Semantic Markup in TFX/IATFX

Michael Kohlhase FAU Erlangen-Nürnberg

http://kwarc.info/kohlhase

November 23, 2019

Abstract

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

The functionality of STEX is distributed over many packages and classes, see the original STEX article [Koh08] for an overview and motivation.

- Content Markup of Mathematical Formulae in TEX/IATEX
 - cmath: Content Markup for Mathematical Formulae
 - presentation: Flexible Notations for Semantic Macros
- Mathematical Statements
 - omtext: Mathematical Text
 - statements: Semantic Markup for Mathematical Statements
 - sproof: Semantic Markup for Proofs
 - problem: Semantic Markup for Problems and Exercises
- Context Markup
 - modules: Object-Oriented Mathematics
 - smultiling: Support for Multilingual Mathematics
 - structview: Advanced Inheritance and Interpretation in Mathematics
- Mathematical Document Classes
 - omdoc: Open Mathematical Documents
 - smglom: Semantic Multilingual Glossary of Mathematics
 - mikoslides: Slides and Lectures for STEX
 - hwexam: Content Markup for Homeworks and Exams
- Metadata
 - dcm: Dublin Core Metadata
 - workaddress: FOAF-like Metadata (persons/affiliations)
- Auxiliary Packages
 - metakeys: Extended Key/Value Management
 - pathsuris: Absolute Path Handling for STEX
 - sref: Semantic References in STEX
 - tikzinput: Diagrams as Standalone Images
 - mathhub: Support for STFX on MathHub.info

Contents

1	Introduction		3
	1.1	The XML vs. TFX/LATFX Formats and Workflows	3
	1.2	A LATEX-based Workflow for XML-based Mathematical Documents	5
	1.3	Generating OMDoc from STEX	5
	1.4	Conclusion	5
	1.5	Licensing, Download and Setup	6
2	The Packages of the STEX Collection		
	2.1	The STEX Distribution	7
	2.2	Content Markup of Mathematical Formulae in TFX/IATFX	7
	2.3	Mathematical Statements	8
	2.4	Context Markup for Mathematics	8
	2.5	Mathematical Document Classes	9
	2.6	Metadata	10
	$\frac{2.7}{2.7}$	Support for MathHub	10
	2.8	Auxiliary Packages	10
3	Workflows and Best Practices		
	3.1	The "Little Modules" Approach	12
	3.2	Basic Utilities & Makefiles	13
	3.3	MathHub: a Portal for Active Mathematical Documents	13
	3.4	lmh: MathHub's Build System Locally	13
	5.4	Imil. Mathrido's Build System Locally	10
4	The	e Implementation	15
	4.1	Package Options	15
	4.2	The STEX Logo	15
	4.3	The STEX Basics	15

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. Open-Math [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 [] 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 structure¹, 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. T_EX/\(\text{L}\)T_EX 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 presentation MathML and content MathML content

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

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,
- accessibility to disabled persons, e.g. (well-written) MathML contains enough structural information to supports screen readers.
- reusability, searchability 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/ETEX/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, and if needed, the community develops IaTeX 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/IATeX 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 T_EX/I^AT_EX language that make authoring content easier: many editors have special modes for I^AT_EX, 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.

²We 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.

1.2 A Lagrangian Texture 1.2 A Lagrangian 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 TexleTex documents to their corresponding XML-based counterparts. That way, XML-documents can be authored and prototyped in the LeTex workflow, and transformed to XML for publication and added-value services, combining the two workflows.

There are various attempts to solve the TEX/IFTEX to XML transformation problem (see [Sta+09] for an overview); the most mature is probably Bruce Miller's IFTEXML system [LTX]. 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 IFTEX style files are (ultimately) programmed in TEX, the TEX analyzer can handle all TEX extensions, including all of IFTEX. Thus the IFTEXML parser can handle all of TEX/IFTEX, if the emitter is extensible, which is guaranteed by the IFTEXML binding language: To transform a TEX/IFTEX document to a given XML format, all TEX extensions must have "IFTEXML bindings" binding, i.e. a directive to the IFTEXML emitter that specifies the target representation in XML.

1.3 Generating OMDoc from STEX

The STEX packages (see Section 2) provide functionalities for marking up the functional structure of mathematical documents, so that the LATEX sources contain enough information that can be exported to the OMDoc format (Open Mathematical Documents; see [Koh06]). For the actual transformation, we use a LATEXML plugin [L2O] that provides the LATEXML bindings for the STEX packages.

1.4 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, an SMGloM 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.

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

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 OM-Doc MKM format.

1.5 Licensing, Download and Setup

The STEX packages are licensed under the LATEX Project Public License [Pro07], 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).

1.5.1 The STEX Distribution

The STEX packages and classes are available from the Comprehensive TeX Archive Network (CTAN [CTAN]) and are part of the primary TeX/LATeX distributions (e.g. TeXlive [TL] and MikTeX [MiK]). The development version is on GitHub [sTeX], it can cloned or forked from the repository URL

```
https://github.com/sLaTeX/sTeX.git
```

Warning: the STEX is under continuous development and CTAN updates are as infrequent as they are tedious, so the GIT version is almost always preferable.

It is usually a good idea to enlarge the internal memory allocation of the TEX/IATEXexecutables. This can be done by adding the following configurations in texmf.cnf (or changing them, if they already exist). Note that you will probably need sudo to do this.

After that, you have to run the

```
sudo fmtutil-sys --all
```

With this installation using STEX is as painless as using IATEX, just make sure the STEX distribution is where the latex can find it and run pdflatex over the main file.

1.5.2 The STEX Plugin for LATEXML

For the OMDoc transformation of STEX documents we use a LaTEXML plugin that provides the LaTEXML bindings for the STEX packages. For installation and setup follow the instructions at [L2O]

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/IATEX workflow. We will group them by the conceptual level they address. Figure 1 gives an overview.

2.1 The STeX Distribution

The stex package provides stex.sty that just loads all packages below and passes around the package options accordingly and stex-logo.sty that provides the macros \sTeX and \stex that typeset the STeX logo.

\sTeX

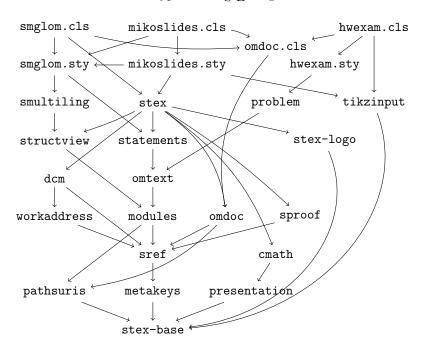


Figure 1: The STEX packages and their dependencies.

2.2 Content Markup of Mathematical Formulae in T_FX/I^LT_FX

2.2.1 cmath: Building Content Math Representations

The cmath package (see [Koh18a]) supplies an interface for building content math representations. It gives special macros for marking up variables, applications and bindings. It supports the transformation into both OpenMath [Bus+04] and content MathML [Aus+03].

2.2.2 presentation: Flexible Notations for Semantic Macros

The presentation package (see [KG18a]) 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 IATEX. Moreover, the notation definitions can be used by MKM systems for added-value services, either directly from the STEX sources, or after translation.

2.3 Mathematical Statements

2.3.1 omtext: Mathematical Text

The omtext package (see[Koh18f]) supplies an infrastructure for writing mathematical text fragments in STEX. The basic omtext environment can be given various metadata that are relevant for content-oriented mathematics. omtext is specialized to various specially layouted text forms like quotes and such. Furthermore, the omtext package provides content-oriented index markup.

2.3.2 statements: Definition, Theorem, Example and the like

The statements package (see[Koh18j]) 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.

2.3.3 sproof: sTeX Proofs

The sproof package (see [Koh18h]) 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.

2.3.4 problem: Problems and Exercises

The problem package (see [Koh18g]) supplies macros and environment that allow to annotate the structure of mathematical and exercises in STEX files. These structures can be used for homework assignments and exams; see section 2.5.2.

2.4 Context Markup for Mathematics

2.4.1 modules: Extending Content Macros for Mathematical Notation

The modules package (see [KGA18a]) supplies a definition mechanism for semantic macros and a non-standard scoping construct for them, which is oriented at the semantic dependency relation rather than the document structure. This structure can be used by MKM systems for added-value services, either directly from the

SIEX sources, or after translation. A side effect of this is that we have an "object-oriented" inheritance mechanism for semantic macros: the semantic macros for the mathematical objects described in a module come with the module itself. As a consequence, the **modules signatures** (only the macro definitions, not the descriptions) need to be loaded before they can be used somewhere else.

2.4.2 smultiling: Multilingual Mathematical Modules

In multilingual settings, i.e. where we have multiple STEX documents that are translations of each other, it is better to separate the module signature from the descriptive document. The smultiling package (see [KG18b]) provides the necessary infrastructure.

2.4.3 structview: Structures and Views

OMDoc/MMT structures add advanced inheritance and interpretation methods for mathematical knowledge management – see [RK13] for a motivation and the grisly details. The structview package (see [Koh18k]) provides the necessary infrastructure.

2.5 Mathematical Document Classes

The STEX distribution provides four top-level document classes for various types of mathematical documents. In all cases, the functionality of the $\langle class \rangle$.cls is mainly provided by $\langle class \rangle$.sty, allowing the LATEX document class $\langle class \rangle$.cls to be minimal, only setting up the required packages and calling corresponding package $\langle class \rangle$.sty.

2.5.1 OMDoc Documents

The omdoc class and package (see [Koh18e]) provides an infrastructure that allows to markup up generic mathematical documents with OMDoc structures. It provides the omgroup environment for reusable document fragments.

2.5.2 hwexam: Homeworks and Exams

The hwexam package [Koh18c] infrastructure for marking up homework assignments, and exams. The content markup strategy employed in STEX allows to specify – and profit from – administrative metadata such as time and point counts. This package relies on the problem package [Koh18g] which provides markup for problems, hints, and solutions.

2.5.3 smglom: Mathematical Glossary

The smglom package provides a document class for the Semantic Multilingual Glossary of Mathematics. This bakes in special file/module conventions that allow the concise and convenient markup of multilingual glossary modules.

2.5.4 mikoslides: Slides and Course Notes

The mikoslides package provides a document class from which we can generate both course slides – via the beamer class – and course notes – via the omdoc class – in a transparent way.

2.6 Metadata

The metadata packages allow embedding linked-data-style annotations into texts. They are tied to well-known ontologies.

2.6.1 dcm: Dublin Core Metadata

The dcm package (see [Koh18b]) provides keys and macros for annotating Dublin Core Metadata [DC].

2.6.2 workaddress: Markup for FOAF Metadata

The workaddress package (see [Koh18m]) provides keys and macros for annotating metadata from a part of the FOAF (Friend-of-a-Friend; persons, their relations, and affiliations) ontology [FOAF14].

2.7 Support for MathHub

The mathhub package provides the supplementary packages mikoslides-mh, modules-mh.sty, omtext-mh.sty, problem-mh.sty, smultiling-mh.sty, structview-mh.sty, and tikzinput-mh.sty with variants of the user-visible macros that are adapted to the MathHub system - see Section 3.3 for details.

2.8 Auxiliary Packages

2.8.1 metakeys: An extended key/value Interface for STFX

metakeys is a purely syntactic utility package, see [Koh18d], it should eventually, probably be replaced by something more standard and maintained.

2.8.2 pathsuris: Managing Relative/Absolute File Paths

Experience shows that distributing STEX content into small IATEX files makes content much more reusable. The price we pay is that we have to manage many file names from which a document is composed. Practical experience shows that system-internally, we want to use absolute file URIs (as canonical file names) to get around redundant file loading which would lead to exponential blowups. The pathsURIs package (see [KGA18b]) provides the necessary infrastructure.

2.8.3 sref: Semantic References

 $\mbox{ST}_{\mbox{E}}\mbox{X}$ uses its own (semantic) infrastructure for internal (and cross-document!) references: the \mbox{sref} package (see [Koh18i]). It should eventually, probably be replaced by – or at least based on – something more standard and maintained.

2.8.4 tikzinput: External TIKZ Pictures as Standalone Images

tikz is a diagram package for LATEX, which can be used for mixed (i.e. with STEX islands) diagrams in mathematical documents. However LATEXML treatment of tikz is still experimental, so we use the tikzinput package (see [Koh18l]) to externalize the tikz material into standalone files, which can be easily re-used and converted to e.g. SVG independently and then input as regular images.

3 Workflows and Best Practices

3.1 The "Little Modules" Approach

One of the key advantages of semantic markup with STEX is that the STEX sources are highly reusable by the "object-oriented" inheritance model induced by STEX modules. It turned out to be useful to divide STEX documents into three kinds of files:

- 1. **module files**: files that essentially contain a collection of STEX modules [KGA18a] usually a single one whose module name coincides the file name base.
- 2. **fragment files**: files that contain a group of input references to module- or fragment files usually one group deep for flexibility, transition text, and additional remarks.
- 3. **driver files** that set up the document class, contain the preambles, and input reference fragment files.

These correspond to the STEX documents, but can reuse and share STEX fragments and modules. Figure 2 shows a situation, where we have two courses given over multiple years, which results in five course notes documents given by driver files, wich share quite a few components. As drivers and fragment files are mostly content-free – they only contribute document structure, this lets all documents contribute from the development of the modules.



Figure 2: Reuse of Fragments and Modules in a Course Notes Setting

The downside of this "object-oriented" inheritance mechanism is that we need to keep the module signatures (see Section 2.4.1) up to date adding to the complexity of document management.

Another advantage of the "little modules approach" is that modules can be developed separately. Indeed all modules of a given subject share common preand post-ambles which can be developed in special files – usually named pre.tex, post.tex, and preamble.tex (the latter is included in pre.tex). Given such a setup, the call to pdflatex can be suitably adapted to handle the pre/postfixes.

3.2 Basic Utilities & Makefiles

The STEX distribution contains three basic command line utilities to manage STEX documents in the bin directory of the distribution.

sms computes the STEX module signatures for a given STEX file (see [KGA18a] details).

filedate and checksum that help keeping the metadata of the self-documenting LATEX packages in the STEX distribution up to date.

installFonts.sh that installs the fonts necessary for Chinese STEX documents.

These are supplemented by a set of UNIX Makefiles in the lib/make directory. The way to use them is to include them into a Makefile in the directory and then run one of the targets pdf and mpdf to make the PDF versions of the drivers and modules and omdoc and mods to generate OMDoc. Note that we need to make sms in order to make the respective STFX module signatures for the modules.

3.3 MathHub: a Portal for Active Mathematical Documents

MathHub (http://mathhub.info see [Ian+14]) is a portal for Active Mathematical Documents – documents that are made context-aware and interactive by semantic annotations. STEX is one of the main input formats for informal active documents. MathHub supports STEX documents in three ways:

- MathHub offers free/open hosting in document repositories for (mathematical) STEX document collections.
- 2. the backend system supports the large-scale change- and error-management for STFX documents in the "little modules" paradigm.
- 3. the front-end displays interactive (HTML5) documents generated from the STEX sources (via OMDoc).

The MathHub system is probably the best way of developing and hosting larger ST_EX document collections. It offers two authoring workflows an online authoring workflow via a direct web interface [MH] or casual users and an offline authoring workflow that we describe next.

3.4 lmh: MathHub's Build System Locally

As direct web editing workflows are not efficient for larger document collections, the MathHub system offers an offline authoring system. This uses GIT repositories for distribution – the author develops the document collection on a local working copy and then commits for inclusion to MathHub. The MathHub build system can be used locally for efficient development via the localmh system [lmh]. In a nutshell – see [LMH] for details –

- 1. localmh is installed in a docker container that supplies the build system and provides the lmh command suite.
- 2. 1mh pdf formats STEX modules to PDF building all dependencies, e.g. module signatures, first.
- 3. 1mh omdoc generates OMDoc for STEX documents again with dependencies
- 4. lmh xhtml generates active documents (in XHTML5) from the STEX sources or their OMDoc versions.
- 5. $lmh \langle gitsc \rangle$ distributes the git subcommand $\langle gitsc \rangle$ over multiple repositories.

Various other 1mh subcommands help with large-scale editing problems like renaming or moving modules, translations in multilingual settings, etc.

4 The Implementation

4.1 Package Options

The first step is to declare (a few) package options that handle whether certain information is printed or not. They all come with their own conditionals that are set by the options.

```
1 \*package\
2 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{statements}}
3 \PassOptionsToPackage{\CurrentOption}{structview}
4 \PassOptionsToPackage{\CurrentOption}{sproofs}
5 \PassOptionsToPackage{\CurrentOption}{omdoc}
6 \PassOptionsToPackage{\CurrentOption}{cmath}
7 \PassOptionsToPackage{\CurrentOption}{dcm}}
8 \ProcessOptions
```

Then we make sure that the necessary packages are loaded (in the right versions).

```
9 \RequirePackage{stex-logo}
10 \RequirePackage{omdoc}
11 \RequirePackage{statements}
12 \RequirePackage{structview}
13 \RequirePackage{sproof}
14 \RequirePackage{cmath}
15 \RequirePackage{dcm}
16 \langle / package \rangle
```

4.2 The STEX Logo

```
17 \langle *\logo \rangle
18 \RequirePackage{stex-base}
19 \RequirePackage{xspace}
20 \def\stex{%
21  \@ifundefined{texorpdfstring}%
22  {\let\texorpdfstring\@firstoftwo}%
23  {}%
24  \texorpdfstring{\raisebox{-.5ex}S\kern-.5ex\TeX}{sTeX}\xspace%
25 }
26 \def\sTeX{\stex}
27 \langle /\logo \rangle
```

4.3 The STEX Basics

Basic setup stuff

We define a conditional \ifClatexml to determine whether we are running LATEXML or pdflatex. We set it to false here, and in the LATEXML binding we set it to true.

```
28 \langle *base \rangle
29 \newif\if@latexml\@latexmlfalse
30 \langle /base \rangle
```

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

 k , 4 , 5 LaTeXML, 5

References

- [] Meetings and Conferences on Mathematical Knowledge Management. Project homepage. URL: http://www.mkm-ig.org/meetings/ (visited on 03/12/2007).
- [Aus+03] Ron Ausbrooks et al. Mathematical Markup Language (MathML) Version 2.0 (second edition). W3C Recommendation. World Wide Web Consortium (W3C), 2003. URL: http://www.w3.org/TR/MathML2.
- [Ber89] J. A. Bergstra. Algebraic specification. ACM Press, 1989.
- [Bus+04] Stephen Buswell et al. *The Open Math Standard*, *Version 2.0*. Tech. rep. The OpenMath Society, 2004. URL: http://www.openmath.org/standard/om20.
- [CTAN] CTAN the Comprehensive T_EX Archive Network. URL: http://ctan.org (visited on 12/11/2012).
- [DC] Dublin Core Metadata Initiative. URL: http://www.dublincore.org (visited on 11/09/2010).
- [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.
- [FOAF14] FOAF Vocabulary Specification 0.99. Namespace Document. The FOAF Project, Jan. 14, 2014. URL: http://xmlns.com/foaf/spec/.
- [Ian+14] Mihnea Iancu et al. "System Description: MathHub.info". In: Intelligent Computer Mathematics. Conferences on Intelligent Computer Mathematics (Coimbra, Portugal, July 7-11, 2014). Ed. by Stephan Watt et al. LNCS 8543. Springer, 2014, pp. 431-434. ISBN: 978-3-319-08433-6. URL: http://kwarc.info/kohlhase/papers/cicm14-mathhub.pdf.
- [Kay00] Michael Kay. XSLT Programmers Reference. Wrox, 2000.
- [KG18a] Michael Kohlhase and Deyan Ginev. presentation.sty: An Infrastructure for Presenting Semantic Macros in sTeX. Tech. rep. 2018.
 URL: https://github.com/sLaTeX/sTeX/raw/master/sty/presentation/presentation.pdf.
- [KG18b] Michael Kohlhase and Deyan Ginev. smultiling.sty: Multilinguality Support for sTeX. Tech. rep. 2018. URL: https://github.com/ sLaTeX/sTeX/raw/master/sty/smultiling/smultiling.pdf.
- [KGA18a] Michael Kohlhase, Deyan Ginev, and Rares Ambrus. modules.sty:

 Semantic Macros and Module Scoping in sTeX. Tech. rep. 2018. URL:

 https://github.com/sLaTeX/sTeX/raw/master/sty/modules/
 modules.pdf.

- [KGA18b] Michael Kohlhase, Deyan Ginev, and Rares Ambrus. *modules.sty:*Semantic Macros and Module Scoping in sTeX. Tech. rep. 2018. URL:

 https://github.com/sLaTeX/sTeX/raw/master/sty/pathsuris/pathsuris.pdf.
- [KGA18c] Michael Kohlhase, Deyan Ginev, and Rares Ambrus. modules.sty: Semantic Macros and Module Scoping in sTeX. Self-documenting LATEX package. 2018.
- [Knu84] Donald E. Knuth. The TEXbook. Addison Wesley, 1984.
- [Koh04] Michael Kohlhase. "Semantic Markup for TeX/IATeX". In: Mathematical User Interfaces Workshop 2004. Ed. by Paul Libbrecht. 2004. URL: http://www.activemath.org/~paul/MathUI04.
- [Koh06] Michael Kohlhase. OMDoc An open markup format for mathematical documents [Version 1.2]. LNAI 4180. Springer Verlag, Aug. 2006. URL: http://omdoc.org/pubs/omdoc1.2.pdf.
- [Koh08] Michael Kohlhase. "Using LaTeX as a Semantic Markup Format". In: Mathematics in Computer Science 2.2 (2008), pp. 279-304. URL: https://kwarc.info/kohlhase/papers/mcs08-stex.pdf.
- [Koh18a] Michael Kohlhase. cmath.sty: An Infrastructure for building Inline Content Math in sTeX. Tech. rep. 2018. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/cmath/cmath.pdf.
- [Koh18b] Michael Kohlhase. dcm.sty: An Infrastructure for marking up Dublin Core Metadata in LATEX documents. Tech. rep. 2018. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/stex/dcm/dcm.pdf.
- [Koh18c] Michael Kohlhase. hwexam.sty/cls: An Infrastructure for formatting Assignments and Exams. Self-documenting LATEX package. 2018. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/hwexam/hwexam.pdf.
- [Koh18d] Michael Kohlhase. metakeys.sty: A generic framework for extensible Metadata in IATEX. Tech. rep. 2018. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/metakeys/metakeys.pdf.
- [Koh18e] Michael Kohlhase. omdoc.sty/cls: Semantic Markup for Open Mathematical Documents in LATEX. Tech. rep. 2018. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/omdoc/omdoc.pdf.
- [Koh18f] Michael Kohlhase. omtext: Semantic Markup for Mathematical Text Fragments in LATEX. Tech. rep. 2018. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/omtext/omtext.pdf.
- [Koh18g] Michael Kohlhase. problem.sty: An Infrastructure for formatting Problems. Self-documenting LATEX package. 2018. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/problem/problem.pdf.

- [Koh18h] Michael Kohlhase. sproof.sty: Structural Markup for Proofs. Tech. rep. 2018. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/sproof/sproof.pdf.
- [Koh18i] Michael Kohlhase. sref.sty: Semantic Crossreferencing in LATEX. Self-documenting LATEX package. 2018. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/sref/sref.pdf.
- [Koh18j] Michael Kohlhase. statements.sty: Structural Markup for Mathematical Statements. Tech. rep. 2018. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/statements/statements.pdf.
- [Koh18k] Michael Kohlhase. structview.sty: Structures and Views in sT_EX. Tech. rep. 2018. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/structview/structview.pdf.
- [Koh18l] Michael Kohlhase. tikzinput: Selective Input of TIKZ Diagrams. Tech. rep. 2018. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/tikzinput/tikzinput.pdf.
- [Koh18m] Michael Kohlhase. workaddress.sty: An Infrastructure for marking up Dublin Core Metadata in LATEX documents. Tech. rep. 2018. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/stex/workaddress/workaddress.pdf.
- [L2O] GitHub repository. URL: https://github.com/sLaTeX/LaTeXML-Plugin-sTeX.
- [Lam94] Leslie Lamport. LaTeX: A Document Preparation System, 2/e. Addison Wesley, 1994.
- [LMH] MathHub Offline Authoring. URL: http://mathhub.info/help/offline-authoring.html (visited on 04/07/2016).
- [lmh] KWARC/lmh. URL: https://github.com/KWARC/lmh (visited on 05/15/2015).
- [LTX] Bruce Miller. LaTeXML: A LATEX to XML Converter. URL: http://dlmf.nist.gov/LaTeXML/ (visited on 03/12/2013).
- [MH] MathHub Online Authoring. URL: http://mathhub.info/help/online-authoring.html (visited on 04/07/2016).
- [MiK] MiKTEX...typesetting beautiful documents... URL: http://miktex.org (visited on 12/11/2012).
- [Pro07] LATEX Project. The LATEX project public license. Software License. 2007. URL: http://www.latex-project.org/lppl/.
- [RK13] Florian Rabe and Michael Kohlhase. "A Scalable Module System". In: Information & Computation 0.230 (2013), pp. 1-54. URL: http://kwarc.info/frabe/Research/mmt.pdf.
- [RV01] Alan Robinson and Andrei Voronkov, eds. *Handbook of Automated Reasoning*. Vol. I and II. Elsevier Science and MIT Press, 2001.

- [Sta+09] Heinrich Stamerjohanns et al. "MathML-aware article conversion from LATEX, A comparison study". In: Towards Digital Mathematics Library, DML 2009 workshop. Ed. by Petr Sojka. Masaryk University, Brno, 2009, pp. 109–120. URL: http://kwarc.info/kohlhase/papers/dml09-conversion.pdf.
- [sTeX] sTeX: A semantic Extension of TeX/LaTeX. URL: https://github.com/sLaTeX/sTeX (visited on 05/15/2015).
- [TL] $TeX\ Live.\ URL:\ http://www.tug.org/texlive/$ (visited on 12/11/2012).