# MathHub Support for STEX\*

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

November 4, 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
<b>2</b>	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	smultiling-mh: MH Variants for Multilinguality	4
	2.5	mikoslides-mh: Support for MiKo Slides	4
	2.6	problem-mh: Support for Problems	5
	2.7	hwexam-mh: Support for Assignments	5
3	Lim	itations	5
4	Imp	plementation	6
	4.1	General Infrastructure	6
	4.2	modules-mh: MH Variants for Modules	7
	4.3	omtext-mh: MH Variants for OMText	12
	4.4	smultiling-mh: MH Variants for Multilinguality	13
	4.5	mikoslides-mh: Support for MiKo Slides	15
	4.6	problem-mh: Support for Problems	16

<sup>\*</sup>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 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 \adoptmhmodule \mhinputref \mhinput The  $\usemmand$  and  $\adoptmhmodule$  macros are the analogs to  $\usemodule$  and  $\adoptmodule$ .

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

\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 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}



 $<sup>^{1}\</sup>mathrm{EdNote}$ : needs to be documented

 $<sup>^2\</sup>mathrm{EdNote}\colon$  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

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 STEX 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 an the base package.

```
1 (*Itxml | modules.Itxml | omtext.Itxml | smultiling.Itxml | mikoslides.Itxml | problem.Itxml | hwexam.Itxml)
2 package LaTeXML::Package::Pool;
3 use strict;
4 use LaTeXML::Package;
5 (/ltxml | modules.ltxml | omtext.ltxml | smultiling.ltxml | mikoslides.ltxml | problem.ltxml | hwexam.ltxml)
6 (package)\ProvidesPackage{mathhub}[2015/11/04 v1.0 sTeX Support for MathHub.info]
7 (*package)
8 \DeclareOption*{}
9 \ProcessOptions
10 (/package)
11 (*ltxml)
12 DeclareOption(undef, sub {});
13 ProcessOptions();
14 \langle /ltxml \rangle
   Then we need to set up the packages by requiring the metakeys pack-
age [Koh15] to be loaded (in the right version).
15 (*package)
16 \RequirePackage{keyval}
17 (/package)
18 (*ltxml)
19 RequirePackage('keyval');
20 (/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.

```
21 (*package)
22 \newrobustcmd\mhcurrentrepos[1]{%
    \edef\@test{#1}%
23
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
    \@mhcurrentrepos{#1}% define mh@currentrepos
29
31 \newrobustcmd\@mhcurrentrepos[1]{\edef\mh@currentrepos{#1}}%
```

```
32 (/package)
                                33 (*ltxml)
                                34 DefMacro('\mhcurrentrepos{}','\@mhcurrentrepos{#1}');
                                35 DefMacro('\@mhcurrentrepos{#1}','\def\mh@currentrepos{#1}\@mhcurrentrepos{#1}');
                                36 DefConstructor('\@@mhcurrentrepos{}','',
                                             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}
                                \label{limits} 42 \end{libinput} [1] {\end{libinjut} [1] {\end{libinjut} (\end{libinjut} )} % $$ $ (\end{libinjut} [1] {\end{libinjut} (\end{libinjut} )} $$ $ (\end{libinjut} [1] {\end{libinjut} (\end{libinjut} )} $$ $ (\end{libinjut} [1] {\end{libinjut} )} $$ $ (\end{libinjut} )} $$ $ (\end{lib
                                43 \IfFileExists{\@libfile}{\input\@libfile}%
                                44 {\edef\@@group{\expandafter\modules@@first\mh@currentrepos;}
                                45 \edef\@inffile{\MathHub{\@@group/meta-inf/lib/#1}}
                                46 \IfFileExists{\@inffile}{\input{\@inffile}}%
                                47 {\PackageError{modules}
                                             {Library file missing, cannot input #1\MessageBreak%
                                49
                                                    Both \@libfile.tex\MessageBreak and \@inffile.tex\MessageBreak do not exit}%
                                             {Check whether the file name is correct}}}}
                                51 \langle /package \mid ltxml \rangle
                                52 \langle \mathsf{ltxml} \rangle,;
```

#### 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{modules}
59 \sqrt{modules}
60 \left\{\text{modules.ltxml}\right\}
61 \DeclareOption(undef,sub{\PassOptions('modules','sty',ToString(Digest(T_CS('\CurrentOption'))));}
62 \ProcessOptions();
63 \RequirePackage('modules');
64 \RequirePackage('mathhub');
65 \sqrt{\modules.ltxml}
```

\importmhmodule

The \importmhmodule[ $\langle key=value\ list\rangle$ ] {module} saves the current value of \mh@currentrepos in a local macro \mh@Grepos, 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@Grepos. 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.

```
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}%
                    \ifx\importmhmodule@path\@empty% if module name is not set
                74
                      \importmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
                75
                    \else%
                76
                77
                      \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
                      \ifx\importmhmodule@repos\@empty% if in the same repos
                78
                        \relax% no need to change mh@currentrepos, i.e, current dirctory.
                79
                      \else%
                80
                        81
                82
                      %,{\importmodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},
                83
                      ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
                84
                85
                      \mhcurrentrepos{\mh@@repos}% after importing, reset to old value
                86
                    \ignorespaces%
                87
                88 }%
                and now the analogs
  \usemhmodule
                89 \newrobustcmd\usemhmodule[2][]{%
                    \metasetkeys{importmhmodule}{#1}%
                91
                    \ifx\importmhmodule@path\@empty%
                      \usemodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
                92
                93
                    \else%
                      \edef\mh@@repos{\mh@currentrepos}%
                94
                95
                      \ifx\importmhmodule@repos\@empty%
                96
                      \else%
                        \mhcurrentrepos{\importmhmodule@repos}%
                97
                98
                      \usemodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodule@
                99
                      \mhcurrentrepos\mh@@repos%
               100
                    \fi%
               101
               102
                    \ignorespaces%
               103 }%
\adoptmhmodule
               104 \newrobustcmd\adoptmhmodule[2][]{%
                    \metasetkeys{importmhmodule}{#1}%
               105
                    \ifx\importmhmodule@path\@empty
               106
                      \adoptmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
               107
               108
                    \else%
               109
                      \edef\mh@currentrepos}%
```

```
\ifx\importmhmodule@repos\@empty%
                  110
                         \else%
                  111
                            \mhcurrentrepos{\importmhmodule@repos}%
                  112
                  113
                          \adoptmodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodul
                  114
                  115
                          \mhcurrentrepos\mh@@repos%
                  116
                       \fi%
                       \ignorespaces%
                  117
                  118 }%
      \mhinputref
                  119 \newrobustcmd\mhinputref[2][]{%
                       \def\@repos{#1}%
                  120
                  121
                       \edef\mh@@repos{\mh@currentrepos}%
                       \ifx\@repos\@empty%
                  123
                       \else%
                  124
                          \mhcurrentrepos{#1}%
                       \fi%
                  125
                       \inputref{\MathHub{\mh@currentrepos/source/#2}}%
                  126
                       \mhcurrentrepos\mh@@repos%
                  127
                  128
                       \ignorespaces%
                  129 }%
         \mhinput
                  130 \let\mhinput\mhinputref%
importmhmodulevia
                  131 \newenvironment{importmhmodulevia}[3][]{%
                       \ifmod@show\par\noindent importing module #2 via \@@doit\fi
                  134 }{%
                       \aftergroup\@@doit\ifmod@show end import\fi%
                  135
                  136 }%
                  137 \srefaddidkey{mhview}
                  138 \addmetakey{mhview}{display}
                  139 \addmetakey{mhview}{creators}
                  140 \addmetakey{mhview}{contributors}
                  141 \addmetakey{mhview}{srccite}
                  142 \addmetakey*{mhview}{title}
                  143 \addmetakey{mhview}{fromrepos}
                  144 \addmetakey{mhview}{torepos}
                  145 \addmetakey{mhview}{frompath}
                  146 \verb| \addmetakey{mhview}{topath}|
                  147 \addmetakey[sms]{mhview}{ext}
           mhview the MathHub version
                  148 \mbox{ newenvironment{mhview}[3][]{% keys, from, to}
                       \metasetkeys{mhview}{#1}%
```

```
\sref@target%
             150
                  \begin{@mhview}{#2}{#3}%
             151
                  152
             153 }{%
                  \end{@mhview}%
             154
                  \ignorespaces%
             156 }%
             157 \ifmod@show\surroundwithmdframed{mhview}\fi
     Omhview The Omhview does the actual bookkeeping at the module level.
             158 \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}%
             161 }{}%
mhviewsketch The mhviewsketch environment behaves like mhview, but only has text contents.
             162 \newenvironment{mhviewsketch}[3][]{%
                  \metasetkeys{mhview}{#1}%
             163
             164
                  \sref@target%
             165
                  \begin{@mhview}{#2}{#3}%
                  \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
             166
             167 }{%
             168
                  \end{@mhview}%
                  \ignorespaces%
             169
             171 \ \texttt{ifmod@show} \\ \text{surroundwithmdframed{mhviewsketch}} \\ \text{fi}
             172 (/modules)
             173 (*modules.ltxml)
             174 DefKeyVal('mhview','id','Semiverbatim');
             175 DefKeyVal('mhview', 'fromrepos', 'Semiverbatim');
             176 DefKeyVal('mhview', 'torepos', 'Semiverbatim');
             177 DefKeyVal('mhview', 'frompath', 'Semiverbatim');
             178 DefKeyVal('mhview','topath','Semiverbatim');
             179 DefKeyVal('mhview', 'title', 'Semiverbatim');
             180 DefKeyVal('mhview', 'creators', 'Semiverbatim');
             181 DefKeyVal('mhview', 'contributors', 'Semiverbatim');
             182 DefKeyVal('mhview', 'display', 'Semiverbatim');
             183 DefKeyVal('mhview', 'ext', 'Semiverbatim');
             184 DefMacroI(T_CS('\begin{mhview}'),'OptionalKeyVals:mhview {}{}', sub {
                  my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
             185
                  my $from = ToString(Digest($from_arg));
             186
                  my $to = ToString(Digest($to_arg));
             187
                  AssignValue(from_module => $from);
                  AssignValue(to_module => $to);
                  my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
                  my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
             191
```

EdN:3

 $^3\mathrm{EdNote}\colon\, \mathrm{MK} \colon$  sort these into the rest.

```
my $repos = LookupValue('current_repos');
192
     my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
193
     my $to_path = ToString(GetKeyVal($keyvals,'topath'));
194
    my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
195
     $ext = 'sms' unless $ext;
196
     my $current_repos = LookupValue('current_repos');
197
     if (!$from_repos) { $from_repos = $current_repos; }
198
199
     if (!$to_repos) { $to_repos = $current_repos; }
200
     return (
       Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
201
       Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
202
203
       Invocation(T_CS('\begin{viewenv}'), $keyvals, $from_arg, $to_arg)->unlist
    );
204
205 }):
206 DefMacroI('\end{mhview}',undef,'\end{viewenv}');
207
208 DefMacroI(T_CS('\begin{mhviewsketch}'),'OptionalKeyVals:mhview {}{}', sub {
209
     my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
     my $from = ToString(Digest($from_arg));
     my $to = ToString(Digest($to_arg));
211
212
    my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
     my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
213
     my $repos = LookupValue('current_repos');
214
     my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
215
     my $to_path = ToString(GetKeyVal($keyvals,'topath'));
216
     my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
     $ext = 'sms' unless $ext;
218
     my $current_repos = LookupValue('current_repos');
219
     if (!$from_repos) { $from_repos = $current_repos; }
220
     if (!$to_repos) { $to_repos = $current_repos; }
221
222
     return (
223
       Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
224
       Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
225
       Invocation(T_CS('\begin{viewsketchenv}'), $keyvals, $from_arg, $to_arg)->unlist
    );
226
227 });
228 DefMacroI('\end{mhviewsketch}',undef,'\end{viewsketchenv}');
230 DefConstructor('\importmhmodule OptionalKeyVals:importmhmodule {}',
231
          "<omdoc:imports "
232
          . "from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))()###2'"
                   . "?&defined(&GetKeyVal(#1,'conservative'))(load='&GetKeyVal(#1,'conservative')'
233
      afterDigest => \&importMHmoduleI);
234
235
236 DefConstructor('\usemhmodule OptionalKeyVals:importmhmodule {}',
237
      "<omdoc:uses from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))()###</pre>
238
      afterDigest => \&importMHmoduleI);
239
240 DefConstructor('\adoptmhmodule OptionalKeyVals:importmhmodule {}',
      "<omdoc:adopts from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))()#</pre>
```

241

```
afterDigest => \&importMHmoduleI);
242
243
244 RawTeX('
245 \newcommand\mhinputref [2] [] {\def\@repos{#1}%
246 \edef\mh@@repos{\mh@currentrepos}\%
247 \ifx\@repos\@empty\else\mhcurrentrepos{#1}\fi%
248 \inputref{\MathHub{\mh@currentrepos/source/#2}}%
249 \mhcurrentrepos\mh@@repos}
250 \newcommand\mhinput[2][]{\def\@repos{#1}%
251 \edef\mh@currentrepos}%
252 \ifx\@repos\@empty\else\mhcurrentrepos{#1}\fi%
253 \input{\MathHub{\mh@currentrepos/source/#2}}%
254 \mhcurrentrepos\mh@@repos}
255 \newenvironment{importmhmodulevia}[3][]{\def\@repos{#1}%
256 \edef\mh@@repos{\mh@currentrepos}%
257 \ifx\@repos\@empty\else\mhcurrentrepos{#1}\fi%
258 \end{figure} $$258 \end{fi
259 \begin{importmoduleenv}[load=\MathHub{\mh@currentrepos/source/#2}]{#3}}
260 {\end{importmoduleenv}\aftergroup\@@doit}
261 ');
262 (/modules.ltxml)
```

#### 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.

```
263 (*omtext)
             264 \ProvidesPackage{omtext-mh}[2015/11/04 v1.0 MathHub support for the sTeX omtext package]
             265 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{omtext}}
             266 \ProcessOptions
             267 \RequirePackage{omtext}
             268 \RequirePackage{mathhub}
             269 (/omtext)
             270 (*omtext.ltxml)
             271 DeclareOption(undef,sub{PassOptions('omtext','sty',ToString(Digest(T_CS('\CurrentOption')))); }
             272 ProcessOptions();
             273 RequirePackage('omtext');
             274 RequirePackage('mathhub');
             275 (/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.
             276 (*omtext)
             277 \addmetakey{Gin}{mhrepos}
             278 \newcommand\mhgraphics[2][]{\metasetkeys{Gin}{#1}%
             279 \edef\mh@@repos{\mh@currentrepos}%
             280 \ifx\Gin@mhrepos\@empty\mygraphics[#1]{\MathHub{\mh@currentrepos/source/#2}}%
             281 \else\mygraphics[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
             282 \def\Gin@mhrepos{}\mhcurrentrepos\mh@@repos}
```

```
283 \newcommand\mhcgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}}
284 \newcommand\mhbgraphics[2][]{\fbox{\mhgraphics[#1]{#2}}}
285 \newcommand\mhcbgraphics [2] [] \begin{center} fbox{\mhgraphics [#1] {#2}} \end{center} for each of the property o
286 (/omtext)
287 (*omtext.ltxml)
288 sub mhgraphics {
              my ($gullet,$keyval,$arg2) = 0_;
290
              my $repo_path;
              if ($keyval) {
291
                     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
292
293
              if (! $repo_path) {
                     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
294
               else {
295
                     $keyval->setValue('mhrepos',undef); }
296
             my $mathhub_base = ToString(Digest('\MathHub{}'));
297
             my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
298
299 return Invocation(T_CS('\@includegraphicx'), $keyval, T_OTHER($finalpath)); }#$
300 DefKeyVal('Gin', 'mhrepos', 'Semiverbatim');
301 DefMacro('\mhgraphics OptionalKeyVals:Gin {}', \&mhgraphics);
302 DefMacro('\mhcgraphics []{}','\begin{center}\mhgraphics[#1]{#2}\end{center}');
303 DefMacro('\mhbgraphics []{}','\fbox{\mhgraphics[#1]{#2}}');
304 (/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.
```

```
305 (*smultiling)
          306 \ProvidesPackage{smultiling-mh}[2015/11/04 v1.0 MathHub support for the sTeX smultiling package
          307 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{smultiling}}
          308 \ProcessOptions
          309 \RequirePackage{smultiling}
          310 \RequirePackage{mathhub}
          311 (/smultiling)
          312 (*smultiling.ltxml)
          313 DeclareOption(undef,sub{PassOptions('smultiling','sty',ToString(Digest(T_CS('\CurrentOption')))
          314 ProcessOptions();
          315 RequirePackage('smultiling');
          316 RequirePackage('mathhub');
          317 (/smultiling.ltxml)
mhmodnl:*
          318 (*smultiling)
          319 \addmetakey{mhmodnl}{repos}
          320 \addmetakey{mhmodnl}{path}
          321 \addmetakey*{mhmodnl}{title}
```

322 \addmetakey\*{mhmodnl}{creators}
323 \addmetakey\*{mhmodnl}{contributors}
324 \addmetakey{mhmodnl}{srccite}

```
325 \addmetakey{primary}{mhmodnl}[yes]
        326 (/smultiling)
        327 (*smultiling.ltxml)
        328 DefKeyVal('mhmodnl','title','Semiverbatim');
        329 DefKeyVal('mhmodnl', 'repos', 'Semiverbatim');
        330 DefKeyVal('mhmodnl','path','Semiverbatim');
        331 DefKeyVal('mhmodnl','creators','Semiverbatim');
        332 DefKeyVal('mhmodnl', 'contributors', 'Semiverbatim');
        333 DefKeyVal('mhmodnl', 'primary', 'Semiverbatim');
        334 (/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.
        335 (*smultiling)
        336 \newenvironment{mhmodnl}[3][]{\metasetkeys{mhmodnl}{#1}\%
        338 \edef\@repos{\ifx\mhmodnl@repos\@empty\mh@currentrepos\else\mhmodnl@repos}
        339 \if@langfiles\importmhmodule[repos=\@repos,load=#2,ext=tex]{#2}\else
        340 \ifx\mhmodnl@load\@empty\importmodule{#2}\else\importmodule[ext=tex,load=\mhmodnl@load]{#2}\fi%
        341 \fi}
        342 {\end{module}}
        343 (/smultiling)
        344 (*smultiling.ltxml)
        345 DefEnvironment('{mhmodnl} OptionalKeyVals:mhmodnl {}{}',
                    "?#excluded()(<omdoc:theory xml:id='#2.#3' >"
        346
                        "?&defined(&GetKeyVal(#1,'creators'))(<dc:creator>&GetKeyVal(#1,'creators')</dc:cr
        347
                        "?&defined(&GetKeyVal(#1,'title'))(<dc:title>&GetKeyVal(#1,'title')</dc:title>)()"
        348
                        "?&defined(&GetKeyVal(#1,'contributors'))(<dc:contributor>&GetKeyVal(#1,'contribut
        349
                        "<omdoc:imports from='?%GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'
        350
                        "#body"
        351
                    . "</omdoc:theory>)",
        352
             afterDigestBegin=>sub {
        353
               my ($stomach, $whatsit) = @_;
        354
               my $keyval = $whatsit->getArg(1);
        355
               my $signature = ToString($whatsit->getArg(2));
        356
               my $language = ToString($whatsit->getArg(3));
        357
        358
               my $repos = ToString(GetKeyVal($keyval,'torepos'));
               my $current_repos = LookupValue('current_repos');
        359
               if (!$repos) { $repos = $current_repos; }
        360
               my $defpaths = LookupValue('defpath');
        361
               my $load_path = ($$defpaths{MathHub}).$repos.'/source/'.$signature;
        362
        363
               if ($keyval) {
        364
                 # If we're not given load, AND the langfiles option is in effect,
        365
        366
                 # default to #2
                 if ((! $keyval->getValue('path')) && (LookupValue('smultiling_langfiles'))) {
        367
                   $keyval->setValue('load',$load_path); }
        368
                 # Always load a TeX file
        369
        370
                 $keyval->setValue('ext','tex');
        371
                 $keyval->setValue('id', "$signature.$language"); }
```

```
module_afterDigestBegin(@_);
          372
          373
                  importmoduleI(@_);
                  return; },
          374
                afterDigest=>sub {
          375
                  module_afterDigest(@_); });
          376
          377 (/smultiling.ltxml)%$
mhviewsig The mhviewsig environment is just a layer over the mhview environment with the
           keys suitably adapted.
          378 \newenvironment{mhviewsig}[4][]{\def\@test{#1}\ifx\@test\@empty%
          379 \begin{mhview}[id=#2,ext=tex]{#3}{#4}\else\begin{mhview}[id=#2,#1,ext=tex]{#3}{#4}\fi}
          380 {\end{mhview}}
          381 (*smultiling | smultiling.ltxml)
          382 (smultiling.ltxml)');
           The mhviewnl environment is just a layer over the mhviewsketch environment
           with the keys and language suitably adapted.<sup>4</sup>
          383 \newenvironment{mhviewnl}[5][]{\def\@test{#1}\ifx\@test\@empty%
          384 \pm 1.43{#5}\else%
          385 \begin{mhviewsketch}[id=#2.#3,#1,ext=tex]{#4}{#5}\fi}
          386 {\end{mhviewsketch}}
          387 \; \langle / \mathsf{smultiling} \; | \; \mathsf{smultiling.ltxml} \rangle
          388 (smultiling.ltxml)');
                   mikoslides-mh: Support for MiKo Slides
           4.5
           We set up package options and pass them on to the mikoslides package, which
           we also load.
          389 (*mikoslides)
          390 \ProvidesPackage{mikoslides-mh}[2015/11/04 v1.0 MathHub support for the sTeX mikoslides package
          391 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{mikoslides}}
          392 \ProcessOptions
          393 \RequirePackage{mikoslides}
          394 \RequirePackage{mathhub}
          395 (/mikoslides)
          396 (*mikoslides.ltxml)
          397 DeclareOption(undef, sub{PassOptions('mikoslides','sty',ToString(Digest(T_CS('\CurrentOption')))
          398 ProcessOptions();
          399 RequirePackage('mikoslides');
          400 RequirePackage('mathhub');
```

\mhframeimage

401 (/mikoslides.ltxml)

EdN:4

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

```
402 \( mikoslides \) \( \addmetakey \) \( \Gin \) \( \frac{mhrepos}{mikoslides. \] \( \text{txml} \) \( \Def Key Val ('Gin', 'mhrepos', 'Semiverbatim') \) \( \text{ty} \) \(
```

 $<sup>^4\</sup>mathrm{EDNote}$ : 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  $.\langle lang \rangle$ !

```
404 (mikoslides.ltxml)RawTeX('
405 (*mikoslides.ltxml | mikoslides)
406 \newcommand\mhframeimage[2][]{%
     \metasetkeys{Gin}{#1}%
407
     \edef\mh@currentrepos}%
408
409
     \ifx\Gin@mhrepos\@empty%
410
       \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%
411
     \else%
       \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%
412
     \fi%
413
414 }%
415 (/mikoslides.ltxml | mikoslides)
416 (mikoslides.ltxml)');
```

### 4.6 **problem-mh**: Support for Problems

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

```
417 (*problem)
418 \ProvidesPackage{problem-mh}[2015/11/04 v1.0 MathHub support for the sTeX problem package]
419 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{problem}}
420 \ProcessOptions
421 \RequirePackage{problem}
422 \RequirePackage{mathhub}
423 \langle/problem\text{423 \langle problem}
424 \langle*problem.ltxml\text{425 DeclareOption(undef,sub{PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))));
426 \text{ProcessOptions();
427 \text{RequirePackage('problem');
428 \text{RequirePackage('mathhub');
429 \langle/problem.ltxml\rangle}
```

\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.

```
430 \*problem\>
431 \newcommand\includemhproblem[2][]{\metasetkeys{inclprob}{#1}%
432 \edef\mh@@repos{\mh@currentrepos}%
433 \ifx\inclprob@mhrepos\@empty\else\mhcurrentrepos\inclprob@mhrepos\fi%
434 \input{\MathHub{\mh@currentrepos/source/#2}}%
435 \mhcurrentrepos\mh@@repos\clear@inclprob@keys}
436 \/problem\>
437 \{*problem.ltxml\}
438 sub includemhproblem {
439  my ($gullet,$keyval,$arg2) = @_;
440  my $repo_path;
411  if ($keyval) {
422  $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
```

### 4.7 hwexam-mh: Support for Assignments

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

```
453 \*hwexam\\
454 \ProvidesPackage{hwexam-mh}[2015/11/04 v1.0 MathHub support for the sTeX hwexam package]
455 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{hwexam}}
456 \ProcessOptions
457 \RequirePackage{hwexam}
458 \RequirePackage{mathhub}
459 \/hwexam\\
460 \*hwexam.ltxml\\
461 DeclareOption(undef,sub{PassOptions('hwexam','sty',ToString(Digest(T_CS('\CurrentOption'))));}
462 ProcessOptions();
463 RequirePackage('hwexam');
464 RequirePackage('mathhub');
465 \/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.

```
466 (*package)
467 \newcommand\includemhassignment[2][]{\metasetkeys{inclassig}{#1}\%
468 \edef\mh@currentrepos}%
469 \ifx\inclassig@mhrepos\@empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
470 \includeassignment [#1] {\MathHub{\mh@currentrepos/source/#2}}%
471 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
472 (/package)
473 (*ltxml)
474 sub includemhassignment {
    my ($gullet,$keyval,$arg2) = @_;
    my $repo_path;
477
    if ($keyval) {
478
       $repo_path = ToString(GetKeyVal($keyval, 'mhrepos')); }
479
    if (! $repo_path) {
       $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
480
481
     else {
```

```
$keyval->setValue('mhrepos',undef); }
                                                  482
                                                               my $mathhub_base = ToString(Digest('\MathHub{}'));
                                                  483
                                                               my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
                                                  485 return Invocation(T_CS('\includeassignment'), $keyval, T_OTHER($finalpath)); }#$
                                                  486 DefKeyVal('inclprob', 'mhrepos', 'Semiverbatim');
                                                  487 DefMacro('\includemhassignment OptionalKeyVals:inclprob {}', \&includemhassignment);
                                                  488 (/ltxml)
\inputmhassignment analogous
                                                  489 (*package)
                                                  490 \newcommand\inputmhassignment[2][]{\metasetkeys{inclassig}{#1}%
                                                  491 \edef\mh@currentrepos}%
                                                  492 \verb|\| ifx\| classig@mhrepos\\| else\\| mhcurrentrepos\\| inclassig@mhrepos\\| fi\%| else\\| else\\| fi\%| else\\| else| else\\| else| else\\| else\\| else\\| else| else\\| else| else| else| else\\| else| else| else| else\\| else| else
                                                  493 \inputassignment[#1] {\MathHub{\mh@currentrepos/source/#2}}%
                                                  494 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
                                                  495 (/package)
                                                  496 (*ltxml)
                                                  497 sub inputmhassignment {
                                                               my ($gullet,$keyval,$arg2) = 0_;
                                                  499
                                                                my $repo_path;
                                                  500
                                                                if ($keyval) {
                                                                      $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
                                                                if (! $repo_path) {
                                                  502
                                                                      $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
                                                  503
                                                  504
                                                                      $keyval->setValue('mhrepos',undef); }
                                                  505
                                                               my $mathhub_base = ToString(Digest('\MathHub{}'));
                                                              my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
                                                             return Invocation(T_CS('\inputassignment'), $keyval, T_OTHER($finalpath)); }#$
                                                  509 DefMacro('\inputmhassignment OptionalKeyVals:inclprob {}', \&inputmhassignment);
                                                  510 (/ltxml)
```

#### 4.8 Finale

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

```
511 (*ltxml)
512 1;
513 (/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 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).