MathHub Support for STEX*

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

November 11, 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

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

4.5	smultiling-mh: MH Variants for Multilinguality	13
4.6	structview-mh: MH Variants for Structures and Views	15
4.7	mikoslides-mh: Support for MiKo Slides	18
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 [HorlacJuc:cscpnrr11]. 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:github:on].

1. none reported yet.

4 Implementation

The sref package generates two files: the IATEX package (all the code between <code>*package</code>) and <code>*(package)</code>) and the IATEXML bindings (between <code>*Itxml</code>) and <code>*(Itxml</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 # -*- CPERL -*-
3 package LaTeXML::Package::Pool;
4 use strict;
5 use LaTeXML::Package;
6 (/ltxml | modules.ltxml | structview.ltxml | omtext.ltxml | statements.ltxml | smultiling.ltxml | mikoslides.ltxml | probler
7 (package)\ProvidesPackage{mathhub}[2015/11/04 v1.0 sTeX Support for MathHub.info]
8 (*package)
9 \DeclareOption*{}
10 \ProcessOptions
11 (/package)
12 (*ltxml)
13 use LaTeXML::Util::Pathname;
14 DeclareOption(undef, sub {});
15 ProcessOptions();
16 (/ltxml)
   Then we need to set up the packages by requiring the metakeys pack-
age [Kohlhase:metakeys:ctan] to be loaded (in the right version).
17 (*package)
18 \RequirePackage{keyval}
19 (/package)
20 (*ltxml)
21 RequirePackage('keyval');
22 (/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.

```
23 \*package\
24 \newcommand\mhcurrentrepos[1]{%
25 \edef\Qtest{#1}%
26 \ifx\Qtest\mhQcurrentrepos% if new dir = old dir
27 \relax% no need to change
28 \else%
29 \protectedQwrite\Qauxout{}{\string\Qmhcurrentrepos{#1}}%
30 \fi%
31 \Qmhcurrentrepos{#1}% define mhQcurrentrepos
```

```
33 \newcommand\@mhcurrentrepos[1]{\edef\mh@currentrepos{#1}}%
           34 (/package)
           35 (*ltxml)
           36 DefMacro('\mhcurrentrepos{}','\@mhcurrentrepos{#1}');
           37 DefMacro('\@mhcurrentrepos{#1}','\def\mh@currentrepos{#1}\@mhcurrentrepos{#1}');
           38 DefConstructor('\@@mhcurrentrepos{}','',
               afterDigest => sub{ AssignValue('current_repos',ToString($_[1]->getArg(1)),'global'); } );
           40 \langle /ltxml \rangle #$
\libinput
          the \libinput macro inputs from the lib directory of the MathHub repository
           or the meta-inf/lib repos of the group.
           41 (*package)
           42 \def\modules@@first#1/#2;{#1}
           43 \newcommand\libinput[1]{\def\@libfile{\MathHub{\mh@currentrepos/lib/#1}}%
           44 \IfFileExists{\@libfile}{\input\@libfile}%
           45 {\edef\@@group{\expandafter\modules@@first\mh@currentrepos;}
           46 \edgroup/meta-inf/lib/#1\}}
           47 \IfFileExists{\@inffile}{\input{\@inffile}}%
           48 {\PackageError{modules}
               {Library file missing, cannot input #1\MessageBreak%
                 Both \@libfile.tex\MessageBreak and \@inffile.tex\MessageBreak do not exist}%
           50
               {Check whether the file name is correct}}}}
           51
           52 (/package)
           53 (*ltxml)
           54 DefMacro('\modules@@first#1/#2;','#1');
           55 DefMacro('\libinput {}', sub{
                 my ($gullet, $name) = @_;
                 $name = ToString($name);
           57
                 #Relative paths for recursive search
           58
                 my $libpath = "../../lib";
           59
                 my $inffile = "../../../meta-inf/lib";
                 my $file = pathname_find($name, types => ['tex'], paths =>$libpath);
           61
                 $file = pathname_find($name, types=>['tex'], paths=>$inffile) unless $file;
           62
                 # Singal error if the file cannot be found
           63
                 LaTeXML::Package::InputContent($file, noerror=>1); });
           64
           65 (/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.

```
66 \( \section{a} \)
67 \ProvidesPackage{modules-mh}[2015/11/04 v1.0 MathHub support for the sTeX modules package]
68 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{modules}}
69 \ProcessOptions
70 \RequirePackage{modules}
71 \RequirePackage{mathhub}
72 \( /modules \)
```

```
73 (*modules.ltxml)
                 74 DeclareOption(undef,sub{PassOptions('modules','sty',ToString(Digest(T_CS('\CurrentOption'))));
                 75 ProcessOptions();
                 76 RequirePackage('modules');
                 77 RequirePackage('mathhub');
                 78 (/modules.ltxml)
                 The \infty = \frac{ist}{lso} [module] saves the current value of
\importmhmodule
                 \mh@currentrepos in a local macro \mh@@repos, resets \mh@currentrepos to
                 the new value if one is given in the optional argument, and after importing resets
                 \mh@currentrepos to the old value in \mh@@repos. We do all the \ifx compar-
                 ison with an \expandafter, since the values may be passed on from other key
                 bindings. Parameters will be passed to \importmodule.
                 79 (*modules)
                 80 \srefaddidkey{importmhmodule}%
                 81 \addmetakey{importmhmodule}{repos}% saves the repo's path. E.g: smglom/numberfield
                 82 \addmetakey{importmhmodule}{path}% saves the module name. E.g: naturalnumbers
                 83 \addmetakey[sms]{importmhmodule}{ext}% saves the extension: E.g: tex
                 84 \addmetakey[false]{importmhmodule}{conservative}[true]%
                 85 \newcommand\importmhmodule[2][]{%
                 86
                     \metasetkeys{importmhmodule}{#1}%
                      \ifx\importmhmodule@path\@empty% if module name is not set
                 87
                        \importmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
                 88
                 89
                        \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
                 90
                        \ifx\importmhmodule@repos\@empty% if in the same repos
                 91
                 92
                          \relax% no need to change mh@currentrepos, i.e, current dirctory.
                 93
                          \mhcurrentrepos{\importmhmodule@repos}% change it.
                 94
                 95
                        \importmodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},%
                 96
                        ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
                 97
                 98
                        \mhcurrentrepos{\mh@@repos}% after importing, reset to old value
                 99
                      \fi%
                     \ignorespaces%
                100
                101 }%
                102 (/modules)
                103 (*modules.ltxml)
                104 DefKeyVal('importmhmodule','id','Semiverbatim');
                105 DefKeyVal('importmhmodule', 'repos', 'Semiverbatim');
                106 DefKeyVal('importmhmodule', 'path', 'Semiverbatim');
                107 DefKeyVal('importmhmodule','ext','Semiverbatim');
                108 DefKeyVal('importmhmodule', 'conservative', 'Semiverbatim');
                109 DefConstructor('\importmhmodule OptionalKeyVals:importmhmodule {}',
                110
                           "<omdoc:imports "
                           . "from='?%GetKeyVal(#1,'load'))(&canonical_omdoc_path(&GetKeyVal(#1,'load')))()###2'"
                111
                                    "?&defined(&GetKeyVal(#1,'conservative'))(load='&GetKeyVal(#1,'conservative')'
                112
                113
                      afterDigest => \&importMHmoduleI);
```

114

```
115 sub importMHmoduleI {
                  my ($stomach, $whatsit) = @_;
                  my $keyval = $whatsit->getArg(1);
             117
                  my $id = $whatsit->getArg(2);
             118
             119
                  if ($keyval) {
             120
                     my $repos = ToString($keyval->getValue('repos'));
             121
                     my $path = ToString($keyval->getValue('path'));
                     my $current_repos = LookupValue('current_repos');
             122
                     if (!$repos) { # Use the implicit current repository
             123
                       $repos = $current_repos; }
             124
                     my $defpaths = LookupValue('defpath');
             125
             126
                     my $load_path = ($$defpaths{MathHub}).$repos.'/source/'.$path;
                     $keyval->setValue('load',$load_path);
             127
                     AssignValue('current_repos' => $repos, 'global');
             128
                     importmoduleI($stomach,$whatsit);
             129
                     AssignValue('current_repos' => $current_repos, 'global'); }
             130
             131
                     importmoduleI($stomach,$whatsit); }
             132
             133
                  return; }
             134
             135 DefConstructor('\importMHmoduleI OptionalKeyVals:importmhmodule {}', '',
                    afterDigest=> \&importMHmoduleI );#$
             136
             137 (/modules.ltxml)
              and now the analogs
\usemhmodule
             138 (*modules)
             139 \newcommand\usemhmodule[2][]{%
             140
                   \metasetkeys{importmhmodule}{#1}%
             141
                   \ifx\importmhmodule@path\@empty%
             142
                     \usemodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
             143
                   \else%
             144
                     \edef\mh@@repos{\mh@currentrepos}%
             145
                     \ifx\importmhmodule@repos\@empty%
             146
                     \else%
             147
                       \mhcurrentrepos{\importmhmodule@repos}%
             148
                     \usemodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodule@
             149
             150
                     \mhcurrentrepos\mh@@repos%
             151
                   \fi%
                   \ignorespaces%
             152
             153 }%
             154 (/modules)
             155 (*modules.ltxml)
             156 DefConstructor('\usemhmodule OptionalKeyVals:importmhmodule {}',
                    "<omdoc:uses from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))()###</pre>
             157
                    afterDigest => \&importMHmoduleI);
             159 (/modules.ltxml)
```

```
160 (modules.ltxml)RawTeX('
            161 (*modules | modules.ltxml)
            162 \newcommand\mhinputref[2][]{%
            163
                 \def\@repos{#1}%
            164
                 \edef\mh@@repos{\mh@currentrepos}%
                 \ifx\@repos\@empty%
            165
                 \else%
            166
                    \mhcurrentrepos{#1}%
            167
            168
                 \inputref{\MathHub{\mh@currentrepos/source/#2}}%
            169
                 \mhcurrentrepos\mh@@repos%
            170
                 \ignorespaces%
            171
            172 }%
            173 (/modules | modules.ltxml)
            174 (modules.ltxml)');
   \mhinput
            175 (*modules)
            176 \let\mhinput\mhinputref%
             177 (/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.
            178 (*omtext)
             179 \ProvidesPackage{omtext-mh}[2015/11/04 v1.0 MathHub support for the sTeX omtext package]
            181 \ProcessOptions
            182 \RequirePackage{mathhub}
            183 \RequirePackage{omtext}
            184 \RequirePackage{modules-mh}
            185 (/omtext)
            186 (*omtext.ltxml)
            187 DeclareOption(undef,sub{PassOptions('omtext','sty',ToString(Digest(T_CS('\CurrentOption')))); }
            188 ProcessOptions();
             189 RequirePackage('mathhub');
            190 RequirePackage('omtext');
            191 RequirePackage('modules-mh');
            192 (/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.
            193 (*omtext)
            194 \addmetakey{Gin}{mhrepos}
            195 \newcommand\mhgraphics[2][]{\metasetkeys{Gin}{#1}%
            196 \edef\mh@@repos{\mh@currentrepos}%
```

\mhinputref

```
197 \ifx\Gin@mhrepos\@empty\mygraphics[#1]{\MathHub{\mh@currentrepos/source/#2}}%
198 \else\mygraphics[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
199 \def\Gin@mhrepos{}\mhcurrentrepos\mh@@repos}
200 \newcommand\mhcgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}}
201 \newcommand\mhbgraphics[2][]{\fbox{\mhgraphics[#1]{#2}}}
202 \newcommand\mhcbgraphics[2][]{\begin{center}\fbox{\mhgraphics[#1]{#2}}\end{center}}
203 (/omtext)
204 (*omtext.ltxml)
205 \; \mathrm{sub} \; \; \mathrm{mhgraphics} \; \{
    my ($gullet,$keyval,$arg2) = @_;
206
     my $repo_path;
207
     if ($keyval) {
208
       $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
209
210
    if (! $repo_path) {
       $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
211
212
    else {
       $keyval->setValue('mhrepos',undef); }
213
    my $mathhub_base = ToString(Digest('\MathHub{}'));
214
215 my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
216 return Invocation(T_CS('\@includegraphicx'), $keyval, T_OTHER($finalpath)); }#$
217 DefKeyVal('Gin', 'mhrepos', 'Semiverbatim');
218 DefMacro('\mhgraphics OptionalKeyVals:Gin {}', \&mhgraphics);
219 DefMacro('\mhcgraphics []{}','\begin{center}\mhgraphics[#1]{#2}\end{center}');
220 DefMacro('\mhbgraphics []{}','\fbox{\mhgraphics[#1]{#2}}');
221 (/omtext.ltxml)
```

4.4 statements-mh: MH Variants for Statements

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

```
222 (*statements)
223 \ProvidesPackage{statements-mh}[2015/11/04 v1.0 MathHub support for the sTeX statements package
225 \ProcessOptions
226 \RequirePackage{mathhub}
227 \RequirePackage{statements}
228 \RequirePackage{omtext-mh}
229 (/statements)
230 (*statements.ltxml)
231 DeclareOption(undef, sub{PassOptions('statements','sty',ToString(Digest(T_CS('\CurrentOption')))
232 ProcessOptions();
233 RequirePackage('mathhub');
234 RequirePackage('statements');
235 RequirePackage('omtext-mh');
236 (/statements.ltxml)
237 (*statements)
238 \let\usemhvocab=\usemhmodule
```

 $239 \langle \text{/statements} \rangle$ $240 \langle \text{*statements.ltxml} \rangle$

```
241 DefMacro('\usemhvocab','\usemhmodule'); 242 \langle/statements.ltxml\rangle
```

4.5 smultiling-mh: MH Variants for Multilinguality

```
We set up package options and pass them on to the smultiling package, which
           we also load.
          243 (*smultiling)
          244 \ProvidesPackage{smultiling-mh}[2015/11/04 v1.0 MathHub support for the sTeX smultiling package
          245 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{smultiling}}
          246 \ProcessOptions
          247 \RequirePackage{mathhub}
          248 \RequirePackage{smultiling}
          249 \verb|\RequirePackage{structview-mh}|
          250 (/smultiling)
          251 (*smultiling.ltxml)
          252 DeclareOption(undef, sub{PassOptions('smultiling','sty',ToString(Digest(T_CS('\CurrentOption')))
          253 ProcessOptions();
          254 RequirePackage('mathhub');
          255 RequirePackage('smultiling');
          256 RequirePackage('structview-mh');
          257 (/smultiling.ltxml)
mhmodnl:*
          258 (*smultiling)
          259 \addmetakey{mhmodnl}{repos}
          260 \addmetakey{mhmodnl}{path}
          261 \addmetakey*{mhmodnl}{title}
          262 \addmetakey*{mhmodnl}{creators}
          263 \addmetakey*{mhmodnl}{contributors}
          264 \addmetakey{mhmodnl}{srccite}
          265 \addmetakey{primary}{mhmodnl}[yes]
          266 (/smultiling)
          267 (*smultiling.ltxml)
          268 DefKeyVal('mhmodnl','title','Semiverbatim');
          269 DefKeyVal('mhmodnl', 'repos', 'Semiverbatim');
          270 DefKeyVal('mhmodnl','path','Semiverbatim');
          271 DefKeyVal('mhmodnl','creators','Semiverbatim');
          272 DefKeyVal('mhmodnl', 'contributors', 'Semiverbatim');
          273 DefKeyVal('mhmodnl', 'primary', 'Semiverbatim');
          274 (/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.
          276 \newenvironment{mhmodnl}[3][]{\metasetkeys{mhmodnl}{#1}%
          277 \def\Ctest{#1}\ifx\Ctest\Cempty\begin{module}[id=#2.#3]\else\begin{module}[id=#2.#3,#1]\fi%
          278 \edef\@repos{\ifx\mhmodnl@repos\@empty\mh@currentrepos\else\mhmodnl@repos}
          279 \if@langfiles\importmhmodule[repos=\@repos,load=#2,ext=tex] \{#2\}\else
```

```
281 \fi}
          282 {\end{module}}
          283 \langle / smultiling \rangle
          284 (*smultiling.ltxml)
          285 DefEnvironment('{mhmodnl} OptionalKeyVals:mhmodnl {}{}',
                        "?#excluded()(<omdoc:theory xml:id='#2.#3' >"
                            "?&defined(&GetKeyVal(#1,'creators'))(<dc:creator>&GetKeyVal(#1,'creators')</dc:cr
          287
                            "?&defined(&GetKeyVal(#1,'title'))(<dc:title>&GetKeyVal(#1,'title')</dc:title>)()"
          288
                            "?&defined(&GetKeyVal(#1,'contributors'))(<dc:contributor>&GetKeyVal(#1,'contribut
          289
                            "<omdoc:imports from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'
           290
                            "#body"
           291
                        . "</omdoc:theory>)",
           292
                afterDigestBegin=>sub {
           293
                  my ($stomach, $whatsit) = @_;
          294
                  my $keyval = $whatsit->getArg(1);
          295
                  my $signature = ToString($whatsit->getArg(2));
          296
                  my $language = ToString($whatsit->getArg(3));
          297
          298
                  my $repos = ToString(GetKeyVal($keyval,'torepos'));
           299
                  my $current_repos = LookupValue('current_repos');
                  if (!$repos) { $repos = $current_repos; }
          300
                  my $defpaths = LookupValue('defpath');
          301
                  my $load_path = ($$defpaths{MathHub}).$repos.'/source/'.$signature;
          302
          303
          304
                  if ($keyval) {
                    # If we're not given load, AND the langfiles option is in effect,
           305
                    # default to #2
           306
                    if ((! $keyval->getValue('path')) && (LookupValue('smultiling_langfiles'))) {
          307
                       $keyval->setValue('load',$load_path); }
          308
                    # Always load a TeX file
          309
                    $keyval->setValue('ext','tex');
          310
          311
                    $keyval->setValue('id', "$signature.$language"); }
          312
                  module_afterDigestBegin(@_);
          313
                  importmoduleI(@_);
                  return; },
          314
                afterDigest=>sub {
          315
                  module_afterDigest(@_); });
          316
          317 (/smultiling.ltxml)%$
mhviewsig The mhviewsig environment is just a layer over the mhview environment with the
           keys suitably adapted.
          318 \(\smultiling.ltxml\)\(\text{RawTeX('}\)
          319 (*smultiling | smultiling.ltxml)
          320 \newenvironment{mhviewsig}[4][]{\def\@test{#1}\ifx\@test\@empty%
          321 \begin{mhview}[id=#2,ext=tex]{#3}{#4}\else%
          322 \begin{mhview}[id=#2,#1,ext=tex]{#3}{#4}\fi}
          323 {\end{mhview}}
```

280 \ifx\mhmodnl@load\@empty\importmodule{#2}\else\importmodule[ext=tex,load=\mhmodnl@load]{#2}\fi%

mhviewnl The mhviewnl environment is just a layer over the mhviewsketch environment

```
EdN:4
```

```
with the keys and langauge suitably adapted.<sup>4</sup>
324 \newenvironment{mhviewnl}[5][]{\def\@test{#1}\ifx\@test\@empty%
325 \begin{mhviewsketch}[id=#2.#5,ext=tex]{#3}{#4}\else%
326 \begin{mhviewsketch}[id=#2.#5,#1,ext=tex]{#3}{#4}\fi}
327 {\end{mhviewsketch}}
328 \/smultiling | smultiling.ltxml\\)
329 \/smultiling.ltxml\\);
```

4.6 structview-mh: MH Variants for Structures and Views

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

```
330 (*structview)
331 \ProvidesPackage{structview-mh}[2015/11/04 v1.0 MathHub support for the sTeX structview package
332 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{structview}}
333 \ProcessOptions
334 \RequirePackage{mathhub}
335 \RequirePackage{structview}
336 \RequirePackage{structview}
337 \/structview\
338 \*structview.ltxml\
339 \DeclareOption(undef,sub{PassOptions('structview','sty',ToString(Digest(T_CS('\CurrentOption')))))
340 \ProcessOptions();
341 \RequirePackage('mathhub');
342 \RequirePackage('structview');
343 \RequirePackage('modules-mh');
344 \/structview.ltxml\
```

importmhmodulevia

```
345 \structview.ltxml\RawTeX('
346 (*structview | structview.ltxml)
347 \newenvironment{importmhmodulevia}[3][]{%
     \ifmod@show\par\noindent importing module #2 via \@@doit\fi
349
350 }{%
351 \aftergroup\@@doit\ifmod@show end import\fi%
352 }%
353 (/structview | structview.ltxml)
354 (structview.ltxml)');
355 (*structview)
356 \srefaddidkey{mhview}
357 \addmetakey{mhview}{display}
358 \addmetakey{mhview}{creators}
359 \addmetakey{mhview}{contributors}
360 \addmetakey{mhview}{srccite}
```

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

```
361 \addmetakey*{mhview}{title}
       362 \addmetakey{mhview}{fromrepos}
       363 \addmetakey{mhview}{torepos}
       364 \addmetakey{mhview}{frompath}
       365 \addmetakey{mhview}{topath}
       366 \addmetakey[sms]{mhview}{ext}
       367 (/structview)
       368 (*structview.ltxml)
       369 DefKeyVal('mhview','id','Semiverbatim');
       370 DefKeyVal('mhview', 'display', 'Semiverbatim');
       371 DefKeyVal('mhview', 'creators', 'Semiverbatim');
       372 DefKeyVal('mhview','contributors','Semiverbatim');
       373 DefKeyVal('mhview', 'srccite', 'Semiverbatim');
       374 DefKeyVal('mhview','title','Semiverbatim');
       375 DefKeyVal('mhview', 'fromrepos', 'Semiverbatim');
       376 DefKeyVal('mhview','torepos','Semiverbatim');
       377 DefKeyVal('mhview', 'frompath', 'Semiverbatim');
       378 DefKeyVal('mhview', 'topath', 'Semiverbatim');
       379 DefKeyVal('mhview', 'ext', 'Semiverbatim');
       380 (/structview.ltxml)
mhview the MathHub version
       381 (*structview)
       382 \newenvironment{mhview}[3][]{% keys, from, to
            \metasetkeys{mhview}{#1}%
       383
            \sref@target%
       384
            \begin{@mhview}{#2}{#3}%
       385
            386
       387 }{%
             \end{@mhview}%
       389
            \ignorespaces%
       390 }%
       391 \ifmod@show\surroundwithmdframed{mhview}\fi
       392 (/structview)
       393 (*structview.ltxml)
       394 DefMacroI(T_CS('\begin{mhview}'),'OptionalKeyVals:mhview {}{}', sub {
            my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
            my $from = ToString(Digest($from_arg));
       396
            my $to = ToString(Digest($to_arg));
       397
            AssignValue(from_module => $from);
       398
            AssignValue(to_module => $to);
       399
            my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
       400
            my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
            my $repos = LookupValue('current_repos');
            my $from_path = ToString(GetKeyVal($keyvals, 'frompath'));
       403
            my $to_path = ToString(GetKeyVal($keyvals,'topath'));
       404
            my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
       405
            $ext = 'sms' unless $ext;
       406
       407
            my $current_repos = LookupValue('current_repos');
            if (!$from_repos) { $from_repos = $current_repos; }
```

```
if (!$to_repos) { $to_repos = $current_repos; }
             409
                  return (
             410
                     Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
             411
                     Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
             412
                     Invocation(T_CS('\begin{viewenv}'), $keyvals, $from_arg, $to_arg) ->unlist
             413
             414
                  );
             415 });
             416 DefMacroI('\end{mhview}',undef,'\end{viewenv}');
             417 (/structview.ltxml)
     Omhview The Omhview does the actual bookkeeping at the module level.
             418 (*structview)
             419 \newenvironment{@mhview}[2]{%from, to
                  \importmhmodule[repos=\mhview@fromrepos,path=\mhview@frompath,ext=\mhview@ext]{#1}%
                   \importmhmodule[repos=\mhview@torepos,path=\mhview@topath,ext=\mhview@ext]{#2}%
             422 }{}%
             423 (/structview)
mhviewsketch The mhviewsketch environment behaves like mhview, but only has text contents.
             424 (*structview)
             425 \newenvironment{mhviewsketch}[3][]{\%
             426
                   \metasetkeys{mhview}{#1}%
             427
                   \sref@target%
                   \begin{@mhview}{#2}{#3}%
                   \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
             429
             430 }{%
                   \end{@mhview}%
             431
                  \ignorespaces%
             432
             433 }%
             434 \ifmod@show\surroundwithmdframed{mhviewsketch}\fi
             435 (/structview)
             436 (*structview.ltxml)
             437 DefMacroI(T_CS('\begin{mhviewsketch}'),'OptionalKeyVals:mhview {}{}', sub {
                  my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
                  my $from = ToString(Digest($from_arg));
             439
                  my $to = ToString(Digest($to_arg));
             440
             441
                  my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
                  my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
             442
                  my $repos = LookupValue('current_repos');
             443
                  my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
             444
             445
                  my $to_path = ToString(GetKeyVal($keyvals,'topath'));
                  my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
             446
                  $ext = 'sms' unless $ext;
             447
                  my $current_repos = LookupValue('current_repos');
             448
                  if (!$from_repos) { $from_repos = $current_repos; }
             449
                  if (!$to_repos) { $to_repos = $current_repos; }
             450
                  return (
             451
                     Tokenize("\\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
             452
             453
                     Tokenize("\\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
             454
                     Invocation(T_CS('\begin{viewsketchenv}'), $keyvals, $from_arg, $to_arg)->unlist
```

```
455 );
456 });
457 DefMacroI('\end{mhviewsketch}',undef,'\end{viewsketchenv}');
458 (/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.
```

```
459 (*mikoslides)
460 \ProvidesPackage{mikoslides-mh}[2015/11/04 v1.0 MathHub support for the sTeX mikoslides package
461 \ensuremath{\mbox{\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{}\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{$\mbox{
462 \ProcessOptions
463 \RequirePackage{mathhub}
464 \RequirePackage{mikoslides}
465 \RequirePackage{statements-mh}
466 (/mikoslides)
467 (*mikoslides.ltxml)
468 \ \texttt{DeclareOption(undef,sub\{PassOptions('mikoslides','sty',ToString(Digest(T\_CS('\CurrentOption'))))} \\
469 ProcessOptions();
470 RequirePackage('mathhub');
471 RequirePackage('mikoslides');
472 RequirePackage('statements-mh');
473 (/mikoslides.ltxml)
Use the current value of \mh@currentrepos or the value of the mhrepos key if it
474 \(\text{mikoslides}\)\\addmetakey{Gin}\{\text{mhrepos}\}
```

\mhframeimage

is given in \frameimage.

```
475 (mikoslides.ltxml)DefKeyVal('Gin', 'mhrepos', 'Semiverbatim');
476 (mikoslides.ltxml)RawTeX('
477 (*mikoslides.ltxml | mikoslides)
478 \newcommand\mhframeimage[2][]{%
     \metasetkeys{Gin}{#1}%
479
     \edef\mh@currentrepos}%
480
     \ifx\Gin@mhrepos\@empty%
481
       \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%
482
     \else%
483
        \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%
484
     \fi%
485
486 }%
487 (/mikoslides.ltxml | mikoslides)
488 (mikoslides.ltxml),;
```

problem-mh: Support for Problems 4.8

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

```
489 (*problem)
```

```
491 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{problem}}
                                                     492 \ProcessOptions
                                                     493 \RequirePackage{mathhub}
                                                     494 \RequirePackage{problem}
                                                     495 \RequirePackage{omtext-mh}
                                                     496 (/problem)
                                                     497 (*problem.ltxml)
                                                     498 DeclareOption(undef,sub{PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))));
                                                     499 ProcessOptions();
                                                     500 RequirePackage('mathhub');
                                                     501 RequirePackage('problem');
                                                     502 RequirePackage('omtext-mh');
                                                     503 (/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.
                                                     504 (*problem)
                                                     505 \newcommand\includemhproblem[2][]{\metasetkeys{inclprob}{#1}%
                                                     506 \edef\mh@@repos{\mh@currentrepos}%
                                                     507 \ \texttt{\fim} 
                                                     508 \input{\MathHub{\mh@currentrepos/source/#2}}%
                                                     509 \mhcurrentrepos\mh@@repos\clear@inclprob@keys}
                                                     510 (/problem)
                                                     511 (*problem.ltxml)
                                                     512 sub includemhproblem {
                                                     513 my ($gullet,$keyval,$arg2) = 0_;
                                                     514 my $repo_path;
                                                     515 if ($keyval) {
                                                                          $repo_path = ToString(GetKeyVal($keyval, 'mhrepos')); }
                                                     516
                                                                  if (! $repo_path) {
                                                     517
                                                     518
                                                                          $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
                                                     519
                                                                  else {
                                                     520
                                                                           $keyval->setValue('mhrepos',undef); }
                                                     521 my $mathhub_base = ToString(Digest('\MathHub{}'));
                                                                  my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
                                                                  return Invocation(T_CS('\includeproblem'), $keyval, T_OTHER($finalpath)); }#$
                                                     524 DefKeyVal('inclprob', 'mhrepos', 'Semiverbatim');
                                                     525 DefMacro('\includemhproblem OptionalKeyVals:inclprob {}', \&includemhproblem);
                                                     526 (/problem.ltxml)
```

490 \ProvidesPackage{problem-mh}[2015/11/04 v1.0 MathHub support for the sTeX problem package]

4.9 hwexam-mh: Support for Assignments

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

```
527\ \langle*hwexam\rangle 528\ \ProvidesPackage\{hwexam-mh\}[2015/11/04\ v1.0\ MathHub\ support\ for\ the\ sTeX\ hwexam\ package]
```

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

```
571 (/hwexam)
572 \langle *hwexam.ltxml \rangle
573 \; {\tt sub} \; {\tt inputmhassignment} \; \{
     my ($gullet,$keyval,$arg2) = @_;
     my $repo_path;
575
576
     if ($keyval) {
        $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
577
     if (! $repo_path) {
578
        $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
579
     else {
580
        $keyval->setValue('mhrepos',undef); }
581
     my $mathhub_base = ToString(Digest('\MathHub{}'));
582
     my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
     return Invocation(T_CS('\inputassignment'), $keyval, T_OTHER($finalpath)); }#$
585\ \texttt{DefMacro('\input mhassignment\ Optional Key Vals: inclprob\ \{\}',\ \& input mhassignment)};
586 \langle /hwexam.ltxml \rangle
```

4.10 Finale

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

 $587 \ \langle ltxml \ | \ modules.ltxml \ | \ structview.ltxml \ | \ omtext.ltxml \ | \ statements.ltxml \ | \ smultiling.ltxml \ | \ mikoslides.ltxml \ | \ problem$