System Description
sTeX3 – A LaTeX-based Ecosystem for Semantic/Active Mathematical Documents

Michael Kohlhase, Dennis Müller

Computer Science, FAU Erlangen-Nürnberg

Abstract. We report on sTeX3 – a complete redesign and reimplementation (using LaTeX3) from the ground up of the sTeX ecosystem for semantic markup of mathematical documents. Specifically, we present: *i*) The sTeX package that allows declaring semantic macros and provides a module system for organizing and importing semantic macros using logical identifiers. sTeX3 is a (now) standard LaTeX package with minimal dependencies and is compatible with arbitrary document classes and packages. *ii*) The RuSTeX system, an implementation of the core TeX-engine in Rust. It allows for converting arbitrary LaTeX-documents to XHTML– for sTeX3-documents enriched with semantic annotations based on the OMDoc ontology. *iii*) An Mmt integration: The RuSTeX-generated XHTML can be imported and served by the Mmt system for further semantic knowledge management services.

This paper uses sTeX3. The semantically annotated XHTML version of this paper is available at `https://tinyurl.com/cicm22stex`

1 Introduction and History

In the sTeX project [sLX], we explore how established communication and publication workflows – this mainly means LaTeX in Mathematics and theoretical sciences – can be extended semantically for computer support. The central element of this endeavour is the sTeX package [Koh08; sTeX] which allows to *semantically preload* LaTeX documents via special (semantic) macros.

sTeX documents can be processed by `pdflatex` in the usual way. Additionally, in sTeX1 they could initially be processed by LaTeXML [LTX], a LaTeX-to-XML transformer, using a dedicated sTeX plugin producing OMDoc [Koh06]. Unfortunately, this plugin was elaborate, implementing the OMDoc-specific behaviour via dedicated Perl bindings for the majority of sTeX-macros and was correspondingly difficult to maintain. It was also invasive with respect to the LaTeXML code base and quickly became incompatible with newer versions of LaTeXML. Furthermore, conversion to OMDoc required the usage of dedicated document classes, rendering sTeX incompatible with existing and established authoring workflows, and setting up sTeX to work in the first place was prohibitively difficult, involving manually changing core parameters of a user's TeX system.

Nevertheless, the sTeX package (and associated classes) have been used to produce extensive course materials (3000+ pages of slides and integrated narrative), ca. 2500 exercise/exam problems, and the SMGloM, a multilingual mathematical glossary [SMG], currently containing $\geq$ 2250 concepts in English (93%), German (71%) and Chinese (11%). This sTeX-**corpus**, together with the OMDoc format, have informed the development of the sTeX packages and document model. All sTeX content is available as *mathematical archives* [Hor+11] on `https://MathHub.info` and can be browsed on `https://mmt.beta.vollki.kwarc.info/:sTeX`.

While the original sTeX architecture and realization showed that semantic preloading of mathematical documents and the deployment of active documents based on this is possible given enough motivation, the technical/practical problems mentioned above quickly became a showstopper. Not surprisingly, the use of sTeX never quite gained much traction outside the authors' research group and collaborative projects. Additionally, the continuing development of the MMT system [RK13; MMT] over the last years similarly drove development of its own variant of the OMDoc format and ontology (MMT/OMDoc).

Consequently, we decided to rethink and reimplement sTeX from the ground up, using LaTeX3, with both the problems with sTeX1 and the developments of the MMT/OMDoc ontology in mind. The result is the sTeX3 package and system, which we present in this system description (extensive documentation available at [KM]).

Notably, this very document uses sTeX3 and its module system within the `llncs` document class, and is available (and compiles) on Overleaf at `https://www.overleaf.com/read/tcnwysdzthwx`. We occasionally refer to the source files available there for clarification. In this document, sTeX was configured such that every semantic macro generates a link to our document server, but it should be noted that this behavior can be fully customized.

2 The sTeX3 package

The design of the sTeX3 system was based on the following guiding principles:

1. *Ease of set-up*: The sTeX3 package should work with a vanilla, unmodified TeX/LaTeX system – e.g. a sufficiently recent TeX Live installation – without the need of changing any TeX-parameters and without any external software.
2. *Universality*: The sTeX3 package should be compatible with arbitrary TeX document classs, packages, and authoring workflows. Semantically annotating existing environments (*theorems*, *definitions*, *proofs* etc.) should not impact document layout: their layout should be fully customizable.
3. *No code duplication*: The functionality of sTeX3 macros and environments should be governed by the LaTeX-code of the package alone (as opposed to dedicated macro bindings for OMDoc export that implement the same functionality with a different output format) to help maintainability.
4. MMT-*completeness*: sTeX3 should be a full surface language for MMT/OMDoc.

Let us see how the current system is doing on these accounts.

1. Ease of set-up Indeed, sTeX3 now works with any unmodified TeX system with a LaTeX3 kernel later than February 2022. For older, but not too outdated LaTeX3 versions (up to TeX Live 2018 as running on Overleaf), the missing functionality can be easily added (in this document via the `stex-expl-compat` package).

2. Universality The sTeX3 package can be imported in the usual manner (via `\usepackage{stex}`) and only depends on three other packages, namely `ltxcmds`, `standalone` and `xspace`, all of which are ubiquitous, non-invasive and do not take package options that might lead to conflicts.

To allow for collaboration (e.g. via git) and compatibility with submission systems (e.g. arxiv.org) sTeX3 can "persist" all semantic macros and other module content into a `.sms`-file during compilation (similar to the `.toc`-file), which can be used in subsequent compilations, obviating the need for the (potentially many) original modules to be physically present. This file can then be put under version control or distributed alongside the document.

To be adaptable to document styles, sTeX3 determines the specific highlighting for symbols via four macros, which can be redefined, namely `\compemph{}`, `\symrefemph{}`, `\defemph{}` and `\varemph{}`. For this document, these are defined in `highlights.tex`.

While sTeX1 declared its own environments `definition`, `example`, `theorem` etc., doing so necessarily made sTeX1 incompatible with document classes that predefine these environments (like `llncs`), or a user's preferences. However, this was necessary to allow for providing additional semantic information, e.g. as in `\begin{definition}[for=foo,type=inductive]`.

In sTeX3, we instead use environments `sdefinition`, `sexample`, `sassertion` etc., that take care of the semantic information provided, but whose typesetting can be customized. For example, by setting `\stexpatchdefinition{\begin{definition}}{\end{definition}}`, every `\begin{sdefinition}[...]` will process the arguments provided, and then delegate to the `definition`-environment for layout and numbering. Analogously, `\stexpatchassertion[theorem]{\begin{theorem}}{\end{theorem}}` will delegate every `\begin{sassertion}[type=theorem,...]` to the `theorem`-environment.

3. No code duplication This principle lead to the following design choice: Rather than converting sTeX documents to OMDoc directly, we have the sTeX package insert semantic annotations into a non-`PDF` output format; e.g. XHTML. The package itself determines the full Mmt URIs for all symbols, governs the OpenMath syntax tree and introduces annotations via merely three macros that a "backend" of choice should provide:

- `\stex_annotate:nnn{key}{value}{code}` annotates `code` with `key=value` (e.g. by wrapping `code` in a `<span property="key" resource="value">...`).
- `\stex_invisible:n{code}` exports `code`, but hides it in the presentation (e.g. by setting `style="display:none"`).

– Lastly, `\begin{stex_annotate_env}{key}{value}` acts like `\stex_annotate:nnn{key}{value}{code}`, but as an environment.

The file `stex-backend-pdflatex.cfg` contains the implementations for these macros for the standard `pdflatex` backend.

4. MMT *completeness* with respect to the sTeX3 package is a complex issue – at least when trying to avoid code duplication – since the MMT/OMDOC ontology supplies very powerful representational primitives, and will therefore be treated in a regular companion paper submitted to CICM.

3 OMDoc and Mmt

sTeX-XHTML: In sTeX1, translating sTeX content to OMDOC was achieved directly via LaTeXML. In sTeX3, we instead translate sTeX content to XHTML, augmented with annotations via XML attributes corresponding to the OMDOC ontology. In principle, this workflow allows for a plurality of systems as translators, such as LaTeXML or TeX4ht. In practice, unfortunately, it has turned out to be difficult to preserve the intended attribute annotations using a current version of LaTeXML in math-mode, where they are most important. For now, we therefore implemented our own plain TeX-interpreter from the ground up using Rust, for converting LaTeX documents to XHTML. The resulting RusTeX software[1] uses a user's local LaTeX system, keeping the number of required primitives to implement to a reasonable minimum, and can therefore handle (in principle) arbitrary TeX code to the virtually same degree as the user's TeX system (`pdflatex`, to be precise), at the cost of (a priori) no special treatment of higher-level LaTeX macros (although RusTeX allows for providing dedicated bindings for these, too). MMT bundles and interfaces with RusTeX via the JNI to convert to XHTML using MMT's build system and cache sTeX modules across conversion tasks.

XHTML-MMT/OMDOC*:* Having obtained semantically annotated XHTML, we have implemented a new XHTML import in MMT to extract the semantic annotations and map them directly to the corresponding MMT/OMDOC concepts. In addition to thus converting sTeX content to OMDOC, the MMT system can host the generated XHTML in a semantically informed manner and offer the full suite of available knowledge management services for sTeX, up to and including type checking and inference.

4 Ongoing and Future Work

Having solved many of the previous problems surrounding sTeX1 that discouraged users from using sTeX, the most pressing issues now are related to finding,

[1] `https://github.com/slatex/RusTeX`

managing, and reusing *existing* sTeX content. We are therefore working on a dedicated IDE for sTeX in the form of a Language Server Protocol server and a plugin for VS Code that bundles the MMT system, offers convenient interfaces to interact with it, allows for searching available sTeX content (online and locally) and generally helps with semantically annotating documents.[2]

References

[Hor+11] Fulya Horozal et al. "Combining Source, Content, Presentation, Narration, and Relational Representation". In: *Intelligent Computer Mathematics*. Ed. by James Davenport et al. LNAI 6824. Springer Verlag, 2011, pp. 212–227. URL: https://kwarc.info/frabe/Research/HIJKR_dimensions_11.pdf.

[KM] Michael Kohlhase and Dennis Müller. *The sTeX3 Package Collection*. Tech. rep. URL: https://github.com/slatex/sTeX/blob/main/doc/stex-doc.pdf (visited on 04/24/2022).

[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.

[LTX] Bruce Miller. *LaTeXML: A LaTeX to XML Converter*. URL: http://dlmf.nist.gov/LaTeXML/ (visited on 03/12/2021).

[MMT] *MMT – Language and System for the Uniform Representation of Knowledge*. URL: https://uniformal.github.io/ (visited on 01/15/2019).

[RK13] Florian Rabe and Michael Kohlhase. "A Scalable Module System". In: *Information & Computation* 0.230 (2013), pp. 1–54. URL: https://kwarc.info/frabe/Research/mmt.pdf.

[sLX] *sLaTeX: An Ecosystem for Semantically Enhanced LaTeX*. URL: https://github.com/sLaTeX (visited on 03/11/2021).

[SMG] *SMGloM: A Semantic, Multilingual Terminology for Mathematics*. URL: http://smglom.mathhub.info (visited on 04/21/2014).

[sTeX] *sTeX: A semantic Extension of TeX/LaTeX*. URL: https://github.com/sLaTeX/sTeX (visited on 05/11/2020).

[2] Available at https://github.com/slatex/sTeX-IDE