MathHub Support for sTeX*

Michael Kohlhase
FAU Erlangen-Nürnberg
http://kwarc.info/kohlhase

November 9, 2020

Abstract

The `mathhub` package collection 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 `mathhub` packages extend sTeX with support for MathHub file system layout, which has co-eevolved with the MathHub.info portal for active documents, but is useful for organizing collections of sTeX documents in its own right.

Contents

---

*Version v1.2 (last revised 2010/10/01)

1 Introduction

As STEX files tend to be highly interlinked semantically one of the most important practical problems to solve for managing larger collections is the management of (relative or absolute) paths. The `mathhub` package provides an infrastructure for supporting a regular ↝ manageable file system layout schema that has co-eveolved with the MathHub.info portal for active documents, but is useful for organizing collections of STEX documents in its own right. In particular, since the layout scheme is supported by the `lmh` (local mathhub []), and `make` (a build system [] for MathHub archives) in MMT [MMT] which automates many management tasks. For instance, after installing the `mmt.jar`, the shell command `mmt lmh install` ⟨*group*⟩/⟨*arch*⟩ installs the installs the MathHub archive ⟨*group*⟩/⟨*arch*⟩ together with all of its dependencies and `mmt make pdflatex` ⟨*file*⟩ generates PDF for the file ⟨*file*⟩ (and generates/updates all auxiliary files necessary along the way).

MathHub (`http://MathHub.info`), is a portal and archive for flexiformal mathematics. It hosts much of the STEX content MathHub on GIT repositories (public and private escrow) for mathematical documentation projects. MathHub supports online and offline (via `lmh`) authoring and document development infrastructure, and a rich, interactive reading interface.

The **MathHub file system layout** has a **MathHub root folder** (e.g. `~/localmh/MathHub`) which conibetains all STEX sources, which are in turn organized in **MathHub archives** [Hor+11]. These are organized in a two-level folder system that is compatible by GIT repository managers like GitHub [GH] and GitLab [GL]. Even though it is not necessary for the `mathhub` package we will assume that these are GIT repositories, which have names of the form ⟨*group*⟩/⟨*arch*⟩, where ⟨*group*⟩ is a MathHub-unique repository group and ⟨*arch*⟩ a MathHub archive name that is ⟨*group*⟩-unique.

The MathHub archives have a prescribed structure; see [Hor+11] for details. For our purposes, we only need two aspects:

- the STEX sources are all in a top-level subdirectory `source`,

- there is a top-level sub-directory `META-INF` with a manifest file `MANIFEST.MF` which consists of lines of the form ⟨*key*⟩: ⟨*values*⟩.

For the purposes of the `mathhub` package we assume that the `MANIFEST.MF` file has at least the `id` key specified and the the value is exactly ⟨*group*⟩/⟨*arch*⟩. Furthermore, we assume that the `MATHHUB` environment variable is set with the system path to the MathHub root folder.

With this information the mechanics of the MathHub archive structure can be hidden from the STEX author with MathHub-enabled versions of the STEX macros (let's call them **mh-variants**), which are defined in the mh-packages of the `mathhub` bundle, which we document in this manual. The mechanics of the `mathhub` bundle is as follows: For most STEX package ⟨*pack*⟩`.sty` there is a mh-variant ⟨*pack*⟩`-mh.sty`, and ⟨*pack*⟩`.sty` takes the option `mh`. When that is given (by calling `\usepackage[mh]{`⟨*pack*⟩`}`), then ⟨*pack*⟩`.sty` inputs ⟨*pack*⟩`-mh.sty` from the `mathhub` bundle, which augments the ⟨*pack*⟩ package with MathHub functionality.

2   The User Interface

We now document mh-variants of the sTEX packages that have MathHub functionality.

2.1   `mathhub.sty`: General Infrastructure

For the generation of absolute file paths, the `mathhub` package keeps track of the current archive. If this ever needs to be set manually, it can be declared by the `\mhcurrentrepos` macro relative to the MathHub root path. `\mhcurrentrepos{group/repos}` declare that it resides at the path `/user/foo/localmh/MathHub/group/repos` given that the MathHub root path is `/user/foo/localmh/mathhub`.

    Given a systematic grouping in the MathHub file layout scheme, sTEX files in the same repository (and often even in the same group) share much of the preamble material. Thus it makes sense to centralize that in external (shared) files and situate it at the group and repository levels: at the group level. Fort the group level, the MathHub file system layout uses a specical repository $\langle group\rangle$/`meta-inf/lib` and at the repository level we use $\langle group\rangle$/$\langle repos\rangle$/`lib` for such files. The `\libinput` macro supports this practice: `\libinput{`$\langle filename\rangle$`}` macro inputs the files $\langle group\rangle$/`meta-inf/lib/`$\langle filename\rangle$ and then $\langle group\rangle$/$\langle repos\rangle$/`lib/`$\langle filename\rangle$ if they exist. Thus a typical top-level sTEX file has the following lines in the preamble:

`\libinput{preamble}`

The `\libusepackage` is analogous it allows to share LATEX package between MathHub archives.

2.2   `omdoc-mh.sty`: MH Document Infrastructure

The `\addmhbibresource` macro is a variant of `\addbibresource` from bibLATEX with repository support. Concretely, `\addmhbibresource[`$\langle repos\rangle$`]{`$\langle path\rangle$`}` expands to `\addbibresource{`$\langle MathHub\rangle$`/`$\langle reponame\rangle$`/`$\langle path\rangle$`}`, where $\langle reponame\rangle$ is $\langle repo\rangle$ if that is non-empty and the current repository else. Note that in contrast to the other MH variants, this does not add the `/source/` into the path, since bibTEX files are often put into the `lib` directory parallel to `source`.

2.3   `modules-mh.sty`: MH Variants for Modules

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

`\importmodule[load=\MathHub{fooMH/bar/source/baz/foobar}]{foobar}`

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

```
\importmhmodule[mhrepos=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 one of the following forms

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

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

Of course, LaTeX does not know about the repositories when they are called from a file system, so we can use the `\mhcurrentrepos` macro to tell them. But this is usually 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.

\mhcurrentrepos

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

\usemhmodule

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

\mhinputref
\mhinput

Caveat if you want to use the MathHub support macros, then every time a module is imported or a document fragment is included from another repository, 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.

2.4 `omtext-mh.sty`: MH Variants for OMText

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

\cmhgraphics

```
\mycgraphics{\MathHub{fooMH/bar/source/baz/foobar}}
```

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

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

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

2.5 `smultiling-mh.sty`: MH Variants for Multilinguality

<div style="margin-left:auto">mhmodsig
mhmodnl</div>

The `mhmodsig` and `mhmodnl` environments are the MH variants of the `modsig` and `modnl` environments from the `smultiing` package. Just as in the other MH packages, `mhmodnl` takes additional `mhrepos` and `path` keys and combine them to `load` key of `modnl`. Instead of writing

```
\begin{modnl}[load=\MathHub{fooMH/bar/source/baz/foobar}]{foobar}{en}
```

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

```
\begin{modnl}[mhrepos=fooMH/bar,path=baz/foobar]{foobar}{en}
```

`mhmodsig` is just a notational variant of `modsig` that allows to keep the sources uniform.

2.6 `structview-mh.sty`: MH Variants for Structures and Views

EdN:1

[1]

2.7 **mikoslides-mh.sty**: Support for MiKo Slides

`\mhframeimage`

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

```
\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}
```

`\mhinputref*`

If we want to transclude a the contents of a file as a note, we can use the `\mhinputref*` macro: `\mhinputref*[foo]{bar}` is equivalent to

```
\begin{note}
\mhinputref[foo]{bar}
\end{note}
```

---

[1]EDNOTE: needs to be documented

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

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

```
\includeproblem[pts=7]{\MathHub{fooMH/bar/source/baz/foobar}}
```

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

```
\includemhproblem[mhrepos=fooMH/bar,pts=7]{baz/foobar}
```

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

2.9 **hwexam-mh.sty**: Support for Assignments

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

```
\includeassignment[number=3]{\MathHub{fooMH/bar/source/baz/foobar}}
```

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

```
\includemhassignment[mhrepos=fooMH/bar,number=3]{baz/foobar}
```

2.10 **lstmh.sty**: Support for Listings

\lstinputmhlisting
The `\lstinputmhlisting` macro is a variant of `\lstinputlisting` with repository support. Instead of writing

```
\lstinputlisting[language=XML]{\MathHub{fooMH/bar/source/baz/foobar.xml}}
```

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

```
\lstinputmhlisting[mhrepos=fooMH/bar,language=XML]{baz/foobar.xml}
```

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

We need to set up the packages by requiring the `metakeys` package [Koh20] to be loaded (in the right version).

```
1 ⟨*package⟩
2 \ProvidesPackage{mathhub}[2010/10/01 v1.2 Basic MathHub functionality]
3 \RequirePackage{keyval}
4 \RequirePackage{pathsuris}
```

4.1 `mathhub.sty`: General Infrastructure

\mhcurrentrepos   \mhcurrentrepos is used to initialize the current repository.

```
5 \newcommand\mhcurrentrepos[1]{\edef\mh@currentrepos{#1}}
```

\libinput   the \libinput macro inputs from the `lib` directory of the MathHub repository and then the `meta-inf/lib` repository of the group, if they exist. Since in practice nested libinputs may occur, we make sure that we stash the old values of \mh@inffile and \mh@libfile and restore them at the end.

```
 6 \def\modules@@first#1/#2;{#1}
 7 \newcommand\libinput[1]{%
 8 \ifcsvoid{mh@currentrepos}{%
 9   \PackageError{mathhub}{current MathHub repository not found}{}}%
10   {}
11 \edef\@mh@group{\expandafter\modules@@first\mh@currentrepos;}
12 \let\orig@inffile\mh@inffile\let\orig@libfile\mh@libfile%
13 \def\mh@inffile{\MathHub{\@mh@group/meta-inf/lib/#1}}%
14 \def\mh@libfile{\MathHub{\mh@currentrepos/lib/#1}}%
15 \if@iswindows@%
16   \path@to@windows\mh@inffile%
17   \path@to@windows\mh@libfile%
18 \fi%
19 \IfFileExists\mh@inffile{\input\mh@inffile}{}%
20 \IfFileExists\mh@inffile{}{\IfFileExists\mh@libfile{}{%
21   {\PackageError{mathhub}
22     {Library file missing; cannot input #1.tex\MessageBreak%
23     Both \mh@libfile.tex\MessageBreak and \mh@inffile.tex\MessageBreak%
24     do not exist}%
25   {Check whether the file name is correct}}}}
26 \IfFileExists\mh@libfile{\input\mh@libfile\relax}{}
27 \let\mh@inffile\orig@inffile\let\mh@libfile\orig@libfile}
```

\libusepackage   the \libusepackage is analogous to \libinput

```
28 \newcommand\libusepackage[2][]{%
29 \edef\@mh@group{\expandafter\modules@@first\mh@currentrepos;}
30 \let\orig@inffile\mh@inffile\let\orig@libfile\mh@libfile
31 \edef\mh@inffile{\MathHub{\@mh@group/meta-inf/lib/#2}}
32 \edef\mh@libfile{\MathHub{\mh@currentrepos/lib/#2}}%
33 \if@iswindows@%
```

```
34    \path@to@windows\mh@inffile%
35    \path@to@windows\mh@libfile%
36 \fi%
37 \IfFileExists{\mh@inffile.sty}{\usepackage[#1]{\mh@inffile}}{}%
38 \IfFileExists {\mh@inffile.sty}{}{\IfFileExists{\mh@libfile.sty}{}{%
39    {\PackageError{mathhub}
40      {Library file missing; cannot use package #2.sty\MessageBreak%
41      Both \mh@libfile.sty\MessageBreak and \mh@inffile.sty\MessageBreak%
42      do not exist}%
43    {Check whether the file name is correct}}}}
44 \IfFileExists\mh@libfile{\input\mh@libfile\relax}{}
45 \let\mh@inffile\orig@inffile\let\mh@libfile\orig@libfile}
```

Generally, the TEX formatter `pdflatex` needs to know the file system paths of the referenced STEX files: usually long relative paths. The `pathsuris` package [KGA20] from the STEX bundle makes this somewhat more palatable by supplying the `\defpath` macro, which we can use to set the MathHub root path, e.g. by `\defpath{MathHub}{/user/foo/localmh/MathHub}` (we will assume this setting for all examples below). Fortunately, we can compute this automatically.

We parse the `MATHHUB` environment variable via `kpsewhich` (LATEX can run this even in paranoid mode) and then set the MathMub path using `\defpath`.

```
46
47 \kpsewhich\mathhub@path{--var-value MATHHUB}
48 \if@iswindows@\windows@to@path\mathhub@path\fi
49 \ifx\mathhub@path\@empty%
50 \PackageError{mathhub}
51    {MATHHUB system variable not found or wrongly set}
52    {use export MATHHUB="<path>", where <path> points your MathHub direcctory}
53 \else\defpath{MathHub}{\mathhub@path}\fi
54 \message{^^JMATHHUB:>>\meaning\mathhub@path<<^^J}
55
```

`\findmanifest` `\findmanifest{⟨path⟩}` searches for a file `MANIFEST.MF` up and over ⟨path⟩ in the file system tree.

```
56 \def\findmanifest#1{
57    \@cpath{#1}
58    \ifx\@CanPath\@Slash
59      \def\manifest@mf{}
60    \else\ifx\@CanPath\@empty
61        \def\manifest@mf{}
62    \else
63      \edef\@findmanifest@path{\@CanPath/MANIFEST.MF}
64      \if@iswindows@\path@to@windows\@findmanifest@path\fi
65      %\message{^^JHere: \@findmanifest@path^^J}
66      \IfFileExists{\@findmanifest@path}{
67        %\message{MANIFEST.MF found at \@findmanifest@path}
68        \edef\manifest@mf{\@findmanifest@path}
69        \xdef\temp@archive@dir{\expandafter\detokenize\expandafter{\@CanPath}}
70      }{
```

```
71    \edef\@findmanifest@path{\@CanPath/META-INF/MANIFEST.MF}
72    \if@iswindows@\path@to@windows\@findmanifest@path\fi
73    \IfFileExists{\@findmanifest@path}{
74      %\message{MANIFEST.MF found at \@findmanifest@path}
75      \edef\manifest@mf{\@findmanifest@path}
76      \xdef\temp@archive@dir{\expandafter\detokenize\expandafter{\@CanPath}}
77    }{
78      \edef\@findmanifest@path{\@CanPath/meta-inf/MANIFEST.MF}
79      \if@iswindows@\path@to@windows\@findmanifest@path\fi
80      \IfFileExists{\@findmanifest@path}{
81        %\message{MANIFEST.MF found at \@findmanifest@path}
82        \edef\manifest@mf{\@findmanifest@path}
83        \xdef\temp@archive@dir{\expandafter\detokenize\expandafter{\@CanPath}}
84      }{
85        \findmanifest{\@CanPath/..}
86      }}}
87  \fi\fi
88 }
```

the next macro is a helper function for parsing `MANIFEST.MF`

```
89 \def\split@manifest@key{
90   \IfSubStr{\manifest@line}{\@Colon}{
91       \StrBefore{\manifest@line}{\@Colon}[\manifest@key]
92       \StrBehind{\manifest@line}{\@Colon}[\manifest@line]
93       \trimstring\manifest@line
94       \trimstring\manifest@key
95   }{
96       \def\manifest@key{}
97   }
98 }
```

the next helper function iterates over lines in `MANIFEST.MF`

```
 99 \def\parse@manifest@loop{
100   \ifeof\@manifest
101   \else
102     \read\@manifest to \manifest@line\relax
103     \edef\manifest@line{\expandafter\detokenize\expandafter{\manifest@line}}
104     \split@manifest@key
105     % id
106     \IfStrEq\manifest@key{\detokenize{id}}{
107         %\message{archive id: \manifest@line}
108         \xdef\manifest@mf@id{\manifest@line}
109     }{
110     % narration-base
111     \IfStrEq\manifest@key{\detokenize{narration-base}}{
112         %\message{archive narration-base: \manifest@line}
113         \xdef\manifest@mf@narr{\manifest@line}
114     }{
115     % namespace
116     \IfStrEq\manifest@key{\detokenize{source-base}}{
```

```
117       % \message{archive source-base: \manifest@line}
118         \xdef\manifest@mf@ns{\manifest@line}
119      }{
120      \IfStrEq\manifest@key{\detokenize{ns}}{
121         %\message{archive ns: \manifest@line}
122         \xdef\manifest@mf@ns{\manifest@line}
123      }{
124      % dependencies
125      \IfStrEq\manifest@key{\detokenize{dependencies}}{
126         %\message{archive dependencies: \manifest@line}
127         \xdef\manifest@mf@deps{\manifest@line}
128      }{
129      }}}}}
130      \parse@manifest@loop
131    \fi
132 }
```

\parsemanifest   \parsemanifest{⟨*macroname*⟩}{⟨*path*⟩} finds MANIFEST.MF via \findmanifest{⟨*path*⟩},
and parses the file, storing the individual fields (id, narr, ns and dependencies)
in ⟨*macroname*⟩id, ⟨*macroname*⟩narr, etc.

```
133 \newread\@manifest
134 \def\parsemanifest#1#2{
135    \gdef\temp@archive@dir{}
136    \findmanifest{#2}
137    \begingroup
138      \gdef\manifest@mf@id{}
139      \gdef\manifest@mf@narr{}
140      \gdef\manifest@mf@ns{}
141      \gdef\manifest@mf@deps{}
142      \openin\@manifest\manifest@mf
143      \parse@manifest@loop
144      \closein\@manifest
145    \endgroup
146    \if@iswindows@\windows@to@path\manifest@mf\fi
147    \cslet{#1id}\manifest@mf@id
148    \cslet{#1narr}\manifest@mf@narr
149    \cslet{#1ns}\manifest@mf@ns
150    \cslet{#1deps}\manifest@mf@deps
151    \cslet{#1dir}\temp@archive@dir
152 }
```

\setcurrentreposinfo   \setcurrentreposinfo{⟨*id*⟩} sets the current repository to ⟨*id*⟩, checks if the
MANIFEST.MF of this repository has already been read, and if not, find it, parses
it and stores the values in \currentrepos@⟨*key*⟩@⟨*id*⟩ for later retrieval.

```
153 \def\setcurrentreposinfo#1{%
154    \ifcsdef{currentrepos@dir@#1}{%
155      \mhcurrentrepos{#1}%
156    }{%
157      \parsemanifest{mathhub@archive@}{\MathHub{#1}}%
```

11

```
158    \@setcurrentreposinfo%
159  }%
160 }
161
162 \def\@setcurrentreposinfo{%
163  \mhcurrentrepos{\mathhub@archive@id}%
164  \csxdef{currentrepos@dir@\mathhub@archive@id}{\mathhub@archive@dir}%
165  \csxdef{currentrepos@narr@\mathhub@archive@id}{\mathhub@archive@narr}%
166  \csxdef{currentrepos@ns@\mathhub@archive@id}{\mathhub@archive@ns}%
167  \csxdef{currentrepos@deps@\mathhub@archive@id}{\mathhub@archive@deps}%
168 }
```

Finally – and that is the ultimate goal of all of the above, we set the current repos.

```
169 \parsemanifest{mathhub@archive@}\stex@maindir
170 \@setcurrentreposinfo
171 ⟨/package⟩
```

4.2 `omdoc--mh.sty`: MH Document Infrastructure

```
172 ⟨∗omdoc⟩
173 \ProvidesPackage{omdoc-mh}[2019/03/20 v1.1 MathHub support for OMDoc Documents]
174 \RequirePackage{mathhub}
```

\addmhbibresource

```
175 \newcommand\addmhbibresource[2][]{%
176  \def\@repos{#1}%
177  \ifx\@repos\@empty%
178  \addbibresource{\MathHub{\mh@currentrepos/#2}}%
179  \else
180  \addbibresource{\MathHub{\@repos/#2}}%
181  \fi%
182  \ignorespacesandpars}%
183 ⟨/omdoc⟩
```

4.3 `modules-mh.sty`: MH Variants for Modules

```
184 ⟨∗modules⟩
185 \ProvidesPackage{modules-mh}[2019/03/20 v1.1 MathHub support for the sTeX modules package]
186 \RequirePackage{mathhub}
```

\importmhmodule   The \importmhmodule[⟨*key=value list*⟩]{module} saves the current value of
\mh@currentrepos in a local macro \mh@@repos, resets \mh@currentrepos to
the new value if one is given in the optional argument, and after importing resets
\mh@currentrepos to the old value in \mh@@repos. We do all the \ifx compar-
ison with an \expandafter, since the values may be passed on from other key
bindings. Parameters will be passed to \importmodule.

```
187 \srefaddidkey{importmhmodule}%
188 \addmetakey{importmhmodule}{mhrepos}%
189 \addmetakey{importmhmodule}{path}%
```

```
190 \addmetakey{importmhmodule}{ext}% why does this exist?
191 \addmetakey{importmhmodule}{dir}%
192 \addmetakey[false]{importmhmodule}{conservative}[true]%
193 \newcommand\importmhmodule[2][]{%
194   \usemodule@maybesetcodes
195   \metasetkeys{importmhmodule}{#1}%
196   \ifx\importmhmodule@dir\@empty%
197     \edef\@path{\importmhmodule@path}%
198   \else\edef\@path{\importmhmodule@dir/#2}\fi%
199   \ifx\@path\@empty% if module name is not set
200     \importmodule[id=\importmhmodule@id]{#2}%
201   \else%
202     \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
203     \ifx\importmhmodule@mhrepos\@empty% if in the same repos
204       \relax% no need to change mh@currentrepos, i.e, current directory.
205     \else%
206       \setcurrentreposinfo{\importmhmodule@mhrepos}% change it.
207       \addto@thismodulex{\noexpand\setcurrentreposinfo{\importmhmodule@mhrepos}}%
208     \fi%
209     \importmodule[load=\MathHub{\mh@currentrepos/source/\@path},
210                        id=\importmhmodule@id]{#2}%
211     \setcurrentreposinfo{\mh@@repos}% after importing, reset to old value
212     \addto@thismodulex{\noexpand\setcurrentreposinfo{\mh@@repos}}%
213   \fi%
214   \ignorespacesandpars%
215 }
```

and now the analogs

\usemhmodule

```
216 \newcommand\usemhmodule[2][]{%
217 \metasetkeys{importmhmodule}{#1}%
218 \ifx\importmhmodule@dir\@empty%
219   \edef\@path{\importmhmodule@path}%
220 \else\edef\@path{\importmhmodule@dir/#2}\fi%
221 \ifx\@path\@empty%
222   \usemodule[id=\importmhmodule@id]{#2}%
223 \else%
224   \edef\mh@@repos{\mh@currentrepos}%
225   \ifx\importmhmodule@mhrepos\@empty%
226   \else\setcurrentreposinfo{\importmhmodule@mhrepos}\fi%
227   \usemodule[load=\MathHub{\mh@currentrepos/source/\@path},%
228                      id=\importmhmodule@id]{#2}%
229   \setcurrentreposinfo\mh@@repos%
230 \fi%
231 \ignorespacesandpars}
```

\mhinputref

```
232 \newcommand\mhinputref[2][]{%
233 \def\@repos{#1}%
```

```
234 \edef\mh@@repos{\mh@currentrepos}%
235 \ifx\@repos\@empty\else\setcurrentreposinfo{#1}\fi%
236 \inputref{\MathHub{\mh@currentrepos/source/#2}}%
237 \setcurrentreposinfo\mh@@repos%
238 \ignorespacesandpars}
```

\mhinput

```
239 \let\mhinput\mhinputref%
240 ⟨/modules⟩
```

4.4 `omtext-mh.sty`: MH Variants for OMText

```
241 ⟨∗omtext⟩
242 \ProvidesPackage{omtext-mh}[2019/03/20 v1.1 MathHub support for the sTeX omtext package]
243 \RequirePackage{mathhub}
```

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

```
244 \def\Gin@mhrepos{}
245 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
246 \newcommand\mhgraphics[2][]{\setkeys{Gin}{#1}%
247 \edef\mh@@repos{\mh@currentrepos}%
248 \ifx\Gin@mhrepos\@empty\edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%\includegraphics[#
249 \else\edef\temp@path{\MathHub{\Gin@mhrepos/source/#2}}\fi%\includegraphics[#1]{\MathHub{\Gin@mh
250 \if@iswindows@\path@to@windows\temp@path\fi
251 \includegraphics[#1]{\temp@path}
252 \def\Gin@mhrepos{}\setcurrentreposinfo\mh@@repos}
253 \newcommand\cmhgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}}
```

The following macros are deprecated.

```
254 \newcommand\mhcgraphics[2][]{\begin{center}\mhgraphics[#1]{#2}\end{center}
255   \PackageWarning{omtext-mh}{\protect\mhcgraphics\space is deprecated, use \protect\cmhgraphics
256 \newcommand\mhbgraphics[2][]{\fbox{\mhgraphics[#1]{#2}}
257   \PackageWarning{omtext-mh}{\protect\mhbgraphics\space is deprecated, use
258     \protect\mhgraphics\space and {center} instead}}
259 \newcommand\mhcbgraphics[2][]{\begin{center}\fbox{\mhgraphics[#1]{#2}}\end{center}
260   \PackageWarning{omtext-mh}{\protect\mhcbgraphics\space is deprecated, use
261     \protect\mhgraphics,\space {center}, and \protect\fbox\space instead}}
262 ⟨/omtext⟩
```

4.5 `smultiling-mh.sty`: MH Variants for Multilinguality

```
263 ⟨∗smultiling⟩
264 \ProvidesPackage{smultiling-mh}[2019/03/20 v1.1 MathHub support for the sTeX smultiling package
265 \RequirePackage{mathhub}
```

mhmodsig

```
266 \newenvironment{mhmodsig}{\begin{modsig}}{\end{modsig}}
```

mhmodnl:*

```
267 \addmetakey{mhmodnl}{mhrepos}
268 \addmetakey{mhmodnl}{path}
269 \addmetakey*{mhmodnl}{title}
270 \addmetakey*{mhmodnl}{creators}
271 \addmetakey*{mhmodnl}{contributors}
272 \addmetakey{mhmodnl}{srccite}
273 \addmetakey{primary}{mhmodnl}[yes]
```

mhmodnl   The mhmodnl environment is just a layer over the module environment and the
           \importmhmodule macro with the keys and language suitably adapted.

```
274 \newenvironment{mhmodnl}[3][]{\metasetkeys{mhmodnl}{#1}\def\@test{#1}%
275 \edef\@repos{\ifx\mhmodnl@mhrepos\@empty\mh@currentrepos\else\mhmodnl@mhrepos\fi}%
276 \edef\@load{\MathHub{\@repos/source/\ifx\mhmodnl@path\@empty #2\else\mhmodnl@path\fi}}%
277 \ifx\@test\@empty\begin{modnl}[load=\@load]{#2}{#3}\else\begin{modnl}[load=\@load,#1]{#2}{#3}\f
278 \ignorespacesandpars}
279 {\end{modnl}\ignorespacesandparsafterend}
```

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

```
280 \newenvironment{mhviewsig}[4][]{% keys, id, from, to
281 \def\@test{#1}\ifx\@test\@empty%
282 \begin{mhview}[id=#2]{#3}{#4}\else%
283 \begin{mhview}[id=#2,#1]{#3}{#4}\fi%
284 \ignorespacesandpars}
285 {\end{mhview}\ignorespacesandparsafterend}
```

mhviewnl   The mhviewnl environment is just a layer over the mhview environment with the
EdN:2   keys and language suitably adapted.[2]

```
286 \newenvironment{mhviewnl}[5][]{% keys, id, lang, from, to
287 \def\@test{#1}\ifx\@test\@empty%
288 \begin{mhview}[id=#2.#3]{#4}{#5}\else%
289 \begin{mhview}[id=#2.#3,#1]{#4}{#5}\fi%
290 \ignorespacesandpars}
291 {\end{mhview}\ignorespacesandparsafterend}
292 ⟨/smultiling⟩
```

4.6 structview-mh.sty: MH Variants for Structures and Views

```
293 ⟨*structview⟩
294 \ProvidesPackage{structview-mh}[2019/03/20 v1.1 MathHub support for the sTeX structview package
295 \RequirePackage{mathhub}
```

mhstructure

```
296 \newenvironment{mhstructure}[3][]{%
```

---

[2]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 .⟨lang⟩!

```
297    \gdef\@@doit{\importmhmodule[#1]{#3}}%
298    \ifmod@show\par\noindent structure import "#2" from module #3 \@@doit\fi%
299    \ignorespacesandpars}
300 {\aftergroup\@@doit\ifmod@show end import\fi%
301    \ignorespacesandparsafterend}
```

importmhmodulevia   this is now deprecated, we give an error

```
302 \newenvironment{importmhmodulevia}[2][]%
303 {\PackageError{structview-mh}%
304    {The {importmhmodulevia} environment is deprecated}{use the {mhstructure} instead!}%
305    \begin{mhstructure}[#1]{missing}{#2}}
306 {\end{mhstructure}}
```

```
307 \srefaddidkey{mhview}
308 \addmetakey{mhview}{display}
309 \addmetakey{mhview}{creators}
310 \addmetakey{mhview}{contributors}
311 \addmetakey{mhview}{srccite}
312 \addmetakey*{mhview}{title}
313 \addmetakey{mhview}{type}
314 \addmetakey{mhview}{fromrepos}
315 \addmetakey{mhview}{torepos}
316 \addmetakey{mhview}{frompath}
317 \addmetakey{mhview}{topath}
```

mhview   the MathHub version

```
318 \newenvironment{mhview}[3][]%  keys, from, to
319 {\metasetkeys{mhview}{#1}%
320    \sref@target%
321    \begin{@mhview}{#2}{#3}%
322    \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
323    \ignorespacesandpars}
324 {\end{@mhview}\ignorespacesandparsafterend}
325 \ifmod@show\surroundwithmdframed{mhview}\fi
```

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

```
326 \newenvironment{@mhview}[2]{%from, to
327    \usemhmodule[mhrepos=\mhview@fromrepos,path=\mhview@frompath]{#1}%
328    \usemhmodule[mhrepos=\mhview@torepos,path=\mhview@topath]{#2}%
329 }{}%
```

mhviewsketch   The `mhviewsketch` environment is deprecated, we give an error

```
330 \newenvironment{mhviewsketch}[3][]%
331 {\PackageError{structview}%
332    {The {mhviewsketch} environment is deprecated}{use the {mhview} instead!}%
333    \begin{mhview}[#1]{#2}{#3}}
334 {\end{mhview}}
```

mhinlineView   Analogous modification to `inlineView`

```
335 \newenvironment{mhinlineView}[2][]% keys, source
336 {\metasetkeys{mhview}{#1}\sref@target%
337   \importmhmodule[mhrepos=\mhview@fromrepos,path=\mhview@frompath]{#2}%
338   \ignorespacesandpars}
339 {\ignorespacesandpars}
```

mhinlineview

```
340 \newcommand\mhinlineview[3][]{\begin{mhinlineView}[#1]{#2}{\module@id}#3\end{mhinlineView}}
341 ⟨/structview⟩
```

4.7   mikoslides-mh.sty: Support for MiKo Slides

```
342 ⟨∗mikoslides⟩
343 \ProvidesPackage{mikoslides-mh}[2019/03/20 v1.1 MathHub support for the sTeX mikoslides package]
344 \RequirePackage{mathhub}
```

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

```
345 \def\Gin@mhrepos{}
346 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
347 \newcommand\mhframeimage[2][]{%
348   \setkeys{Gin}{#1}%
349   \edef\mh@@repos{\mh@currentrepos}%
350   \ifx\Gin@mhrepos\@empty%
351     \edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%
352   \else%
353     \edef\temp@path{\MathHub{\Gin@mhrepos/source/#2}}%
354   \fi%
355   \if@iswindows@\path@to@windows\temp@path\fi%
356   \frameimage[#1]{\temp@path}%
357 }%
```

\mhinputref*

```
358 \let\orig@mhinputref\mhinputref
359 \def\mhinputref{\@ifstar\nmhinputref\orig@mhinputref}
360 \newcommand\nmhinputref[2][]{\ifnotes\orig@mhinputref[#1]{#2}\fi}
361 ⟨/mikoslides⟩
```

4.8   problem-mh.sty: Support for Problems

```
362 ⟨∗problem⟩
363 \ProvidesPackage{problem-mh}[2019/03/20 v1.1 MathHub support for the sTeX problem package]
364 \RequirePackage{mathhub}
```

\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.

```
365 \addmetakey{inclprob}{mhrepos}
```

```
366 \newcommand\includemhproblem[2][]{\metasetkeys{inclprob}{#1}%
367 \edef\mh@@repos{\mh@currentrepos}%
368 \ifx\inclprob@mhrepos\@empty\else\setcurrentreposinfo\inclprob@mhrepos\fi%
369 \edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}
370 \if@iswindows@\path@to@windows\temp@path\fi
371 \input{\temp@path}%
372 \setcurrentreposinfo\mh@@repos\clear@inclprob@keys}
373 ⟨/problem⟩
```

4.9 **hwexam-mh.sty:** Support for Assignments

```
374 ⟨∗hwexam⟩
375 \ProvidesPackage{hwexam-mh}[2019/03/20 v1.1 MathHub support for the sTeX hwexam package]
376 \RequirePackage{mathhub}
```

\inputmhassignment   The \inputmhassignment 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.

```
377 \newcommand\inputmhassignment[2][]{\metasetkeys{inclassig}{#1}%
378 \edef\mh@@repos{\mh@currentrepos}%
379 \ifx\inclassig@mhrepos\@empty\else\setcurrentreposinfo\inclassig@mhrepos\fi%
380 \edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%
381 \if@iswindows@\path@to@windows\temp@path\fi%
382 \inputassignment[#1]{\temp@path}%
383 \setcurrentreposinfo\mh@@repos\clear@inclassig@keys}%
384 \newcommand\includemhassignment[2][]{\newpage\inputmhassignment[#1]{#2}}
385 ⟨/hwexam⟩
```

4.10 **tikzinput-mh.sty:** Support for Assignments

```
386 ⟨∗tikzinput⟩
387 \ProvidesPackage{tikzinput-mh}[2019/03/20 v1.1 MathHub support for the sTeX tikzinput package]
388 \RequirePackage{mathhub}
389 \RequirePackage{pathsuris}

390 \define@key{Gin}{mhrepos}{\def\Gin@mhrepos{#1}}
391 \newcommand\mhtikzinput[2][]{\def\Gin@mhrepos{}\setkeys{Gin}{#1}%
392 \edef\mh@@repos{\mh@currentrepos}%
393 \ifx\Gin@mhrepos\@empty\edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%
394 \else\setcurrentreposinfo\Gin@mhrepos\edef\temp@path{\MathHub{\Gin@mhrepos/source/#2}}\fi%
395 \if@iswindows@\path@to@windows\temp@path\fi%
396 \tikzinput[#1]{\temp@path}
397 \def\Gin@mhrepos{}\setcurrentreposinfo\mh@@repos}
398 \newcommand\cmhtikzinput[2][]{\begin{center}\mhtikzinput[#1]{#2}\end{center}}
399 ⟨/tikzinput⟩
```

4.11 **lstmh.sty:** Support for Listings

```
400 ⟨∗lst⟩
401 \ProvidesPackage{lstmh}[2019/03/20 v1.1 MathHub support for the listings package]
```

```
402 \RequirePackage{mathhub}
403 \RequirePackage{pathsuris}
404 \RequirePackage{listings}

405 \define@key{lst}{mhrepos}{\def\lst@mhrepos{#1}}
406 \newcommand\lstinputmhlisting[2][]{\def\lst@mhrepos{}\setkeys{lst}{#1}%
407 \edef\mh@@repos{\mh@currentrepos}%
408 \ifx\lst@mhrepos\@empty\edef\temp@path{\MathHub{\mh@currentrepos/source/#2}}%
409 \else\edef\temp@path{\MathHub{\lst@mhrepos/source/#2}}\fi%
410 \if@iswindows@\path@to@windows\temp@path\fi%
411 \lstinputlisting[#1]{\temp@path}
412 \def\lst@mhrepos{}\setcurrentreposinfo\mh@@repos}
413 \newcommand\clstinputmhlisting[2][]{\begin{center}\lstinputmhlisting[#1]{#2}\end{center}}
414 ⟨/lst⟩
```

Change History

References

[ ]   *The Local MathHub Tool (LMH)*. URL: `https://uniformal.github.io/doc/archives/Mathhub/lmh.html` (visited on 10/02/2020).

[ ]   *The MMT Language and System; Building Documents*. URL: `http://uniformal.github.io/doc/applications/building` (visited on 10/02/2020).

[GH]   *GitHub: Build software better, together*. URL: `http://github.com` (visited on 02/24/2014).

[GL]   *The first single application for the entire DevOps lifecycle – GitLab*. URL: `http://gitlab.com` (visited on 01/12/2019).

[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: `https://kwarc.info/frabe/Research/HIJKR_dimensions_11.pdf`.

[KGA20]   Michael Kohlhase, Deyan Ginev, and Rares Ambrus. *modules.sty: Semantic Macros and Module Scoping in sTeX*. Tech. rep. 2020. URL: `https://github.com/sLaTeX/sTeX/raw/master/sty/pathsuris/pathsuris.pdf`.

[Koh20]   Michael Kohlhase. *metakeys.sty: A generic framework for extensible Metadata in LaTeX*. Tech. rep. 2020. URL: `https://github.com/sLaTeX/sTeX/raw/master/sty/metakeys/metakeys.pdf`.

[MMT]   *MMT – Language and System for the Uniform Representation of Knowledge*. Project web site. URL: `https://uniformal.github.io/` (visited on 01/15/2019).

[sTeX]   *sTeX: A semantic Extension of TeX/LaTeX*. URL: `https://github.com/sLaTeX/sTeX` (visited on 05/11/2020).