aiming at supporting the review process of the OpenMath Society but also the more dynamic preceding phase, when, for example, a working group in some company or research institute is developing a new CD that will then be submitted to the OpenMath Society for approval. Thus, both to keep wiki pages small, and to support a more dynamic workflow, we chose to map symbol definitions and even their subelements (formal and informal properties, and examples) to wiki pages. The latter choice is influenced

[4] These aspects were discussed in more detail on the mailing list of the Semantic Wiki Interest Group [28] in two threads named "Modeling 'third party' relations on Semantic Mediawiki page?" and "Creating Triples Anywhere in a Semantic Wiki" in April 2007.

from our experience with OMDoc [13], where axioms and theorems—defining or asserting additional properties of symbols—and examples are modeled as entities separate from the declarations or definitions of the symbols, allowing for greater flexibility.

Now that this design choice has been found reasonable for SWiM, it should not violate the compatibility with other OpenMath applications. SWiM allows for importing and exporting OpenMath files from and to the local file system or the OpenMath Subversion repository. On import, CDs are split into their subparts as mentioned above, every part being stored on its own wiki page. The containment relations are preserved as XIncludes [21], which are resolved on export again. The XIncludes are also resolved when a document is rendered, so that a whole CD can be viewed at once.

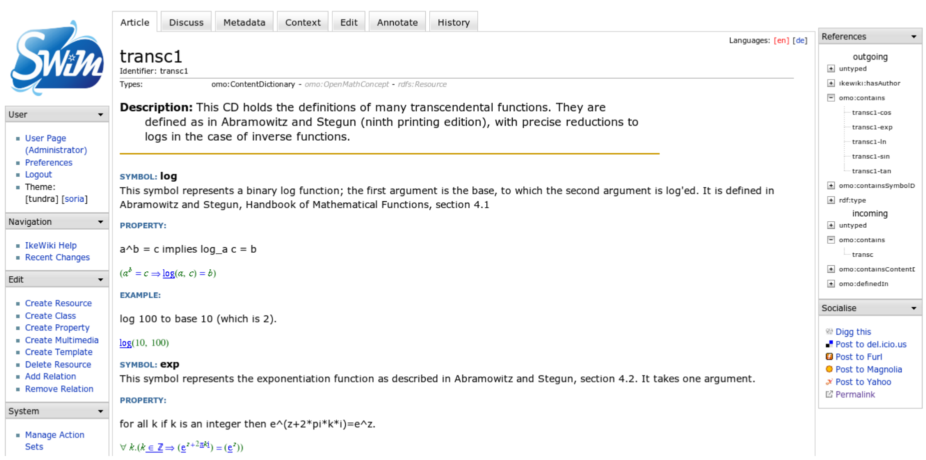

Fig. 2. A content dictionary in SWiM

SWiM supports editing OpenMath and other semantic markup in a semi-WYSIWYG way. Plain text can be edited and styled visually (but styles are lost on export), for OpenMath objects there is a simple linear ASCII syntax, and other XML structures are made accessible as special HTML tables (see fig. 3).

The semantics of the OpenMath CD language is documented in a human-readable specification [7, chap. 4], which is not explicit enough to make OpenMath CDs directly usable by semantic web applications. Instead, an ontology had to be developed to allow for making RDF statements about resources in CDs, and an automated extraction of RDF statements from the OpenMath markup had to be developed. The OpenMath document ontology models classes and properties for all structural entities found in OpenMath's CD groups, CDs, type signatures, and notation definitions in OWL-DL. Properties from common on-

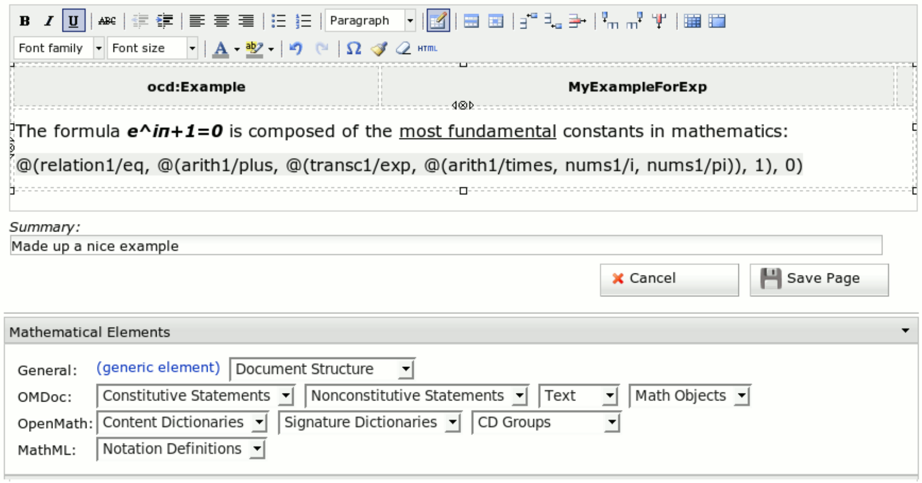

Fig. 3. Editing OpenMath in SWiM

tologies like Dublin Core were reused where appropriate. The inner structures of formulæ (also known as "OpenMath objects") were not modeled, for two reasons: First, formulæ are not accessible as entities of their own in SWiM but as children of structural entities like FMPs, which are represented in the document ontology, and secondly, RDF combined with logics used for semantic web reasoning is not expressive enough for capturing the full semantics of mathematical formulæ. The latter should instead be left to a theorem prover or computer algebra system working with OpenMath objects. If we had an ontology for expressing the *syntactic* structure of an OpenMath object like $\forall x.x \in \mathbb{R} \Rightarrow e^x > 0$ in RDF, as discussed in [20], we would be able to make the references to the bound variable x explicit (they would point to the same URI), but still we would not be able to express the notion of α-equivalence (meaning that we could have used any other name for x as well) in the first order logic subsets commonly used for reasoning on the semantic web. There is, however, a property that states that an FMP or an example uses a symbol—which is contained in some OpenMath object inside the FMP or example—and points to the definition of that symbol in some CD. This is not only useful for determining dependencies among CDs (What other CDs do I need to load in order to get a self-contained collection?), but also for rendering formulæ according to notation definitions (see sect. 5).

Whenever a wiki page is stored in SWiM, i. e. whenever it is saved in the editor or imported, an RDF representation in terms of the document ontology is extracted from the markup. Consider the following CD:

```
<CD xml:id="sample">
  <CDName>sample</CDName>
  <CDDate>2008-03-05</CDDate>
  <CDStatus>private</CDStatus>
```

```
  <Description>A sample CD</Description>
  <xi:include href="url/of/CDDefinition"/></CD>
```

From this, the following RDF would be extracted (in Turtle syntax [4]), where `omo` is the prefix of the OpenMath document ontology namespace:

```
<#sample>
  rdf:type omo:ContentDictionary ;
  dc:identifier "sample" ;
  dc:date 2008-03-05 ;
  omo:status "private" ;
  dc:description "A sample CD" ;
  omo:containsSymbolDefinition <url/of/cddefinition> .
```

The extracted RDF is stored in the Jena store built into IkeWiki, where the OpenMath document ontology resides as well. IkeWiki uses Jena's builtin RDFS reasoner that implements the RDFS semantics but understands the OWL syntax as well [26]. IkeWiki currently utilizes the RDF graph in order to generate a list of incoming and outgoing links for the current page, grouped by property (shown on the right side in fig. 2), to feed a graphical RDF browser, to preselect properties of pages and links an author would probably want to annotate, and it supports embedding arbitrary inline SPARQL queries [25] into pages.

5 Defining Notations and (Re-)Rendering Formulæ

For defining the notation of a symbol, SWiM employs the pattern-based language proposed for MathML 3 [2, sect. 8.6, "Rendering of Content Elements"]. A notation definition for a symbol consists of a *prototype* that matches a fragment of semantic markup, either matching elements literally or matching subtrees against so-called content metavariables, and a *rendering*, which is a template of presentation markup with so-called element metavariables in those places where the results of rendering the XML trees matched against the correspondent content metavariables are to be inserted. This is, for example, a notation definition for the root operator in the *arith1* CD, specifying the rendering $\sqrt[n]{\mathrm{arg}}$:

```
<notation xml:id="ntn-root">
  <prototype>
    <om:OMA>
      <om:OMS cd="arith1" name="root"/>
      <expr name="arg"/>
      <expr name="n"/></om:OMA></prototype>
  <rendering>
    <m:mroot>
      <render name="arg"/>
      <render name="n"/>
    </m:mroot></rendering></notation>
```

From this, the RDF triple <#ntn-root> omo:rendersSymbol <url/of/arith1/root> . would be extracted.

SWiM employs the Java-based mmlproc rendering library [14] for rendering OpenMath objects to presentational (i. e. non-semantic) MathML, which can be viewed in recent versions of the Firefox or Opera browsers. Whenever a wiki page containing notation definitions is saved or imported, the notation definitions are put into a cache read by mmlproc. To symbols without a notation definition, mmlproc applies a default rendering like root(arg,n).

If a notation definition has been added, deleted, or changed, the affected documents have to be re-rendered. In order to do this properly, SWiM has to

1. identify changes to notation definitions
2. identify documents affected by a change

(1) is done by computing an XML diff between the cached and the newly inserted version of a notation definition. (2) is done by querying the RDF graph for all FMPs and examples using the symbol rendered by the respective notation definition, as shown in fig. 4 and 5. Not only the wiki pages holding these FMPs and examples have to be re-rendered, but also those pages (symbol definitions and CDs) that directly or indirectly *include* these fragments.

```
SELECT DISTINCT ?page WHERE {
  <changed-ntn-def> omo:rendersSymbol ?sym .
  { ?page omo:usesSymbol ?sym } UNION
  { ?exOrFMP omo:usesSymbol ?sym .
    { ?page omo:contains ?exOrFMP } UNION
    { ?page omo:contains ?symDef .
      ?symDef omo:contains ?exOrFMP } } }
```

Fig. 4. SPARQL query determining the effect of changing a notation definition; see fig. 5 for a graphical representation.

6 Related Work

SWiM was originally motivated by deficiencies in **mathematical wikis** like Wikipedia or PlanetMath (see sect. 1). For MediaWiki, a semantic web extension has been developed [29], which aims at being used in the MediaWiki-powered Wikipedia. se(ma)^{2}wi [31] was a Wikipedia-independent experiment with a Semantic MediaWiki fed with OMDoc-formatted mathematical knowledge from the ActiveMath learning environment. While the ActiveMath learning metadata are displayed in the wiki, most of the structural semantics explicitly given in OMDoc is, however, lost during this import: The formulæ are converted to

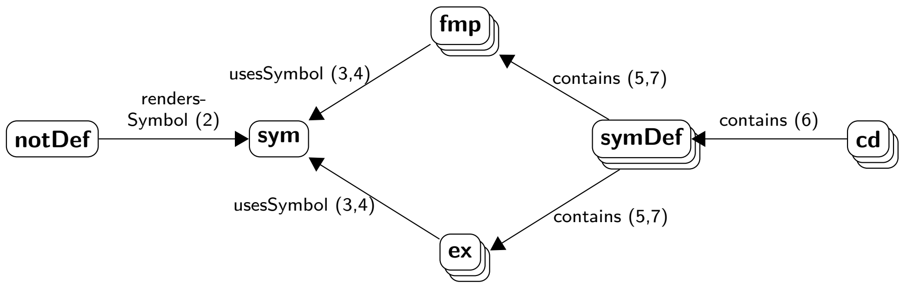

Fig. 5. Finding pages (depicted as stacks of nodes) affected by changes to a notation definition. Numbers refer to lines of listing 4. Note that both *sym* and the *symDef*s are instances of the class *SymbolDefinition*.

presentational-only LaTeX, and the links between wiki pages that represent mathematical statements, for example a link from a theorem to its proof, are not typed and therefore cannot be queried.

In its role as a CD editor, SWiM is comparable to an **ontology editor**. Actually, the IkeWiki base system provides a simple editor for RDFS and OWL ontologies as well. OntoWiki is a more comprehensive, agile collaborative editor for ontologies and instance data that is inspired by wiki principles like ease of use for non-experts and versioning [1]. In OntoWiki, one edits a resource like a database record, namely in a table containing edit boxes for all properties of the resource, whereas a resource in a semantic wiki is represented as a semi-structured document, which the user can enrich with annotations. While an OpenMath CD has a mostly record-like structure, this does not hold for mathematical documents in general; consider e. g. the OpenMath objects inside a CD, or a mathematical lecture written in the more versatile OMDoc language [13].

While SWiM is the first wiki that supports editing **notation definitions**, this has been investigated for text editors before. In the PLATΩ system, the TeXmacs editor has been extended towards semantic markup [3]. The developers focus on notations that use natural language and on parsing text and formulæ the user writes in a presentational style back to a semantic representation. Both features have not yet been investigated in SWiM; here the focus is rather on making the semantic markup editable in a convenient way. As a change to a notation definition in PLATΩ/TeXmacs involves regenerating parser rules, special attention is paid to making this efficient by only regenerating those rules that are affected by a change. Improving this by computing minimal diffs w. r. t. extended equivalence relations for structured document formats (such as ignoring changes to whitespace) and computing and previewing long-range effects of changes is further elaborated in [22].

Formula rendering is a special case of inline query processing. Many semantic wikis support inline queries as a means of automatically generating lists [15]; usually an inline query consists of a predicate $p\colon \text{Page} \to \mathbb{B}$, a specification of the information that is desired for every page satisfying p (e. g. its title), and a formatting style for the result. An OpenMath object in SWiM can be considered a query for the notation definitions of the symbols used in the object, where for every symbol only the most appropriate rendering[5] is included in the result set and the result is "formatted" by rendering the symbols according to the rendering specifications in the result set. In this setting, we can determine whether a change to a page (here: a notation definition) affects the result set of a query (here: a formula) by checking whether the formula contains a particular fixed symbol, which requires linear time w. r. t. the size of the formula. For general queries[6], this is far more complex, as the satisfiability problem for propositional boolean expressions is $\mathcal{NP}$-complete.

The **document ontology** presented here is the first one that has been developed for OpenMath CDs. In previous work, we have introduced a similar ontology for a subset of OMDoc [17]. A different ontology for OpenMath has been developed in the MONET project. It does not model the *document structure* of CDs but can be used to relate OpenMath objects to certain web services operating on them; e. g. one can specify that there is a web service for computing definite integrals, which can operate on any object that applies the *defint* symbol from the *calculus1* CD to certain arguments [9].

7 Conclusion and Outlook

Having motivated that mathematical semantic markup languages can help to make semantic wikis aware of mathematics, we showed how the OpenMath content dictionary language was integrated into the semantic wiki SWiM by choosing an appropriate page granularity, modeling a document ontology, and extracting relevant facts from the markup into RDF. We motivated the need for supporting the maintenance of notation definitions for mathematical symbols and showed how to utilize the information from the RDF graph in order to improve the performance of the system and the usability in terms of navigation and orientation when editing notation definitions.

So far, SWiM assumes that there is at most one notation definition per symbol. The mmlproc renderer supports callbacks to an algorithm that selects the most appropriate out of a set of multiple possible renderings for a symbol. A default implementation considering the static context of a formula (such as the language of the document [section]) is provided with mmlproc [14]. In future, it is planned to provide a user interface inside SWiM that lets the user select his preferred rendering for every symbol.

A visual editor for formulæ will be provided as well. Available editors will be evaluated w. r. t. their extensibility by new symbols and notation definitions. Ideally, the tool palette of a visual formula editor would be supplied dynamically with all known instances of the *SymbolDefinition* class.

---

5 See sect. 7 for dealing with multiple renderings per symbol.

6 In database research, the area of problems touched on here is known as "materialized view maintenance" [11].

Additional reasoning tasks need to be investigated to allow for more powerful queries. For example, the dependency relation between CDs (see sect. 4) and the containment relation between CDs and their subparts (see sect. 5) are transitive, but RDFS cannot express transitivity, and SPARQL cannot compute transitive closures.

Finally, the relationship between the structural semantics of documents and domain knowledge is worth investigating. If we define the Landau symbols $\mathcal{O}$ and Ω in a CD, probably including their type declarations, we have not gained more domain knowledge than that two mathematical concepts $\mathcal{O}$ and Ω exist that map integer functions to sets of integer functions. In the more expressive OMDoc language we could provide a definition of Ω as $\forall f, g. f \in \Omega(g) :\Leftrightarrow g \in \mathcal{O}(f)$ and use that knowledge e. g. to customize presentation: Formulæ using Ω could be rewritten to their equivalents using more the familiar $\mathcal{O}$. Relationships between mathematical concepts are not only given by definitions: Commonly a differentiable function f is defined as a function that has a derivative, but the fact that f also is continuous is only observed afterwards as a theorem. From such definitions or theorems one could extract a domain ontology use it for reasoning. In DL this might look as follows (class names abbreviated) [5]:

$$\text{ContFunc} \sqsubseteq \text{Func} \tag{1}$$
$$\text{DiffableFunc} = \text{Func} \sqcap \exists \text{hasDeriv}.\text{Func} \tag{2}$$
$$\text{DiffableFunc} \sqsubseteq \text{ContFunc} \tag{3}$$

This information could then be utilized e. g. to display a general theorem about continuous functions when the user searches for a theorem about differentiable functions.

Acknowledgments The author would like to thank Michael Kohlhase for inspiration by his vision of a semantic web for mathematics, and James Davenport, representing the OpenMath society, for sharing his insights about CD metadata.

References

1. S. Auer, S. Dietzold, and T. Riechert. Ontowiki – A tool for social, semantic collaboration. In I. F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, editors, *5th International Semantic Web Conference*, volume 4273 of *LNCS*. Springer, 2006.
2. R. Ausbrooks, B. Bos, O. Caprotti, D. Carlisle, G. Chavchanidze, A. Coorg, S. Dalmas, S. Devitt, S. Dooley, M. Hinchcliffe, P. Ion, M. Kohlhase, A. Lazrek, D. Leas, P. Libbrecht, M. Mavrikis, B. Miller, R. Miner, M. Sargent, K. Siegrist, N. Soiffer, S. Watt, and M. Zergaoui. Mathematical Markup Language (MathML) version 3.0. W3C working draft, W3C, 2007. `http://www.w3.org/TR/MathML3`.
3. S. Autexier, A. Fiedler, T. Neumann, and M. Wagner. Supporting user-defined notations when integrating scientific text-editors with proof assistance systems. In M. Kauers, M. Kerber, R. Miner, and W. Windsteiger, editors, *Towards Mechanized*

Mathematical Assistants. MKM/Calculemus, number 4573 in LNAI. Springer, 2007.
4. D. Beckett. Turtle – terse RDF triple language, 2007. `http://www.dajobe.org/2004/01/turtle/`.
5. M. Bröcheler. The mathematical semantic web. Bachelor's thesis, Computer Science, Jacobs University, Bremen, 2007.
6. M. Buffa, F. Gandon, G. Ereteo, P. Sander, and C. Faron. Sweetwiki: A semantic wiki. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2008. in press.
7. S. Buswell, O. Caprotti, D. P. Carlisle, M. C. Dewar, M. Gaetano, and M. Kohlhase. The Open Math standard, version 2.0. Technical report, The Open Math Society, 2004. `http://www.openmath.org/standard/om20`.
8. F. Cajori. *A History of Mathematical Notations*. Courier Dover Publications, 1993. Originally published in 1929.
9. O. Caprotti, M. Dewar, and D. Turi. Mathematical service matching using description logic and OWL. In A. Asperti, G. Bancerek, and A. Trybulec, editors, *Mathematical Knowledge Management*, number 3119 in LNAI. Springer, 2004.
10. J. H. Davenport. OpenMath in a (semantic) web. In O. Caprotti, S. Xambó, M.-A. Huertas, M. Kohlhase, and M. Seppälä, editors, *3rd JEM Workshop – Joining International Mathematics*, 2008. `http://jem-thematic.net/workshop3`.
11. A. Gupta and I. S. Mumick, editors. *Materialized views: techniques, implementations, and applications*. MIT Press, Cambridge, MA, USA, 1999.
12. M. Kay. XSL Transformations (XSLT) Version 2.0. W3C Recommendation, W3C, 2007. `http://www.w3.org/TR/2007/REC-xslt20-20070123/`.
13. M. Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in LNAI. Springer, 2006.
14. M. Kohlhase, C. Müller, and F. Rabe. Notations for living mathematical documents. In *Mathematical Knowledge Management, MKM'08*, LNAI. Springer Verlag, 2008. in press.
15. 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.
16. A. Krowne. An architecture for collaborative math and science digital libraries. Master's thesis, Virginia Tech, 2003. `http://scholar.lib.vt.edu/theses/available/etd-09022003-150851/`.
17. C. Lange. SWiM – a semantic wiki for mathematical knowledge management. Technical Report 5, Jacobs University Bremen, 2007. `http://kwarc.info/projects/swim/pubs/tr-swim.pdf`.
18. C. Lange. SWiM: A semantic wiki for mathematical knowledge management. `http://kwarc.info/projects/swim/`, 2008.
19. S. Manzoor, P. Libbrecht, C. Ullrich, and E. Melis. Authoring Presentation for OpenMath. In M. Kohlhase, editor, *Mathematical Knowledge Management*, number 3863 in LNAI. Springer, 2005.
20. M. Marchiori. The mathematical semantic web. In A. Asperti, B. Buchberger, and J. H. Davenport, editors, *Mathematical Knowledge Management*, number 2594 in LNCS. Springer, 2003.
21. J. Marsh, D. Orchard, and D. Veillard. XML inclusions (XInclude) version 1.0 (second edition). W3C Recommendation, World Wide Web Consortium (W3C), Nov. 2006. Available at `http://www.w3.org/TR/2006/REC-xinclude-20061115/`.

22. N. Müller and M. Wagner. Towards Improving Interactive Mathematical Authoring by Ontology-driven Management of Change. In A. Hinneburg, editor, *LWA (Lernen, Wissensentdeckung und Adaptivität)*, 2007.
23. P. Murray-Rust, H. S. Rzepa, and M. Wright. Development of chemical markup language (cml) as a system for handling complex chemical content. *New Journal of Chemistry Articles*, 25:618–634, 2001.
24. E. Oren, R. Delbru, K. Möller, M. Völkel, and S. Handschuh. Annotation and navigation in semantic wikis. In M. Völkel, S. Schaffert, and S. Decker, editors, *1st Workshop on Semantic Wikis*, volume 206 of *CEUR Workshop Proceedings*, 2006.
25. E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF. W3C Recommendation, W3C, 2006. `http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/`.
26. S. Schaffert. IkeWiki: A semantic wiki for collaborative knowledge management. In *1st International Workshop on Semantic Technologies in Collaborative Applications (STICA)*, 2006.
27. E. Smirnova and S. M. Watt. Notation Selection in Mathematical Computing Environments. In *Proc. Transgressive Computing (TC) 2006*, 2006.
28. Mailing list of the semantic wiki interest group. swikig@aifb.uni-karlsruhe.de, `http://www.aifb.uni-karlsruhe.de/mailman/listinfo/swikig`.
29. M. Völkel, M. Krötzsch, D. Vrandečić, H. Haller, and R. Studer. Semantic Wikipedia. In *15th WWW conference*, 2006.
30. Wikipedia, the free encyclopedia. `http://www.wikipedia.org`, 2001–2007.
31. C. Zinn. Bootstrapping a semantic wiki application for learning mathematics. In Y. Sure and S. Schaffert, editors, *Semantics: From Visions to Applications*, 2006.