Integrating sTeX into llncs.sty

Michael Kohlhase[0000−0002−9859−6337], Dennis Müller[0000−0002−4482−4912], and Jan Frederick Schaefer

Computer Science, FAU Erlangen-Nürnberg

Abstract. Paper shows how to integrate sTeX stuff into regular papers, e.g. via the `llncs.cls` and get them through submission systems like EasyChair.

1 Introduction

One of the useful new features of sTeX is that we can mix in its semantic annotation functionality into arbitrary document classes. This paper is an example of how this can be done. Indeed it uses the llncs class as we would submit it to a conference with LNCS or LNAI proceedings.

In Section 2 show how we can use the sTeX functionality to write an sTeX module. This presupposes that the author has cloned the necessary MathHub archives from `https://gl.mathhub.info`, say under the local path `/Users/kohlhase/localmh/MathHub`. One of the past hindrances was that it was virtually impossible to submit a paper that includes MathHub archives via a conference tool like EasyChair.org. Therefore the sTeX distribution contains a python script `mhlocalize` that copies a snapshot of the sTeX files from the MathHub archives into the current directory, so they can be zipped for submission with the paper sources.

2 The Meat of the Matter: An sTeX module in a Paper

Here we make use of the sTeX functionality: We initialize the preamble with

```
\documentclass{llncs}
\usepackage[mh,mathhub=/Users/kohlhase/localmh/MathHub]{smglom}
```

The first line contains the `llncs` document class and the second calls the `smglom` package from the sTeX distribution. We give it the `mathhub` option that allows to specify the system path to the local MathHub archives.

This is sufficient solitary authoring. For collaborative authoring, the approach of supplying the respective sTeX packages with a `path=` option will not work, since the path to the local MathHub home is system-dependent and can therefore not be committed. There are two ways of fixing this. One is to have an unversioned file `lmh.sty` with contents

```
\def\mathhub@path{/path/to/mathhub}
```

and add a `\usepackage{lmh}` before the `\usepackage[mh]{smglom}`. For the other see Section 3.

Either way, somewhere in the paper we can just write an sTeX module, here one about the expected utility of an action in a partially observable environment.

```
\begin{module}[id=exputil]
\importmhmodule[mhrepos=MiKoMH/AI,dir=rational-decisions/en]{Ramsey-thm}
\importmhmodule[mhrepos=MiKoMH/AI,dir=probabilistic-reasoning/en]{condprob}
\gimport[smglom/probability]{expectation}
\symdef{expectedUtilityOp}{EU}
\symdef{expectedUtility}[1]{\prefix\expectedUtilityOp{#1}}
\symdef{expectedUtilityGiven}[2]{\prefix\expectedUtilityOp{#1|#2}}
\symdef{agactres}[1]{R_{#1}}

\begin{definition}
The \trefii{expected}{utility} $\expectedUtilityGiven{a}{\mathbf{e}}$ of an
\trefi[agent-math]{action} $a$ (given evidence $\mathbf{e}$) is
\[\fundefeq{a,\mathbf{e}}{\expectedUtilityGiven{a}{\mathbf{e}}}
    {\SumInColl{s}\Omega
    {\realtimes[cdot]{\CondProb{\agactres{a}=s}{a,\mathbf{e}},\utilityof{s}}}}
\]
where $\agactres{a}$  is a \trefii[unconditional-prob]{random}{variable} whose
values are the results of performing $a$ in the current state.
\end{definition}
\end{module}
```

The first line introduces a module, the next three lines make available the semantic macros introduced by two modules in the sTeX-based AI lecture hosted at `https://gl.mathhub.info/MiKoMH/AI` and another from the SMGloM glossary archive at `https://gl.mathhub.info/smglom/probability`. The next three lines introduce semantic macros we use in the definition itself. With all the semantic macros (included and directly provided), we can mark up the formulae and technical terms in the usual sTeX way. This gives the following result:[1]

EdN:1

Definition 21 The expected utility $EU(a|\mathbf{e})$ of an action a (given evidence $\mathbf{e}$) is

$$EU(a|\mathbf{e}) := \sum_{s\in\Omega} P(R_a = s \mid a, \mathbf{e}) \cdot U(s)$$

where R_a is a random variable whose values are the results of performing a in the current state.

Note that we are taking

3 Localizing the Papers

To make the paper sources into a self-contained folder, the sTeX tools distribution at `https://github.com/sLaTeX/lmhtools` provides the python script

[1] EdNote: there is still a problem here, the `definition` environment does not produce a good label.

`localize` that copies (the necessary parts) of MathHub archives into the `lmh` subdir in the current directory.

```
python3 localize --localmh /path/to/MathHub paper.tex
```

extracts all archive inclusions from the sTeX document with main file `paper.tex` and recursively copies (or updates) all used files into the local `lmh` directory respecting the archive structure. This has two applications:

1. **Collaborative Authoring in GIT**: Instead of requiring all co-authors have the MathHub archives cloned, you can commit the `localize`d snapshot and use that. This has the advantage that the late-binding problem inherent in using central archives is mitigated as all co-authors use the same `lmh` folder. And it is sufficient that one co-author has the MathHub archives cloned, and `localize`s it.
2. **Submission via a Conference or Journal Submission system**: For submission, we need a self-contained zip file which we can now create by `localize`ing and `zip`ping.

4 Conclusion

With the new functionality developed in 2020, mixing sTeX into generic LaTeX document classes is quite feasible. This considerably lowers the entry barrier to using sTeX from the state where you had to use the sTeX document classes.

One of the advantages of sTeX is the availability of well-thought out modules with semantic macros for many areas of mathematics, which can be included directly from a local clone of the archives. For collaborative authoring and paper submission we add a tool for localizing the MathHub archives.