

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>structview-mh</code> : MH Variants for Structures and Views	4
2.6	<code>mikoslides-mh</code> : Support for MiKo Slides	4
2.7	<code>problem-mh</code> : Support for Problems	5
2.8	<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	10
4.4	<code>smultiling-mh</code> : MH Variants for Multilinguality	11
4.5	<code>structview-mh</code> : MH Variants for Structures and Views	13

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

4.6	mikoslides-mh: Support for MiKo Slides	17
4.7	problem-mh: Support for Problems	18
4.8	hwexam-mh: Support for Assignments	19
4.9	Finale	21

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 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 in each module, since the `\importmhmodule` macro sets the current repository automatically.

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

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

2.3 omtex-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 structview-mh: MH Variants for Structures and Views

3

2.6 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}}
```

¹EDNOTE: needs to be documented

²EDNOTE: mhmodsig seems to be missing what happened?

³EDNOTE: needs to be documented

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}
```

2.7 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`.

2.8 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 `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 `\endpackage`) and the \LaTeX ML bindings (between `\beginltxml` and `\endltxml`). 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 \LaTeX ML binding files and the base package.

```

1 \beginltxml | modules.ltxml | omtex.ltxml | smultiling.ltxml | mikoslides.ltxml | problem.ltxml | hwexam.ltxml
2 \package \LaTeXML::Package::Pool;
3 use strict;
4 use \LaTeXML::Package;
5 \endltxml | modules.ltxml | omtex.ltxml | smultiling.ltxml | mikoslides.ltxml | problem.ltxml | hwexam.ltxml
6 \package\ProvidesPackage{mathhub}[2015/11/04 v1.0 sTeX Support for MathHub.info]

7 \beginpackage
8 \DeclareOption*{}
9 \ProcessOptions
10 \endpackage
11 \beginltxml
12 \DeclareOption(undef,sub {});
13 \ProcessOptions();
14 \endltxml

```

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

```

15 \beginpackage
16 \RequirePackage{keyval}
17 \endpackage
18 \beginltxml
19 \RequirePackage('keyval');
20 \endltxml

```

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 \beginpackage
22 \newcommand\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 \newcommand\@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` 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 \newcommand\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 }%
89 </modules>
90 <*modules.ltxml>
91 DefKeyVal('importmhmodule','id','Semiverbatim');
92 DefKeyVal('importmhmodule','repos','Semiverbatim');
93 DefKeyVal('importmhmodule','path','Semiverbatim');
94 DefKeyVal('importmhmodule','ext','Semiverbatim');
95 DefKeyVal('importmhmodule','conservative','Semiverbatim');
96 DefConstructor('\importmhmodule OptionalKeyVals:importmhmodule {}',
97   "<omdoc:imports "
98   . "from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))(###2'"
99   . "?&defined(&GetKeyVal(#1,'conservative'))(load='&GetKeyVal(#1,'conservative'))'"
100   afterDigest => \&importMHmoduleI);
101
102 DefConstructor('\usemhmodule OptionalKeyVals:importmhmodule {}',
103   "<omdoc:uses from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))(###"
104   afterDigest => \&importMHmoduleI);
105
106 sub importMHmoduleI {
107   my ($stomach, $whatsit) = @_;
108   my $keyval = $whatsit->getArg(1);
109   my $id = $whatsit->getArg(2);
110   if ($keyval) {
111     my $repos = ToString($keyval->getValue('repos'));
112     my $path = ToString($keyval->getValue('path'));
113     my $current_repos = LookupValue('current_repos');
114     if (!$repos) { # Use the implicit current repository
115       $repos = $current_repos; }

```



```

116 my $defpaths = LookupValue('defpath');
117 my $load_path = ($$defpaths[MathHub]).$repos.'/source/'$.path;
118 $keyval->setValue('load',$load_path);
119 AssignValue('current_repos' => $repos, 'global');
120 importmoduleI($stomach,$whatsit);
121 AssignValue('current_repos' => $current_repos, 'global'); }
122 else {
123 importmoduleI($stomach,$whatsit); }
124 return; }
125
126 DefConstructor('\importMHmoduleI OptionalKeyVals:importmhmodule {}', '',
127 afterDigest=> \&importMHmoduleI );#$
128 </modules.ltxml>

```

and now the analogs

\usemhmodule

```

129 \newcommand\usemhmodule[2] []{%
130 \metasetkeys{importmhmodule}{#1}%
131 \ifx\importmhmodule@path\empty%
132 \usemodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
133 \else%
134 \edef\mh@@repos{\mh@currentrepos}%
135 \ifx\importmhmodule@repos\empty%
136 \else%
137 \mhcurrentrepos{\importmhmodule@repos}%
138 \fi%
139 \usemodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodule@
140 \mhcurrentrepos\mh@@repos%
141 \fi%
142 \ignorespaces%
143 }%

```

\mhinputref

```

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

```

\mhinput

```
159 \let\mhinput\mhinputref%
```

4.3 omtex-mh: MH Variants for OMTex

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

```
160 <omtex>
161 \ProvidesPackage{omtex-mh}[2015/11/04 v1.0 MathHub support for the sTeX omtex package]
162 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{omtex}}
163 \ProcessOptions
164 \RequirePackage{omtex}
165 \RequirePackage{mathhub}
166 </omtex>
167 <omtex.ltxml>
168 \DeclareOption(undef,sub{PassOptions('omtex','sty',ToString(Digest(T_CS('CurrentOption')))); }
169 \ProcessOptions();
170 \RequirePackage('omtex');
171 \RequirePackage('mathhub');
172 </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.

```
173 <omtex>
174 \addmetakey{Gin}{mhrepos}
175 \newcommand\mhgraphics[2][]{\metasetkeys{Gin}{#1}%
176 \edef\mh@crepos{\mh@currentrepos}%
177 \ifx\Gin@mhrepos\empty\mygraphics[#1]{\MathHub{\mh@currentrepos/source/#2}}%
178 \else\mygraphics[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
179 \def\Gin@mhrepos{\mhcurrentrepos\mh@crepos}
180 \newcommand\mhgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}}
181 \newcommand\mhgraphics[2][]{\fbox{\mhgraphics[#1]{#2}}}
182 \newcommand\mhgraphics[2][]{\begin{center}\fbox{\mhgraphics[#1]{#2}}\end{center}}
183 </omtex>
184 <omtex.ltxml>
185 sub mhgraphics {
186   my ($gullet,$keyval,$arg2) = @_;
187   my $repo_path;
188   if ($keyval) {
189     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
190   if (! $repo_path) {
191     $repo_path = ToString(Digest(T_CS('mh@currentrepos'))); }
192   else {
193     $keyval->setValue('mhrepos',undef); }
194   my $mathhub_base = ToString(Digest('\MathHub{'}));
195   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
196   return Invocation(T_CS('@includegraphicx'), $keyval, T_OTHER($finalpath)); }#
197 DefKeyVal('Gin','mhrepos','Semiverbatim');
198 DefMacro('mhgraphics OptionalKeyVals:Gin {}', \&mhgraphics);
```

```

199 DefMacro('mhcgraphics []{}','\begin{center}\mhgraphics[#1]{#2}\end{center}');
200 DefMacro('mhbgraphics []{}','\fbox{\mhgraphics[#1]{#2}}');
201 </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.

```

202 <*smultiling>
203 \ProvidesPackage{smultiling-mh}[2015/11/04 v1.0 MathHub support for the sTeX smultiling package]
204 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{smultiling}}
205 \ProcessOptions
206 \RequirePackage{smultiling}
207 \RequirePackage{mathhub}
208 </smultiling>
209 <*smultiling.ltxml>
210 DeclareOption(undef,sub{PassOptions('smultiling','sty',ToString(Digest(T_CS('CurrentOption'))))
211 ProcessOptions();
212 RequirePackage('smultiling');
213 RequirePackage('mathhub');
214 </smultiling.ltxml>

mhmodnl:*

215 <*smultiling>
216 \addmetakey{mhmodnl}{repos}
217 \addmetakey{mhmodnl}{path}
218 \addmetakey*{mhmodnl}{title}
219 \addmetakey*{mhmodnl}{creators}
220 \addmetakey*{mhmodnl}{contributors}
221 \addmetakey{mhmodnl}{srccite}
222 \addmetakey{primary}{mhmodnl}[yes]
223 </smultiling>
224 <*smultiling.ltxml>
225 DefKeyVal('mhmodnl','title','Semiverbatim');
226 DefKeyVal('mhmodnl','repos','Semiverbatim');
227 DefKeyVal('mhmodnl','path','Semiverbatim');
228 DefKeyVal('mhmodnl','creators','Semiverbatim');
229 DefKeyVal('mhmodnl','contributors','Semiverbatim');
230 DefKeyVal('mhmodnl','primary','Semiverbatim');
231 </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.

232 <*smultiling>
233 \newenvironment{mhmodnl}[3][{}]{\metasetkeys{mhmodnl}{#1}%
234 \def\@test{#1}\ifx\@test\@empty\begin{module}[id=#2.#3]\else\begin{module}[id=#2.#3,#1]\fi%
235 \edef\@repos{\ifx\mhmodnl@repos\@empty\mh@currentrepos\else\mhmodnl@repos}
236 \if@langfiles\importmhmodule[repos=\@repos,load=#2,ext=tex]{#2}\else
237 \ifx\mhmodnl@load\@empty\importmodule{#2}\else\importmodule[ext=tex,load=\mhmodnl@load]{#2}\fi%

```

```

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

```

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

```

275 \smultiling.ltxml)RawTeX(
276 \smultiling | smultiling.ltxml)
277 \newenvironment{mhviewsig}[4][\def\@test{#1}\ifx\@test\empty%
278 \begin{mhview}[id=#2,ext=tex]{#3}{#4}\else\begin{mhview}[id=#2,#1,ext=tex]{#3}{#4}\fi}
279 {\end{mhview}}

```

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

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*!

```

280 \newenvironment{mhviewnl}[5][]{\def\@test{#1}\ifx\@test\@empty%
281 \begin{mhviewsketch}[id=#2.#3,ext=tex]{#4}{#5}\else%
282 \begin{mhviewsketch}[id=#2.#3,#1,ext=tex]{#4}{#5}\fi}
283 {\end{mhviewsketch}}
284 \</smultiling | smultiling.ltxml>
285 \<smultiling.ltxml>');

```

4.5 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.

```

286 \<structview>
287 \ProvidesPackage{structview-mh}[2015/11/04 v1.0 MathHub support for the sTeX structview package]
288 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{structview}}
289 \ProcessOptions
290 \RequirePackage{structview}
291 \RequirePackage{mathhub}
292 \</structview>
293 \<structview.ltxml>
294 \DeclareOption{undef,sub{PassOptions('structview','sty',ToString(Digest(T_CS('\CurrentOption'))))}}
295 \ProcessOptions();
296 \RequirePackage('structview');
297 \RequirePackage('mathhub');
298 \</structview.ltxml>

```

`importmhmodulevia`

```

299 \<modules.ltxml>RawTeX('
300 \<modules | modules.ltxml>
301 \newenvironment{importmhmodulevia}[3][]{%
302 \gdef\@doit{\importmhmodule[#1]{#2}{#3}}%
303 \ifmod@show\par\noindent importing module #2 via \@doit\fi
304 }{%
305 \aftergroup\@doit\ifmod@show end import\fi%
306 }%
307 \</modules | modules.ltxml>
308 \<modules.ltxml>');

309 \<structview>
310 \srefaddidkey{mhview}
311 \addmetakey{mhview}{display}
312 \addmetakey{mhview}{creators}
313 \addmetakey{mhview}{contributors}
314 \addmetakey{mhview}{srccite}
315 \addmetakey*{mhview}{title}
316 \addmetakey{mhview}{fromrepos}
317 \addmetakey{mhview}{torepos}
318 \addmetakey{mhview}{frompath}
319 \addmetakey{mhview}{topath}
320 \addmetakey[sms]{mhview}{ext}

```

mhview the MathHub version

```

321 \newenvironment{mhview}[3][]{% keys, from, to
322   \metasetkeys{mhview}{#1}%
323   \sref@target%
324   \begin{@mhview}{#2}{#3}%
325   \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
326 }{%
327   \end{@mhview}%
328   \ignorespaces%
329 }%
330 \ifmod@show\surroundwithmdframed{mhview}\fi

```

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

```

331 \newenvironment{@mhview}[2]{%from, to
332   \importmhmodule[repos=\mhview@fromrepos,path=\mhview@frompath,ext=\mhview@ext]{#1}%
333   \importmhmodule[repos=\mhview@torepos,path=\mhview@topath,ext=\mhview@ext]{#2}%
334 }{}%

```

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

```

335 \newenvironment{mhviewsketch}[3][]{%
336   \metasetkeys{mhview}{#1}%
337   \sref@target%
338   \begin{@mhview}{#2}{#3}%
339   \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
340 }{%
341   \end{@mhview}%
342   \ignorespaces%
343 }%
344 \ifmod@show\surroundwithmdframed{mhviewsketch}\fi
345 \</structview>

346 <*modules>
347 \srefaddidkey{mhview}
348 \addmetakey{mhview}{display}
349 \addmetakey{mhview}{creators}
350 \addmetakey{mhview}{contributors}
351 \addmetakey{mhview}{srccite}
352 \addmetakey*{mhview}{title}
353 \addmetakey{mhview}{fromrepos}
354 \addmetakey{mhview}{torepos}
355 \addmetakey{mhview}{frompath}
356 \addmetakey{mhview}{topath}
357 \addmetakey[sms]{mhview}{ext}

```

mhview the MathHub version

```

358 \newenvironment{mhview}[3][]{% keys, from, to
359   \metasetkeys{mhview}{#1}%
360   \sref@target%
361   \begin{@mhview}{#2}{#3}%

```

```

362 \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
363 }{%
364 \end{@mhview}%
365 \ignorespaces%
366 }%
367 \ifmod@show\surroundwithmdframed{mhview}\fi

```

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

```

368 \newenvironment{@mhview}[2]{%from, to
369 \importmhmodule[repos=\mhview@fromrepos,path=\mhview@frompath,ext=\mhview@ext]{#1}%
370 \importmhmodule[repos=\mhview@torepos,path=\mhview@topath,ext=\mhview@ext]{#2}%
371 }{}%

```

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

```

372 \newenvironment{mhviewsketch}[3][]{%
373 \metasetkeys{mhview}{#1}%
374 \sref@target%
375 \begin{@mhview}{#2}{#3}%
376 \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
377 }{%
378 \end{@mhview}%
379 \ignorespaces%
380 }%
381 \ifmod@show\surroundwithmdframed{mhviewsketch}\fi
382 </modules>

```

5

```

383 <*modules.ltxml>
384 DefKeyVal('mhview','id','Semiverbatim');
385 DefKeyVal('mhview','fromrepos','Semiverbatim');
386 DefKeyVal('mhview','torepos','Semiverbatim');
387 DefKeyVal('mhview','frompath','Semiverbatim');
388 DefKeyVal('mhview','topath','Semiverbatim');
389 DefKeyVal('mhview','title','Semiverbatim');
390 DefKeyVal('mhview','creators','Semiverbatim');
391 DefKeyVal('mhview','contributors','Semiverbatim');
392 DefKeyVal('mhview','display','Semiverbatim');
393 DefKeyVal('mhview','ext','Semiverbatim');
394 DefMacroI(T_CS('\begin{mhview}'),'OptionalKeyVals:mhview {}{}', sub {
395 my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
396 my $from = ToString(Digest($from_arg));
397 my $to = ToString(Digest($to_arg));
398 AssignValue(from_module => $from);
399 AssignValue(to_module => $to);
400 my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
401 my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
402 my $repos = LookupValue('current_repos');
403 my $from_path = ToString(GetKeyVal($keyvals,'frompath'));

```

⁵EDNOTE: MK: sort these into the rest.

```

404 my $to_path = ToString(GetKeyVal($keyvals,'topath'));
405 my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
406 $ext = 'sms' unless $ext;
407 my $current_repos = LookupValue('current_repos');
408 if (!$from_repos) { $from_repos = $current_repos; }
409 if (!$to_repos) { $to_repos = $current_repos; }
410 return (
411     Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
412     Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
413     Invocation(T_CS('\begin{viewenv}'),$keyvals,$from_arg,$to_arg)->unlist
414 );
415 });
416 DefMacroI('\end{mhview}',undef,'\end{viewenv}');
417
418 DefMacroI(T_CS('\begin{mhviewsketch}'),'OptionalKeyVals:mhview {}{}', sub {
419     my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
420     my $from = ToString(Digest($from_arg));
421     my $to = ToString(Digest($to_arg));
422     my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
423     my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
424     my $repos = LookupValue('current_repos');
425     my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
426     my $to_path = ToString(GetKeyVal($keyvals,'topath'));
427     my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
428     $ext = 'sms' unless $ext;
429     my $current_repos = LookupValue('current_repos');
430     if (!$from_repos) { $from_repos = $current_repos; }
431     if (!$to_repos) { $to_repos = $current_repos; }
432     return (
433         Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
434         Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
435         Invocation(T_CS('\begin{viewsketchenv}'),$keyvals,$from_arg,$to_arg)->unlist
436     );
437 });
438 DefMacroI('\end{mhviewsketch}',undef,'\end{viewsketchenv}');
439 </modules.ltxml>
440 <*structview.ltxml>
441 DefKeyVal('mhview','id','Semiverbatim');
442 DefKeyVal('mhview','fromrepos','Semiverbatim');
443 DefKeyVal('mhview','torepos','Semiverbatim');
444 DefKeyVal('mhview','frompath','Semiverbatim');
445 DefKeyVal('mhview','topath','Semiverbatim');
446 DefKeyVal('mhview','title','Semiverbatim');
447 DefKeyVal('mhview','creators','Semiverbatim');
448 DefKeyVal('mhview','contributors','Semiverbatim');
449 DefKeyVal('mhview','display','Semiverbatim');
450 DefKeyVal('mhview','ext','Semiverbatim');
451 DefMacroI(T_CS('\begin{mhview}'),'OptionalKeyVals:mhview {}{}', sub {
452     my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
453     my $from = ToString(Digest($from_arg));

```



```

454 my $to = ToString(Digest($to_arg));
455 AssignValue(from_module => $from);
456 AssignValue(to_module => $to);
457 my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
458 my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
459 my $repos = LookupValue('current_repos');
460 my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
461 my $to_path = ToString(GetKeyVal($keyvals,'topath'));
462 my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
463 $ext = 'sms' unless $ext;
464 my $current_repos = LookupValue('current_repos');
465 if (!$from_repos) { $from_repos = $current_repos; }
466 if (!$to_repos) { $to_repos = $current_repos; }
467 return (
468     Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
469     Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
470     Invocation(T_CS('\begin{viewenv}'),$keyvals,$from_arg,$to_arg)->unlist
471 );
472 });
473 DefMacroI('\end{mhview}',undef,'\end{viewenv}');
474
475 DefMacroI(T_CS('\begin{mhviewsketch}'),'OptionalKeyVals:mhview {}{}', sub {
476     my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
477     my $from = ToString(Digest($from_arg));
478     my $to = ToString(Digest($to_arg));
479     my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
480     my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
481     my $repos = LookupValue('current_repos');
482     my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
483     my $to_path = ToString(GetKeyVal($keyvals,'topath'));
484     my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
485     $ext = 'sms' unless $ext;
486     my $current_repos = LookupValue('current_repos');
487     if (!$from_repos) { $from_repos = $current_repos; }
488     if (!$to_repos) { $to_repos = $current_repos; }
489     return (
490         Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
491         Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
492         Invocation(T_CS('\begin{viewsketchenv}'),$keyvals,$from_arg,$to_arg)->unlist
493     );
494 });
495 DefMacroI('\end{mhviewsketch}',undef,'\end{viewsketchenv}');
496 </structview.ltxml>

```

4.6 mikoslides-mh: Support for MiKo Slides

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

```

497 <*mikoslides>

```

```

498 \ProvidesPackage{mikoslides-mh}[2015/11/04 v1.0 MathHub support for the sTeX mikoslides package]
499 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{mikoslides}}
500 \ProcessOptions
501 \RequirePackage{mikoslides}
502 \RequirePackage{mathhub}
503 </mikoslides>
504 <*mikoslides.ltxml>
505 DeclareOption(undef,sub{PassOptions('mikoslides','sty',ToString(Digest(T_CS('\CurrentOption'))))}
506 ProcessOptions();
507 RequirePackage('mikoslides');
508 RequirePackage('mathhub');
509 </mikoslides.ltxml>

\mhframeimage Use the current value of \mh@currentrepos or the value of the mhrepos key if it
is given in \frameimage.
510 <mikoslides>\addmetakey{Gin}{mhrepos}
511 <mikoslides.ltxml>DefKeyVal('Gin','mhrepos','Semiverbatim');
512 <mikoslides.ltxml>RawTeX('
513 <*mikoslides.ltxml | mikoslides>
514 \newcommand\mhframeimage[2][{}]{%
515 \metasetkeys{Gin}{#1}%
516 \edef\mh@repos{\mh@currentrepos}%
517 \ifx\Gin@mhrepos\empty%
518 \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%
519 \else%
520 \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%
521 \fi%
522 }%
523 </mikoslides.ltxml | mikoslides>
524 <mikoslides.ltxml>');

```

4.7 problem-mh: Support for Problems

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

```

525 <*problem>
526 \ProvidesPackage{problem-mh}[2015/11/04 v1.0 MathHub support for the sTeX problem package]
527 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{problem}}
528 \ProcessOptions
529 \RequirePackage{problem}
530 \RequirePackage{mathhub}
531 </problem>
532 <*problem.ltxml>
533 DeclareOption(undef,sub{PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))))});
534 ProcessOptions();
535 RequirePackage('problem');
536 RequirePackage('mathhub');
537 </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`.

```

538 <*problem>
539 \newcommand\includemhproblem[2][\metasetkeys{inclprob}{#1}%
540 \edef\mh@@repos{\mh@currentrepos}%
541 \ifx\inclprob@mhrepos\empty\else\mhcurrentrepos\inclprob@mhrepos\fi%
542 \input{\MathHub{\mh@currentrepos/source/#2}}%
543 \mhcurrentrepos\mh@@repos\clear@inclprob@keys}
544 </problem>
545 <*problem.ltxml>
546 sub includemhproblem {
547   my ($gullet,$keyval,$arg2) = @_ ;
548   my $repo_path;
549   if ($keyval) {
550     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
551   if (! $repo_path) {
552     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
553   else {
554     $keyval->setValue('mhrepos',undef); }
555   my $mathhub_base = ToString(Digest('\MathHub{ }'));
556   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
557   return Invocation(T_CS('\includeproblem'), $keyval, T_OTHER($finalpath)); }#$
558 DefKeyVal('inclprob','mhrepos','Semiverbatim');
559 DefMacro('\includemhproblem OptionalKeyVals:inclprob { }', \&includemhproblem);
560 </problem.ltxml>

```

4.8 hwexam-mh: Support for Assignments

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

```

561 <*hwexam>
562 \ProvidesPackage{hwexam-mh}[2015/11/04 v1.0 MathHub support for the sTeX hwexam package]
563 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{hwexam}}
564 \ProcessOptions
565 \RequirePackage{hwexam}
566 \RequirePackage{mathhub}
567 </hwexam>
568 <*hwexam.ltxml>
569 DeclareOption(undef,sub{PassOptions('hwexam','sty',ToString(Digest(T_CS('\CurrentOption')))); }
570 ProcessOptions();
571 RequirePackage('hwexam');
572 RequirePackage('mathhub');
573 </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`.

```

574 <*package>
575 \newcommand\includemhassignment[2][\metasetkeys{inclassig}{#1}%
576 \edef\mh@@repos{\mh@currentrepos}%
577 \ifx\inclassig@mhrepos\empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
578 \includeassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}%
579 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
580 </package>
581 <*ltxml>
582 sub includemhassignment {
583   my ($gullet,$keyval,$arg2) = @_;
584   my $repo_path;
585   if ($keyval) {
586     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
587   if (! $repo_path) {
588     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
589   else {
590     $keyval->setValue('mhrepos',undef); }
591   my $mathhub_base = ToString(Digest('\MathHub{'}));
592   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
593   return Invocation(T_CS('\includeassignment'), $keyval, T_OTHER($finalpath)); }#$
594 DefKeyVal('inclprob','mhrepos','Semiverbatim');
595 DefMacro('\includemhassignment OptionalKeyVals:inclprob {}', \&includemhassignment);
596 </ltxml>

```

`\inputmhassignment` analogous

```

597 <*package>
598 \newcommand\inputmhassignment[2][\metasetkeys{inclassig}{#1}%
599 \edef\mh@@repos{\mh@currentrepos}%
600 \ifx\inclassig@mhrepos\empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
601 \inputassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}%
602 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
603 </package>
604 <*ltxml>
605 sub inputmhassignment {
606   my ($gullet,$keyval,$arg2) = @_;
607   my $repo_path;
608   if ($keyval) {
609     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
610   if (! $repo_path) {
611     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
612   else {
613     $keyval->setValue('mhrepos',undef); }
614   my $mathhub_base = ToString(Digest('\MathHub{'}));
615   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
616   return Invocation(T_CS('\inputassignment'), $keyval, T_OTHER($finalpath)); }#$
617 DefMacro('\inputmhassignment OptionalKeyVals:inclprob {}', \&inputmhassignment);
618 </ltxml>

```

4.9 Finale

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

```
619 <*!xml>  
620 1;  
621 </!xml>
```

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).