

hwexam.sty/cls: An Infrastructure for formatting Assignments and Exams*

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

October 31, 2015

Abstract

The `hwexam` package and class allows individual course assignment sheets and compound assignment documents using problem files marked up with the `problem` package.

Contents

1	Introduction	2
2	The User Interface	2
2.1	Package and Class Options	2
2.2	Assignments	2
2.3	Typesetting Exams	2
2.4	Including Assignments	3
2.5	Support for MathHub	4
3	Limitations	4
4	Implementation: The hwexam Class	6
4.1	Class Options	6
5	Implementation: The hwexam Package	7
5.1	Package Options	7
5.2	Assignments	8
5.3	Including Assignments	11
5.4	Typesetting Exams	11
5.5	Support for MathHub	13
5.6	Leftovers	15

*Version v1.0 (last revised 2013/12/12)

1 Introduction

The `hwexam` package and class supplies an infrastructure that allows to format nice-looking assignment sheets by simply including problems from problem files marked up with the `problem` package [Koh15c]. It is designed to be compatible with `problems.sty`, and inherits some of the functionality.

2 The User Interface

2.1 Package and Class Options

The `hwexam` package and class take the options `solutions`, `notes`, `hints`, `pts`, `min`, and `boxed` that are just passed on to the `problems` package (cf. its documentation for a description of the intended behavior).

`showmeta` If the `showmeta` option is set, then the metadata keys are shown (see [Koh15a] for details and customization options).

The `hwexam` class additionally accepts the options `report`, `book`, `chapter`, `part`, and `showignores`, of the `omdoc` package [Koh15b] on which it is based and passes them on to that. For the `extrefs` option see [Koh15d].

2.2 Assignments

`assignment` This package supplies the `assignment` environment that groups problems into assignment sheets. It takes an optional `KeyVal` argument with the keys `number` (for the assignment number; if none is given, 1 is assumed as the default or — in multi-assignment documents — the ordinal of the `assignment` environment), `title` (for the assignment title; this is referenced in the title of the assignment sheet), `type` (for the assignment type; e.g. “quiz”, or “homework”), `given` (for the date the assignment was given), and `due` (for the date the assignment is due).

2.3 Typesetting Exams

`multiple` Furthermore, the `hwexam` package takes the option `multiple` that allows to combine multiple assignment sheets into a compound document (the assignment sheets are treated as section, there is a table of contents, etc.).

`test` Finally, there is the option `test` that modifies the behavior to facilitate formatting tests. Only in `test` mode, the macros `\testspace`, `\testnewpage`, and `\testemptypage` have an effect: they generate space for the students to solve the given problems. Thus they can be left in the \LaTeX source.

`\testspace` `\testspace` takes an argument that expands to a dimension, and leaves vertical space accordingly. `\testnewpage` makes a new page in `test` mode, and `\testemptypage` generates an empty page with the cautionary message that this page was intentionally left empty.

`testheading` Finally, the `\testheading` takes an optional keyword argument where the keys `duration` specifies a string that specifies the duration of the test, `min` specifies the equivalent in number of minutes, and `reqpts` the points that are required for a

perfect grade.

```

\title{320101 General Computer Science (Fall 2010)}
\begin{testheading}[duration=one hour,min=60,reqpts=27]
  Good luck to all students!
\end{testheading}

```

formats to

Name:

Matriculation Number:

320101 General Computer Science (Fall 2010)

October 31, 2015

You have one hour(sharp) for the test;
Write the solutions to the sheet.
The estimated time for solving this exam is 58 minutes, leaving you 2 minutes for revising your exam.
You can reach 30 points if you solve all problems. You will only need 27 points for a perfect score, i.e. 3 points are bonus points.

Different problems test different skills and knowledge, so do not get stuck on one problem.

	To be used for grading, do not write here								
prob.	1.1	2.1	2.2	2.3	3.1	3.2	3.3	Sum	grade
total	4	4	6	6	4	4	2	30	
reached									

good luck

Example 1: A generated test heading.

2.4 Including Assignments

```

\includeassignment
number
title
type
given
due

```

The `\includeassignment` macro can be used to include an assignment from another file. It takes an optional `KeyVal` argument and a second argument which is a path to the file containing the problem (the macro assumes that there is only one `assignment` environment in the included file). The keys `number`, `title`, `type`, `given`, and `due` are just as for the `assignment` environment and (if given) overwrite the ones specified in the `assignment` environment in the included file.

2.5 Support for MathHub

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 `modules` macros.

`\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}
```

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

```
\includemhassignment{baz/foobar}
```

Of course, neither \LaTeX nor $\text{\LaTeX}ML$ know about the repositories when they are called from a file system, so we can use the `\mhcurrentrepos` macro from the `modules` package 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.

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.

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 \LaTeX GitHub repository [sTeX].

1. none reported yet.

4 Implementation: The hwexam Class

The functionality is spread over the `hwexam` class and package. The class provides the `document` environment and pre-loads some convenience packages, whereas the package provides the concrete functionality.

`hwexam.dtx` generates four files: `hwexam.cls` (all the code between `<*cls>` and `</cls>`), `hwexam.sty` (between `<*package>` and `</package>`) and their L^AT_EXML bindings (between `<*ltxml.cls>` and `</ltxml.cls>` and `<*ltxml.sty>` and `</ltxml.sty>` respectively). We keep the corresponding code fragments together, since the documentation applies to both of them and to prevent them from getting out of sync.

first the general setup for L^AT_EXML(for the class and package)

```
1 <*ltxml.cls | ltxml.sty>
2 # -*- CPERL -*-
3 package LaTeXML::Package::Pool;
4 use strict;
5 use LaTeXML::Package;
6 use LaTeXML::Util::Pathname;
7 use Cwd qw(cwd abs_path);
8 </ltxml.cls | ltxml.sty>
```

4.1 Class Options

To initialize the `hwexam` class, we declare and process the necessary options by passing them to the respective packages and classes they come from.

```
9 <*cls>
10 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{hwexam}}
11 \PassOptionsToClass{\CurrentOption}{omdoc}}
12 \ProcessOptions
13 </cls>
14 <*ltxml.cls>
15 DeclareOption(undef,sub {PassOptions('hwexam','sty',ToString(Digest(T_CS('\CurrentOption'))));});
16 DeclareOption(undef,sub {PassOptions('omdoc','cls',ToString(Digest(T_CS('\CurrentOption'))));});
17 ProcessOptions();
18 </ltxml.cls>
```

We load `omdoc.cls`, and the desired packages. For the L^AT_EXML bindings, we make sure the right packages are loaded.

```
19 <*cls>
20 \LoadClass{omdoc}
21 \RequirePackage{stex}
22 \RequirePackage{hwexam}
23 \RequirePackage{tikzinput}
24 \RequirePackage{graphicx}
25 \RequirePackage{a4wide}
26 \RequirePackage{amssymb}
27 \RequirePackage{amstext}
28 \RequirePackage{amsmath}
29 </cls>
```

```

30 <*ltxml.cls>
31 LoadClass('omdoc');
32 RequirePackage('stex');
33 RequirePackage('hwexam');
34 RequirePackage('tikzinput', options => ['image']);
35 RequirePackage('graphicx');
36 RequirePackage('amssymb');
37 RequirePackage('amstext');
38 RequirePackage('amsmath');
39 </ltxml.cls>

```

EdN:1 Finally, we register another keyword for the document environment¹

```

40 <*cls>
41 \newcommand\assig@default@type{Assignment}
42 \addmetakey[\assig@default@type]{document}{hwexamtype}
43 </cls>

```

5 Implementation: The hwexam Package

5.1 Package Options

The first step is to declare (a few) package options that handle whether certain information is printed or not. Some come with their own conditionals that are set by the options, the rest is just passed on to the `problems` package.

```

44 <*package>
45 \newif\iftest\testfalse
46 \DeclareOption{test}{\testtrue}
47 \newif\ifmultiple\multiplefalse
48 \DeclareOption{multiple}{\multipletrue}
49 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{problem}}
50 \ProcessOptions
51 </package>

```

Then we make sure that the necessary packages are loaded (in the right versions).

```

52 <*package>
53 \RequirePackage{keyval}[1997/11/10]
54 \RequirePackage{problem}
55 </package>

```

Here comes the equivalent header information for L^AT_EXML, we also initialize the package inclusions. Since L^AT_EXML does not handle options yet, we have nothing to do.

```

56 <*ltxml>
57 DeclareOption('test', '');
58 DeclareOption('multiple', '');
59 DeclareOption(undef, sub {PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))))}
60 ProcessOptions();

```

¹EdNOTE: MK: this still needs to be internationalized.

```

61 RequirePackage('problem');
62 </ltxml>

    Then we register the namespace of the requirements ontology
63 <*ltxml>
64 RegisterNamespace('assig'=>"http://omdoc.org/ontology/assignments#");
65 RegisterDocumentNamespace('assig'=>"http://omdoc.org/ontology/assignments#");
66 </ltxml>

```

5.2 Assignments

We will prepare the keyval support for the `assignment` environment.

```

67 <*package>
68 \srefaddidkey{assig}
69 \addmetakey{assig}{number}
70 \addmetakey*{assig}{title}
71 \addmetakey{assig}{type}
72 \addmetakey{assig}{given}
73 \addmetakey{assig}{due}
74 \addmetakey[false]{assig}{loadmodules}[true]

```

The next three macros are intermediate functions that handle the case gracefully, where the respective token registers are undefined.

The `\given@due` macro prints information about the given and due status of the assignment. Its arguments specify the brackets.

```

75 \newcommand\given@due[2]{%
76 \ifx \inclassig@given\@empty
77 \ifx \assig@given\@empty
78 \ifx \inclassig@due\@empty
79 \ifx \assig@due\@empty% all empty do nothing
80 \else #1%
81 \fi
82 \else #1%
83 \fi
84 \else #1%
85 \fi
86 \else #1%
87 \fi
88 \ifx\inclassig@given\@empty
89 \ifx\assig@given\@empty% do nothing
90 \else Given \assig@given%
91 \fi
92 \else Given \inclassig@given%
93 \fi
94 \ifx \inclassig@due\@empty
95 \ifx \assig@due\@empty% do nothing
96 \else
97 \ifx \inclassig@given\@empty
98 \ifx \assig@given\@empty% do nothing

```



```

99 \else ,~%
100 \fi
101 \else ,~%
102 \fi
103 \fi
104 \else
105 \ifx \inclassig@given\@empty
106 \ifx \assig@given\@empty% do nothing
107 \else ,~%
108 \fi
109 \else ,~%
110 \fi
111 \fi
112 \ifx \inclassig@due\@empty
113 \ifx \assig@due\@empty% do nothing
114 \else Due \assig@due%
115 \fi
116 \else Due \inclassig@due%
117 \fi
118 \ifx \inclassig@given\@empty
119 \ifx \assig@given\@empty
120 \ifx \inclassig@due\@empty
121 \ifx \assig@due\@empty% all empty do nothing
122 \else #2%
123 \fi
124 \else #2%
125 \fi
126 \else #2%
127 \fi
128 \else #2%
129 \fi
130 }

```

`\assignment@title` This macro prints the title of an assignment, the local title is overwritten, if there is one from the `\includeassignment`. `\assignment@title` takes three arguments the first is the fallback when no title is given at all, the second and third go around the title, if one is given.

```

131 \newcommand\assignment@title[3]
132 {\ifx\inclassig@title\@empty% if there is no outside title
133 \ifx\assig@title\@empty{#1}\else{#2\assig@title{#3}}\fi
134 \else{#2}\inclassig@title{#3}\fi}% else show the outside title

```

`\assignment@number` Like `\assignment@title` only for the number, and no around part.

```

135 \newcommand\assignment@number%
136 {\ifx\inclassig@number\@empty% if there is no outside number
137 \ifx\assig@number\@empty\else\assig@number\fi
138 \else\inclassig@number\fi}% else show the outside number

```

With them, we can define the central `assignment` environment. This has two forms (separated by `\ifmultiple`) in one we make a title block for an assignment

sheet, and in the other we make a section heading and add it to the table of contents. We first define an assignment counter

```
assignment For the assignment environment we delegate the work to the @assignment envi-
            ronment that depends on whether multiple option is given.
139 \newenvironment{assignment}[1] [] {\metasetkeys{assig}{#1}\sref{target%
140 \edef\@@num{\assignment@number}%
141 \ifx\@@num\empty\stepcounter{assignment}\else\setcounter{assignment}{\@@num}\fi%
142 \setcounter{problem}{0}%
143 \def\current@section@level{\document@hwexamtype}%
144 \sref{label@id{\document@hwexamtype \thesection}%
145 \begin{@assignment}}
146 {\end{@assignment}}
```

In the multi-assignment case we just use the omdoc environment for suitable sectioning.

```
147 \ifmultiple
148 \newenvironment{@assignment}%
149 {\ifx\assig@loadmodules\@true
150 \begin{omgroup}[loadmodules]{\protect\document@hwexamtype~\arabic{assignment}%
151 \assignment@title{\;\;(\;)\;\;}\given@due{\;}}
152 \else
153 \begin{omgroup}{\protect\document@hwexamtype~\arabic{assignment}%
154 \assignment@title{\;\;(\;)\;\;}\given@due{\;}}
155 \fi%
156 {\protect\document@hwexamtype~\arabic{assignment}%
157 \assignment@title{\;\;(\;)\;\;}\given@due{\;}}
158 {\end{omgroup}}
```

for the single-page case we make a title block from the same components.

```
159 \else
160 \newenvironment{@assignment}
161 {\begin{center}\bf
162 \Large\@title\strut\
163 \document@hwexamtype~\arabic{assignment}\assignment@title{\;\;(\;)\;\;}\
164 \large\given@due{--\;\;}\;\;--}
165 \end{center}}
166 {}
167 \fi% multiple
168 \end{package}
```

```
169 \*ltxml>
170 DefEnvironment('assignment' OptionalKeyVals:assig',
171 " <omdoc:omgroup ?&GetKeyVal(#1,'id')(xml:id='&GetKeyVal(#1,'id')')() "
172 . " assig:dummy='for the namespace'"
173 . " <omdoc:metadata>"
174 . " <dc:title>"
175 . " Assignment ?&GetKeyVal(#1,'num')(&GetKeyVal(#1,'num')).() )"
176 . " ?&GetKeyVal(#1,'title')(&GetKeyVal(#1,'title')))"
177 . " </dc:title>"
178 . " ?&GetKeyVal(#1,'given')(<omdoc:meta property='assig:given'>&GetKeyVal(#1,'given')</omdoc:meta>)"
```

```

179 .      "?&GetKeyVal(#1,'due')(<omdoc:meta property='assig:due'>&GetKeyVal(#1,'due')</omdoc:meta
180 .      "?&GetKeyVal(#1,'pts')(<omdoc:meta property='assig:pts'>&GetKeyVal(#1,'pts')</omdoc:meta
181 .      "</omdoc:metadata>"
182 .      "#body"
183 . "</omdoc:omgroup>\n"#,
184 # afterDigest=> sub {
185 #   my ($stomach, $kv) = @_;
186 #   my $kvi = LookupValue('inclassig');
187 #   my @keys = qw(id num title pts given due);
188 #   my @vals = $kvi && map($kvi->getValue($_), @keys);
189 #   foreach my $i(0..$#vals) {
190 #     $kv->setValue($keys[$i],$vals[$i]) if $vals[$i];
191 #   }
192 #};#$
193 </ltxml>

```

5.3 Including Assignments

`\in*assignment` This macro is essentially a glorified `\include` statement, it just sets some internal macros first that overwrite the local points. Importantly, it resets the `inclassig` keys after the input.

```

194 <*package>
195 \addmetakey{inclassig}{number}
196 \addmetakey*{inclassig}{title}
197 \addmetakey{inclassig}{type}
198 \addmetakey{inclassig}{given}
199 \addmetakey{inclassig}{due}
200 \addmetakey{inclassig}{mhrepos}
201 \clear@inclassig@keys%initially
202 \newcommand\includeassignment[2][{}]{\metasetkeys{inclassig}{#1}%
203 \include{#2}\clear@inclassig@keys}
204 \newcommand\inputassignment[2][{}]{\metasetkeys{inclassig}{#1}%
205 \input{#2}\clear@inclassig@keys}
206 </package>
207 <*ltxml>
208 DefMacro('includeassignment [ ] {}', sub {
209   my ($stomach, $arg1, $arg2) = @_;
210   AssignValue('inclassig',$arg1) if $arg1;
211   (Invocation(T_CS('input'),$arg2)->unlist);
212 });
213 DefMacro('inputassignment [ ] {}','includeassignment[#1]{#2}');
214 </ltxml>

```

5.4 Typesetting Exams

`\quizheading`

```

215 <*package>
216 \addmetakey{quizheading}{tas}
217 \newcommand\quizheading[1]{\def\@tas{#1}%

```

```

218 \large\noindent NAME: \hspace{8cm} MAILBOX:\\[2ex]%
219 \ifx\@tas\@empty\else%
220 \noindent TA: \@for\@I:=\@tas\do{\Large$\Box$}\@I\hspace*{1em}}\\[2ex]\fi}

\testheading
221 \addmetakey{testheading}{min}
222 \addmetakey{testheading}{duration}
223 \addmetakey{testheading}{reqpts}
224 \newenvironment{testheading}[1][\metasetkeys{testheading}{#1}
225 {\noindent\large{Name: \hfill Matriculation Number:\hspace*{2cm}\strut\\[1ex]
226 \begin{center}\Large\textbf{\@title}\\[1ex]\large\@date\\[3ex]\end{center}
227 {\textbf{You have
228 \ifx\test@heading@duration\@empty\testheading@min minutes\else\testheading@duration\fi
229 (sharp) for the test}};\ Write the solutions to the sheet.}\par\noindent
230
231 \newcount\check@time\check@time=\testheading@min
232 \advance\check@time by -\theassignment@totalmin
233 The estimated time for solving this exam is {\theassignment@totalmin} minutes,
234 leaving you {\the\check@time} minutes for revising your exam.
235
236 \newcount\bonus@pts\bonus@pts=\theassignment@totalpts
237 \advance\bonus@pts by -\testheading@reqpts
238 You can reach {\theassignment@totalpts} points if you solve all problems. You will only need
239 {\testheading@reqpts} points for a perfect score, i.e.\ {\the\bonus@pts} points are
240 bonus points. \vfill
241 \begin{center}
242 {\Large\em
243 % You have ample time, so take it slow and avoid rushing to mistakes!\\[2ex]
244 Different problems test different skills and knowledge, so do not get stuck on
245 one problem.}\vfill\par\correction@table \\[3ex]
246 \end{center}}
247 {\newpage}
248 \end{package}
249 \end{*xml}
250 DefEnvironment('{testheading}OptionalKeyVals:omdoc','');
251 \end{*xml}

\testspace
252 \begin{package}
253 \newcommand\testspace[1]{\iftest\vspace*{#1}\fi}
254 \end{package}
255 \end{*xml}
256 DefConstructor('\testspace{}','');
257 \end{*xml}

\testnewpage
258 \begin{package}
259 \newcommand\testnewpage{\iftest\newpage\fi}
260 \end{package}

```

```

261 <*txml>
262 DefConstructor('\testnewpage','');
263 </txml>

\testemptypage
264 <*package>
265 \newcommand\testemptypage[1][\iftest\begin{center}This page was intentionally left
266     blank for extra space\end{center}\vfill\eject\else\fi]
267 </package>
268 <*txml>
269 DefConstructor('\testemptypage','');
270 </txml>

\@problem This macro acts on a problem's record in the *.aux file. Here we redefine it to
generate the correction table.
271 <*package>
272 \renewcommand\@problem[3]{\stepcounter{assignment@probs}
273 \def\@@pts{#2}\ifx\@@pts\@empty\else\addtocounter{assignment@totalpts}{#2}\fi
274 \def\@@min{#3}\ifx\@@min\@empty\else\addtocounter{assignment@totalmin}{#3}\fi
275 \xdef\correction@probs{\correction@probs & #1}%
276 \xdef\correction@pts{\correction@pts & #2}
277 \xdef\correction@reached{\correction@reached &}}
278 </package>

\correction@table This macro generates the correction table
279 <*package>
280 \newcounter{assignment@probs}
281 \newcounter{assignment@totalpts}
282 \newcounter{assignment@totalmin}
283 \newcommand\correction@probs{prob.}%
284 \newcommand\correction@pts{total}%
285 \newcommand\correction@reached{reached}%
286 \stepcounter{assignment@probs}
287 \newcommand\correction@table{\begin{tabular}{|l|*{\theassignment@probs}{c|}|l|}\hline%
288 &\multicolumn{\theassignment@probs}{c|}|}%
289 {\footnotesize To be used for grading, do not write here} &\\\hline
290 \correction@probs & Sum & grade\\\hline
291 \correction@pts & \theassignment@totalpts & \\\hline
292 \correction@reached & & \\\hline
293 \end{tabular}}
294 </package>

```

5.5 Support for MathHub

`\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`.

```

295 <*package>

```

```

296 \newcommand\includemhassignment[2][\metasetkeys{inclassig}{#1}%
297 \edef\mh@@repos{\mh@currentrepos}%
298 \ifx\inclassig@mhrepos\@empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
299 \includeassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}%
300 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
301 \</package>
302 \<!--xml-->
303 sub includemhassignment {
304   my ($gullet,$keyval,$arg2) = @_;
305   my $repo_path;
306   if ($keyval) {
307     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
308   if (! $repo_path) {
309     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
310   else {
311     $keyval->setValue('mhrepos',undef); }
312   my $mathhub_base = ToString(Digest('\MathHub{ }'));
313   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
314   return Invocation(T_CS('\includeassignment'), $keyval, T_OTHER($finalpath)); }#$
315 DefKeyVal('inclprob','mhrepos','Semiverbatim');
316 DefMacro('\includemhassignment OptionalKeyVals:inclprob { }', \&includemhassignment);
317 \</xml-->

```

\inputmhassignment analogous

```

318 \<!--xml-->
319 \newcommand\inputmhassignment[2][\metasetkeys{inclassig}{#1}%
320 \edef\mh@@repos{\mh@currentrepos}%
321 \ifx\inclassig@mhrepos\@empty\else\mhcurrentrepos\inclassig@mhrepos\fi%
322 \inputassignment[#1]{\MathHub{\mh@currentrepos/source/#2}}%
323 \mhcurrentrepos\mh@@repos\clear@inclassig@keys}
324 \</package>
325 \<!--xml-->
326 sub inputmhassignment {
327   my ($gullet,$keyval,$arg2) = @_;
328   my $repo_path;
329   if ($keyval) {
330     $repo_path = ToString(GetKeyVal($keyval,'mhrepos')); }
331   if (! $repo_path) {
332     $repo_path = ToString(Digest(T_CS('\mh@currentrepos'))); }
333   else {
334     $keyval->setValue('mhrepos',undef); }
335   my $mathhub_base = ToString(Digest('\MathHub{ }'));
336   my $finalpath = $mathhub_base.$repo_path.'/source/'.ToString($arg2);
337   return Invocation(T_CS('\inputassignment'), $keyval, T_OTHER($finalpath)); }#$
338 DefMacro('\inputmhassignment OptionalKeyVals:inclprob { }', \&inputmhassignment);
339 \</xml-->

```

5.6 Leftovers

at some point, we may want to reactivate the logos font, then we use

```
here we define the logos that characterize the assignment
\font\bierfont=../assignments/bierglas
\font\denkerfont=../assignments/denker
\font\uhrfont=../assignments/uhr
\font\warnschildfont=../assignments/achtung

\newcommand\bierglas{{\bierfont\char65}}
\newcommand\denker{{\denkerfont\char65}}
\newcommand\uhr{{\uhrfont\char65}}
\newcommand\warnschild{{\warnschildfont\char 65}}
\newcommand\hardA{\warnschild}
\newcommand\longA{\uhr}
\newcommand\thinkA{\denker}
\newcommand\discussA{\bierglas}
```

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

```
340 <ltxml>1;
```

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

L^AT_EX^{ML}, 4, 6, 7

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.
- [Koh15a] Michael Kohlhase. *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>.
- [Koh15b] Michael Kohlhase. *omdoc.sty/cls: Semantic Markup for Open Mathematical Documents in L^AT_EX*. Tech. rep. Comprehensive T_EX Archive Network (CTAN), 2015. URL: <http://www.ctan.org/tex-archive/macros/latex/contrib/stex/omdoc/omdoc.pdf>.
- [Koh15c] Michael Kohlhase. *problem.sty: An Infrastructure for formatting Problems*. Tech. rep. Comprehensive T_EX Archive Network (CTAN), 2015. URL: <http://www.ctan.org/tex-archive/macros/latex/contrib/stex/problem/problem.pdf>.
- [Koh15d] Michael Kohlhase. *sref.sty: Semantic Crossreferencing in L^AT_EX*. Tech. rep. Comprehensive T_EX Archive Network (CTAN), 2015. URL: <http://www.ctan.org/tex-archive/macros/latex/contrib/stex/sref/sref.pdf>.
- [sTeX] *KWARC/sTeX*. URL: <https://svn.kwarc.info/repos/stex> (visited on 05/15/2015).