MathHub Support for STEX*

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

November 18, 2015

Abstract

The sref package is part of the STEX collection, a version of TEX/LATEX that allows to markup TEX/LATEX documents semantically without leaving the document format, essentially turning TEX/LATEX into a document format for mathematical knowledge management (MKM).

The ${\tt mathhub}$ packages extend SIEX with support for the MathHub.info portal

Contents

1	Intr	roduction	3
2	The	e User Interface	3
	2.1	Package Options	3
	2.2	modules-mh: MH Variants for Modules	3
	2.3	omtext-mh: MH Variants for OMText	4
	2.4	statements-mh: MH Variants for Statements	4
	2.5	smultiling-mh: MH Variants for Multilinguality	4
	2.6	structview-mh: MH Variants for Structures and Views	4
	2.7	mikoslides-mh: Support for MiKo Slides	5
	2.8	problem-mh: Support for Problems	5
	2.9	hwexam-mh: Support for Assignments	5
3	Lim	itations	6
4	Imp	plementation	7
	4.1	General Infrastructure	7
	4.2	modules-mh: MH Variants for Modules	8
	4.3	omtext-mh: MH Variants for OMText	11
	4.4	statements-mh: MH Variants for Statements	12

^{*}Version v1.0 (last revised 2015/11/04)

4.5	smultiling-mh: MH Variants for Multilinguality	12
4.6	structview-mh: MH Variants for Structures and Views	14
4.7	mikoslides-mh: Support for MiKo Slides	17
4.8	problem-mh: Support for Problems	17
4.9	hwexam-mh: Support for Assignments	18
4.10	tikzinput-mh: Support for Assignments	19
4.11	Finale	20

1 Introduction

Much of the STEX content is hosted on MathHub (http://MathHub.info), a portal and archive for flexiformal mathematics. MathHub offers GIT repositories (public and private escrow) for mathematical documentation projects, online and offline authoring and document development infrastructure, and a rich, interactive reading interface. The modules package supports repository-sensitive operations on MathHub.

Note that MathHub has two-level repository names of the form $\langle group \rangle / \langle repo \rangle$, where $\langle group \rangle$ is a MathHub-unique repository group and $\langle repo \rangle$ a repository name that is $\langle group \rangle$ -unique. The file and directory structure of a repository is arbitrary – except that it starts with the directory source because they are Math Archives in the sense of [Hor+11]. But this structure can be hidden from the STEX author with MathHub-enabled versions of the STEX macros, which are defined in this package.

Caveat if you want to use the MathHub support macros (let's call them mh-variants), then every time a module is imported or a document fragment is included from another repos, the mh-variant \importmhmodule must be used, so that the "current repository" is set accordingly. To be exact, we only need to use mh-variants, if the imported module or included document fragment use mh-variants.

2 The User Interface

2.1 Package Options

none so far

2.2 modules-mh: MH Variants for Modules

\importmhmodule

The importmhmodule macro is a variant of \importmodule with repository support. Instead of writing

\defpath{MathHub}{/user/foo/lmh/MathHub}
\importmodule[load=\MathHub{fooMH/bar/source/baz/foobar}]{foobar}

we can simply write (assuming that \MathHub is defined as above)

\importmhmodule[repos=fooMH/bar,path=baz/foobar]{foobar}

Note that the **\importmhmodule** form is more semantic, which allows more advanced document management features in MathHub.

If baz/foobar is the "current module", i.e. if we are on the MathHub path ...MathHub/fooMH/bar..., then stating the repository in the first optional argument is redundant, so we can just use

\importmhmodule[path=baz/foobar]{foobar}

if no file needs to loaded, \importmhmodule is the same as \importmodule.

\mhcurrentrepos

Of course, neither LATEX nor LATEXMLknow about the repositories when they are called from a file system, so we can use the \mhcurrentrepos macro to tell them. But this is only needed to initialize the infrastructure in the driver file. In particular, we do not need to set it in in each module, since the \importmhmodule macro sets the current repository automatically.

\usemhmodule \mhinputref \mhinput The \usemhmodule is the analog to \usemodule.

For this, the modules package supplies the mh-variants $\mbox{mhinput}$ and $\mbox{mhinput}$ of the \mbox{input} macro introduced above and normal \mbox{ETEX} \mbox{input} macro.

2.3 omtext-mh: MH Variants for OMText

\mhcgraphics

The \mhcgraphics macro is a variant of \mycgraphics with repository support. Instead of writing

```
\defpath{MathHub}{/user/foo/lmh/MathHub}
\mycgraphics{\MathHub{fooMH/bar/source/baz/foobar}}
```

we can simply write (assuming that \MathHub is defined as above)

\mhcgraphics[fooMH/bar]{baz/foobar}

Note that the \mhcgraphics form is more semantic, which allows more advanced document management features in MathHub.

2.4 statements-mh: MH Variants for Statements

this only provides \usembrocab a variant of \usevocab (which might go away at some time)

2.5 smultiling-mh: MH Variants for Multilinguality

1 2

2.6 structview-mh: MH Variants for Structures and Views

3

EdN:1

EdN:3

2.7 mikoslides-mh: Support for MiKo Slides

\mhframeimage

The \mhframeimage macro is a variant of \frameimage with repository support. Instead of writing

\defpath{MathHub}{/user/foo/lmh/MathHub}
\frameimage{\MathHub{fooMH/bar/source/baz/foobar}}

we can simply write (assuming that \MathHub is defined as above)

\mhframeimage[fooMH/bar]{baz/foobar}

Note that the \mhframeimage form is more semantic, which allows more advanced document management features in MathHub.

If baz/foobar is the "current module", i.e. if we are on the MathHub path ...MathHub/fooMH/bar..., then stating the repository in the first optional argument is redundant, so we can just use

\mhframeimage{baz/foobar}

2.8 **problem-mh**: Support for Problems

\includemhproblem

The \includemhproblem macro is a variant of \importmodule with repository support. Instead of writing

\defpath{MathHub}{/user/foo/lmh/MathHub}
\includeproblem[pts=7]{\MathHub{fooMH/bar/source/baz/foobar}}

we can simply write (assuming that \MathHub is defined as above)

\includemhproblem[fooMH/bar]{baz/foobar}

Note that the \importmhproblem form is more semantic, which allows more advanced document management features in MathHub.

2.9 hwexam-mh: Support for Assignments

\includemhassignment

The \includemhassignment macro is a variant of \includeassignment with repository support. Instead of writing

\defpath{MathHub}{/user/foo/lmh/MathHub}
\includeassignment[pts=7]{\MathHub{fooMH/bar/source/baz/foobar}}

we can simply write (assuming that \MathHub is defined as above)

\includemhassignment[fooMH/bar]{baz/foobar}

 $^{^1\}mathrm{Ed}\mathrm{Note}\colon$ needs to be documented

 $^{^2\}mathrm{EdNote}$: mhmodsig seems to be missing what happened?

 $^{^3\}mathrm{EdNote}$: needs to be documented

3 Limitations

In this section we document known limitations. If you want to help alleviate them, please feel free to contact the package author. Some of them are currently discussed in the STEX GitHub repository [sTeX].

1. none reported yet.

4 Implementation

The sref package generates two files: the LATEX package (all the code between <code>*package</code>) and <code>\/package</code>) and the LATEXML bindings (between <code>*ltxml</code>) and <code>\/ltxml</code>). We keep the corresponding code fragments together, since the documentation applies to both of them and to prevent them from getting out of sync.

We first set up header information for the LATEXML binding files an the base package.

```
1 (*Itxml | modules.ltxml | structview.ltxml | omtext.ltxml | statements.ltxml | smultiling.ltxml | mikoslides.ltxml | probler
2 # -*- CPERL -*-
3 package LaTeXML::Package::Pool;
4 use strict;
5 use LaTeXML::Package;
6 use LaTeXML::Util::Pathname;
7 (/ltxml | modules.ltxml | structview.ltxml | omtext.ltxml | statements.ltxml | smultiling.ltxml | mikoslides.ltxml | probler
Then we need to set up the packages by requiring the metakeys pack-
age [Koh15] to be loaded (in the right version).
9 (*package)
10 \RequirePackage{keyval}
11 (/package)
12 (*ltxml)
13 RequirePackage('keyval');
14 (/ltxml)
```

4.1 General Infrastructure

\mhcurrentrepos \@mhcurrentrepos \mhcurrentrepos is used to initialize the current repository. If the repos has changed, it writes a call to the internal macro \@mhcurrentrepos for the aux file and calls it. So that the \importmodule calls there work with the correct repos.

```
15 (*package)
16 \newcommand\mhcurrentrepos[1]{%
    \ensuremath{\texttt{def}\@\text{test{#1}}\%}
17
    \ifx\@test\mh@currentrepos% if new dir = old dir
18
19
      \relax% no need to change
20
      \protected@write\@auxout{}{\string\@mhcurrentrepos{#1}}%
21
22
    \@mhcurrentrepos{#1}% define mh@currentrepos
23
24 }%
25 \newcommand\@mhcurrentrepos[1]{\edef\mh@currentrepos{#1}}%
26 (/package)
27 (*ltxml)
28 DefMacro('\mhcurrentrepos{}','\@mhcurrentrepos{#1}');
29 DefMacro('\@mhcurrentrepos{#1}\0@mhcurrentrepos{#1}');
30 DefConstructor('\@@mhcurrentrepos{}','',
    afterDigest => sub{ AssignValue('current_repos',ToString($_[1]->getArg(1)),'global'); } );
32 (/ltxml)#$
```

```
\libinput the \libinput macro inputs from the lib directory of the MathHub repository
           or the meta-inf/lib repos of the group.
           33 (*package)
           34 \def\modules@@first#1/#2;{#1}
           35 \newcommand\libinput[1]{\def\@libfile{\MathHub{\mh@currentrepos/lib/#1}}%
           36 \IfFileExists{\@libfile}{\input\@libfile}%
           37 {\edef\@@group{\expandafter\modules@@first\mh@currentrepos;}
           38 \edef\@inffile{\MathHub{\@@group/meta-inf/lib/#1}}
           39 \IfFileExists{\@inffile}{\input{\@inffile}}%
           40 {\PackageError{modules}
               {Library file missing, cannot input #1\MessageBreak%
           42
                 Both \@libfile.tex\MessageBreak and \@inffile.tex\MessageBreak do not exist}%
               {Check whether the file name is correct}}}}
           43
           44 (/package)
           45 (*ltxml)
           46 DefMacro('\modules@@first#1/#2;','#1');
           47 DefMacro('\libinput {}', sub{
           48
                 my ($gullet, $name) = @_;
                 $name = ToString($name);
           49
                 #Relative paths for recursive search
           50
           51
                 my $FIRSTLIB = ('/../../lib');
                 my $SECONDLIB = ('/../../meta-info/lib');
           53
                 my $file = pathname_find($name, types => ['tex'], paths =>[$FIRSTLIB]);
                 $file = pathname_find($name, types=>['tex'], paths=>[$SECONDLIB]) unless $file;
           54
                 # Singal error if the file cannot be found
           55
                 LaTeXML::Package::InputContent($file, noerror=>1); });
           56
           57 (/ltxml)
```

4.2 modules-mh: MH Variants for Modules

We set up package options and pass them on to the modules package, which we also load.

```
58 \( \section \) \ \text{FrovidesPackage{modules-mh}[2015/11/04 v1.0 MathHub support for the sTeX modules package] } \( \text{60 \RequirePackage{mathhub}} \) \( \lambda \) \( \lambda \) \( \lambda \) \( \text{modules} \rangle \) \( \lambda \) \( \text{modules.ltxml} \rangle \) \( \text{63 RequirePackage('mathhub');} \) \( \text{64 } \lambda / \) \( \text{modules.ltxml} \rangle \)
```

\importmhmodule

The $\infty = value \ list$] {module} saves the current value of \mh@currentrepos in a local macro \mh@@repos, resets \mh@currentrepos to the new value if one is given in the optional argument, and after importing resets \mh@currentrepos to the old value in \mh@@repos. We do all the \ifx comparison with an \expandafter, since the values may be passed on from other key bindings. Parameters will be passed to \importmodule.

```
65 (*modules)
```

66 \srefaddidkey{importmhmodule}%

```
67 \addmetakey{importmhmodule}{repos}% saves the repo's path. E.g: smglom/numberfield
68 \addmetakey{importmhmodule}{path}% saves the module name. E.g: naturalnumbers
69 \addmetakey[sms]{importmhmodule}{ext}\% saves the extension: E.g. tex
70 \addmetakey[false]{importmhmodule}{conservative}[true]%
71 \newcommand\importmhmodule[2][]{%
     \metasetkeys{importmhmodule}{#1}%
73
     \ifx\importmhmodule@path\@empty% if module name is not set
74
       \importmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
     \else%
75
       \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
76
       \ifx\importmhmodule@repos\@empty% if in the same repos
77
         \relax% no need to change mh@currentrepos, i.e, current dirctory.
 78
 79
       \else%
         \mhcurrentrepos{\importmhmodule@repos}% change it.
80
81
       %,{\importmodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},
82
       ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
83
       \mhcurrentrepos{\mh@@repos}% after importing, reset to old value
84
     \fi%
85
86
     \ignorespaces%
87 }%
88 (/modules)
89 (*modules.ltxml)
90 DefKeyVal('importmhmodule', 'id', 'Semiverbatim');
91 DefKeyVal('importmhmodule', 'repos', 'Semiverbatim');
92 DefKeyVal('importmhmodule', 'path', 'Semiverbatim');
93 DefKeyVal('importmhmodule', 'ext', 'Semiverbatim');
94 DefKeyVal('importmhmodule', 'conservative', 'Semiverbatim');
95 DefConstructor('\importmhmodule OptionalKeyVals:importmhmodule {}',
          "<omdoc:imports "
96
          . "from='?%GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))()###2'"
97
98
                   . "?&defined(&GetKeyVal(#1,'conservative'))(load='&GetKeyVal(#1,'conservative')'
99
      afterDigest => \&importMHmoduleI);
100
101 sub importMHmoduleI {
    my ($stomach, $whatsit) = @_;
102
    my $keyval = $whatsit->getArg(1);
103
     my $id = $whatsit->getArg(2);
104
     if ($keyval) {
105
       my $repos = ToString($keyval->getValue('repos'));
106
107
       my $path = ToString($keyval->getValue('path'));
       my $current_repos = LookupValue('current_repos');
108
       if (!$repos) { # Use the implicit current repository
109
110
         $repos = $current_repos; }
       my $defpaths = LookupValue('defpath');
111
112
       my $load_path = ($$defpaths{MathHub}).$repos.'/source/'.$path;
113
       $keyval->setValue('load',$load_path);
114
       AssignValue('current_repos' => $repos, 'global');
       importmoduleI($stomach,$whatsit);
115
       AssignValue('current_repos' => $current_repos, 'global'); }
116
```

```
117
                      importmoduleI($stomach,$whatsit); }
              118
                   return; }
              119
              120
              121 DefConstructor('\importMHmoduleI OptionalKeyVals:importmhmodule {}', '',
                     afterDigest=> \&importMHmoduleI );#$
              123 \langle \mathsf{/modules.ltxml} \rangle
               and now the analogs
\usemhmodule
              124 (*modules)
              125 \newcommand\usemhmodule[2][]{%
                    \metasetkeys{importmhmodule}{#1}%
              126
              127
                    \ifx\importmhmodule@path\@empty%
              128
                      \usemodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
              129
                    \else%
              130
                      \edef\mh@@repos{\mh@currentrepos}%
                      \ifx\importmhmodule@repos\@empty%
              131
                      \else%
              132
              133
                        \mhcurrentrepos{\importmhmodule@repos}%
              134
              135
                      \usemodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodule@
                      \mhcurrentrepos\mh@@repos%
              136
              137
                    \fi%
                    \ignorespaces%
              138
              139 }%
              140 (/modules)
              141 (*modules.ltxml)
              142 DefConstructor('\usemhmodule OptionalKeyVals:importmhmodule {}',
                     "<omdoc:uses from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))()###</pre>
                     afterDigest => \&importMHmoduleI);
              144
              145 (/modules.ltxml)
\mhinputref
              146 \(\rangle\) modules.ltxml\(\rangle\) RawTeX('
              147 (*modules | modules.ltxml)
              148 \newcommand\mhinputref[2][]{%
                   \def\@repos{#1}%
              149
                   \edef\mh@currentrepos}%
              150
                   \verb|\ifx@repos@empty|| % \\
              151
              152
                   \else%
              153
                      \mhcurrentrepos{#1}%
              154
                    \inputref{\MathHub{\mh@currentrepos/source/#2}}%
              155
                    \mhcurrentrepos\mh@@repos%
              156
                    \ignorespaces%
              157
              158 }%
              159 (/modules | modules.ltxml)
              160 (modules.ltxml)');
```

```
161 (*modules)
                             162 \let\mhinput\mhinputref%
                             163 (/modules)
                                              omtext-mh: MH Variants for OMText
                               4.3
                               We set up package options and pass them on to the omtext package, which we
                               also load.
                             164 (*omtext)
                             165 \ProvidesPackage{omtext-mh}[2015/11/04 v1.0 MathHub support for the sTeX omtext package]
                             166 \RequirePackage{mathhub}
                             167 (/omtext)
                             168 (*omtext.ltxml)
                             169 RequirePackage('mathhub');
                             170 (/omtext.ltxml)
\mh*graphics Use the current value of \mh@currentrepos or the value of the mhrepos key if it
                               is given in \my*graphics.
                             171 (*omtext)
                             172 \def\Gin@mhrepos{}
                             173 \define@key{Gin}{mhrepos}{\csxdef\Gin@mhrepos{#1}}
                             174 \newcommand\mhgraphics[2][]{\setkeys{Gin}{#1}%
                             175 \edef\mh@@repos{\mh@currentrepos}%
                             176 \ifx\Gin@mhrepos\@empty\mygraphics[#1]{\MathHub{\mh@currentrepos/source/#2}}%
                             177 \else\mygraphics[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
                             178 \def\Gin@mhrepos{}\mhcurrentrepos\mh@@repos}
                             179 \verb| newcommand \verb| mhcgraphics[2][]{\end{center}} | 179 \verb| newcommand \verb| new
                             180 \newcommand\mhbgraphics[2][]{\fbox{\mhgraphics[#1]{#2}}}
                             181 \newcommand\mhcbgraphics[2][]{\begin{center}\fbox{\mhgraphics[#1]{#2}}\end{center}}
                             182 (/omtext)
                             183 (*omtext.ltxml)
                             184 sub mhgraphics {
                                        my ($gullet,$keyval,$arg2) = @_;
                             185
                                         my $repo_path;
                             186
                             187
                                         if ($keyval) {
                                              $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
                             188
                                        if (! $repo_path) {
                             189
                                              $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
                             190
                             191
                                         else {
                                              $keyval->setValue('mhrepos',undef); }
                             192
                                        my $mathhub_base = ToString(Digest('\MathHub{}'));
                                        my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
                             return Invocation(T_CS('\@includegraphicx'), $keyval, T_OTHER($finalpath)); }#$
                             196 DefKeyVal('Gin', 'mhrepos', 'Semiverbatim');
                             197 DefMacro('\mhgraphics OptionalKeyVals:Gin {}', \&mhgraphics);
                             198 DefMacro('\mhcgraphics []{}','\begin{center}\mhgraphics[#1]{#2}\end{center}');
                             199 DefMacro('\mhbgraphics []{}','\fbox{\mhgraphics[#1]{#2}}');
```

\mhinput

200 (/omtext.ltxml)

4.4 statements-mh: MH Variants for Statements

We set up package options and pass them on to the statements package, which we also load.

```
201 \( *statements \)
202 \ProvidesPackage{statements-mh}[2015/11/04 v1.0 MathHub support for the sTeX statements package 203 \RequirePackage{mathhub}
204 \( /statements \)
205 \( *statements.ltxml \)
206 \( RequirePackage('mathhub'); \)
207 \( /statements.ltxml \)
208 \( *statements \)
209 \\ let\usemhvocab=\usemhmodule \)
210 \( /statements \)
211 \( *statements.ltxml \)
212 \( DefMacro('\usemhvocab', '\usemhmodule'); \)
213 \( /statements.ltxml \)
```

4.5 smultiling-mh: MH Variants for Multilinguality

We set up package options and pass them on to the smultiling package, which we also load.

```
we also load.
214 \(*smultiling\)
215 \ProvidesPackage{smultiling-mh}[2015/11/04 v1.0 MathHub support for the sTeX smultiling package
216 \RequirePackage{mathhub}
217 \(\smultiling\)
218 \(*smultiling.ltxml\)
219 RequirePackage('mathhub');
220 \(\smultiling.ltxml\)

mhmodnl:*

221 \(*smultiling\)
222 \addmetakey{mhmodnl}{repos}
223 \addmetakey{mhmodnl}{path}
224 \addmetakey*{mhmodnl}{title}
```

```
223 \addmetakey\{mhmodnl}\{path\}
224 \addmetakey*\{mhmodnl}\{title\}
225 \addmetakey*\{mhmodnl}\{creators\}
226 \addmetakey*\{mhmodnl}\{creators\}
227 \addmetakey*\{mhmodnl}\{creators\}
228 \addmetakey\{mhmodnl}\{srccite\}
228 \addmetakey\{primary\{mhmodnl\}\{ps\}
229 \square\{smultiling\}
230 \square\{smultiling\}
231 \DefKeyVal('mhmodnl','title','Semiverbatim');
232 \DefKeyVal('mhmodnl','repos','Semiverbatim');
233 \DefKeyVal('mhmodnl','path','Semiverbatim');
234 \DefKeyVal('mhmodnl','creators','Semiverbatim');
235 \DefKeyVal('mhmodnl','contributors','Semiverbatim');
236 \DefKeyVal('mhmodnl','primary','Semiverbatim');
237 \square\{smultiling.ltxml\}
```

mhmodnl The mhmodnl environment is just a layer over the module environment and the \importmhmodule macro with the keys and language suitably adapted.

```
238 (*smultiling)
239 \newenvironment{mhmodnl}[3][]{\metasetkeys{mhmodnl}{#1}%
240 \def\@test{#1}\ifx\@test\@empty\begin{module}[id=#2.#3]\else\begin{module}[id=#2.#3,#1]\fi%
241 \edef\@repos{\ifx\mhmodnl@repos\@empty\mh@currentrepos\else\mhmodnl@repos}
242 \if@langfiles\importmhmodule[repos=\@repos,load=#2,ext=tex]{#2}\else
243 \ ifx\mhmodnl@load\\\end{empty} import\module {#2}\else\import\module [ext=tex,load=\mhmodnl@load] {\#2}\fi%\mhmodnl@load\end{empty} import\module {$\#2$}\else\mhmodnl@load\end{empty} import\mhmodnl@load\end{empty} import\end{empty} import\mhmodnl@load\end{empty} imp
244 \fi}
245 {\end{module}}
246 (/smultiling)
247 (*smultiling.ltxml)
248 DefEnvironment('{mhmodnl} OptionalKeyVals:mhmodnl {}{}', and the state of the contract of the state of the contract of th
                                      "?#excluded()(<omdoc:theory xml:id='#2.#3' >"
249
                                                   "?&defined(&GetKeyVal(#1,'creators'))(<dc:creator>&GetKeyVal(#1,'creators')</dc:cr
250
251
                                                   "?&defined(&GetKeyVal(#1,'title'))(<dc:title>&GetKeyVal(#1,'title')</dc:title>)()"
252
                                                   "?&defined(&GetKeyVal(#1,'contributors'))(<dc:contributor>&GetKeyVal(#1,'contribut
253
                                                   "<omdoc:imports from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'</pre>
                                                  "#bodv"
254
                                         "</omdoc:theory>)",
255
256
               afterDigestBegin=>sub {
257
                      my ($stomach, $whatsit) = @_;
258
                      my $keyval = $whatsit->getArg(1);
259
                     my $signature = ToString($whatsit->getArg(2));
                      my $language = ToString($whatsit->getArg(3));
260
                      my $repos = ToString(GetKeyVal($keyval,'torepos'));
261
262
                      my $current_repos = LookupValue('current_repos');
263
                      if (!$repos) { $repos = $current_repos; }
264
                      my $defpaths = LookupValue('defpath');
                      my $load_path = ($$defpaths{MathHub}).$repos.'/source/'.$signature;
265
266
                      if ($keyval) {
267
                            # If we're not given load, AND the langfiles option is in effect,
268
                            # default to #2
269
270
                            if ((! $keyval->getValue('path')) && (LookupValue('smultiling_langfiles'))) {
271
                                   $keyval->setValue('load',$load_path); }
272
                            # Always load a TeX file
273
                            $keyval->setValue('ext','tex');
                            $keyval->setValue('id', "$signature.$language"); }
274
                      module_afterDigestBegin(@_);
275
276
                      importmoduleI(@_);
277
                      return; },
                afterDigest=>sub {
278
279
                      module_afterDigest(@_); });
280 (/smultiling.ltxml)%$
```

mhviewsig The mhviewsig environment is just a layer over the mhview environment with the keys suitably adapted.

281 \(\smultiling.ltxml\)\(\RawTeX(')

```
283 \newenvironment{mhviewsig}[4][]{\def\@test{#1}\ifx\@test\@empty%
                   284 \begin{mhview}[id=#2,ext=tex]{#3}{#4}\else%
                   285 \begin{mhview}[id=#2,#1,ext=tex]{#3}{#4}\fi}
                   286 {\end{mhview}}
         mhviewnl The mhviewnl environment is just a layer over the mhviewsketch environment
                    with the keys and language suitably adapted.<sup>4</sup>
                   287 \newenvironment{mhviewnl}[5][]{\def\@test{#1}\ifx\@test\@empty%
                   288 \begin{mhviewsketch}[id=#2.#5,ext=tex]{#3}{#4}\else%
                   289 \begin{mhviewsketch}[id=#2.#5,#1,ext=tex]{#3}{#4}\fi}
                   290 {\end{mhviewsketch}}
                   291 \langle /\text{smultiling} \mid \text{smultiling.ltxml} \rangle
                   292 (smultiling.ltxml)');
                           structview-mh: MH Variants for Structures and Views
                    4.6
                    We set up package options and pass them on to the structview package, which
                    we also load.
                   293 (*structview)
                   294 \ProvidesPackage{structview-mh}[2015/11/04 v1.0 MathHub support for the sTeX structview package
                   295 \RequirePackage{mathhub}
                   296 (/structview)
                   297 (*structview.ltxml)
                   298 RequirePackage('mathhub');
                   299 (/structview.ltxml)
importmhmodulevia
                   300 \(\structview.ltxml\)\(\RawTeX(')
                   301 (*structview | structview.ltxml)
                   302 \newenvironment{importmhmodulevia}[3][]{%
                        \ifmod@show\par\noindent importing module #2 via \@@doit\fi
                   304
                   305 }{%
                         \aftergroup\@@doit\ifmod@show end import\fi%
                   308 (/structview | structview.ltxml)
                   309 (structview.ltxml),;
                   310 (*structview)
                   311 \srefaddidkey{mhview}
                   312 \addmetakey{mhview}{display}
                   313 \addmetakey{mhview}{creators}
                   314 \addmetakey{mhview}{contributors}
                   315 \addmetakey{mhview}{srccite}
                   316 \addmetakey*{mhview}{title}
                   317 \addmetakey{mhview}{fromrepos}
                       ^4\mathrm{EdNote}: MK: we have to do something about the if@langfiles situation here. But this is
```

282 (*smultiling | smultiling.ltxml)

EdN:4

```
318 \addmetakey{mhview}{torepos}
       319 \addmetakey{mhview}{frompath}
       320 \addmetakey{mhview}{topath}
       321 \addmetakey[sms]{mhview}{ext}
       322 (/structview)
       323 (*structview.ltxml)
       324 DefKeyVal('mhview','id','Semiverbatim');
       325 DefKeyVal('mhview','display','Semiverbatim');
       326 DefKeyVal('mhview', 'creators', 'Semiverbatim');
       327 DefKeyVal('mhview','contributors','Semiverbatim');
       328 DefKeyVal('mhview', 'srccite', 'Semiverbatim');
       329 DefKeyVal('mhview','title','Semiverbatim');
       330 DefKeyVal('mhview', 'fromrepos', 'Semiverbatim');
       331 DefKeyVal('mhview', 'torepos', 'Semiverbatim');
       332 DefKeyVal('mhview','frompath','Semiverbatim');
       333 DefKeyVal('mhview', 'topath', 'Semiverbatim');
       334 DefKeyVal('mhview','ext','Semiverbatim');
       335 (/structview.ltxml)
mhview the MathHub version
       336 \ \langle *structview \rangle
       337 \newenvironment{mhview}[3][]{% keys, from, to
             \metasetkeys{mhview}{#1}%
       339
             \sref@target%
             \begin{@mhview}{#2}{#3}%
       340
             \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
       341
       342 }{%
             \end{@mhview}%
       343
             \ignorespaces%
       344
       345 }%
       346 \ifmod@show\surroundwithmdframed{mhview}\fi
       347 (/structview)
       348 (*structview.ltxml)
       349\;DefMacroI(T_CS('\begin{mhview}'),'OptionalKeyVals:mhview {}{}', sub {}
            my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
       350
       351
             my $from = ToString(Digest($from_arg));
            my $to = ToString(Digest($to_arg));
             AssignValue(from_module => $from);
       353
             AssignValue(to_module => $to);
       354
            my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
       355
            my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
       356
            my $repos = LookupValue('current_repos');
       357
            my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
            my $to_path = ToString(GetKeyVal($keyvals, 'topath'));
       359
            my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
       360
             $ext = 'sms' unless $ext;
       361
            my $current_repos = LookupValue('current_repos');
       362
            if (!$from_repos) { $from_repos = $current_repos; }
       363
       364
            if (!$to_repos) { $to_repos = $current_repos; }
       365
            return (
```

```
Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
             366
             367
                     Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
                     Invocation(T_CS('\begin{viewenv}'), $keyvals, $from_arg, $to_arg) -> unlist
             368
                  );
             369
             370 });
             371 DefMacroI('\end{mhview}',undef,'\end{viewenv}');
             372 (/structview.ltxml)
     Omhview The Omhview does the actual bookkeeping at the module level.
             373 (*structview)
             374 \newenvironment{@mhview}[2]{%from, to
                  \importmhmodule[repos=\mhview@fromrepos,path=\mhview@frompath,ext=\mhview@ext]{#1}%
                  \importmhmodule[repos=\mhview@torepos,path=\mhview@topath,ext=\mhview@ext]{#2}%
             377 }{}%
             378 (/structview)
mhviewsketch The mhviewsketch environment behaves like mhview, but only has text contents.
             379 (*structview)
             380 \newenvironment{mhviewsketch}[3][]{%
                   \metasetkeys{mhview}{#1}%
             381
                   \sref@target%
             382
                   \begin{@mhview}{#2}{#3}%
             383
                   \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
             384
             385 }{%
                   \end{@mhview}%
             386
                  \ignorespaces%
             387
             388 }%
             389 \ifmod@show\surroundwithmdframed{mhviewsketch}\fi
             390 (/structview)
             391 (*structview.ltxml)
             392 DefMacroI(T_CS('\begin{mhviewsketch}'),'OptionalKeyVals:mhview {}{}', sub {
                  my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
                  my $from = ToString(Digest($from_arg));
             394
                  my $to = ToString(Digest($to_arg));
             395
                  my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
             396
                  my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
             397
                  my $repos = LookupValue('current_repos');
                  my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
             399
                  my $to_path = ToString(GetKeyVal($keyvals, 'topath'));
             400
                  my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
             401
                  $ext = 'sms' unless $ext;
             402
                  my $current_repos = LookupValue('current_repos');
             403
                  if (!$from_repos) { $from_repos = $current_repos; }
                  if (!$to_repos) { $to_repos = $current_repos; }
             405
             406
                  return (
                     Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
             407
                     Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
             408
                     Invocation(T_CS('\begin{viewsketchenv}'), $keyvals, $from_arg, $to_arg)->unlist
             409
             410 );
             411 });
```

```
412 DefMacroI('\end{mhviewsketch}',undef,'\end{viewsketchenv}'); 413 \langlestructview.ltxml\rangle
```

4.7 mikoslides-mh: Support for MiKo Slides

```
We set up package options and pass them on to the mikoslides package, which
                we also load.
               414 (*mikoslides)
               415 \ProvidesPackage{mikoslides-mh}[2015/11/04 v1.0 MathHub support for the sTeX mikoslides package
               416 \RequirePackage{mathhub}
               417 (/mikoslides)
               418 (*mikoslides.ltxml)
               419 RequirePackage('mathhub');
               420 \langle /mikoslides.ltxml \rangle
\mhframeimage Use the current value of \mh@currentrepos or the value of the mhrepos key if it
                is given in \frameimage.
               421 (*mikoslides)
               422 \def\Gin@mhrepos{}
               423 \define@key{Gin}{mhrepos}{\csxdef\Gin@mhrepos{#1}}
               424 (/mikoslides)
               425 (mikoslides.ltxml)DefKeyVal('Gin', 'mhrepos', 'Semiverbatim');
               426 (mikoslides.ltxml)RawTeX('
               427 (*mikoslides.ltxml | mikoslides)
               428 \newcommand\mhframeimage[2][]{%
                    \setkeys{Gin}{#1}%
               429
               430
                    \edef\mh@currentrepos}%
               431
                     \ifx\Gin@mhrepos\@empty%
               432
                       \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%
               433
                       \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%
               434
                    \fi%
               435
               436 }%
               437 (/mikoslides.ltxml | mikoslides)
               438 (mikoslides.ltxml),;
```

4.8 **problem-mh**: Support for Problems

We set up package options and pass them on to the problem package, which we also load.

```
439 (*problem)
440 \ProvidesPackage{problem-mh}[2015/11/04 v1.0 MathHub support for the sTeX problem package]
441 \RequirePackage{mathhub}
442 (/problem)
443 (*problem.ltxml)
444 RequirePackage('mathhub');
445 (/problem.ltxml)
```

\includemhproblem

The \includemhproblem saves the current value of \mh@currentrepos in a local macro \mh@@repos, resets \mh@currentrepos to the new value if one is given in the optional argument, and after importing resets \mh@currentrepos to the old value in \mh@@repos.

```
446 (*problem)
447 \newcommand\includemhproblem[2][]{\metasetkeys{inclprob}{#1}%
448 \edef\mh@@repos{\mh@currentrepos}%
449 \verb|\inclprob@mhrepos\\empty\\else\\mhcurrentrepos\\inclprob@mhrepos\\fi%
450 \input{\MathHub{\mh@currentrepos/source/#2}}%
451 \mhcurrentrepos\mh@@repos\clear@inclprob@keys}
452 (/problem)
453 (*problem.ltxml)
454 sub includemhproblem {
    my ($gullet,$keyval,$arg2) = @_;
    my $repo_path;
456
457
    if ($keyval) {
       $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
458
    if (! $repo_path) {
       $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
460
    else {
461
       $keyval->setValue('mhrepos',undef); }
462
    my $mathhub_base = ToString(Digest('\MathHub{}'));
463
    my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
464
     return Invocation(T_CS('\includeproblem'), $keyval, T_OTHER($finalpath)); }#$
466 DefKeyVal('inclprob', 'mhrepos', 'Semiverbatim');
467 DefMacro('\includemhproblem OptionalKeyVals:inclprob {}', \&includemhproblem);
468 (/problem.ltxml)
```

4.9 hwexam-mh: Support for Assignments

We set up package options and pass them on to the hwexam package, which we also load.

```
469 (*hwexam)
470 \ProvidesPackage{hwexam-mh} [2015/11/04 v1.0 MathHub support for the sTeX hwexam package]
471 \RequirePackage{mathhub}
472 (/hwexam)
473 (*hwexam.ltxml)
474 RequirePackage('mathhub');
475 (/hwexam.ltxml)
```

\includemhassignment

The \includemhassignment saves the current value of \mh@currentrepos in a local macro \mh@@repos, resets \mh@currentrepos to the new value if one is given in the optional argument, and after importing resets \mh@currentrepos to the old value in \mh@@repos.

```
\label{thm:command} 476 \end{center} $$477 \rightarrow \frac{2][]_{\text{metasetkeys}_{inclassig}_{\#1}}% $$478 \end{center} $$479 \ifx\inclassig@mhrepos\\empty\\else\mhcurrentrepos\inclassig@mhrepos\\fi%
```

```
480 \includeassignment [#1] {\MathHub{\mh@currentrepos/source/#2}}%
                   481 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
                   482 (/hwexam)
                   483 (*hwexam.ltxml)
                   484 sub includemhassignment {
                        my ($gullet,$keyval,$arg2) = @_;
                   486
                        my $repo_path;
                   487
                        if ($keyval) {
                          $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
                   488
                        if (! $repo_path) {
                   489
                          $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
                   490
                   491
                        else {
                           $keyval->setValue('mhrepos',undef); }
                   492
                        my $mathhub_base = ToString(Digest('\MathHub{}'));
                   493
                        my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
                   494
                        return Invocation(T_CS('\includeassignment'), $keyval, T_OTHER($finalpath)); }#$
                   496 DefKeyVal('inclprob', 'mhrepos', 'Semiverbatim');
                   497 DefMacro('\includemhassignment OptionalKeyVals:inclprob {}', \&includemhassignment);
                   498 (/hwexam.ltxml)
\inputmhassignment analogous
                   499 (*hwexam)
                   500 \newcommand\inputmhassignment[2][]{\metasetkeys{inclassig}{\#1}%
                   501 \edef\mh@@repos{\mh@currentrepos}%
                   502 \ifx\inclassig@mhrepos\@empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
                   503 \inputassignment[#1] {\MathHub{\mh@currentrepos/source/#2}}%
                   504 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
                   505 (/hwexam)
                   506 (*hwexam.ltxml)
                   507 sub inputmhassignment {
                       my ($gullet,$keyval,$arg2) = @_;
                   508
                        my $repo_path;
                   509
                   510
                        if ($keyval) {
                          $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
                   511
                        if (! $repo_path) {
                          $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
                   513
                        else {
                   514
                           $keyval->setValue('mhrepos',undef); }
                   515
                        my $mathhub_base = ToString(Digest('\MathHub{}'));
                   517 my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
                   518 return Invocation(T_CS('\inputassignment'), $keyval, T_OTHER($finalpath)); }#$
                   519 DefMacro('\inputmhassignment OptionalKeyVals:inclprob {}', \&inputmhassignment);
                   520 (/hwexam.ltxml)
```

4.10 tikzinput-mh: Support for Assignments

We set up package options and pass them on to the tikzinput package, which we also load.

```
521 \langle *tikzinput \rangle
```

```
522 \ProvidesPackage{tikzinput-mh}[2015/11/04 v1.0 MathHub support for the sTeX tikzinput package]
523 \RequirePackage{mathhub}
524 \langle /tikzinput \rangle
525 \langle *tikzinput.ltxml \rangle
526 RequirePackage('mathhub');
527 (/tikzinput.ltxml)
528 \tikzinput.ltxml\RawTeX('
529 (*tikzinput | tikzinput.ltxml)
530 \end{fine} \end{fine} \hrepos} {\csxdef\Gin@mhrepos{\#1}} \label{fine}
531 \end{area} $$1 \rightarrow [2] [] {\end{area}} \end{area} $$1 \rightarrow [2] [] {\end{area}} $$
532 \edef\mh@currentrepos}%
533 \ \texttt{\GinQmhrepos} \ \texttt{\G
534 \le tikzinput [#1] {\MathHub{\Gin@mhrepos/source/#2}} fi
535 \def\Gin@mhrepos{}\mhcurrentrepos\mh@@repos}
536 \newcommand\cmhtikzinput[2][]{\begin{center}\mhtikzinput[#1]{#2}\end{center}}
537 \; \langle / \text{tikzinput} \; | \; \text{tikzinput.ltxml} \rangle
_{538} \langle tikzinput.ltxml \rangle,;
```

4.11 Finale

Finally, we need to terminate the file with a success mark for perl.

539 (ltxml | modules.ltxml | structview.ltxml | omtext.ltxml | statements.ltxml | smultiling.ltxml | mikoslides.ltxml | problem

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. ISBN: 978-3-642-22672-4. URL: http://kwarc.info/frabe/Research/HIJKR_dimensions_11.pdf.
- [Koh15] Michael Kohlhase. metakeys.sty: A generic framework for extensible Metadata in LATEX. Tech. rep. Comprehensive TEX Archive Network (CTAN), 2015. URL: http://www.ctan.org/tex-archive/macros/latex/contrib/stex/metakeys/metakeys.pdf.
- [sTeX] KWARC/sTeX. URL: https://svn.kwarc.info/repos/stex (visited on 05/15/2015).