

MathHub Support for \LaTeX^*

Michael Kohlhasse
Jacobs University, Bremen
<http://kwarc.info/kohlhasse>

January 14, 2016

Abstract

The `sref` package is part of the \LaTeX collection, a version of $\text{\TeX}/\text{\LaTeX}$ that allows to markup $\text{\TeX}/\text{\LaTeX}$ documents semantically without leaving the document format, essentially turning $\text{\TeX}/\text{\LaTeX}$ into a document format for mathematical knowledge management (MKM).

The `mathhub` packages extend \LaTeX with support for the MathHub.info portal

Contents

1	Introduction	3
2	The User Interface	3
2.1	Package Options	3
2.2	<code>modules-mh</code> : MH Variants for Modules	3
2.3	<code>omtext-mh</code> : MH Variants for OMText	4
2.4	<code>smultiling-mh</code> : MH Variants for Multilinguality	4
2.5	<code>structview-mh</code> : MH Variants for Structures and Views	4
2.6	<code>mikoslides-mh</code> : Support for MiKo Slides	4
2.7	<code>problem-mh</code> : Support for Problems	5
2.8	<code>hwexam-mh</code> : Support for Assignments	5
3	Limitations	5
4	Implementation	6
4.1	General Infrastructure	6
4.2	<code>modules-mh</code> : MH Variants for Modules	6
4.3	<code>omtext-mh</code> : MH Variants for OMText	8
4.4	<code>smultiling-mh</code> : MH Variants for Multilinguality	9
4.5	<code>structview-mh</code> : MH Variants for Structures and Views	10

*Version ? (last revised ?)

4.6	<code>mikoslides-mh</code> : Support for MiKo Slides	11
4.7	<code>problem-mh</code> : Support for Problems	12
4.8	<code>hwexam-mh</code> : Support for Assignments	12
4.9	<code>tikzinput-mh</code> : Support for Assignments	13

1 Introduction

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

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

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

2 The User Interface

2.1 Package Options

none so far

2.2 modules-mh: MH Variants for Modules

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

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

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

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

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

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

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

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

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

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

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

2.3 omtex-mh: MH Variants for OMText

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

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

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

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

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

2.4 smultiling-mh: MH Variants for Multilinguality

1 2

2.5 structview-mh: MH Variants for Structures and Views

3

2.6 mikosides-mh: Support for MiKo Slides

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

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

¹EDNOTE: needs to be documented

²EDNOTE: mhmodsig seems to be missing what happened?

³EDNOTE: needs to be documented

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

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

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

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

```
\mhframeimage{baz/foobar}
```

2.7 problem-mh: Support for Problems

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

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

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

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

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

2.8 hwexam-mh: Support for Assignments

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

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

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

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

3 Limitations

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

1. none reported yet.

4 Implementation

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

```
1 <*package>
2 \RequirePackage{keyval}
3 </package>
```

4.1 General Infrastructure

`\mhcurrentrepos` `\mhcurrentrepos` is used to initialize the current repository. If the repos has changed, it writes a call to the internal macro `\@mhcurrentrepos` for the aux file and calls it. So that the `\importmodule` calls there work with the correct repos.

```
4 <*package>
5 \newcommand\mhcurrentrepos[1]{%
6   \edef\@test{#1}%
7   \ifx\@test\mhcurrentrepos% if new dir = old dir
8     \relax% no need to change
9   \else%
10    \protected@write\@auxout{}\string\@mhcurrentrepos{#1}%
11    \fi%
12    \@mhcurrentrepos{#1}% define mhcurrentrepos
13 }%
14 \newcommand\@mhcurrentrepos[1]{\edef\mhcurrentrepos{#1}}%
15 </package>
```

`\libinput` the `\libinput` macro inputs from the `lib` directory of the MathHub repository or the `meta-inf/lib` repos of the group.

```
16 <*package>
17 \def\modules@@first#1/#2;{#1}
18 \newcommand\libinput[1]{\def\@libfile{\MathHub{\mhcurrentrepos/lib/#1}}%
19 \IfFileExists{\@libfile}{\input\@libfile}%
20 {\edef\@group{\expandafter\modules@@first\mhcurrentrepos;}
21 \edef\@infile{\MathHub{\@group/meta-inf/lib/#1}}
22 \IfFileExists{\@infile}{\input{\@infile}}%
23 {\PackageError{modules}
24   {Library file missing, cannot input #1\MessageBreak%
25     Both \@libfile.tex\MessageBreak and \@infile.tex\MessageBreak do not exist}%
26   {Check whether the file name is correct}}}%
27 </package>
```

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.

```
28 <*modules>
29 \ProvidesPackage{modules-mh}[2015/11/22 v1.0 MathHub support for the sTeX modules package]
30 \RequirePackage{mathhub}
```

```

31 </modules>

\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.

32 <*modules>
33 \srefaddidkey{importmhmodule}%
34 \addmetakey{importmhmodule}{repos}% saves the repo's path. E.g: smglom/numberfield
35 \addmetakey{importmhmodule}{path}% saves the module name. E.g: naturalnumbers
36 \addmetakey[sms]{importmhmodule}{ext}% saves the extension: E.g: tex
37 \addmetakey[false]{importmhmodule}{conservative}[true]%
38 \newcommand\importmhmodule[2][]{%
39   \metasetkeys{importmhmodule}{#1}%
40   \ifx\importmhmodule@path\@empty% if module name is not set
41     \importmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
42   \else%
43     \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
44     \ifx\importmhmodule@repos\@empty% if in the same repos
45       \relax% no need to change mh@currentrepos, i.e, current dirctory.
46     \else%
47       \mhcurrentrepos{\importmhmodule@repos}% change it.
48     \fi%
49     \importmodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},%
50     ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
51     \mhcurrentrepos{\mh@@repos}% after importing, reset to old value
52   \fi%
53   \ignorespaces%
54 }%
55 </modules>

```

and now the analogs

\usemhmodule

```

56 <*modules>
57 \newcommand\usemhmodule[2][]{%
58   \metasetkeys{importmhmodule}{#1}%
59   \ifx\importmhmodule@path\@empty%
60     \usemodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
61   \else%
62     \edef\mh@@repos{\mh@currentrepos}%
63     \ifx\importmhmodule@repos\@empty%
64     \else%
65       \mhcurrentrepos{\importmhmodule@repos}%
66     \fi%
67     \usemodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodule@
68     \mhcurrentrepos\mh@@repos%

```

```

69 \fi%
70 \ignorespaces%
71 }%
72 </modules>

\mhinputref
73 <*modules>
74 \newcommand\mhinputref[2] [] {%
75 \def\@repos{#1}%
76 \edef\mh@repos{\mh@currentrepos}%
77 \ifx\@repos\empty%
78 \else%
79 \mhcurrentrepos{#1}%
80 \fi%
81 \inputref{\MathHub{\mh@currentrepos/source/#2}}%
82 \mhcurrentrepos\mh@repos%
83 \ignorespaces%
84 }%
85 </modules>

```

```

\mhinput
86 <*modules>
87 \let\mhinput\mhinputref%
88 </modules>

```

4.3 omtex-mh: MH Variants for OMTex

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

```

89 <*omtex>
90 \ProvidesPackage{omtex-mh}[2015/11/22 v1.0 MathHub support for the sTeX omtex package]
91 \RequirePackage{mathhub}
92 </omtex>

\mh*graphics Use the current value of \mh@currentrepos or the value of the mhrepos key if it
is given in \my*graphics.
93 <*omtex>
94 \def\Gin@mhrepos{}
95 \define@key{Gin}{mhrepos}{\csxdef\Gin@mhrepos{#1}}
96 \newcommand\mhgraphics[2] [] {\setkeys{Gin}{#1}%
97 \edef\mh@repos{\mh@currentrepos}%
98 \ifx\Gin@mhrepos\empty\mygraphics[#1]{\MathHub{\mh@currentrepos/source/#2}}%
99 \else\mygraphics[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
100 \def\Gin@mhrepos{\mhcurrentrepos\mh@repos}
101 \newcommand\mhgraphics[2] [] {\begin{center}\mhgraphics[#1]{#2}\end{center}}
102 \newcommand\mhgraphics[2] [] {\fbox{\mhgraphics[#1]{#2}}}
103 \newcommand\mhgraphics[2] [] {\begin{center}\fbox{\mhgraphics[#1]{#2}}\end{center}}
104 </omtex>

```


4.4 smultiling-mh: MH Variants for Multilinguality

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

```
105 <*smultiling>
106 \ProvidesPackage{smultiling-mh}[2015/11/22 v1.0 MathHub support for the sTeX smultiling package]
107 \RequirePackage{mathhub}
108 </smultiling>
```

`mhmodnl:`

```
109 <*smultiling>
110 \addmetakey{mhmodnl}{repos}
111 \addmetakey{mhmodnl}{path}
112 \addmetakey*{mhmodnl}{title}
113 \addmetakey*{mhmodnl}{creators}
114 \addmetakey*{mhmodnl}{contributors}
115 \addmetakey{mhmodnl}{srccite}
116 \addmetakey{primary}{mhmodnl}[yes]
117 </smultiling>
```

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

```
118 <*smultiling>
119 \newenvironment{mhmodnl}[3][\metasetkeys{mhmodnl}{#1}%
120 \def\@test{#1}\ifx\@test\@empty\begin{module}[id=#2.#3]\else\begin{module}[id=#2.#3,#1]\fi%
121 \edef\@repos{\ifx\mhmodnl@repos\@empty\mh@currentrepos\else\mhmodnl@repos}
122 \if@langfiles\importmhmodule[repos=\@repos,load=#2,ext=tex]{#2}\else
123 \ifx\mhmodnl@load\@empty\importmodule{#2}\else\importmodule[ext=tex,load=\mhmodnl@load]{#2}\fi%
124 \fi}
125 {\end{module}}
126 </smultiling>
```

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

```
127 <*smultiling>
128 \newenvironment{mhviewsig}[4][\def\@test{#1}\ifx\@test\@empty%
129 \begin{mhview}[id=#2,ext=tex]{#3}{#4}\else%
130 \begin{mhview}[id=#2,#1,ext=tex]{#3}{#4}\fi}
131 {\end{mhview}}
```

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

```
132 \newenvironment{mhviewnl}[5][\def\@test{#1}\ifx\@test\@empty%
133 \begin{mhviewsketch}[id=#2.#3,ext=tex]{#4}{#5}\else%
134 \begin{mhviewsketch}[id=#2.#3,#1,ext=tex]{#4}{#5}\fi}
135 {\end{mhviewsketch}}
136 </smultiling>
```

⁴EdNOTE: MK: we have to do something about the `if@langfiles` situation here. But this is non-trivial, since we do not know the current path, to which we could append `.(lang)`!

4.5 structview-mh: MH Variants for Structures and Views

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

```

137 <*structview>
138 \ProvidesPackage{structview-mh}[2015/11/22 v1.0 MathHub support for the sTeX structview package]
139 \RequirePackage{mathhub}
140 </structview>

importmhmodulevia

141 <*structview>
142 \newenvironment{importmhmodulevia}[3][]{%
143   \gdef\@doit{\importmhmodule[#1]{#2}{#3}}%
144   \ifmod@show\par\noindent importing module #2 via \@doit\fi
145 }{%
146   \aftergroup\@doit\ifmod@show end import\fi%
147 }%
148 </structview>

149 <*structview>
150 \srefaddidkey{mhview}
151 \addmetakey{mhview}{display}
152 \addmetakey{mhview}{creators}
153 \addmetakey{mhview}{contributors}
154 \addmetakey{mhview}{srccite}
155 \addmetakey*{mhview}{title}
156 \addmetakey{mhview}{fromrepos}
157 \addmetakey{mhview}{torepos}
158 \addmetakey{mhview}{frompath}
159 \addmetakey{mhview}{topath}
160 \addmetakey[sms]{mhview}{ext}
161 </structview>

mhview the MathHub version

162 <*structview>
163 \newenvironment{mhview}[3][]{% keys, from, to
164   \metasetkeys{mhview}{#1}%
165   \sref@target%
166   \begin{@mhview}{#2}{#3}%
167   \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
168 }{%
169   \end{@mhview}%
170   \ignorespaces%
171 }%
172 \ifmod@show\surroundwithmdframed{mhview}\fi
173 </structview>

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

174 <*structview>
175 \newenvironment{@mhview}[2]{%from, to

```

```

176 \importmhmodule[repos=\mhview@fromrepos,path=\mhview@frompath,ext=\mhview@ext]{#1}%
177 \importmhmodule[repos=\mhview@torepos,path=\mhview@topath,ext=\mhview@ext]{#2}%
178 }{}%
179 \end{structview}

```

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

```

180 \newenvironment{mhviewsketch}[3][]{%
181 \metasetkeys{mhview}{#1}%
182 \sref@target%
183 \begin{@mhview}{#2}{#3}%
184 \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
185 }{}%
186 \end{@mhview}%
187 \ignorespaces%
188 }{}%
189 \ifmod@show\surroundwithmdframed{mhviewsketch}\fi
190 \end{structview}

```

4.6 mikosides-mh: Support for MiKo Slides

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

```

192 \newpackage{mikosides}
193 \ProvidesPackage{mikosides-mh}[2015/11/22 v1.0 MathHub support for the sTeX mikosides package]
194 \RequirePackage{mathhub}
195 \end{package}

```

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

```

196 \newpackage{mikosides}
197 \def\Gin@mhrepos{}
198 \define@key{Gin}{mhrepos}{\csxdef\Gin@mhrepos{#1}}
199 \end{package}
200 \newpackage{mikosides}
201 \newcommand\mhframeimage[2][]{%
202 \setkeys{Gin}{#1}%
203 \edef\mh@crepos{\mh@currentrepos}%
204 \ifx\Gin@mhrepos\empty%
205 \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%
206 \else%
207 \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%
208 \fi%
209 }{}%
210 \end{package}

```

4.7 problem-mh: Support for Problems

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

```
211 <*problem>
212 \ProvidesPackage{problem-mh}[2015/11/22 v1.0 MathHub support for the sTeX problem package]
213 \RequirePackage{mathhub}
214 </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 the optional argument, and after importing resets `\mh@currentrepos` to the old value in `\mh@@repos`.

```
215 <*problem>
216 \newcommand\includemhproblem[2] [] {\metasetkeys{inclprob}{#1}%
217 \edef\mh@@repos{\mh@currentrepos}%
218 \ifx\inclprob@mhrepos\@empty\else\mhcurrentrepos\inclprob@mhrepos\fi%
219 \input{\MathHub{\mh@currentrepos/source/#2}}}%
220 \mhcurrentrepos\mh@@repos\clear@inclprob@keys}
221 </problem>
```

4.8 hwexam-mh: Support for Assignments

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

```
222 <*hwexam>
223 \ProvidesPackage{hwexam-mh}[2015/11/22 v1.0 MathHub support for the sTeX hwexam package]
224 \RequirePackage{mathhub}
225 </hwexam>
```

`\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`.

```
226 <*hwexam>
227 \newcommand\includemhassignment[2] [] {\metasetkeys{inclassig}{#1}%
228 \edef\mh@@repos{\mh@currentrepos}%
229 \ifx\inclassig@mhrepos\@empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
230 \includeassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}}%
231 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
232 </hwexam>
```

`\inputmhassignment` analogous

```
233 <*hwexam>
234 \newcommand\inputmhassignment[2] [] {\metasetkeys{inclassig}{#1}%
235 \edef\mh@@repos{\mh@currentrepos}%
236 \ifx\inclassig@mhrepos\@empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
237 \inputassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}}%

```

```

238 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
239 </hwexam>

```

4.9 tikzinput-mh: Support for Assignments

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

```

240 <*tikzinput>
241 \ProvidesPackage{tikzinput-mh}[2015/11/22 v1.0 MathHub support for the sTeX tikzinput package]
242 \RequirePackage{mathhub}
243 </tikzinput>

244 <*tikzinput>
245 \define@key{Gin}{mhrepos}{\csxdef\Gin@mhrepos{#1}}
246 \newcommand\mhtikzinput[2][]{\def\Gin@mhrepos{}\setkeys{Gin}{#1}%
247 \edef\mh@@repos{\mh@currentrepos}%
248 \ifx\Gin@mhrepos\empty\tikzinput[#1]{\MathHub{\mh@currentrepos/source/#2}}%
249 \else\tikzinput[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
250 \def\Gin@mhrepos{}\mhcurrentrepos\mh@@repos}
251 \newcommand\cmhtikzinput[2][]{\begin{center}\mhtikzinput[#1]{#2}\end{center}}
252 </tikzinput>

```

References

- [Hor+11] Fulya Horozal et al. “Combining Source, Content, Presentation, Narration, and Relational Representation”. In: *Intelligent Computer Mathematics*. Ed. by James Davenport et al. LNAI 6824. Springer Verlag, 2011, pp. 212–227. ISBN: 978-3-642-22672-4. URL: http://kwarc.info/frabe/Research/HIJKR_dimensions_11.pdf.
- [Koh15] Michael Kohlhasse. *metakeys.sty: A generic framework for extensible Metadata in L^AT_EX*. Tech. rep. Comprehensive T_EX Archive Network (CTAN), 2015. URL: <http://www.ctan.org/tex-archive/macros/latex/contrib/stex/metakeys/metakeys.pdf>.
- [sTeX] *KWARC/sTeX*. URL: <https://svn.kwarc.info/repos/stex> (visited on 05/15/2015).