MathHub Support for STEX*

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

November 10, 2015

Abstract

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

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

Contents

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

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

4.5	smultiling-mh: MH Variants for Multilinguality	12
4.6	structview-mh: MH Variants for Structures and Views	14
4.7	mikoslides-mh: Support for MiKo Slides	17
4.8	problem-mh: Support for Problems	18
4.9	hwexam-mh: Support for Assignments	19
4.10	Finale	21

1 Introduction

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

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

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

2 The User Interface

2.1 Package Options

none so far

2.2 modules-mh: MH Variants for Modules

\importmhmodule

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

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

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

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

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

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

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

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

\mhcurrentrepos

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

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

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

2.3 omtext-mh: MH Variants for OMText

\mhcgraphics

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

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

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

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

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

2.4 statements-mh: MH Variants for Statements

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

2.5 smultiling-mh: MH Variants for Multilinguality

1 2

2.6 structview-mh: MH Variants for Structures and Views

3

EdN:1

EdN:3

2.7 mikoslides-mh: Support for MiKo Slides

\mhframeimage

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

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

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

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

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

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

\mhframeimage{baz/foobar}

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

\includemhproblem

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

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

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

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

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

2.9 hwexam-mh: Support for Assignments

\includemhassignment

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

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

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

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

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

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

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

3 Limitations

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

1. none reported yet.

4 Implementation

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

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

```
1 (*Itxml | modules.ltxml | structview.ltxml | omtext.ltxml | statements.ltxml | smultiling.ltxml | mikoslides.ltxml | probler
2 package LaTeXML::Package::Pool;
3 use strict;
4 use LaTeXML::Package;
5 (/ltxml | modules.ltxml | structview.ltxml | omtext.ltxml | statements.ltxml | smultiling.ltxml | mikoslides.ltxml | probler
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 (/ltxml)
   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 \newcommand\mhcurrentrepos[1]{%
23  \edef\0test\#1}%
24  \ifx\0test\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{#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.

```
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[$\langle key=value\ list\rangle$] {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 \newcommand\importmhmodule[2][]{%
     \metasetkeys{importmhmodule}{#1}%
73
     \ifx\importmhmodule@path\@empty% if module name is not set
74
       \importmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
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
         \mhcurrentrepos{\importmhmodule@repos}% change it.
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 }%
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'"
                   . "?&defined(&GetKeyVal(#1,'conservative'))(load='&GetKeyVal(#1,'conservative')'
99
100
      afterDigest => \&importMHmoduleI);
101
102 sub importMHmoduleI {
     my ($stomach, $whatsit) = @_;
103
    my $keyval = $whatsit->getArg(1);
     my $id = $whatsit->getArg(2);
105
106
     if ($keyval) {
       my $repos = ToString($keyval->getValue('repos'));
107
       my $path = ToString($keyval->getValue('path'));
108
       my $current_repos = LookupValue('current_repos');
109
110
       if (!$repos) { # Use the implicit current repository
111
         $repos = $current_repos; }
112
       my $defpaths = LookupValue('defpath');
113
       my $load_path = ($$defpaths{MathHub}).$repos.'/source/'.$path;
       $keyval->setValue('load',$load_path);
114
       AssignValue('current_repos' => $repos, 'global');
115
```

```
importmoduleI($stomach,$whatsit);
             116
                     AssignValue('current_repos' => $current_repos, 'global'); }
             117
             118
                     importmoduleI($stomach,$whatsit); }
             119
                   return; }
             120
             121
             122 DefConstructor('\importMHmoduleI OptionalKeyVals:importmhmodule {}', '',
                    afterDigest=> \&importMHmoduleI );#$
             124 (/modules.ltxml)
              and now the analogs
\usemhmodule
              _{125} \; \langle * modules \rangle
             126 \newcommand\usemhmodule[2][]{%
              127
                   \metasetkeys{importmhmodule}{#1}%
             128
                   \ifx\importmhmodule@path\@empty%
             129
                     \usemodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
             130
                   \else%
                     \edef\mh@@repos{\mh@currentrepos}%
             131
             132
                     \ifx\importmhmodule@repos\@empty%
             133
             134
                       \mhcurrentrepos{\importmhmodule@repos}%
             135
             136
                     \usemodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodule@
             137
                     \mhcurrentrepos\mh@@repos%
             138
                   \fi%
             139
                   \ignorespaces%
             140 }%
             141 (/modules)
             142 (*modules.ltxml)
             143 DefConstructor('\usemhmodule OptionalKeyVals:importmhmodule {}',
                    "<omdoc:uses from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))()###</pre>
             144
             145
                    afterDigest => \&importMHmoduleI);
             146 (/modules.ltxml)
\mhinputref
              147 (modules.ltxml)RawTeX('
             148 (*modules | modules.ltxml)
             149 \newcommand\mhinputref[2][]{%
                   \def\@repos{#1}%
             150
                   \edef\mh@@repos{\mh@currentrepos}%
             151
                   \ifx\@repos\@empty%
             153
                   \else%
                     \mhcurrentrepos{#1}%
             154
             155
                   \inputref{\MathHub{\mh@currentrepos/source/#2}}%
             156
                   \mhcurrentrepos\mh@@repos%
             157
              158
                   \ignorespaces%
             159 }%
```

```
161 (modules.ltxml)');
   \mhinput
            162 (*modules)
            163 \let\mhinput\mhinputref%
            164 (/modules)
                   omtext-mh: MH Variants for OMText
             4.3
             We set up package options and pass them on to the omtext package, which we
             also load.
            165 (*omtext)
            166 \ProvidesPackage{omtext-mh}[2015/11/04 v1.0 MathHub support for the sTeX omtext package]
            167 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{omtext}}
            168 \ProcessOptions
            169 \RequirePackage{mathhub}
            170 \RequirePackage{omtext}
            171 \RequirePackage{modules-mh}
            172 (/omtext)
            173 (*omtext.ltxml)
            174 DeclareOption(undef, sub{PassOptions('omtext', 'sty', ToString(Digest(T_CS('\CurrentOption')))); }
            175 ProcessOptions();
            176 RequirePackage('mathhub');
            177 RequirePackage('omtext');
            178 RequirePackage('modules-mh');
            179 (/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.
            180 (*omtext)
            181 \addmetakey{Gin}{mhrepos}
            182 \newcommand\mhgraphics[2][]{\metasetkeys{Gin}{#1}%
            183 \edef\mh@currentrepos}%
            184 \ifx\Gin@mhrepos\@empty\mygraphics[#1]{\MathHub{\mh@currentrepos/source/#2}}%
            185 \else\mygraphics[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
            186 \def\Gin@mhrepos{}\mhcurrentrepos\mh@@repos}
            187 \newcommand\mhcgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}}
            188 \newcommand\mhbgraphics[2][]{\fbox{\mhgraphics[#1]{#2}}}
            190 (/omtext)
            191 (*omtext.ltxml)
            192 sub mhgraphics {
            193 my ($gullet,$keyval,$arg2) = @_;
                my $repo_path;
            194
                if ($keyval) {
            195
                   $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
            196
            197
                 if (! $repo_path) {
                   $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
            198
```

160 (/modules | modules.ltxml)

4.4 statements-mh: MH Variants for Statements

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

```
209 (*statements)
210 \ProvidesPackage{statements-mh}[2015/11/04 v1.0 MathHub support for the sTeX statements package
211 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{statements}}
212 \ProcessOptions
213 \verb|\RequirePackage{mathhub}|
214 \RequirePackage{statements}
215 \RequirePackage{omtext-mh}
216 (/statements)
217 (*statements.ltxml)
218 DeclareOption(undef, sub{PassOptions('statements','sty',ToString(Digest(T_CS('\CurrentOption')))
219 ProcessOptions();
220 RequirePackage('mathhub');
221 RequirePackage('statements');
222 RequirePackage('omtext-mh');
223 (/statements.ltxml)
224 (*statements)
225 \let\usemhvocab=\usemhmodule
226 (/statements)
227 (*statements.ltxml)
```

4.5 smultiling-mh: MH Variants for Multilinguality

228 DefMacro('\usemhvocab','\usemhmodule');

229 (/statements.ltxml)

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

```
230 \*smultiling\
231 \ProvidesPackage{smultiling-mh}[2015/11/04 v1.0 MathHub support for the sTeX smultiling package
232 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{smultiling}}
233 \ProcessOptions
234 \RequirePackage{mathhub}
235 \RequirePackage{smultiling}
236 \RequirePackage{structview-mh}
```

```
237 (/smultiling)
                                238 (*smultiling.ltxml)
                                239 DeclareOption(undef,sub{PassOptions('smultiling','sty',ToString(Digest(T_CS('\CurrentOption')))
                                240 ProcessOptions();
                                241 RequirePackage('mathhub');
                                242 RequirePackage('smultiling');
                                243 RequirePackage('structview-mh');
                                244 (/smultiling.ltxml)
mhmodnl:*
                                245 (*smultiling)
                                246 \addmetakey{mhmodnl}{repos}
                                247 \addmetakey{mhmodnl}{path}
                                248 \addmetakey*{mhmodnl}{title}
                                249 \addmetakey*{mhmodnl}{creators}
                                250 \addmetakey*{mhmodnl}{contributors}
                                251 \addmetakey{mhmodnl}{srccite}
                                252 \addmetakey{primary}{mhmodnl}[yes]
                                253 (/smultiling)
                                254 (*smultiling.ltxml)
                                255 DefKeyVal('mhmodnl','title','Semiverbatim');
                                256 DefKeyVal('mhmodnl','repos','Semiverbatim');
                                257 DefKeyVal('mhmodnl','path','Semiverbatim');
                                258 DefKeyVal('mhmodnl','creators','Semiverbatim');
                                259 DefKeyVal('mhmodnl','contributors','Semiverbatim');
                                260 DefKeyVal('mhmodnl', 'primary', 'Semiverbatim');
                                261 (/smultiling.ltxml)
                                 The mhmodnl environment is just a layer over the module environment and the
      mhmodnl
                                   \importmhmodule macro with the keys and language suitably adapted.
                                262 (*smultiling)
                                263 \newenvironment{mhmodnl}[3][]{\metasetkeys{mhmodnl}{#1}\%
                                264 \det \text{``dtest$\{\#1\}} ifx \text{``dtest$(@empty$begin{module}[id=\#2.\#3]$) else begin{module}[id=\#2.\#3,\#1] fi\% begin{module}[id=\#2.\#3,\#2] fi\% begin{module}[id=\#2.\#4] fi\% begin{module}[id=\#2.\#4] fi\% begin{module}[id=\#2.\#4] fi\% begin{module}[id=\#2.\#4] f
                                265 \edf \edge {\tt lifx\mbmodnl@repos\edge} \edge {\tt lifx\mbmodnl@repos\e
                                266 \if@langfiles\importmhmodule[repos=\@repos,load=#2,ext=tex]{#2}\else
                                267 \ifx\mhmodnl@load\@empty\importmodule{#2}\else\importmodule[ext=tex,load=\mhmodnl@load]{#2}\fi%
                                268 \fi}
                                269 {\end{module}}
                                270 (/smultiling)
                                271 (*smultiling.ltxml)
                                272 DefEnvironment('{mhmodnl} OptionalKeyVals:mhmodnl {}{}',
                                                                       "?#excluded()(<omdoc:theory xml:id='#2.#3' >"
                                273
                                274
                                                                                   "?&defined(&GetKeyVal(#1,'creators'))(<dc:creator>&GetKeyVal(#1,'creators')</dc:cr
                                                                                   "?&defined(&GetKeyVal(#1,'title'))(<dc:title>&GetKeyVal(#1,'title')</dc:title>)()"
                                275
                                                                                   "?&defined(&GetKeyVal(#1,'contributors'))(<dc:contributor>&GetKeyVal(#1,'contribut
                                276
                                                                                   "<omdoc:imports from='?%GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'
                                277
                                                                                   "#body"
                                278
                                                                        . "</omdoc:theory>)",
                                279
                                280
                                                afterDigestBegin=>sub {
                                                       my ($stomach, $whatsit) = @_;
```

```
my $keyval = $whatsit->getArg(1);
          282
                  my $signature = ToString($whatsit->getArg(2));
          283
                  my $language = ToString($whatsit->getArg(3));
          284
                  my $repos = ToString(GetKeyVal($keyval,'torepos'));
          285
                  my $current_repos = LookupValue('current_repos');
          286
          287
                  if (!$repos) { $repos = $current_repos; }
          288
                  my $defpaths = LookupValue('defpath');
                  my $load_path = ($$defpaths{MathHub}).$repos.'/source/'.$signature;
          289
          290
                  if ($keyval) {
          291
                    # If we're not given load, AND the langfiles option is in effect,
          292
                    # default to #2
          293
                    if ((! $keyval->getValue('path')) && (LookupValue('smultiling_langfiles'))) {
          294
                      $keyval->setValue('load',$load_path); }
          295
                    # Always load a TeX file
          296
                    $keyval->setValue('ext','tex');
          297
                    $keyval->setValue('id', "$signature.$language"); }
          298
                  module_afterDigestBegin(@_);
          299
          300
                  importmoduleI(@_);
          301
                  return; },
                afterDigest=>sub {
          302
                  module_afterDigest(@_); });
          303
          304 (/smultiling.ltxml)%$
mhviewsig The mhviewsig environment is just a layer over the mhview environment with the
           keys suitably adapted.
          305 \smultiling.ltxml\RawTeX('
          306 (*smultiling | smultiling.ltxml)
          307 \newenvironment{mhviewsig}[4][]{\def\@test{#1}\ifx\@test\@empty%
          308 \begin{mhview}[id=#2,ext=tex]{#3}{#4}\else%
          309 \begin{mhview}[id=#2,#1,ext=tex]{#3}{#4}\fi}
          310 {\end{mhview}}
 mhviewnl The mhviewnl environment is just a layer over the mhviewsketch environment
           with the keys and language suitably adapted.<sup>4</sup>
          311 \newenvironment{mhviewnl}[5][]{\def\0test{#1}\ifx\0test\0empty%
          312 \begin{mhviewsketch}[id=#2.#5,ext=tex]{#3}{#4}\else%
          313 \begin{mhviewsketch}[id=#2.#5,#1,ext=tex]{#3}{#4}\fi}
          314 {\end{mhviewsketch}}
          315 (/smultiling | smultiling.ltxml)
          316 (smultiling.ltxml),;
```

4.6 structview-mh: MH Variants for Structures and Views

EdN:4

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

 $^{^4\}mathrm{EDNote}$: 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 $.\langle lang \rangle$!

```
317 (*structview)
                  318 \ProvidesPackage{structview-mh}[2015/11/04 v1.0 MathHub support for the sTeX structview package
                  320 \ProcessOptions
                  321 \RequirePackage{mathhub}
                  322 \RequirePackage{structview}
                  323 \RequirePackage{modules-mh}
                  324 (/structview)
                  325 (*structview.ltxml)
                  326 DeclareOption(undef,sub{PassOptions('structview','sty',ToString(Digest(T_CS('\CurrentOption')))
                  327 ProcessOptions();
                  328 RequirePackage('mathhub');
                  329 RequirePackage('structview');
                  330 RequirePackage('modules-mh');
                  331 (/structview.ltxml)
importmhmodulevia
                  332 \structview.ltxml\RawTeX('
                  333 (*structview | structview.ltxml)
                  334 \newenvironment{importmhmodulevia}[3][]{%
                        \gdef\@@doit{\importmhmodule[#1]{#2}{#3}}%
                        \ifmod@show\par\noindent importing module #2 via \@@doit\fi
                  336
                  337 }{%
                        \aftergroup\@@doit\ifmod@show end import\fi%
                  338
                  339 }%
                  340 (/structview | structview.ltxml)
                  341 (structview.ltxml)');
                  342 \langle *structview \rangle
                  343 \srefaddidkey{mhview}
                  344 \addmetakey{mhview}{display}
                  345 \addmetakey{mhview}{creators}
                  346 \addmetakey{mhview}{contributors}
                  347 \addmetakey{mhview}{srccite}
                  348 \verb| \addmetakey*{mhview}{title}|
                  349 \addmetakey{mhview}{fromrepos}
                  350 \texttt{\addmetakey\{mhview\}\{torepos\}}
                  351 \addmetakey{mhview}{frompath}
                  352 \addmetakey{mhview}{topath}
                  353 \addmetakey[sms]{mhview}{ext}
                  354 (/structview)
                  355 (*structview.ltxml)
                  356 DefKeyVal('mhview','id','Semiverbatim');
                  357 DefKeyVal('mhview', 'display', 'Semiverbatim');
                  358 DefKeyVal('mhview','creators','Semiverbatim');
                  359 DefKeyVal('mhview', 'contributors', 'Semiverbatim');
                  360 DefKeyVal('mhview', 'srccite', 'Semiverbatim');
                  361 DefKeyVal('mhview','title','Semiverbatim');
                  362 DefKeyVal('mhview', 'fromrepos', 'Semiverbatim');
```

363 DefKeyVal('mhview', 'torepos', 'Semiverbatim');

```
364 DefKeyVal('mhview', 'frompath', 'Semiverbatim');
        365 DefKeyVal('mhview', 'topath', 'Semiverbatim');
        366 DefKeyVal('mhview','ext','Semiverbatim');
        367 (/structview.ltxml)
mhview the MathHub version
        368 (*structview)
        369 \newenvironment{mhview}[3][]{% keys, from, to
             \metasetkeys{mhview}{#1}%
        371
             \sref@target%
        372
             \begin{@mhview}{#2}{#3}%
             \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
        373
        374 }{%
             \end{@mhview}%
        375
             \ignorespaces%
        376
        377 }%
        378 \ifmod@show\surroundwithmdframed{mhview}\fi
        379 (/structview)
        380 (*structview.ltxml)
        381 DefMacroI(T_CS('\begin{mhview}'),'OptionalKeyVals:mhview {}{}', sub {
             my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
        383
             my $from = ToString(Digest($from_arg));
        384
             my $to = ToString(Digest($to_arg));
        385
             AssignValue(from_module => $from);
             AssignValue(to_module => $to);
             my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
        387
             my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
        388
             my $repos = LookupValue('current_repos');
        389
             my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
        390
             my $to_path = ToString(GetKeyVal($keyvals,'topath'));
        391
             my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
        392
             $ext = 'sms' unless $ext;
        393
             my $current_repos = LookupValue('current_repos');
        394
        395
             if (!$from_repos) { $from_repos = $current_repos; }
             if (!$to_repos) { $to_repos = $current_repos; }
        396
        397
             return (
               Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
        398
               Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
        399
                Invocation(T_CS('\begin{viewenv}'), $keyvals, $from_arg, $to_arg) -> unlist
        400
        401
             );
        402 }):
        403 DefMacroI('\end{mhview}',undef,'\end{viewenv}');
        404 (/structview.ltxml)
Omhview The Omhview does the actual bookkeeping at the module level.
        405 (*structview)
        406 \newenvironment{@mhview}[2]{%from, to
             \importmhmodule[repos=\mhview@fromrepos,path=\mhview@frompath,ext=\mhview@ext]{#1}%
        408
             \importmhmodule[repos=\mhview@torepos,path=\mhview@topath,ext=\mhview@ext]{#2}%
        409 }{}%
```

```
410 (/structview)
mhviewsketch The mhviewsketch environment behaves like mhview, but only has text contents.
             411 (*structview)
             412 \newenvironment{mhviewsketch}[3][]{%
                  \metasetkeys{mhview}{#1}%
             413
             414
                   \sref@target%
                   \begin{@mhview}{#2}{#3}%
                  \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
             417 }{%
                  \end{@mhview}%
             418
                  \ignorespaces%
             419
             420 }%
             421 \ifmod@show\surroundwithmdframed{mhviewsketch}\fi
             422 (/structview)
             423 (*structview.ltxml)
             424 DefMacroI(T_CS('\begin{mhviewsketch}'),'OptionalKeyVals:mhview {}{}', sub {
                  my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
                  my $from = ToString(Digest($from_arg));
             426
             427
                  my $to = ToString(Digest($to_arg));
                  my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
                  my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
                  my $repos = LookupValue('current_repos');
             430
                  my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
             431
                  my $to_path = ToString(GetKeyVal($keyvals,'topath'));
             432
                  my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
             433
                  $ext = 'sms' unless $ext;
                  my $current_repos = LookupValue('current_repos');
                  if (!$from_repos) { $from_repos = $current_repos; }
             436
                  if (!$to_repos) { $to_repos = $current_repos; }
             437
                  return (
             438
                     Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
             439
                     Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
             440
                     Invocation(T_CS('\begin{viewsketchenv}'), $keyvals, $from_arg, $to_arg)->unlist
             441
             442
                  );
             443 });
             444 DefMacroI('\end{mhviewsketch}',undef,'\end{viewsketchenv}');
             445 (/structview.ltxml)
```

4.7 mikoslides-mh: Support for MiKo Slides

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

```
446 \*mikoslides\
447 \ProvidesPackage\{\text{mikoslides-mh}\} [2015/11/04 v1.0 MathHub support for the sTeX mikoslides package
448 \DeclareOption*\{\PassOptionsToPackage\{\CurrentOption\}\{\text{mikoslides}\}\}
449 \ProcessOptions
450 \RequirePackage\{\text{mathhub}\}
451 \RequirePackage\{\text{mikoslides}\}
```

```
452 \RequirePackage{statements-mh}
              453 (/mikoslides)
              454 (*mikoslides.ltxml)
              455 DeclareOption(undef, sub{PassOptions('mikoslides','sty',ToString(Digest(T_CS('\CurrentOption')))
              456 ProcessOptions();
              457 RequirePackage('mathhub');
               458 RequirePackage('mikoslides');
               459 RequirePackage('statements-mh');
              460 (/mikoslides.ltxml)
\mhframeimage Use the current value of \mh@currentrepos or the value of the mhrepos key if it
               is given in \frameimage.
               461 (mikoslides)\addmetakey{Gin}{mhrepos}
               462 (mikoslides.ltxml)DefKeyVal('Gin', 'mhrepos', 'Semiverbatim');
               463 (mikoslides.ltxml)RawTeX('
              464 (*mikoslides.ltxml | mikoslides)
              465 \newcommand\mhframeimage[2][]{%
              466
                    \metasetkeys{Gin}{#1}%
                    \edef\mh@@repos{\mh@currentrepos}%
              467
                    \ifx\Gin@mhrepos\@empty%
              468
                      \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%
               469
               470
              471
                      \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%
              472
                    \fi%
              473 }%
              474 (/mikoslides.ltxml | mikoslides)
              475 (mikoslides.ltxml),;
                      problem-mh: Support for Problems
```

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

```
476 \*problem\*
477 \ProvidesPackage{problem-mh}[2015/11/04 v1.0 MathHub support for the sTeX problem package]
478 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{problem}}
479 \ProcessOptions
480 \RequirePackage{mathhub}
481 \RequirePackage{problem}
482 \RequirePackage{omtext-mh}
483 \langle /problem\*
484 \*problem.ltxml\*
485 DeclareOption(undef,sub{PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))));
486 ProcessOptions();
487 RequirePackage('mathhub');
488 RequirePackage('problem');
```

\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

489 RequirePackage('omtext-mh');

490 (/problem.ltxml)

the optional argument, and after importing resets \mh@currentrepos to the old value in \mh@@repos.

```
491 \langle *problem \rangle
492 \newcommand\includemhproblem[2][]{\metasetkeys{inclprob}{#1}%
493 \edef\mh@@repos{\mh@currentrepos}%
494 \ifx\inclprob@mhrepos\@empty\else\mhcurrentrepos\inclprob@mhrepos\fi%
495 \input{\MathHub{\mh@currentrepos/source/#2}}%
496 \mhcurrentrepos\mh@@repos\clear@inclprob@keys}
497 (/problem)
498 (*problem.ltxml)
499 sub includemhproblem {
    my ($gullet,$keyval,$arg2) = @_;
     my $repo_path;
501
    if ($keyval) {
502
       $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
503
504
     if (! $repo_path) {
505
       $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
506
     else {
       $keyval->setValue('mhrepos',undef); }
507
    my $mathhub_base = ToString(Digest('\MathHub{}'));
508
    my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
510 return Invocation(T_CS('\includeproblem'), $keyval, T_OTHER($finalpath)); }#$
511 DefKeyVal('inclprob', 'mhrepos', 'Semiverbatim');
512 DefMacro('\includemhproblem OptionalKeyVals:inclprob {}', \&includemhproblem);
513 (/problem.ltxml)
```

4.9 hwexam-mh: Support for Assignments

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

```
514 \*hwexam\\
515 \ProvidesPackage{hwexam-mh}[2015/11/04 v1.0 MathHub support for the sTeX hwexam package]
516 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{hwexam}}
517 \ProcessOptions
518 \RequirePackage{mathhub}
519 \RequirePackage{hwexam}
520 \RequirePackage{problem-mh}
521 \langle/hwexam\rangle
522 \*hwexam.ltxml\rangle
523 DeclareOption(undef,sub{PassOptions('hwexam','sty',ToString(Digest(T_CS('\CurrentOption')))); }
524 \ProcessOptions();
525 \RequirePackage('mathhub');
526 \RequirePackage('hwexam');
527 \RequirePackage('problem-mh');
528 \langle/hwexam.ltxml\rangle
```

\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.
                   529 (*hwexam)
                   530 \newcommand\includemhassignment[2][]{\metasetkeys{inclassig}{#1}%
                   531 \edef\mh@@repos{\mh@currentrepos}%
                   532 \ifx\inclassig@mhrepos\@empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
                   533 \includeassignment[#1] {\MathHub{\mh@currentrepos/source/#2}}%
                   534 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
                   535 (/hwexam)
                   536 (*hwexm.ltxml)
                   537 sub includemhassignment {
                        my ($gullet,$keyval,$arg2) = 0_;
                        my $repo_path;
                   539
                        if ($keyval) {
                   540
                           $repo_path = ToString(GetKeyVal($keyval, 'mhrepos')); }
                   541
                   542
                        if (! $repo_path) {
                   543
                           $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
                   544
                        else {
                           $keyval->setValue('mhrepos',undef); }
                   545
                        my $mathhub_base = ToString(Digest('\MathHub{}'));
                   546
                   547 my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
                   548 return Invocation(T_CS('\includeassignment'), $keyval, T_OTHER($finalpath)); }#$
                   549 DefKeyVal('inclprob', 'mhrepos', 'Semiverbatim');
                   550 DefMacro('\includemhassignment OptionalKeyVals:inclprob {}', \&includemhassignment);
                   551 (/hwexm.ltxml)
\inputmhassignment analogous
                   552 (*hwexam)
                   553 \newcommand\inputmhassignment[2][]{\metasetkeys{inclassig}{\#1}%
                   554 \edef\mh@@repos{\mh@currentrepos}%
                   555 \ifx\inclassig@mhrepos\@empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
                   556 \inputassignment[#1] {\MathHub{\mh@currentrepos/source/#2}}%
                   557 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
                   558 (/hwexam)
                   559 (*hwexam.ltxml)
                   560 sub inputmhassignment {
                        my ($gullet,$keyval,$arg2) = @_;
                        my $repo_path;
                   562
                   563
                        if ($keyval) {
                   564
                           $repo_path = ToString(GetKeyVal($keyval, 'mhrepos')); }
                   565
                        if (! $repo_path) {
                           $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
                   566
                   567
                        else {
                           $keyval->setValue('mhrepos',undef); }
                   568
                        my $mathhub_base = ToString(Digest('\MathHub{}'));
                   569
                        my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
                   570
                        return Invocation(T_CS('\inputassignment'), $keyval, T_OTHER($finalpath)); }#$
                   572 DefMacro('\inputmhassignment OptionalKeyVals:inclprob {}', \&inputmhassignment);
```

573 (/hwexam.ltxml)

4.10 Finale

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

 $574 \ \langle ltxml \ | \ modules.ltxml \ | \ structview.ltxml \ | \ omtext.ltxml \ | \ statements.ltxml \ | \ smultiling.ltxml \ | \ mikoslides.ltxml \ | \ problem \ | \ statements.ltxml \ | \ statements.ltxml$

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).