

MathHub Support for \LaTeX^*

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

November 4, 2015

Abstract

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

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

Contents

1	Introduction	2
2	The User Interface	2
2.1	Package Options	2
2.2	<code>mh-modules</code> : MH Variants for Modules	2
2.3	<code>omtext-mh</code> : MH Variants for OMText	3
2.4	<code>mikoslides-mh</code> : Support for MiKo Slides	3
3	Limitations	4
4	Implementation	5
4.1	Package Options	5
4.2	General Infrastructure	5
4.3	MH Variants for Modules	6
4.4	<code>omtext-mh</code> : MH Variants for OMText	11
4.5	<code>mh-mikoslides</code> : Support for MiKo Slides	12
4.6	Finale	12

*Version v1.2 (last revised 2015/10/27)

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 mh-modules: 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` and `\adoptmhmodule` macros are the analogs to `\usemodule` and `\adoptmodule`.

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

2.3 omtex-mh: MH Variants for OMTex

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

```
\defpath{MathHub}{/user/foo/lmh/MathHub}
\mygraphics{\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 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}}
```

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}
```

3 Limitations

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

1. none reported yet.

4 Implementation

The `sref` package generates two files: the `LATEX` package (all the code between `<*package>` and `</package>`) and the `LATEXML` bindings (between `<*ltxml>` and `</ltxml>`). 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 file.

```
1 <*ltxml>
2 package LaTeXML::Package::Pool;
3 use strict;
4 use LaTeXML::Package;
5 </ltxml>
```

4.1 Package Options

We declare some switches which will modify the behavior according to the package options. Generally, an option `xxx` will just set the appropriate switches to true (otherwise they stay false).¹

```
6 <*package>
7 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{metakeys}}
8 \ProcessOptions
9 </package>
10 <*ltxml>
11 DeclareOption(undef,sub {PassOptions('metakeys','sty',ToString(Digest(T_CS('\CurrentOption'))))}
12 </ltxml>
```

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

```
13 <*package>
14 \RequirePackage{keyval}
15 </package>
16 <*ltxml>
17 RequirePackage('keyval');
18 </ltxml>
```

4.2 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.

```
19 <*package>
20 \newrobustcmd\mhcurrentrepos[1]{%
21   \edef\@test{#1}%
22   \ifx\@test\mh@currentrepos% if new dir = old dir
23     \relax% no need to change
24   \else%
```

¹EdNOTE: do we need this?

```

25 \protected@write\@auxout{}\string\mhcurrentrepos{#1}}%
26 \fi%
27 \mhcurrentrepos{#1}% define mhcurrentrepos
28 }%
29 \newrobustcmd\mhcurrentrepos[1]{\edef\mhcurrentrepos{#1}}%
30 \</package>
31 \<txml>
32 DefMacro('mhcurrentrepos{#1}', '\mhcurrentrepos{#1}');
33 DefMacro('mhcurrentrepos{#1}', '\def\mhcurrentrepos{#1}\@mhcurrentrepos{#1}');
34 DefConstructor('\@mhcurrentrepos{#1}', '',
35 afterDigest => sub{ AssignValue('current_repos', ToString($_[1]->getArg(1)), 'global'); } );
36 \</txml>#\$

```

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

```

37 \<txml>RaxTeX'
38 \<package | txml>
39 \def\modules@@first#1/#2;{#1}
40 \newcommand\libinput[1]{\def\@libfile{\MathHub{\mhcurrentrepos/lib/#1}}%
41 \IfFileExists{\@libfile}{\input\@libfile}%
42 {\edef\@group{\expandafter\modules@@first\mhcurrentrepos;}
43 \edef\@inffile{\MathHub{\@group/meta-inf/lib/#1}}
44 \IfFileExists{\@inffile}{\input\@inffile}}%
45 {\PackageError{modules}
46 {Library file missing, cannot input #1\MessageBreak%
47 Both \@libfile.tex\MessageBreak and \@inffile.tex\MessageBreak do not exit}%
48 {Check whether the file name is correct}}}%
49 \</package | txml>
50 \<txml>' );

```

4.3 MH Variants for Modules

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

```

51 \<modules>
52 \srefaddidkey{importmhmodule}%
53 \addmetakey{importmhmodule}{repos}% saves the repo's path. E.g: smglom/numberfield
54 \addmetakey{importmhmodule}{path}% saves the module name. E.g: naturalnumbers
55 \addmetakey[sms]{importmhmodule}{ext}% saves the extension: E.g: tex
56 \addmetakey[false]{importmhmodule}{conservative}[true]%
57 \newrobustcmd\importmhmodule[2] [] {%
58 \metasetkeys{importmhmodule}{#1}%
59 \ifx\importmhmodule@path\empty% if module name is not set
60 \importmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
61 \else%

```

```

62 \edef\mh@@repos{\mh@currentrepos}% remember so that we can reset it.
63 \ifx\importmhmodule@repos\@empty% if in the same repos
64 \relax% no need to change mh@currentrepos, i.e, current dirctory.
65 \else%
66 \mhcurrentrepos{\importmhmodule@repos}% change it.
67 \fi%
68 \importmodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},%
69 ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
70 \mhcurrentrepos{\mh@@repos}% after importing, reset to old value
71 \fi%
72 \ignorespaces%
73 }%

```

and now the analogs

`\usemhmodule`

```

74 \newrobustcmd\usemhmodule[2][\{%
75 \metasetkeys{importmhmodule}{#1}%
76 \ifx\importmhmodule@path\@empty%
77 \usemodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
78 \else%
79 \edef\mh@@repos{\mh@currentrepos}%
80 \ifx\importmhmodule@repos\@empty%
81 \else%
82 \mhcurrentrepos{\importmhmodule@repos}%
83 \fi%
84 \usemodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodule@
85 \mhcurrentrepos\mh@@repos%
86 \fi%
87 \ignorespaces%
88 }%

```

`\adoptmhmodule`

```

89 \newrobustcmd\adoptmhmodule[2][\{%
90 \metasetkeys{importmhmodule}{#1}%
91 \ifx\importmhmodule@path\@empty
92 \adoptmodule[ext=\importmhmodule@ext,id=\importmhmodule@id]{#2}%
93 \else%
94 \edef\mh@@repos{\mh@currentrepos}%
95 \ifx\importmhmodule@repos\@empty%
96 \else%
97 \mhcurrentrepos{\importmhmodule@repos}%
98 \fi%
99 \adoptmodule[load=\MathHub{\mh@currentrepos/source/\importmhmodule@path},ext=\importmhmodul
100 \mhcurrentrepos\mh@@repos%
101 \fi%
102 \ignorespaces%
103 }%

```

`\mhinputref`

```

104 \newrobustcmd\mhinputref[2][]{%
105   \def\@repos{#1}%
106   \edef\mh@crepos{\mh@currentrepos}%
107   \ifx\@repos\@empty%
108   \else%
109     \mhcurrentrepos{#1}%
110   \fi%
111   \inputref{\MathHub{\mh@currentrepos/source/#2}}}%
112   \mhcurrentrepos\mh@crepos%
113   \ignorespaces%
114 }%

\mhinput
115 \let\mhinput\mhinputref%

importmhmodulevia
116 \newenvironment{importmhmodulevia}[3][]{%
117   \gdef\@doit{\importmhmodule[#1]{#2}{#3}}%
118   \ifmod@show\par\noindent importing module #2 via \@doit\fi
119 }{%
120   \aftergroup\@doit\ifmod@show end import\fi%
121 }%

122 \srefaddidkey{mhview}
123 \addmetakey{mhview}{display}
124 \addmetakey{mhview}{creators}
125 \addmetakey{mhview}{contributors}
126 \addmetakey{mhview}{srccite}
127 \addmetakey*{mhview}{title}
128 \addmetakey{mhview}{fromrepos}
129 \addmetakey{mhview}{torepos}
130 \addmetakey{mhview}{frompath}
131 \addmetakey{mhview}{topath}
132 \addmetakey[sms]{mhview}{ext}

mhview the MathHub version
133 \newenvironment{mhview}[3][]{% keys, from, to
134   \metasetkeys{mhview}{#1}%
135   \sref@target%
136   \begin{@mhview}{#2}{#3}%
137   \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
138 }{%
139   \end{@mhview}%
140   \ignorespaces%
141 }%
142 \ifmod@show\surroundwithmdframed{mhview}\fi

@mhview The @mhview does the actual bookkeeping at the module level.
143 \newenvironment{@mhview}[2]{%from, to

```



```

144 \importmhmodule[repos=\mhview@fromrepos,path=\mhview@frompath,ext=\mhview@ext]{#1}%
145 \importmhmodule[repos=\mhview@torepos,path=\mhview@topath,ext=\mhview@ext]{#2}%
146 }{}%

```

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

```

147 \newenvironment{mhviewsketch}[3][]{%
148 \metasetkeys{mhview}{#1}%
149 \sref@target%
150 \begin{@mhview}{#2}{#3}%
151 \view@heading{#2}{#3}{\mhview@display}{\mhview@title}%
152 }{%
153 \end{@mhview}%
154 \ignorespaces%
155 }%
156 \ifmod@show\surroundwithmdframed{mhviewsketch}\fi
157 \</modules>

```

EdN:2

2

```

158 \<modules.ltxml>
159 DefKeyVal('mhview','id','Semiverbatim');
160 DefKeyVal('mhview','fromrepos','Semiverbatim');
161 DefKeyVal('mhview','torepos','Semiverbatim');
162 DefKeyVal('mhview','frompath','Semiverbatim');
163 DefKeyVal('mhview','topath','Semiverbatim');
164 DefKeyVal('mhview','title','Semiverbatim');
165 DefKeyVal('mhview','creators','Semiverbatim');
166 DefKeyVal('mhview','contributors','Semiverbatim');
167 DefKeyVal('mhview','display','Semiverbatim');
168 DefKeyVal('mhview','ext','Semiverbatim');
169 DefMacroI(T_CS('\begin{mhview}'),'OptionalKeyVals:mhview {}{}', sub {
170 my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
171 my $from = ToString(Digest($from_arg));
172 my $to = ToString(Digest($to_arg));
173 AssignValue(from_module => $from);
174 AssignValue(to_module => $to);
175 my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
176 my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
177 my $repos = LookupValue('current_repos');
178 my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
179 my $to_path = ToString(GetKeyVal($keyvals,'topath'));
180 my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
181 $ext = 'sms' unless $ext;
182 my $current_repos = LookupValue('current_repos');
183 if (!$from_repos) { $from_repos = $current_repos; }
184 if (!$to_repos) { $to_repos = $current_repos; }
185 return (
186   Tokenize("\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
187   Tokenize("\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,

```

²EDNOTE: MK: sort these into the rest.

```

188     Invocation(T_CS('\begin{viewenv}'),$keyvals,$from_arg,$to_arg)->unlist
189   );
190 });
191 DefMacroI('\end{mhview}',undef,'\end{viewenv}');
192
193 DefMacroI(T_CS('\begin{mhviewsketch}'),'OptionalKeyVals:mhview {}{}', sub {
194   my ($gullet, $keyvals, $from_arg, $to_arg) = @_;
195   my $from = ToString(Digest($from_arg));
196   my $to = ToString(Digest($to_arg));
197   my $from_repos = ToString(GetKeyVal($keyvals,'fromrepos'));
198   my $to_repos = ToString(GetKeyVal($keyvals,'torepos'));
199   my $repos = LookupValue('current_repos');
200   my $from_path = ToString(GetKeyVal($keyvals,'frompath'));
201   my $to_path = ToString(GetKeyVal($keyvals,'topath'));
202   my $ext = ToString(GetKeyVal($keyvals,'ext')) if $keyvals;
203   $ext = 'sms' unless $ext;
204   my $current_repos = LookupValue('current_repos');
205   if (!$from_repos) { $from_repos = $current_repos; }
206   if (!$to_repos) { $to_repos = $current_repos; }
207   return (
208     Tokenize("\importMHmoduleI[repos=$from_repos,path=$from_path,ext=$ext]{$from}")->unlist,
209     Tokenize("\importMHmoduleI[repos=$to_repos,path=$to_path,ext=$ext]{$to}")->unlist,
210     Invocation(T_CS('\begin{viewsketchenv}'),$keyvals,$from_arg,$to_arg)->unlist
211   );
212 });
213 DefMacroI('\end{mhviewsketch}',undef,'\end{viewsketchenv}');
214
215 DefConstructor('\importmhmodule OptionalKeyVals:importmhmodule {}',
216   "<omdoc:imports "
217   . "from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))(###2'"
218   . "?&defined(&GetKeyVal(#1,'conservative'))(load='&GetKeyVal(#1,'conservative'))'"
219   afterDigest => \&importMHmoduleI);
220
221 DefConstructor('\usemhmodule OptionalKeyVals:importmhmodule {}',
222   "<omdoc:uses from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))(###"
223   afterDigest => \&importMHmoduleI);
224
225 DefConstructor('\adoptmhmodule OptionalKeyVals:importmhmodule {}',
226   "<omdoc:adopts from='?&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))(#"
227   afterDigest => \&importMHmoduleI);
228
229 RawTeX(
230 \newcommand\mhinputref[2][\def\@repos{#1}%
231 \edef\mh@@repos{\mh@currentrepos}%
232 \ifx\@repos\empty\else\mhcurrentrepos{#1}\fi%
233 \inputref{\MathHub{\mh@currentrepos/source/#2}}%
234 \mhcurrentrepos\mh@@repos}
235 \newcommand\mhinput[2][\def\@repos{#1}%
236 \edef\mh@@repos{\mh@currentrepos}%
237 \ifx\@repos\empty\else\mhcurrentrepos{#1}\fi%

```

```

238 \input{\MathHub{\mh@currentrepos/source/#2}}%
239 \mhcurrentrepos\mh@@repos}
240 \newenvironment{importmhmodulevia}[3][\def\@repos{#1}%
241 \edef\mh@@repos{\mh@currentrepos}%
242 \ifx\@repos\empty\else\mhcurrentrepos{#1}\fi%
243 \gdef\@doit{\importmhmodule[#1]{#2}{#3}}
244 \begin{importmoduleenv}[load=\MathHub{\mh@currentrepos/source/#2}]{#3}}
245 {\end{importmoduleenv}\aftergroup\@doit}
246 ');
247 \modules.ltxml)

```

4.4 omtex-mh: MH Variants for OMTex

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

```

248 \omtext)
249 \addmetakey{Gin}{mhrepos}
250 \newcommand\mhgraphics[2][\metasetkeys{Gin}{#1}%
251 \edef\mh@@repos{\mh@currentrepos}%
252 \ifx\Gin@mhrepos\empty\mygraphics[#1]{\MathHub{\mh@currentrepos/source/#2}}%
253 \else\mygraphics[#1]{\MathHub{\Gin@mhrepos/source/#2}}\fi
254 \def\Gin@mhrepos{\mhcurrentrepos\mh@@repos}
255 \newcommand\mhgraphics[2][\begin{center}\mhgraphics[#1]{#2}\end{center}}
256 \newcommand\mhgraphics[2][\fbox{\mhgraphics[#1]{#2}}]
257 \newcommand\mhcbgraphics[2][\begin{center}\fbox{\mhgraphics[#1]{#2}\end{center}}
258 \omtext)
259 \omtext.ltxml)
260 sub mhgraphics {
261   my ($gullet,$keyval,$arg2) = @_ ;
262   my $repo_path;
263   if ($keyval) {
264     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
265   if (! $repo_path) {
266     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
267   else {
268     $keyval->setValue('mhrepos',undef); }
269   my $mathhub_base = ToString(Digest('\MathHub{ }'));
270   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
271   return Invocation(T_CS('@includegraphicx'), $keyval, T_OTHER($finalpath)); }#$
272 DefKeyVal('Gin','mhrepos','Semiverbatim');
273 DefMacro('\mhgraphics OptionalKeyVals:Gin {}', \&mhgraphics);
274 DefMacro('\mhcgraphics []{}', '\begin{center}\mhgraphics[#1]{#2}\end{center}');
275 DefMacro('\mhgraphics []{}', '\fbox{\mhgraphics[#1]{#2}}');
276 \omtext.ltxml)

```

4.5 mh-mikoslides: Support for MiKo Slides

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

```
277 <mikoslides>\addmetakey{Gin}{mhrepos}
278 <mikoslides.ltxml>DefKeyVal('Gin','mhrepos','Semiverbatim');
279 <mikoslides.ltxml>RawTeX('
280 *mikoslides.ltxml | mikoslides)
281 \newcommand\mhframeimage[2][{}]{%
282   \metasetkeys{Gin}{#1}%
283   \edef\mh@@repos{\mh@currentrepos}%
284   \ifx\Gin@mhrepos\@empty%
285     \frameimage[#1]{\MathHub{\mh@currentrepos/source/#2}}%
286   \else%
287     \frameimage[#1]{\MathHub{\Gin@mhrepos/source/#2}}%
288   \fi%
289 }%
290 </mikoslides.ltxml | mikoslides>
291 <mikoslides.ltxml>');
```

4.6 Finale

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

```
292 <*ltxml>
293 1;
294 </ltxml>
```

References

- [Hor+11] Fulya Horozal et al. “Combining Source, Content, Presentation, Narration, and Relational Representation”. In: *Intelligent Computer Mathematics*. Ed. by James Davenport et al. LNAI 6824. Springer Verlag, 2011, pp. 212–227. ISBN: 978-3-642-22672-4. URL: http://kwarc.info/frabe/Research/HIJKR_dimensions_11.pdf.
- [Koh15] Michael 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).