

MathHub Support for \LaTeX^*

Michael Kohlhasse
Jacobs University, Bremen
<http://kwarc.info/kohlhasse>

November 5, 2015

Abstract

The `sref` package is part of the \LaTeX collection, a version of $\text{\TeX}/\text{\LaTeX}$ that allows to markup $\text{\TeX}/\text{\LaTeX}$ documents semantically without leaving the document format, essentially turning $\text{\TeX}/\text{\LaTeX}$ into a document format for mathematical knowledge management (MKM).

The `mathhub` packages extend \LaTeX with support for the MathHub.info portal

Contents

1	Introduction	3
2	The User Interface	3
2.1	Package Options	3
2.2	<code>modules-mh</code> : MH Variants for Modules	3
2.3	<code>omtext-mh</code> : MH Variants for OMText	4
2.4	<code>smultiling-mh</code> : MH Variants for Multilinguality	4
2.5	<code>mikoslides-mh</code> : Support for MiKo Slides	4
2.6	<code>problem-mh</code> : Support for Problems	5
2.7	<code>hwexam-mh</code> : Support for Assignments	5
3	Limitations	5
4	Implementation	6
4.1	General Infrastructure	6
4.2	<code>modules-mh</code> : MH Variants for Modules	7
4.3	<code>omtext-mh</code> : MH Variants for OMText	12
4.4	<code>smultiling-mh</code> : MH Variants for Multilinguality	13
4.5	<code>mikoslides-mh</code> : Support for MiKo Slides	15
4.6	<code>problem-mh</code> : Support for Problems	16

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

4.7	hwexam-mh: Support for Assignments	17
4.8	Finale	18

1 Introduction

Much of the \LaTeX 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 \LaTeX author with **MathHub**-enabled versions of the \LaTeX 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 be loaded, `\importmhmodule` is the same as `\importmodule`.

`\mhcurrentrepos` Of course, neither \LaTeX nor \LaTeXML know 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 each module, since the `\importmhmodule` macro sets the current repository automatically.

`\usemhmodule` The `\usemhmodule` and `\adoptmhmodule` macros are the analogs to `\usemodule` and `\adoptmodule`.

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

2.3 omtext-mh: MH Variants for OMText

`\mhgraphics` The `\mhgraphics` 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)

```
\mhgraphics[fooMH/bar]{baz/foobar}
```

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

2.4 smultiling-mh: MH Variants for Multilinguality

1 2

2.5 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}
```

¹EDNOTE: needs to be documented

²EDNOTE: mhmodsig seems to be missing what happened?

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}
```

2.6 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 `\importmhmodule` form is more semantic, which allows more advanced document management features in **MathHub**.

2.7 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}
```

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 **TeX** GitHub repository [sTeX].

1. none reported yet.

4 Implementation

The `sref` package generates two files: the \LaTeX package (all the code between $\langle *package \rangle$ and $\langle /package \rangle$) and the \LaTeXML bindings (between $\langle *ltxml \rangle$ and $\langle /ltxml \rangle$). 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 and the base package.

```

1  $\langle *ltxml \mid \text{modules.ltxml} \mid \text{omtext.ltxml} \mid \text{smultiling.ltxml} \mid \text{mikoslides.ltxml} \mid \text{problem.ltxml} \mid \text{hwexam.ltxml} \rangle$ 
2 package LaTeXML::Package::Pool;
3 use strict;
4 use LaTeXML::Package;
5  $\langle /ltxml \mid \text{modules.ltxml} \mid \text{omtext.ltxml} \mid \text{smultiling.ltxml} \mid \text{mikoslides.ltxml} \mid \text{problem.ltxml} \mid \text{hwexam.ltxml} \rangle$ 
6  $\langle package \rangle$  \ProvidesPackage{mathhub}[2015/11/04 v1.0 sTeX Support for MathHub.info]

7  $\langle *package \rangle$ 
8 \DeclareOption*{}
9 \ProcessOptions
10  $\langle /package \rangle$ 
11  $\langle *ltxml \rangle$ 
12 DeclareOption(undef,sub {});
13 ProcessOptions();
14  $\langle /ltxml \rangle$ 
```

Then we need to set up the packages by requiring the `metakeys` package [Koh15] to be loaded (in the right version).

```

15  $\langle *package \rangle$ 
16 \RequirePackage{keyval}
17  $\langle /package \rangle$ 
18  $\langle *ltxml \rangle$ 
19 RequirePackage('keyval');
20  $\langle /ltxml \rangle$ 
```

4.1 General Infrastructure

`\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.

```

21  $\langle *package \rangle$ 
22 \newrobustcmd\mhcurrentrepos[1]{%
23   \edef\@test{#1}%
24   \ifx\@test\mh@currentrepos% if new dir = old dir
25     \relax% no need to change
26   \else%
27     \protected@write\@auxout{}{\string\@mhcurrentrepos{#1}}%
28     \fi%
29     \@mhcurrentrepos{#1}% define mh@currentrepos
30   }%
31 \newrobustcmd\@mhcurrentrepos[1]{\edef\mh@currentrepos{#1}}%
```

```

32 </package>
33 <*ltxml>
34 DefMacro('mhcurrentrepos{','\@mhcurrentrepos{#1}');
35 DefMacro('mhcurrentrepos{','\def\mh@currentrepos{#1}\@mhcurrentrepos{#1}');
36 DefConstructor('mhcurrentrepos{','',
37   afterDigest => sub{ AssignValue('current_repos',ToString($_[1]->getArg(1)),'global'); } );
38 </ltxml>#<
\libinput the \libinput macro inputs from the lib directory of the MathHub repository
or the meta-inf/lib repos of the group.
39 <ltxml>RaxTeX('
40 <*package | ltxml>
41 \def\modules@@first#1/#2;{#1}
42 \newcommand\libinput[1]{\def\@libfile{\MathHub{\mh@currentrepos/lib/#1}}%
43 \IfFileExists{\@libfile}{\input\@libfile}%
44 {\edef\@group{\expandafter\modules@@first\mh@currentrepos;}
45 \edef\@infile{\MathHub{\@group/meta-inf/lib/#1}}
46 \IfFileExists{\@infile}{\input{\@infile}}%
47 {\PackageError{modules}
48   {Library file missing, cannot input #1\MessageBreak%
49     Both \@libfile.tex\MessageBreak and \@infile.tex\MessageBreak do not exit}%
50   {Check whether the file name is correct}}}%
51 </package | ltxml>
52 <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.

```

53 <*modules>
54 \ProvidesPackage{modules-mh}[2015/11/04 v1.0 MathHub support for the sTeX modules package]
55 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{modules}}
56 \ProcessOptions
57 \RequirePackage{modules}
58 \RequirePackage{mathhub}
59 </modules>
60 <*modules.ltxml>
61 DeclareOption(undef,sub{PassOptions('modules','sty',ToString(Digest(T_CS('CurrentOption')))));
62 ProcessOptions();
63 RequirePackage('modules');
64 RequirePackage('mathhub');
65 </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.

```

```

66 <*modules>
67 \srefaddidkey{importmhmodule}%
68 \addmetakey{importmhmodule}{repos}% saves the repo's path. E.g: smglom/numberfield
69 \addmetakey{importmhmodule}{path}% saves the module name. E.g: naturalnumbers
70 \addmetakey[sms]{importmhmodule}{ext}% saves the extension: E.g: tex
71 \addmetakey[false]{importmhmodule}{conservative}[true]%
72 \newrobustcmd\importmhmodule[2] []{%
73   \metasetkeys{importmhmodule}{#1}%
74   \ifx\importmhmodule@path\@empty% if module name is not set
75     \importmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
76   \else%
77     \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
78     \ifx\importmhmodule@repos\@empty% if in the same repos
79       \relax% no need to change mh@currentrepos, i.e, current dirctory.
80     \else%
81       \mhcurrentrepos{\importmhmodule@repos}% change it.
82     \fi%
83     \importmodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},%
84     ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
85     \mhcurrentrepos{\mh@@repos}% after importing, reset to old value
86   \fi%
87   \ignorespaces%
88 }%

```

and now the analogs

`\usemhmodule`

```

89 \newrobustcmd\usemhmodule[2] []{%
90   \metasetkeys{importmhmodule}{#1}%
91   \ifx\importmhmodule@path\@empty%
92     \usemodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
93   \else%
94     \edef\mh@@repos{\mh@currentrepos}%
95     \ifx\importmhmodule@repos\@empty%
96     \else%
97       \mhcurrentrepos{\importmhmodule@repos}%
98     \fi%
99     \usemodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodule@
100     \mhcurrentrepos\mh@@repos%
101   \fi%
102   \ignorespaces%
103 }%

```

`\adoptmhmodule`

```

104 \newrobustcmd\adoptmhmodule[2] []{%
105   \metasetkeys{importmhmodule}{#1}%
106   \ifx\importmhmodule@path\@empty
107     \adoptmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
108   \else%
109     \edef\mh@@repos{\mh@currentrepos}%

```



```

110 \ifx\importmhmodule@repos\empty%
111 \else%
112 \mhcurrentrepos{\importmhmodule@repos}%
113 \fi%
114 \adoptmodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodul
115 \mhcurrentrepos\mh@repos%
116 \fi%
117 \ignorespaces%
118 }%
119 </modules>

\mhinputref
120 <modules.ltxml>\RawTeX('
121 <*modules | modules.ltxml>
122 \newcommand\mhinputref[2] [] {%
123 \def\@repos{#1}%
124 \edef\mh@repos{\mh@currentrepos}%
125 \ifx\@repos\empty%
126 \else%
127 \mhcurrentrepos{#1}%
128 \fi%
129 \inputref{\MathHub{\mh@currentrepos/source/#2}}%
130 \mhcurrentrepos\mh@repos%
131 \ignorespaces%
132 }%

\mhinput
133 \let\mhinput\mhinputref%

importmhmodulevia
134 \newenvironment{importmhmodulevia}[3] [] {%
135 \gdef\@doit{\importmhmodule[#1]{#2}{#3}}%
136 \ifmod@show\par\noindent importing module #2 via \@doit\fi
137 }{%
138 \aftergroup\@doit\ifmod@show end import\fi%
139 }%
140 </modules | modules.ltxml>
141 <modules.ltxml>');

142 <*modules>
143 \srefaddidkey{mhview}
144 \addmetakey{mhview}{display}
145 \addmetakey{mhview}{creators}
146 \addmetakey{mhview}{contributors}
147 \addmetakey{mhview}{srccite}
148 \addmetakey*{mhview}{title}
149 \addmetakey{mhview}{fromrepos}
150 \addmetakey{mhview}{torepos}
151 \addmetakey{mhview}{frompath}
152 \addmetakey{mhview}{topath}

```

```

153 \addmetakey[sms]{mhview}{ext}

mhview the MathHub version

154 \newenvironment{mhview}[3][{}]{% keys, from, to
155   \metasetkeys{mhview}{#1}%
156   \sref@target%
157   \begin{@mhview}{#2}{#3}%
158   \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
159 }{%
160   \end{@mhview}%
161   \ignorespaces%
162 }%
163 \ifmod@show\surroundwithmdframed{mhview}\fi

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

164 \newenvironment{@mhview}[2]{%from, to
165   \importmhmodule[repos=\mhview@fromrepos,path=\mhview@frompath,ext=\mhview@ext]{#1}%
166   \importmhmodule[repos=\mhview@torepos,path=\mhview@topath,ext=\mhview@ext]{#2}%
167 }{}%

mhviewsketch The mhviewsketch environment behaves like mhview, but only has text contents.

168 \newenvironment{mhviewsketch}[3][{}]{%
169   \metasetkeys{mhview}{#1}%
170   \sref@target%
171   \begin{@mhview}{#2}{#3}%
172   \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
173 }{%
174   \end{@mhview}%
175   \ignorespaces%
176 }%
177 \ifmod@show\surroundwithmdframed{mhviewsketch}\fi
178 </modules>

```

EdN:3

```

3

179 <*modules.ltxml>
180 DefKeyVal('mhview','id','Semiverbatim');
181 DefKeyVal('mhview','fromrepos','Semiverbatim');
182 DefKeyVal('mhview','torepos','Semiverbatim');
183 DefKeyVal('mhview','frompath','Semiverbatim');
184 DefKeyVal('mhview','topath','Semiverbatim');
185 DefKeyVal('mhview','title','Semiverbatim');
186 DefKeyVal('mhview','creators','Semiverbatim');
187 DefKeyVal('mhview','contributors','Semiverbatim');
188 DefKeyVal('mhview','display','Semiverbatim');
189 DefKeyVal('mhview','ext','Semiverbatim');
190 DefMacroI(T_CS('\begin{mhview}'),'OptionalKeyVals:mhview {}{}', sub {
191   my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
192   my $from = ToString(Digest($from_arg));

```

³EdNOTE: MK: sort these into the rest.

```

193 my $to = ToString(Digest($to_arg));
194 AssignValue(from_module => $from);
195 AssignValue(to_module => $to);
196 my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
197 my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
198 my $repos = LookupValue('current_repos');
199 my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
200 my $to_path = ToString(GetKeyVal($keyvals,'topath'));
201 my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
202 $ext = 'sms' unless $ext;
203 my $current_repos = LookupValue('current_repos');
204 if (!$from_repos) { $from_repos = $current_repos; }
205 if (!$to_repos) { $to_repos = $current_repos; }
206 return (
207     Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
208     Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
209     Invocation(T_CS('\begin{viewenv}'),$keyvals,$from_arg,$to_arg)->unlist
210 );
211 });
212 DefMacroI('\end{mhview}',undef,'\end{viewenv}');
213
214 DefMacroI(T_CS('\begin{mhviewsketch}'),'OptionalKeyVals:mhview {}{}', sub {
215     my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
216     my $from = ToString(Digest($from_arg));
217     my $to = ToString(Digest($to_arg));
218     my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
219     my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
220     my $repos = LookupValue('current_repos');
221     my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
222     my $to_path = ToString(GetKeyVal($keyvals,'topath'));
223     my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
224     $ext = 'sms' unless $ext;
225     my $current_repos = LookupValue('current_repos');
226     if (!$from_repos) { $from_repos = $current_repos; }
227     if (!$to_repos) { $to_repos = $current_repos; }
228     return (
229         Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
230         Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
231         Invocation(T_CS('\begin{viewsketchenv}'),$keyvals,$from_arg,$to_arg)->unlist
232     );
233 });
234 DefMacroI('\end{mhviewsketch}',undef,'\end{viewsketchenv}');
235
236 DefConstructor('\importmhmodule OptionalKeyVals:importmhmodule {}',
237     "<omdoc:imports "
238     . "from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))(###2"
239     . "?&defined(&GetKeyVal(#1,'conservative'))(load='&GetKeyVal(#1,'conservative'))"
240     afterDigest => \&importMHmoduleI);
241
242 DefConstructor('\usemhmodule OptionalKeyVals:importmhmodule {}',

```

```

243   "<omdoc:uses from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))(##)
244   afterDigest => \&importMHmoduleI);
245
246 DefConstructor('\adoptmhmodule OptionalKeyVals:importmhmodule {}',
247   "<omdoc:adopts from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))(##)
248   afterDigest => \&importMHmoduleI);
249 \modules.ltxml)

```

4.3 omtex-mh: MH Variants for OMText

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

```

250 \*omtext)
251 \ProvidesPackage{omtex-mh}[2015/11/04 v1.0 MathHub support for the sTeX omtex package]
252 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{omtex}}
253 \ProcessOptions
254 \RequirePackage{omtex}
255 \RequirePackage{mathhub}
256 \*omtex)
257 \*omtex.ltxml)
258 DeclareOption(undef,sub{PassOptions('omtex','sty',ToString(Digest(T_CS('CurrentOption')))); }
259 ProcessOptions();
260 RequirePackage('omtex');
261 RequirePackage('mathhub');
262 \*omtex.ltxml)

```

`\mh*graphics` Use the current value of `\mh@currentrepos` or the value of the `mhrepos` key if it is given in `\my*graphics`.

```

263 \*omtex)
264 \addmetakey{Gin}{mhrepos}
265 \newcommand\mhgraphics[2][]{\metasetkeys{Gin}{#1}%
266 \edef\mh@@repos{\mh@currentrepos}%
267 \ifx\Gin@mhrepos@empty\mygraphics[#1]{\MathHub{\mh@currentrepos/source/#2}}%
268 \else\mygraphics[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
269 \def\Gin@mhrepos{\mhcurrentrepos\mh@@repos}
270 \newcommand\mhgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}}
271 \newcommand\mhbggraphics[2][]{\fbox{\mhgraphics[#1]{#2}}}
272 \newcommand\mhcbgraphics[2][]{\begin{center}\fbox{\mhgraphics[#1]{#2}}\end{center}}
273 \*omtex)
274 \*omtex.ltxml)
275 sub mhgraphics {
276   my ($gullet,$keyval,$arg2) = @_;
277   my $repo_path;
278   if ($keyval) {
279     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
280   if (! $repo_path) {
281     $repo_path = ToString(Digest(T_CS('mh@currentrepos'))); }
282   else {
283     $keyval->setValue('mhrepos',undef); }

```

```

284 my $mathhub_base = ToString(Digest('\MathHub{}'));
285 my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
286 return Invocation(T_CS('\includegraphicx'), $keyval, T_OTHER($finalpath)); }#$
287 DefKeyVal('Gin', 'mhrepos', 'Semiverbatim');
288 DefMacro('\mhgraphics OptionalKeyVals:Gin {}', \&mhgraphics);
289 DefMacro('\mhcgraphics []{}', '\begin{center}\mhgraphics[#1]{#2}\end{center}');
290 DefMacro('\mhbgraphics []{}', '\fbox{\mhgraphics[#1]{#2}}');
291 \</omtext.ltxml>

```

4.4 smultiling-mh: MH Variants for Multilinguality

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

```

292 \<smultiling>
293 \ProvidesPackage{smultiling-mh}[2015/11/04 v1.0 MathHub support for the sTeX smultiling package]
294 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{smultiling}}
295 \ProcessOptions
296 \RequirePackage{smultiling}
297 \RequirePackage{mathhub}
298 \</smultiling>
299 \<smultiling.ltxml>
300 DeclareOption(undef, sub{PassOptions('smultiling', 'sty', ToString(Digest(T_CS('\CurrentOption'))))}
301 ProcessOptions();
302 RequirePackage('smultiling');
303 RequirePackage('mathhub');
304 \</smultiling.ltxml>

```

`mhmodnl:`

```

305 \<smultiling>
306 \addmetakey{mhmodnl}{repos}
307 \addmetakey{mhmodnl}{path}
308 \addmetakey*{mhmodnl}{title}
309 \addmetakey*{mhmodnl}{creators}
310 \addmetakey*{mhmodnl}{contributors}
311 \addmetakey{mhmodnl}{srccite}
312 \addmetakey{primary}{mhmodnl}[yes]
313 \</smultiling>
314 \<smultiling.ltxml>
315 DefKeyVal('mhmodnl', 'title', 'Semiverbatim');
316 DefKeyVal('mhmodnl', 'repos', 'Semiverbatim');
317 DefKeyVal('mhmodnl', 'path', 'Semiverbatim');
318 DefKeyVal('mhmodnl', 'creators', 'Semiverbatim');
319 DefKeyVal('mhmodnl', 'contributors', 'Semiverbatim');
320 DefKeyVal('mhmodnl', 'primary', 'Semiverbatim');
321 \</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.

```

322 \<smultiling>

```

```

323 \newenvironment{mhmodnl}[3][\metasetkeys{mhmodnl}{#1}%
324 \def\@test{#1}\ifx\@test\@empty\begin{module}[id=#2.#3]\else\begin{module}[id=#2.#3,#1]\fi%
325 \edef\@repos{\ifx\mhmodnl@repos\@empty\mh@currentrepos\else\mhmodnl@repos}
326 \if@langfiles\importmhmodule[repos=\@repos,load=#2,ext=tex]{#2}\else
327 \ifx\mhmodnl@load\@empty\importmodule{#2}\else\importmodule[ext=tex,load=\mhmodnl@load]{#2}\fi%
328 \fi}
329 {\end{module}}
330 \</smultiling>
331 \<*smultiling.ltxml>
332 DefEnvironment(' {mhmodnl} OptionalKeyVals:mhmodnl {}{ }',
333     "?#excluded()(<omdoc:theory xml:id='#2.#3' >"
334     . "?&defined(&GetKeyVal(#1,'creators'))(<dc:creator>&GetKeyVal(#1,'creators')</dc:creator>)"
335     . "?&defined(&GetKeyVal(#1,'title'))(<dc:title>&GetKeyVal(#1,'title')</dc:title>)"
336     . "?&defined(&GetKeyVal(#1,'contributors'))(<dc:contributor>&GetKeyVal(#1,'contributors')</dc:contributor>)"
337     . "<omdoc:imports from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')&GetKeyVal(#1,'load'))>"
338     . "#body"
339     . "</omdoc:theory>)",
340     afterDigestBegin=>sub {
341         my ($stomach, $whatsit) = @_;
342         my $keyval = $whatsit->getArg(1);
343         my $signature = ToString($whatsit->getArg(2));
344         my $language = ToString($whatsit->getArg(3));
345         my $repos = ToString(GetKeyVal($keyval,'torepos'));
346         my $current_repos = LookupValue('current_repos');
347         if (!$repos) { $repos = $current_repos; }
348         my $defpaths = LookupValue('defpath');
349         my $load_path = ($$defpaths{MathHub}).$repos.'/source/'. $signature;
350
351         if ($keyval) {
352             # If we're not given load, AND the langfiles option is in effect,
353             # default to #2
354             if ((! $keyval->getValue('path')) && (LookupValue('smultiling_langfiles'))) {
355                 $keyval->setValue('load',$load_path); }
356             # Always load a TeX file
357             $keyval->setValue('ext','tex');
358             $keyval->setValue('id'," $signature.$language"); }
359         module_afterDigestBegin(@_);
360         importmoduleI(@_);
361         return; },
362     afterDigest=>sub {
363         module_afterDigest(@_); });
364 \</smultiling.ltxml>%$

```

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

```

365 \<smultiling.ltxml>RawTeX(
366 \<*smultiling | smultiling.ltxml>
367 \newenvironment{mhviewsig}[4][\def\@test{#1}\ifx\@test\@empty%
368 \begin{mhview}[id=#2,ext=tex]{#3}{#4}\else\begin{mhview}[id=#2,#1,ext=tex]{#3}{#4}\fi}
369 {\end{mhview}}

```

`mhviewnl` The `mhviewnl` environment is just a layer over the `mhviewsketch` environment with the keys and language suitably adapted.⁴

```
370 \newenvironment{mhviewnl}[5][]{\def\@test{#1}\ifx\@test\@empty%
371 \begin{mhviewsketch}[id=#2.#3,ext=tex]{#4}{#5}\else%
372 \begin{mhviewsketch}[id=#2.#3,#1,ext=tex]{#4}{#5}\fi}
373 {\end{mhviewsketch}}
374 \</smultiling | smultiling.ltxml>
375 \smultiling.ltxml\>);
```

4.5 mikosides-mh: Support for MiKo Slides

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

```
376 \<*mikosides>
377 \ProvidesPackage{mikosides-mh}[2015/11/04 v1.0 MathHub support for the sTeX mikosides package]
378 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{mikosides}}
379 \ProcessOptions
380 \RequirePackage{mikosides}
381 \RequirePackage{mathhub}
382 \</mikosides>
383 \<*mikosides.ltxml>
384 \DeclareOption{undef,sub{PassOptions('mikosides','sty',ToString(Digest(T_CS('\CurrentOption'))))}}
385 \ProcessOptions();
386 \RequirePackage('mikosides');
387 \RequirePackage('mathhub');
388 \</mikosides.ltxml>
```

`\mhframeimage` Use the current value of `\mh@currentrepos` or the value of the `mhrepos` key if it is given in `\frameimage`.

```
389 \<mikosides>\addmetakey{Gin}{mhrepos}
390 \<mikosides.ltxml>\DefKeyVal('Gin','mhrepos','Semiverbatim');
391 \<mikosides.ltxml>\RawTeX(
392 \<*mikosides.ltxml | mikosides>
393 \newcommand\mhframeimage[2][]{%
394   \metasetkeys{Gin}{#1}%
395   \edef\mh@@repos{\mh@currentrepos}%
396   \ifx\Gin@mhrepos\@empty%
397     \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%
398   \else%
399     \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%
400   \fi%
401 }%
402 \</mikosides.ltxml | mikosides>
403 \<mikosides.ltxml>);
```

⁴EDNOTE: 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`!

4.6 problem-mh: Support for Problems

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

```

404 <*problem>
405 \ProvidesPackage{problem-mh}[2015/11/04 v1.0 MathHub support for the sTeX problem package]
406 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{problem}}
407 \ProcessOptions
408 \RequirePackage{problem}
409 \RequirePackage{mathhub}
410 </problem>
411 <*problem.ltxml>
412 DeclareOption(undef,sub{PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))));}
413 ProcessOptions();
414 RequirePackage('problem');
415 RequirePackage('mathhub');
416 </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`.

```

417 <*problem>
418 \newcommand\includemhproblem[2][]{\metasetkeys{inclprob}{#1}%
419 \edef\mh@@repos{\mh@currentrepos}%
420 \ifx\inclprob\mhrepos\empty\else\mhcurrentrepos\inclprob\mhrepos\fi%
421 \input{\MathHub{\mh@currentrepos/source/#2}}%
422 \mhcurrentrepos\mh@@repos\clear@inclprob@keys}
423 </problem>
424 <*problem.ltxml>
425 sub includemhproblem {
426   my ($gullet,$keyval,$arg2) = @_ ;
427   my $repo_path;
428   if ($keyval) {
429     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
430   if (! $repo_path) {
431     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
432   else {
433     $keyval->setValue('mhrepos',undef); }
434   my $mathhub_base = ToString(Digest('\MathHub{'}));
435   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
436   return Invocation(T_CS('\includeproblem'), $keyval, T_OTHER($finalpath)); }#$
437 DefKeyVal('inclprob','mhrepos','Semiverbatim');
438 DefMacro('\includemhproblem OptionalKeyVals:inclprob {}', \&includemhproblem);
439 </problem.ltxml>

```


4.7 hwexam-mh: Support for Assignments

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

```

440 <*hwexam>
441 \ProvidesPackage{hwexam-mh}[2015/11/04 v1.0 MathHub support for the sTeX hwexam package]
442 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{hwexam}}
443 \ProcessOptions
444 \RequirePackage{hwexam}
445 \RequirePackage{mathhub}
446 </hwexam>
447 <*hwexam.ltxml>
448 DeclareOption(undef,sub{PassOptions('hwexam','sty',ToString(Digest(T_CS('\CurrentOption')))); }
449 ProcessOptions();
450 RequirePackage('hwexam');
451 RequirePackage('mathhub');
452 </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`.

```

453 <*package>
454 \newcommand\includemhassignment[2][\metasetkeys{inclassig}{#1}%
455 \edef\mh@@repos{\mh@currentrepos}%
456 \ifx\inclassig@mhrepos\empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
457 \includeassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}%
458 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
459 </package>
460 <*ltxml>
461 sub includemhassignment {
462   my ($gullet,$keyval,$arg2) = @_ ;
463   my $repo_path;
464   if ($keyval) {
465     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
466   if (! $repo_path) {
467     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
468   else {
469     $keyval->setValue('mhrepos',undef); }
470   my $mathhub_base = ToString(Digest('\MathHub{'}));
471   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
472   return Invocation(T_CS('\includeassignment'), $keyval, T_OTHER($finalpath)); }#$
473 DefKeyVal('inclprob','mhrepos','Semiverbatim');
474 DefMacro('\includemhassignment OptionalKeyVals:inclprob {}', \&includemhassignment);
475 </ltxml>

```

`\inputmhassignment` analogous

```

476 <*package>
477 \newcommand\inputmhassignment[2][\metasetkeys{inclassig}{#1}%

```

```

478 \edef\mh@@repos{\mh@currentrepos}%
479 \ifx\inclassig@mhrepos\empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
480 \inputassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}%
481 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
482 \end{package}
483 \end{ltxml}
484 sub inputmhassignment {
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('\inputassignment'), $keyval, T_OTHER($finalpath)); }#$
496 DefMacro('\inputmhassignment OptionalKeyVals:inclprob {}', \&inputmhassignment);
497 \end{ltxml}

```

4.8 Finale

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

```

498 \end{ltxml}
499 1;
500 \end{ltxml}

```

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 Kohlhasse. *metakeys.sty: A generic framework for extensible Metadata in L^AT_EX*. Tech. rep. Comprehensive T_EX 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).