sTeX: Semantic Markup in TeX/LaTeX

Michael Kohlhase
Jacobs University, Bremen
`http://kwarc.info/kohlhase`

June 17, 2010

Abstract

We present a collection of TeX macro packages that allow to markup TeX/LaTeX documents semantically without leaving the document format, essentially turning TeX/LaTeX into a document format for mathematical knowledge management (MKM).

Contents

1 Introduction

The last few years have seen the emergence of various content-oriented Xml-based, content-oriented markup languages for mathematics on the web, e.g. OpenMath [Bus+04], Content-MathML [Aus+03], or our own OMDoc [Koh06]. These representation languages for mathematics, that make the structure of the mathematical knowledge in a document explicit enough that machines can operate on it. Other examples of content-oriented formats for mathematics include the various logic-based languages found in automated reasoning tools (see [RV01] for an overview), program specification languages (see e.g. [Ber89]).

The promise if these content-oriented approaches is that various tasks involved in "doing mathematics" (e.g. search, navigation, cross-referencing, quality control, user-adaptive presentation, proving, simulation) can be machine-supported, and thus the working mathematician is relieved to do what humans can still do infinitely better than machines: The creative part of mathematics — inventing interesting mathematical objects, conjecturing about their properties and coming up with creative ideas for proving these conjectures. However, before these promises can be delivered upon (there is even a conference series [Mkm] studying "Mathematical Knowledge Management (MKM)"), large bodies of mathematical knowledge have to be converted into content form.

Even though MathML is viewed by most as the coming standard for representing mathematics on the web and in scientific publications, it has not not fully taken off in practice. One of the reasons for that may be that the technical communities that need high-quality methods for publishing mathematics already have an established method which yields excellent results: the TeX/LaTeX system: and a large part of mathematical knowledge is prepared in the form of TeX/LaTeX documents.

TeX [Knu84] is a document presentation format that combines complex page-description primitives with a powerful macro-expansion facility, which is utilized in LaTeX (essentially a set of TeX macro packages, see [Lam94]) to achieve more content-oriented markup that can be adapted to particular tastes via specialized document styles. It is safe to say that LaTeX largely restricts content markup to the document structure[1], and graphics, leaving the user with the presentational TeX primitives for mathematical formulae. Therefore, even though LaTeX goes a great step into the direction of an MKM format, it is not, as it lacks infrastructure for marking up the functional structure of formulae and mathematical statements, and their dependence on and contribution to the mathematical context.

1.1 The Xml vs. TeX/LaTeX Formats and Workflows

MathML is an Xml-based markup format for mathematical formulae, it is standardized by the World Wide Web Consortium in [Aus+03], and is supported by the major browsers. The MathML format comes in two integrated components: presentation MathML and content MathML. The former provides a comprehensive set of layout primitives for presenting the visual appearance of mathematical formulae, and the second one the functional/logical structure of the conveyed mathematical objects. For all practical concerns, presentation MathML is equivalent to the math mode of TeX. The text mode facilitates of TeX (and the multitude of LaTeX classes) are relegated to other Xml formats, which embed MathML.

The programming language constructs of TeX (i.e. the macro definition facilities[2]) are relegated to the Xml programming languages that can be used to develop language extensions. transformation language xslt [Dea99; Kay00] or proper Xml-enabled The Xml-based syntax and the separation of the presentational-, functional- and programming/extensibility concerns in MathML has some distinct advantages over the integrated approach in TeX/LaTeX on the services side: MathML gives us better

- integration with web-based publishing,

[1]supplying macros e.g. for sections, paragraphs, theorems, definitions, etc.

[2]We count the parser manipulation facilities of TeX, e.g. category code changes into the programming facilities as well, these are of course impossible for MathML, since it is bound to Xml syntax.

- accessibility to disabled persons, e.g. (well-written) MathML contains enough structural information to supports screen readers.
- reusability, searchabiliby and integration with mathematical software systems (e.g. copy-and-paste to computer algebra systems), and
- validation and plausibility checking.

On the other hand, TEX/LATEX/s adaptable syntax and tightly integrated programming features within has distinct advantages on the authoring side:

- The TEX/LATEX syntax is much more compact than MathML (see the difference in Figure 1 and Equation 1), and if needed, the community develops LATEX packages that supply new functionality in with a succinct and intuitive syntax.
- The user can define ad-hoc abbreviations and bind them to new control sequences to structure the source code.
- The TEX/LATEX community has a vast collection of language extensions and best practice examples for every conceivable publication purpose and an established and very active developer community that supports these.
- There is a host of software systems centered around the TEX/LATEX language that make authoring content easier: many editors have special modes for LATEX, there are spelling/style/-grammar checkers, transformers to other markup formats, etc.

In other words, the technical community is is heavily invested in the whole workflow, and technical know-how about the format permeates the community. Since all of this would need to be re-established for a MathML-based workflow, the technical community is slow to take up MathML over TEX/LATEX, even in light of the advantages detailed above.

1.2 A LATEX-based Workflow for Xml-based Mathematical Documents

An elegant way of sidestepping most of the problems inherent in transitioning from a LATEX-based to an Xml-based workflow is to combine both and take advantage of the respective advantages.

The key ingredient in this approach is a system that can transform TEXLATEX documents to their corresponding Xml-based counterparts. That way, Xml-documents can be authored and prototyped in the LATEX workflow, and transformed to Xml for publication and added-value services, combining the two workflows.

There are various attempts to solve the TEX/LATEX to Xml transformation problem; the most mature is probably Bruce Miller's LaTeXML system [Mil10]. It consists of two parts: a re-implementation of the TEX analyzer with all of it's intricacies, and a extensible Xml emitter (the component that assembles the output of the parser). Since the LATEX style files are (ultimately) programmed in TEX, the TEX analyzer can handle all TEX extensions, including all of LATEX. Thus the LaTeXML parser can handle all of TEX/LATEX, if the emitter is extensible, which is guaranteed by the LaTeXML binding language: To transform a TEX/LATEX document to a given Xml format, all TEX extensions[3] must have "LaTeXML bindings"binding, i.e. a directive to the LaTeXML emitter that specifies the target representation in Xml.

2 The Packages of the sTEX Collection

In the following, we will shortly preview the packages and classes in the sTEX collection. They all provide part of the solution of representing semantic structure in the TEX/LATEX workflow. We will group them by the conceptual level they address[1] EdNote(1)

[3]i.e. all macros, environments, and syntax extensions used int the source document

[1]EdNote: come up with a nice overview figure here!

2.1 Content Markup of Mathematical Formulae in TEX/LATEX

The first two packages are concerned basically with the math mode in TEX, i.e. mathematical formulae. The underlying problem is that run-of-the-mill TEX/LATEX only specifies the presentation (i.e. what formulae look like) and not their content (their functional structure). Unfortunately, there are no good methods (yet) to infer the latter from the former, but there are ways to get presentation from content.

Consider for instance the following "standard notations"[4] for binomial coefficients: $\binom{n}{k}$, ${}_nC^k$, $\mathcal{C}^n_k$, and $\mathcal{C}^k_n$ all mean the same thing: $\frac{n!}{k!(n-k)!}$. This shows that we cannot hope to reliably recover the functional structure (in our case the fact that the expression is constructed by applying the binomial function to the arguments n and k) from the presentation alone.

The solution to this problem is to dump the extra work on the author (after all she knows what she is talking about) and give them the chance to specify the intended structure. The markup infrastructure supplied by the STEX collection lets the author do this without changing[5] the visual appearance, so that the LATEX workflow is not disrupted. . We speak of semantic preloading for this process and call our collection of macro packages STEX (Semantic TEX). For instance, we can now write

`\CSumlLimits{k}1\infty{\Cexp{x}k}` instead of the usual `\sum_{k=1}^\infty x^k` (1)

In the first form, we specify that you are applying a function (—CSumLimits— $\hat{=}$ Sum with Limits) to 4 arguments: (*i*) the bound variable k (that runs from) (*ii*) the number 1 (to) (*iii*) ∞ (to infinity summing the terms) (*iv*) `\Cexp{x}k` (i.e. x to the power k). In the second form, we merely specify hat LATEX should draw a capital Sigma character (σ) with a lower index which is an equation $k = 1$ and an upper index ∞. Then it should place next to it an x with an upper index k.

Of course human readers (that understand the math) can infer the content structure from this presentation, but the LATEXML converter (who does not understand the math) cannot, but we want to have the content MATHML expression in Figure 1

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <apply>
    <sum/>
      <bvar><ci>k</ci></bvar>
      <lowlimit><cn>1</cn></lowlimit>
      <uplimit><infinit/></cn></uplimit>
      <apply><exp/><ci>x</ci><ci>k</ci></apply>
   </apply>
</math>
```

Example 1: Content MATHML Form of $\sum_{k=1}^{\infty} x^k$

Obviously, a converter can infer this from the first LATEX structure with the help of the curly braces that indicate the argument structure, but not from the second (because it does not understand the math). The nice thing about the —cmathml— infrastructure is that you can still run LATEX over the first form and get the same formula in the DVI file that you would have gotten from running it over the second form. That means, if the author is prepared to write the mathematical formulae a little differently in her LATEX sources, then she can use them in XML and LATEX at the same time.

[4]The first one is standard e.g. in Germany and the US, the third one in France, and the last one in Russia

[5]However, semantic annotation will make the author more aware of the functional structure of the document and thus may in fact entice the author to use presentation in a more consistent way than she would usually have.

2.1.1 `cmathml`: Encoding Content MathML in TeX/LaTeX

The `cmathml` package (see [**Kohlhase:tbscml:ctan**]) provides a set of macros that correspond to the K-14 fragment of mathematics (Kindergarten to undergraduate college level ($\widehat{=}14^{th}$ grade)). We have already seen an example above in equation (1), where the content markup in TeX corresponds to a content MathML-expression (and can actually be translated to this by the LaTeXML system.) However, the content MathML vocabulary is fixed in the MathML specification and limited to the K-14 fragment; the notation of mathematics of course is much larger and extensible on the fly.

2.1.2 `presentation`: Flexible Presentation for Semantic Macros

The `presentation` package (see [**Kohlhase:ipsmsl:ctan**]) supplies an infrastructure that allows to specify the presentation of semantic macros, including preference-based bracket elision. This allows to markup the functional structure of mathematical formulae without having to lose high-quality human-oriented presentation in LaTeX. Moreover, the notation definitions can be used by MKM systems for added-value services, either directly from the sTeX sources, or after translation.

2.2 Mathematical Statements

2.2.1 `statements`: Extending Content Macros for Mathematical Notation

The `statements` package (see[**Kohlhase:smms:ctan**]) provides semantic markup facilities for mathematical statements like Theorems, Lemmata, Axioms, Definitions, etc. in sTeX files. This structure can be used by MKM systems for added-value services, either directly from the sTeX sources, or after translation.

2.2.2 `sproof`: Extending Content Macros for Mathematical Notation

The `sproof` package (see [**Kohlhase:smp:ctan**])supplies macros and environment that allow to annotate the structure of mathematical proofs in sTeX files. This structure can be used by MKM systems for added-value services, either directly from the sTeX sources, or after translation.

2.3 Context Markup for Mathematics

2.3.1 `modules`: Extending Content Macros for Mathematical Notation

The `modules` package (see [**KohAmb:smmssl:ctan**]) supplies a definition mechanism for semantic macros and a non-standard scoping construct for them, which is oriented at the semantic depency relation rather than the document structure. This structure can be used by MKM systems for added-value services, either directly from the sTeX sources, or after translation.

2.4 Mathematical Document Classes

2.4.1 Connexions Modules

CNXLaTeX (see [**Kohlhase:clbscm:ctan**]) is a collection of LaTeX macros that allow to write Connexions modules without leaving the LaTeX workflow. Modules are authored in CNXLaTeX using only a text editor, transformed to PDF and proofread as usual. In particular, the LaTeX workflow is independent of having access to the Connexions system, which makes CNXLaTeX attractive for the initial version of single-author modules.

For publication, CNXLaTeX modules are transformed to CNXml via the LaTeXML translator and can be uploaded to the Connexions system.

2.4.2 OMDoc Documents

The `omdoc` package provides an infrastructure that allows to markup OMDoc documents in LaTeX. It provides `omdoc.cls`, a class with the and `omdocdoc.sty`

2.5 Conclusion

The sTeX collection provides a set of semantic macros that extends the familiar and time-tried LaTeX workflow in academics until the last step of Internet publication of the material. For instance, a Connexions module can be authored and maintained in LaTeX using a simple text editor, a process most academics in technical subjects are well familiar with. Only in a last publishing step (which is fully automatic) does it get transformed into the Xml world, which is unfamiliar to most academics.

Thus, sTeX can serve as a conceptual interface between the document author and MKM systems: Technically, the semantically preloaded LaTeX documents are transformed into the (usually Xml-based) MKM representation formats, but conceptually, the ability to semantically annotate the source document is sufficient.

The sTeX macro packages have been validated together with a case study [Koh04], where we semantically preload the course materials for a two-semester course in Computer Science at Jacobs University Bremen and transform them to the OMDoc MKM format.

2.6 Licensing, Download and Setup

The sTeX packages are licensed under the LaTeX Project Public License [**LPPL**], which basically means that they can be downloaded, used, copied, and even modified by anyone under a set of simple conditions (e.g. if you modify you have to distribute under a different name).

The sTeX packages and classes can be obtained as a self-documenting LaTeX packages: To obtain a package ⟨*package*⟩ download the files ⟨*package*⟩`.dtx` and ⟨*package*⟩`.ins` from

`https://svn.kwarc.info/repos/kwarc/projects/stex/sty/stex/`⟨*package*⟩`/`

To extract the LaTeX package ⟨*package*⟩`.sty` and the LaTeXML bindings in ⟨*package*⟩`.ltxml`, run the LaTeX formatter on `cmathml.ins`, e.g. by typing `latex cmathml.ins` to a shell. To extract the documentation (the version of this document that goes with the extracted package) run the LaTeX formatter on `cmathml.dtx` e.g. by typing `latex` ⟨*package*⟩`.dtx` to a shell.

Usually, the sTeX distribution will also have the newest versions of the files ⟨*package*⟩`.sty`, ⟨*package*⟩`.ltxml`, and the documentation ⟨*package*⟩`.pdf` pre-generated for convenience, so they can be downloaded directly from the URL above.

To install the package, copy the file ⟨*package*⟩`.sty` somewhere, where TeX/LaTeX can find it and rebuild TeX's file name database. This is done by running the command `texhash` or `mktexlsr` (they are the same). In `MikTEX`, there is a menu option to do this.

3 Utilities

To simplify dealing with sTeX documents, we are providing a small collection of command line utilities, which we will describe here. For details and downloads go to `http://kwarc.info/projects/stex`.

`msplit` splits an sTeX file into smaller ones (one module per file)

`rf` computes the "reuse factor", i.e. how often sTeX modules are reused over a collection of documents

`sgraph` visualizes the module graph

`sms` computes the sTeX module signatures for a give sTeX file

`bms` proposes a sensible module structure for an un-annotated sTeX file

References

[Aus+03] Ron Ausbrooks et al. *Mathematical Markup Language (MathML) Version 2.0 (second edition)*. W3C Recommendation. World Wide Web Consortium, 2003. URL: `http://www.w3.org/TR/MathML2`. (Cit. on p. 1).

[Ber89] J. A. Bergstra. *Algebraic specification*. ACM Press, 1989. (Cit. on p. 1).

[Bus+04] Stephen Buswell et al. *The Open Math Standard, Version 2.0*. Tech. rep. The Open Math Society, 2004. URL: `http://www.openmath.org/standard/om20`. (Cit. on p. 1).

[Dea99] Stephen Deach. *Extensible Stylesheet Language (XSL) Specification*. W3C Working Draft. World Wide Web Consortium (W3C), 1999. URL: `http://www.w3.org/TR/WD-xsl`. (Cit. on p. 1).

[Kay00] Michael Kay. *XSLT Programmers Reference*. Wrox, 2000. (Cit. on p. 1).

[KG10] Michael Kohlhase and Deyan Ginev. *`presentation.sty`: An Infrastructure for Presenting Semantic Macros in sTeX*. Self-documenting LaTeX package. 2010. URL: `https://svn.kwarc.info/repos/stex/trunk/sty/presentation/presentation.pdf`.

[KGA10] Michael Kohlhase, Deyan Ginev, and Rares Ambrus. *`modules.sty`: Semantic Macros and Module Scoping in sTeX*. Self-documenting LaTeX package. 2010. URL: `https://svn.kwarc.info/repos/stex/trunk/sty/modules/modules.pdf`.

[Knu84] Donald E. Knuth. *The TeXbook*. Addison Wesley, 1984. (Cit. on p. 1).

[Koh04] Michael Kohlhase. "Semantic Markup for TeX/LaTeX". In: *Mathematical User Interfaces*. Ed. by Paul Libbrecht. 2004. URL: `http://www.activemath.org/~paul/MathUI04`. (Cit. on p. 5).

[Koh06] Michael Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. Aug. 2006. URL: `http://omdoc.org/pubs/omdoc1.2.pdf`. (Cit. on p. 1).

[Koh10a] Michael Kohlhase. *CNXLaTeX: A LaTeX-based Syntax for Connexions Modules*. Self-documenting LaTeX package. 2010. URL: `https://svn.kwarc.info/repos/stex/trunk/sty/cnx/cnx.pdf`.

[Koh10b] Michael Kohlhase. *`cmathml.sty`: A TeX/LaTeX-based Syntax for Content MathML*. Self-documenting LaTeX package. 2010. URL: `https://svn.kwarc.info/repos/stex/trunk/sty/cmathml/cmathml.pdf`.

[Koh10c] Michael Kohlhase. *`sproof.sty`: Structural Markup for Proofs*. Self-documenting LaTeX package. 2010. URL: `https://svn.kwarc.info/repos/stex/trunk/sty/sproof/sproof.pdf`.

[Koh10d] Michael Kohlhase. *`statements.sty`: Structural Markup for Mathematical Statements*. Self-documenting LaTeX package. 2010. URL: `https://svn.kwarc.info/repos/stex/trunk/sty/statements/statements.pdf`.

[Lam94] Leslie Lamport. *LaTeX: A Document Preparation System, 2/e*. Addison Wesley, 1994. (Cit. on p. 1).

[Mel+01] E. Melis et al. "The ActiveMath Learning Environment". In: *Artificial Intelligence and Education* 12.4 (2001). (Cit. on p. 5).

[Mil10] Bruce Miller. *`LaTeXML`: A LaTeX to XML Converter*. 2010. URL: `http://dlmf.nist.gov/LaTeXML/` (visited on 05/08/2010). (Cit. on p. 2).

[Mkm] *Meetings and Conferences on Mathematical Knowledge Management*. Project homepage. URL: `http://www.mkm-ig.org/meetings/` (visited on 03/12/2007). (Cit. on p. 1).

[RV01] Alan Robinson and Andrei Voronkov, eds. *Handbook of Automated Reasoning*. Vol. I and II. Elsevier Science and MIT Press, 2001. (Cit. on p. 1).