MathHub Support for sTeX*

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

November 19, 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 mathhub packages extend sTeX with support for the MathHub.info portal

Contents

---

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

1

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`.

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.

`\mhcurrentrepos`

The `\usemhmodule` is the analog to `\usemodule`.

`\usemhmodule`

For this, the `modules` package supplies the mh-variants `\mhinputref` and `\mhinput` of the `\inputref` macro introduced above and normal LaTeX `\input` macro.

`\mhinputref`
`\mhinput`

2.3 `omtext-mh`: MH Variants for OMText

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

`\mhcgraphics`

```
\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 `\usemhvocab` 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]

4

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]EDNOTE: needs to be documented
[2]EDNOTE: mhmodsig seems to be missing what happened?
[3]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
⟨*package⟩ and ⟨/package⟩) and the LaTeXML bindings (between ⟨*ltxml⟩ and
⟨/ltxml⟩). We keep the corresponding code fragments together, since the docu-
mentation 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 ⟨*ltxml | modules.ltxml | structview.ltxml | omtext.ltxml | statements.ltxml | smultiling.ltxml | mikoslides.ltxml | problem
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 | problem
8 ⟨package⟩\ProvidesPackage{mathhub}[2015/11/04 v1.0 sTeX Support for MathHub.info]
```

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 is used to initialize the current repository. If the repos has
\@mhcurrentrepos   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]{%
17   \edef\@test{#1}%
18   \ifx\@test\mh@currentrepos% if new dir = old dir
19     \relax% no need to change
20   \else%
21     \protected@write\@auxout{}{\string\@mhcurrentrepos{#1}}%
22   \fi%
23   \@mhcurrentrepos{#1}% define mh@currentrepos
24 }%
25 \newcommand\@mhcurrentrepos[1]{\edef\mh@currentrepos{#1}}%
26 ⟨/package⟩
27 ⟨*ltxml⟩
28 DefMacro('\mhcurrentrepos{}','\@mhcurrentrepos{#1}');
29 DefMacro('\@mhcurrentrepos{}','\def\mh@currentrepos{#1}\@@mhcurrentrepos{#1}');
30 DefConstructor('\@@mhcurrentrepos{}','',
31   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}
41   {Library file missing, cannot input #1\MessageBreak%
42     Both \@libfile.tex\MessageBreak and \@inffile.tex\MessageBreak do not exist}%
43   {Check whether the file name is correct}}}}
44 ⟨/package⟩
45 ⟨∗ltxml⟩
46 DefMacro('\modules@@first#1/#2;',''#1');
47 DefMacro('\libinput {}', sub{
48     my ($gullet, $name) = @_;
49     $name = ToString($name);
50     #Relative paths for recursive search
51     my $FIRSTLIB = ('/../../../lib');
52     my $SECONDLIB = ('/../../../../meta-info/lib');
53     my $file = pathname_find($name, types => ['tex'], paths =>[$FIRSTLIB]);
54     $file = pathname_find($name, types=>['tex'], paths=>[$SECONDLIB]) unless $file;
55     # Singal error if the file cannot be found
56     LaTeXML::Package::InputContent($file, noerror=>1); });
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 ⟨∗modules⟩
59 \ProvidesPackage{modules-mh}[2015/11/04 v1.0 MathHub support for the sTeX modules package]
60 \RequirePackage{mathhub}
61 ⟨/modules⟩
62 ⟨∗modules.ltxml⟩
63 RequirePackage('mathhub');
64 ⟨/modules.ltxml⟩
```

\importmhmodule The \importmhmodule[⟨*key=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 compar-
ison 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}%
```

8

```
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][]{%
72   \metasetkeys{importmhmodule}{#1}%
73   \ifx\importmhmodule@path\@empty% if module name is not set
74     \importmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
75   \else%
76     \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
77     \ifx\importmhmodule@repos\@empty% if in the same repos
78       \relax% no need to change mh@currentrepos, i.e, current dirctory.
79     \else%
80       \mhcurrentrepos{\importmhmodule@repos}% change it.
81     \fi%
82     \importmodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},%
83     ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
84     \mhcurrentrepos{\mh@@repos}% after importing, reset to old value
85   \fi%
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 {}',
96       "<omdoc:imports "
97       . "from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))()###2'"
98             . "?&defined(&GetKeyVal(#1,'conservative'))(load='&GetKeyVal(#1,'conservative')'"
99   afterDigest => \&importMHmoduleI);
100
101 sub importMHmoduleI {
102   my ($stomach, $whatsit) = @_;
103   my $keyval = $whatsit->getArg(1);
104   my $id = $whatsit->getArg(2);
105   if ($keyval) {
106     my $repos = ToString($keyval->getValue('repos'));
107     my $path = ToString($keyval->getValue('path'));
108     my $current_repos = LookupValue('current_repos');
109     if (!$repos) { # Use the implicit current repository
110       $repos = $current_repos; }
111     my $defpaths = LookupValue('defpath');
112     my $load_path = ($$defpaths{MathHub}).$repos.'/source/'.$path;
113     $keyval->setValue('load',$load_path);
114     AssignValue('current_repos' => $repos, 'global');
115     importmoduleI($stomach,$whatsit);
116     AssignValue('current_repos' => $current_repos, 'global'); }
```

```
117  else {
118     importmoduleI($stomach,$whatsit);  }
119  return; }
120
121 DefConstructor('\importMHmoduleI OptionalKeyVals:importmhmodule {}', '',
122    afterDigest=> \&importMHmoduleI );#$
```
123 ⟨/modules.ltxml⟩

and now the analogs

\usemhmodule

124 ⟨∗modules⟩
```
125 \newcommand\usemhmodule[2][]{%
126   \metasetkeys{importmhmodule}{#1}%
127   \ifx\importmhmodule@path\@empty%
128     \usemodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
129   \else%
130     \edef\mh@@repos{\mh@currentrepos}%
131     \ifx\importmhmodule@repos\@empty%
132     \else%
133       \mhcurrentrepos{\importmhmodule@repos}%
134     \fi%
135     \usemodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodule@
136     \mhcurrentrepos\mh@@repos%
137   \fi%
138   \ignorespaces%
139 }%
```
140 ⟨/modules⟩
141 ⟨∗modules.ltxml⟩
```
142 DefConstructor('\usemhmodule OptionalKeyVals:importmhmodule {}',
143    "<omdoc:uses from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))()###
144    afterDigest => \&importMHmoduleI);
```
145 ⟨/modules.ltxml⟩

\mhinputref

146 ⟨modules.ltxml⟩RawTeX('
147 ⟨∗modules | modules.ltxml⟩
```
148 \newcommand\mhinputref[2][]{%
149   \def\@repos{#1}%
150   \edef\mh@@repos{\mh@currentrepos}%
151   \ifx\@repos\@empty%
152   \else%
153     \mhcurrentrepos{#1}%
154   \fi%
155   \inputref{\MathHub{\mh@currentrepos/source/#2}}%
156   \mhcurrentrepos\mh@@repos%
157   \ignorespaces%
158 }%
```
159 ⟨/modules | modules.ltxml⟩
160 ⟨modules.ltxml⟩');

\mhinput

```
161 ⟨∗modules⟩
162 \let\mhinput\mhinputref%
163 ⟨/modules⟩
```

4.3   `omtext-mh`: MH Variants for OMText

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 \newcommand\mhcgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}}
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 {
185   my ($gullet,$keyval,$arg2) = @_;
186   my $repo_path;
187   if ($keyval) {
188     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
189   if (! $repo_path) {
190     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
191   else {
192     $keyval->setValue('mhrepos',undef); }
193   my $mathhub_base = ToString(Digest('\MathHub{}'));
194   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
195   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}}');
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.

```
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}
225 \addmetakey*{mhmodnl}{creators}
226 \addmetakey*{mhmodnl}{contributors}
227 \addmetakey{mhmodnl}{srccite}
228 \addmetakey{primary}{mhmodnl}[yes]
229 ⟨/smultiling⟩
230 ⟨∗smultiling.ltxml⟩
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 ⟨/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\@empty\importmodule{#2}\else\importmodule[ext=tex,load=\mhmodnl@load]{#2}\fi%
244 \fi}
245 {\end{module}}
246 ⟨/smultiling⟩
247 ⟨∗smultiling.ltxml⟩
248 DefEnvironment('{mhmodnl} OptionalKeyVals:mhmodnl {}{}',
249        "?#excluded()(<omdoc:theory xml:id='#2.#3' >"
250     .   "?&defined(&GetKeyVal(#1,'creators'))(<dc:creator>&GetKeyVal(#1,'creators')</dc:cr
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,'
254     .   "#body"
255     .   "</omdoc:theory>)",
256   afterDigestBegin=>sub {
257     my ($stomach, $whatsit) = @_;
258     my $keyval = $whatsit->getArg(1);
259     my $signature = ToString($whatsit->getArg(2));
260     my $language = ToString($whatsit->getArg(3));
261     my $repos = ToString(GetKeyVal($keyval,'torepos'));
262     my $current_repos = LookupValue('current_repos');
263     if (!$repos) { $repos = $current_repos; }
264     my $defpaths = LookupValue('defpath');
265     my $load_path = ($$defpaths{MathHub}).$repos.'/source/'.$signature;
266
267     if ($keyval) {
268       # If we're not given load, AND the langfiles option is in effect,
269       # default to #2
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');
274       $keyval->setValue('id',"$signature.$language"); }
275     module_afterDigestBegin(@_);
276     importmoduleI(@_);
277     return; },
278   afterDigest=>sub {
279     module_afterDigest(@_); });
280 ⟨/smultiling.ltxml⟩%$
```

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

```
281 ⟨smultiling.ltxml⟩RawTeX('
```

282 ⟨∗smultiling | smultiling.ltxml⟩
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 langauge suitably adapted.[4]

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 ⟨/smultiling | smultiling.ltxml⟩
292 ⟨smultiling.ltxml⟩');

4.6   `structview-mh`: MH Variants for Structures and Views

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][]{%
303   \gdef\@@doit{\importmhmodule[#1]{#2}{#3}}%
304   \ifmod@show\par\noindent importing module #2 via \@@doit\fi
305 }{%
306   \aftergroup\@@doit\ifmod@show end import\fi%
307 }%
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]EDNOTE: MK: we have to do something about the if@langfiles situation here. But this is non-trivial, since we do not know the current path, to which we could append .⟨*lang*⟩!

14

```
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 ⟨∗structview⟩
337 \newenvironment{mhview}[3][]{% keys, from, to
338   \metasetkeys{mhview}{#1}%
339   \sref@target%
340   \begin{@mhview}{#2}{#3}%
341   \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
342 }{%
343   \end{@mhview}%
344   \ignorespaces%
345 }%
346 \ifmod@show\surroundwithmdframed{mhview}\fi
347 ⟨/structview⟩
348 ⟨∗structview.ltxml⟩
349 DefMacroI(T_CS('\begin{mhview}'),'OptionalKeyVals:mhview {}{}', sub {
350   my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
351   my $from = ToString(Digest($from_arg));
352   my $to = ToString(Digest($to_arg));
353   AssignValue(from_module => $from);
354   AssignValue(to_module => $to);
355   my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
356   my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
357   my $repos = LookupValue('current_repos');
358   my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
359   my $to_path = ToString(GetKeyVal($keyvals,'topath'));
360   my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
361   $ext = 'sms' unless $ext;
362   my $current_repos = LookupValue('current_repos');
363   if (!$from_repos) { $from_repos = $current_repos; }
364   if (!$to_repos) { $to_repos = $current_repos; }
365   return (
```

```
366        Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
367        Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
368        Invocation(T_CS('\begin{viewenv}'),$keyvals,$from_arg,$to_arg)->unlist
369    );
370 });
371 DefMacroI('\end{mhview}',undef,'\end{viewenv}');
372 ⟨/structview.ltxml⟩
```

@mhview   The @mhview does the actual bookkeeping at the module level.

```
373 ⟨*structview⟩
374 \newenvironment{@mhview}[2]{%from, to
375    \importmhmodule[repos=\mhview@fromrepos,path=\mhview@frompath,ext=\mhview@ext]{#1}%
376    \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][]{%
381    \metasetkeys{mhview}{#1}%
382    \sref@target%
383    \begin{@mhview}{#2}{#3}%
384    \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
385 }{%
386    \end{@mhview}%
387    \ignorespaces%
388 }%
389 \ifmod@show\surroundwithmdframed{mhviewsketch}\fi
390 ⟨/structview⟩
391 ⟨*structview.ltxml⟩
392 DefMacroI(T_CS('\begin{mhviewsketch}'),'OptionalKeyVals:mhview {}{}', sub {
393    my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
394    my $from = ToString(Digest($from_arg));
395    my $to = ToString(Digest($to_arg));
396    my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
397    my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
398    my $repos = LookupValue('current_repos');
399    my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
400    my $to_path = ToString(GetKeyVal($keyvals,'topath'));
401    my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
402    $ext = 'sms' unless $ext;
403    my $current_repos = LookupValue('current_repos');
404    if (!$from_repos) { $from_repos = $current_repos; }
405    if (!$to_repos) { $to_repos = $current_repos; }
406    return (
407       Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
408       Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
409       Invocation(T_CS('\begin{viewsketchenv}'),$keyvals,$from_arg,$to_arg)->unlist
410    );
411 });
```

16

```
412 DefMacroI('\end{mhviewsketch}',undef,'\end{viewsketchenv}');
413 ⟨/structview.ltxml⟩
```

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 ⟨/mikoslides.ltxml⟩
```

\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][]{%
429   \setkeys{Gin}{#1}%
430   \edef\mh@@repos{\mh@currentrepos}%
431   \ifx\Gin@mhrepos\@empty%
432     \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%
433   \else%
434     \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%
435   \fi%
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 \ifx\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 {
455   my ($gullet,$keyval,$arg2) = @_;
456   my $repo_path;
457   if ($keyval) {
458     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
459   if (! $repo_path) {
460     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
461   else {
462     $keyval->setValue('mhrepos',undef); }
463   my $mathhub_base = ToString(Digest('\MathHub{}'));
464   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
465   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.

```
476 ⟨∗hwexam⟩
477 \newcommand\includemhassignment[2][]{\metasetkeys{inclassig}{#1}%
478 \edef\mh@@repos{\mh@currentrepos}%
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 {
485   my ($gullet,$keyval,$arg2) = @_;
486   my $repo_path;
487   if ($keyval) {
488     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
489   if (! $repo_path) {
490     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
491   else {
492     $keyval->setValue('mhrepos',undef); }
493   my $mathhub_base = ToString(Digest('\MathHub{}'));
494   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
495   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 {
508   my ($gullet,$keyval,$arg2) = @_;
509   my $repo_path;
510   if ($keyval) {
511     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
512   if (! $repo_path) {
513     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
514   else {
515     $keyval->setValue('mhrepos',undef); }
516   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 ⟨∗tikzinput⟩
```

```
522 \ProvidesPackage{tikzinput-mh}[2015/11/04 v1.0 MathHub support for the sTeX tikzinput package]
523 \RequirePackage{mathhub}
524 ⟨/tikzinput⟩
525 ⟨*tikzinput.ltxml⟩
526 RequirePackage('mathhub');
527 ⟨/tikzinput.ltxml⟩

528 ⟨tikzinput.ltxml⟩RawTeX('
529 ⟨*tikzinput | tikzinput.ltxml⟩
530 \define@key{Gin}{mhrepos}{\csxdef\Gin@mhrepos{#1}}
531 \newcommand\mhtikzinput[2][]{\def\Gin@mhrepos{}\setkeys{Gin}{#1}%
532 \edef\mh@@repos{\mh@currentrepos}%
533 \ifx\Gin@mhrepos\@empty\tikzinput[#1]{\MathHub{\mh@currentrepos/source/#2}}%
534 \else\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 ⟨/tikzinput | tikzinput.ltxml⟩
538 ⟨tikzinput.ltxml⟩');
```

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).