Slides and Course Notes for Jacobs University*

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

November 21, 2015

Abstract

We present a document class from which we can generate both course slides and course notes in a transparent way. Furthermore, we present a set of L^AT_EXML bindings for these, so that we can also generate OMDoc-based course materials, e.g. for inclusion in the ActiveMath system.

Contents

---

*Version ? (last revised ?)

1 Introduction

This Document class is derived from `beamer.cls` [Tana], specializes it with Jacobs stuff and adds a notes version that is more suited to printing than the one supplied by `beamer.cls`.

2 The User Interface

The `mikoslides` class takes the notion of a slide frame from Till Tantau's excellent `beamer` class and adapts its notion of frames for use in the $\mathrm{S}\kern-0.1em\mathrm{T}\kern-0.1em\mathrm{E}\kern-0.1em\mathrm{X}$and OMDoc. To support semantic course notes, it extends the notion of mixing frames and explanatory text, but rather than treating the frames as images (or integrating their contents into the flowing text), the `mikoslides` package displays the slides as such in the course notes to give students a visual anchor into the slide presentation in the course (and to distinguish the different writing styles in slides and course notes).

In practice we want to generate two documents from the same source: the slides for presentation in the lecture and the course notes as a narrative document for home study. To achieve this, the `mikoslides` class has two modes: *slides mode* and *notes mode* which are determined by the package option.

2.1 Package Options

EdN:1

The `mikoslides` class takes a variety of class options:[1]

slides
a
- The options `slides`nd notes`notes` switch between slides mode and notes mode (see Section 2.2).

sectocframes
- If the option `sectocframes` is given, then special frames with section table of contents are produced headers [2]

EdN:2

showmeta
- `showmeta`. If this is set, then the metadata keys are shown (see [Koh15] for details and customization options).

frameimages
- If the option `frameimages` is set, then slide mode also shows the `\frameimage`-generated frames.

2.2 Notes and Slides

frame
note
Slides are represented with the `frame` just like in the `beamer` class, see [Tanb] for details. The `mikoslides` class adds the `note` environment for encapsulating the course note fragments.[1]

⚠ Note that it is essential to start and end the `notes` environment at the start of the line – in particular, there may not be leading blanks – else LaTeX becomes confused and throws error messages that are difficult to decipher.

---

[1]EDNOTE: leaving out noproblems for the moment until we decide what to do with it.
[2]EDNOTE: document the functionality
[1]MK: it would be very nice, if we did not need this environment, and this should be possible in principle, but not without intensive LaTeX trickery. Hints to the author are welcome.

```
\begin{note}
  We start this course with ...
\end{note}

\begin{frame}
  \frametitle{The first slide}
  ...
\end{frame}
\begin{note}
  ... and more explanatory text
\end{note}

\begin{frame}
  \frametitle{The second slide}
  ...
\end{frame}
...
```

Example 1: A typical Course Notes File

By interleaving the `frame` and `note` environments, we can build course notes as shown in Figure 1.

Sometimes, we want to integrate slides as images after all – e.g. because we already have a PowerPoint presentation, to which we want to add sTEXnotes. In `\frameimage` this case we can use `\frameimage[`⟨*opt*⟩`]{`⟨*path*⟩`}`, where ⟨*opt*⟩ are the options of `\includegraphics` from the `graphicx` package [CR99] and ⟨*path*⟩ is the file path (extension can be left off like in `\includegraphics`).

2.3 Header and Footer Lines

2.4 Colors and Highlighting

`\textwarning` The `\textwarning` macro generates a warning sign: ⚠

2.5 Front Matter, Titles, etc

2.6 Miscellaneous

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 sTEXGitHub repository [sTeX].

1. when option `book` which uses `\pagestyle{headings}` is given and semantic macros are given in the `omgroup` titles, then they sometimes are not defined

by the time the heading is formatted. Need to look into how the headings are made. This is a problem of the underlying `omdoc` package.

4   The Implementation

The `mikoslides` 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.

The general preamble for LATEXML:

```
1 ⟨∗ltxml.cls | ltxml.sty⟩
2 # -*- CPERL -*-
3 package LaTeXML::Package::Pool;
4 use strict;
5 use LaTeXML::Package;
6 ⟨/ltxml.cls | ltxml.sty⟩
```

4.1   Class and Package Options

We define some Package Options and switches for the `mikoslides` class and activate them by passing them on to `beamer.cls` and `omdoc.cls` and the `mikoslides` package.

```
 7 ⟨∗cls⟩
 8 \newif\ifnotes\notesfalse
 9 \DeclareOption{notes}{\notestrue\PassOptionsToPackage{\CurrentOption}{mikoslides}}
10 \DeclareOption{slides}{\notesfalse\PassOptionsToPackage{\CurrentOption}{mikoslides}}
11 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{omdoc}
12                                    \PassOptionsToClass{\CurrentOption}{beamer}
13                                    \PassOptionsToPackage{\CurrentOption}{mikoslides}}
14 \ProcessOptions
15 ⟨/cls⟩
16 ⟨∗ltxml.cls⟩
17 DeclareOption(undef, sub {PassOptions('omdoc','cls',ToString(Digest(T_CS('\CurrentOption'))))); 
18                                      PassOptions('mikoslides','sty',ToString(Digest(T_CS('
19 ProcessOptions();
20 ⟨/ltxml.cls⟩
```

now we do the same for the `mikoslides` package. Note that we also have to define the same switches[3], since we might use `mikoslides.sty` in a different class.

EdN:3

```
21 ⟨∗package⟩
22 \newif\if@mikoslides@mh@\@mikoslides@mh@false
23 \DeclareOption{mh}{\@mikoslides@mh@true
24   \PassOptionsToPackage{\CurrentOption}{stex}
25   \PassOptionsToPackage{\CurrentOption}{smglom}
26   \PassOptionsToPackage{\CurrentOption}{tikzinput}}
```

---

[3]EDNOTE: MK: we may think about making all of them internal

```
27 \newif\ifnotes\notesfalse
28 \DeclareOption{notes}{\notestrue}
29 \DeclareOption{slides}{\notesfalse}
30 \newif\ifsectocframes\sectocframesfalse
31 \DeclareOption{sectocframes}{\sectocframestrue}
32 \newif\ifframeimages\frameimagesfalse
33 \DeclareOption{frameimages}{\frameimagestrue}
34 \newif\if@part\@partfalse
35 \DeclareOption{report}{\@parttrue\PassOptionsToPackage{\CurrentOption}{omdoc}}
36 \DeclareOption{book}{\@parttrue\PassOptionsToPackage{\CurrentOption}{omdoc}}
37 \newif\ifproblems\problemstrue
38 \DeclareOption{noproblems}{\problemsfalse}
39 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{stex}
40                        \PassOptionsToPackage{\CurrentOption}{smglom}
41                        \PassOptionsToPackage{\CurrentOption}{tikzinput}}
42 \ProcessOptions
43 ⟨/package⟩
44 ⟨∗ltxml.sty⟩
45 DeclareOption('notes', '');
46 DeclareOption('slides', '');
47 DeclareOption('noproblems', '');
48 DeclareOption('sectocframes', '');
49 DeclareOption('frameimages', '');
50 DeclareOption('mh', sub { AssignValue ('@mikoslides' => 1,'global');
51                        PassOptions('stex','sty',ToString(Digest(T_CS('\CurrentOption'))));
52                        PassOptions('tikzinput','sty',ToString(Digest(T_CS('\CurrentOption'))));
53 DeclareOption(undef, sub {PassOptions('stex','sty',ToString(Digest(T_CS('\CurrentOption'))));
54                                   PassOptions('tikzinput','sty',ToString(Digest(T_CS('\
55 ProcessOptions();
56 RawTeX('\newif\ifnotes\notesfalse');
57 RawTeX('\newif\ifproblems\problemsfalse');
58 ⟨/ltxml.sty⟩
```

Depending on the options, we either load the article-based omdoc or the beamer class. In the first case, we also have to make the beamer-specific things available to article via the beamerarticle package. We use options to avoid loading theorem-like environments, since we want to use our own from the sTeX packages. On the LaTeXML side we just load the omdoc class and provide the \usetheme macro that would otherwise from the the beamer class.

```
59 ⟨∗cls⟩
60 \ifnotes
61   \LoadClass{omdoc}
62   \RequirePackage{a4wide}
63   \RequirePackage{marginnote}
64   \RequirePackage{mdframed}
65   \RequirePackage[notheorems,noamsthm,noxcolor]{beamerarticle}
66 \else
67   \LoadClass[notheorems,noamsthm,10pt]{beamer}
68   \newcounter{Item}
69   \newcounter{paragraph}
```

```
70    \newcounter{subparagraph}
71    \newcounter{Hfootnote}
72    \usetheme{Jacobs}
73 \fi
74 \RequirePackage{mikoslides}
75 ⟨/cls⟩
76 ⟨∗ltxml.cls⟩
77 LoadClass('omdoc');
78 RequirePackage('mikoslides');
79 DefConstructor('\usetheme{}','');
80 ⟨/ltxml.cls⟩
```

now, we load the remaining packages for both versions.

```
81 ⟨∗package⟩
82 \if@mikoslides@mh@\RequirePackage{mikoslides-mh}\fi
83 \RequirePackage{stex}
84 \RequirePackage{smglom}
85 \RequirePackage{tikzinput}
86 \RequirePackage{latexml}
87 \RequirePackage{amssymb}
88 \RequirePackage{amsmath}
89 \RequirePackage{comment}
90 \RequirePackage{textcomp}
91 \RequirePackage{url}
92 ⟨/package⟩
93 ⟨∗ltxml.sty⟩
94 if(LookupValue('@mikoslides')) {RequirePackage('mikoslides-mh');}
95 RequirePackage('stex');
96 RequirePackage('smglom');
97 RequirePackage('tikzinput', options => ['image']);
98 RequirePackage('latexml');
99 RequirePackage('amssymb');
100 RequirePackage('amsmath');
101 RequirePackage('graphicx');
102 RequirePackage('url');
103 ⟨/ltxml.sty⟩
```

4.2   Notes and Slides

We define the sizes of slides in the notes. Somehow, we cannot get by with the same here.

```
104 ⟨∗package⟩
105 \newcounter{slide}
106 \newlength{\slidewidth}\setlength{\slidewidth}{12.8cm}
107 \newlength{\slideheight}\setlength{\slideheight}{9cm}
108 ⟨/package⟩
109 ⟨∗ltxml.sty⟩
110 DefRegister('\slidewidth'     => Dimension('13.6cm'));
111 DefRegister('\slideheight'     => Dimension('9cm'));
```

112 ⟨/ltxml.sty⟩

note   The note environment is used to leave out text in the slides mode. It does not
        have a counterpart in OMDoc. So for course notes, we define the note environment
        to be a no-operation otherwise we declare the note environment as a comment via
        the comment package.

113 ⟨∗package⟩
114 \ifnotes%
115   \renewenvironment{note}{\ignorespaces}{}%
116 \else%
117   \excludecomment{note}%
118 \fi%
119 ⟨/package⟩
120 ⟨∗ltxml.sty⟩
121 DefEnvironment('{note}','#body');
122 ⟨/ltxml.sty⟩

        We start by giving the LaTeXML binding for the frame environment from the
        beamer class. We first set up the slide boxes in article mode. We set up sizes
        and provide a box register for the frames and a counter for the slides.

123 ⟨∗package⟩
124 \ifnotes
125   \newlength{\slideframewidth}
126   \setlength{\slideframewidth}{1.5pt}

frame   We first define the keys.
127   \addmetakey{frame}{label}
128   \addmetakey[yes]{frame}{allowframebreaks}
129   \addmetakey{frame}{allowdisplaybreaks}
130   \addmetakey[yes]{frame}{fragile}
131   \addmetakey[yes]{frame}{shrink}
132   \addmetakey[yes]{frame}{squeeze}

        We redefine the itemize environment so that it looks more like the one in beamer
        with Jacobs theme. We create the box with the mdframed environment from the
        equinymous package. Then we define the environment, read them, and construct
        the slide number and label.

133   \renewenvironment{frame}[1][]{%
134   \metasetkeys{frame}{#1}%
135   \stepcounter{slide}%
136   \def\@currentlabel{\theslide}%
137   \ifx\frame@label\@empty%
138   \else%
139   \label{\frame@label}%
140   \fi%

        We redefine the itemize environment so that it looks more like the one in beamer
        with Jacobs theme.

141   \def\itemize@level{outer}%

7

```
142        \def\itemize@outer{outer}%
143        \def\itemize@inner{inner}%
144        \renewcommand\newpage{}%
145        \renewcommand\metakeys@show@keys[2]{\marginnote{{\scriptsize ##2}}}%
146        \renewenvironment{itemize}{%
147          \ifx\itemize@level\itemize@outer%
148            \def\itemize@label{$\rhd$}%
149          \fi%
150          \ifx\itemize@level\itemize@inner%
151            \def\itemize@label{$\scriptstyle\rhd$}%
152          \fi%
153          \begin{list}%
154          {\itemize@label}%
155          {\setlength{\labelsep}{.3em}%
156           \setlength{\labelwidth}{.5em}%
157           \setlength{\leftmargin}{1.5em}%
158          }%
159          \edef\itemize@level{\itemize@inner}%
160        }{%
161          \end{list}%
162        }%
```

We create the box with the `mdframed` environment from the equinymous package.

```
163        \begin{mdframed}[linewidth=\slideframewidth,skipabove=1ex,skipbelow=1ex,userdefinedwidth=\s
164      }{%
165        \medskip\miko@slidelabel\end{mdframed}%
166      }%
167  ⟨/package⟩
168  ⟨∗ltxml.sty⟩
169  DefEnvironment('{frame}[]',
170      "<omdoc:omgroup layout='slide'>"
171      .  "#body\n"
172      ."</omdoc:omgroup>\n\n",
173   afterDigestBegin=>sub {
174      $_[1]->setProperty(theory=>LookupValue('current_module')); });
175  ⟨/ltxml.sty⟩#$
```

Now, we need to redefine the frametitle (we are still in course notes mode).

\frametitle

```
176  ⟨∗package⟩
177    \renewcommand{\frametitle}[1]{{\Large\bf\sf\color{blue}{#1}}\medskip}%
178  \fi %ifnotes
179  ⟨/package⟩
180  ⟨∗ltxml.sty⟩
181  DefConstructor('\frametitle{}',
182    "\n<omdoc:metadata><dc:title>#1</dc:title></omdoc:metadata>");
183  ⟨/ltxml.sty⟩
```

\frameimage We have to make sure that the width is overwritten, for that we check the

\Gin@ewidth macro from the graphicx package[4]

```
184 ⟨∗package⟩
185 \newrobustcmd\frameimage[2][]{%
186   \stepcounter{slide}%
187   \ifframeimages%
188     \def\Gin@ewidth{}\setkeys{Gin}{#1}%
189     \ifnotes%
190     \else%
191       \vfill%
192     \fi%
193     \ifx\Gin@ewidth\@empty%
194       \mycgraphics[width=\slidewidth,#1]{#2}\else\mycgraphics[#1]{#2}%
195     \fi%
196     \par\strut\hfill{\footnotesize Slide \arabic{slide}}%
197     \ifnotes%
198     \else%
199       \vfill%
200     \fi%
201   \fi%
202 }% ifframeimages
203 ⟨/package⟩
204 ⟨∗ltxml.sty⟩
205 DefMacro('\frameimage[]{}','\@frameimage{\includegrahics[#1,width=\slidewidth]{#2}}');
206 DefConstructor('\@frameimage{}',"<omdoc:omgroup layout='slide'>#1</omdoc:omgroup>\n");
207 ⟨/ltxml.sty⟩
```

4.3   Header and Footer Lines

Now, we set up the infrastructure for the footer line of the slides, we use boxes for the logos, so that they are only loaded once, that considerably speeds up processing.

\setslidelogo   The default logo is the logo of Jacobs University. Customization can be done by \setslidelogo{⟨logo name⟩}.

```
208 ⟨∗package⟩
209 \newlength{\slidelogoheight}
210 \ifnotes%
211   \setlength{\slidelogoheight}{.4cm}%
212 \else%
213   \setlength{\slidelogoheight}{1cm}%
214 \fi%
215 \newsavebox{\slidelogo}%
216 \sbox{\slidelogo}{\includegraphics[height=\slidelogoheight]{jacobs-logo}}%
217 \newrobustcmd{\setslidelogo}[1]{%
218   \sbox{\slidelogo}{\includegraphics[height=\slidelogoheight]{#1}}%
219 }%
```

---

[4]EDNOTE: MK@DG; we need to do that in the LaTeXML binding as well!

\setsource \source stores the writer's name. By default it is *Michael Kohlhase* since he is the main user and designer of this package. \setsource{⟨*name*⟩} can change the writer's name.

```
220 \def\source{Michael Kohlhase}% customize locally
221 \newrobustcmd{\setsource}[1]{\def\source{#1}}%
```

\setlicensing Now, we set up the copyright and licensing. By default we use the Creative Commons Attribuition-ShareAlike license to strengthen the public domain. If package hyperref is loaded, then we can attach a hyperlink to the license logo. \setlicensing[⟨*url*⟩]{⟨*logo name*⟩} is used for customization, where ⟨*url*⟩ is optional.

```
222 \def\copyrightnotice{\footnotesize\copyright:\hspace{.3ex}{\source}}%
223 \newsavebox{\cclogo}%
224 \sbox{\cclogo}{\includegraphics[height=\slidelogoheight]{cc_somerights}}%
225 \newif\ifcchref\cchreffalse%
226 \AtBeginDocument{%
227   \@ifpackageloaded{hyperref}{\cchreftrue}{\cchreffalse}
228 }%
229 \def\licensing{%
230   \ifcchref%
231     \href{http://creativecommons.org/licenses/by-sa/2.5/}{\usebox{\cclogo}}%
232   \else%
233     {\usebox{\cclogo}}%
234   \fi%
235 }%
236 \newrobustcmd{\setlicensing}[2][]{%
237   \def\@url{#1}%
238   \sbox{\cclogo}{\includegraphics[height=\slidelogoheight]{#2}}%
239   \ifx\@url\@empty%
240     \def\licensing{{\usebox{\cclogo}}}%
241   \else%
242     \def\licensing{%
243   \ifcchref%
244     \href{#1}{\usebox{\cclogo}}%
245   \else%
246     {\usebox{\cclogo}}%
247   \fi%
248     }%
249   \fi%
250 }%
```

\slidelabel Now, we set up the slide label for the article mode.[5]

```
251 \newrobustcmd\miko@slidelabel{%
252   \vbox to \slidelogoheight{%
253     \vss\hbox to \slidewidth%
254     {\licensing\hfill\copyrightnotice\hfill\arabic{slide}\hfill\usebox{\slidelogo}}%
255   }%
256 }%
```

---

[5]EDNOTE: see that we can use the themes for the slides some day. This is all fake.

4.4 Colors and Highlighting

Now, we set up an infrastructure for highlighting phrases in slides. Note that we use content-oriented macros for highlighting rather than directly using color markup. The first thing to to is to adapt the green so that it is dark enough for most beamers

```
257 \AtBeginDocument{%
258   \definecolor{green}{rgb}{0,.5,0}%
259   \definecolor{purple}{cmyk}{.3,1,0,.17}%
260 }%
```

We customize the \defemph, \notemph, and \stDMemph macros with colors for the use in the statements package. Furthermore we customize the \@@lec macro for the appearance of line end comments in \lec.

```
261 % \def\STpresent#1{\textcolor{blue}{#1}}
262 \def\defemph#1{{\textcolor{magenta}{#1}}}
263 \def\notemph#1{{\textcolor{magenta}{#1}}}
264 \def\stDMemph#1{{\textcolor{blue}{#1}}}
265 \def\@@lec#1{(\textcolor{green}{#1})}
266 ⟨/package⟩
267 ⟨∗ltxml.sty⟩
268 #DefMacro('\defemph{}','{\textcolor{magenta}{#1}}');
269 #DefMacro('\notemph{}','{\textcolor{magenta}{#1}}');
270 ⟨/ltxml.sty⟩
```

I like to use the dangerous bend symbol for warnings, so we provide it here.

\textwarning   as the macro can be used quite often we put it into a box register, so that it is only loaded once.

```
271 ⟨∗package⟩
272 \pgfdeclareimage[width=.9em]{miko@small@dbend}{dangerous-bend}
273 \def\smalltextwarning{%
274   \pgfuseimage{miko@small@dbend}%
275   \xspace%
276 }%
277 \pgfdeclareimage[width=1.5em]{miko@dbend}{dangerous-bend}
278 \newrobustcmd\textwarning{%
279   \raisebox{-.05cm}{\pgfuseimage{miko@dbend}}%
280   \xspace%
281 }%
282 \pgfdeclareimage[width=2.5em]{miko@big@dbend}{dangerous-bend}%
283 \newrobustcmd\bigtextwarning{%
284   \raisebox{-.05cm}{\pgfuseimage{miko@big@dbend}}%
285   \xspace%
286 }%
287 ⟨/package⟩
288 ⟨∗ltxml.sty⟩
289 DefMacro('\textwarning','\@textwarning\xspace');
290 DefConstructor('\@textwarning',"");
291 ⟨/ltxml.sty⟩
```

11

4.5 Front Matter, Titles, etc

We need to redefine the frontmatter macros inherited from the `beamer` class for LaTeXML, since there they take an optional argument.

```
292 ⟨∗ltxml.sty⟩
293 DefMacro('\title[]{}', '\@add@frontmatter{ltx:title}{#1}');
294 DefMacro('\date[]{}', '\@add@frontmatter{ltx:date}[role=creation]{#1}');
295 DefMacro('\author[]{}', sub { andSplit(T_CS('\@author'),$_[1]); });#$
296 ⟨/ltxml.sty⟩

297 %        Must be first command on slide to make positioning work.
298 ⟨∗package⟩
299 \newrobustcmd\putgraphicsat[3]{%
300   \begin{picture}(0,0)\put(#1){\includegraphics[#2]{#3}}\end{picture}%
301 }%
302 \newrobustcmd\putat[2]{%
303   \begin{picture}(0,0)\put(#1){#2}\end{picture}%
304 }%
```

4.6 Sectioning

If the `sectocframes` option is set, then we make section frames. We first define a set of counters

```
305 \ifsectocframes%
306   \if@part%
307     \newcounter{mpart}%
308     \newcounter{mchapter}%
309     \newcounter{msection}[mchapter]%
310   \else%
311     \newcounter{msection}%
312   \fi%
313   \newcounter{msubsection}[msection]%
314   \newcounter{msubsubsection}[msubsection]%
315   \newcounter{msubsubsubsection}[msubsubsection]%
316 \fi% ifsectocframes
```

and then

```
317 \ifnotes\else% only in slides
318   \renewenvironment{omgroup}[2][]{%
319     \metasetkeys{omgroup}{#1}\sref@target%
320     \advance\section@level by 1%
321     \ifsectocframes%
322     \begin{frame}%
323     \vfill\Large\centering%
324     \red{%
325     \ifcase\section@level\or%
326        \stepcounter{mpart}Part \Roman{mpart}\or%
327        \stepcounter{mchapter}Chapter \arabic{mchapter}\or
328        \stepcounter{msection}\if@part\arabic{mchapter}.\fi\arabic{msection}\or
329        \stepcounter{msubsection}\if@part\arabic{mchapter}.\fi\arabic{msection}.\arabic{msubsec
```

```
330        \stepcounter{msubsubsection}\if@part\arabic{mchapter}.\fi\arabic{msection}.\arabic{msub
331        \stepcounter{msubsubsubsection}\if@part\arabic{mchapter}.\fi\arabic{msection}.\arabic{m
332      \fi% end ifcase
333      \quad #2%
334    }%
335    \vfill%
336    \end{frame}%
337    \fi %ifsectocframes
338  }
339  {\advance\section@level by -1}%
340 \fi% ifnotes
341 ⟨/package⟩
```

4.7   Miscellaneous

The following fixes an error I do not understand, this has something to do with
beamer compatibility, which has similar definitions but only up to 1.

```
342 ⟨*package⟩
343 \expandafter\def\csname Parent2\endcsname{}
344 %     \begin{macrocode}
345 %
346 % We need to disregard the columns macros introduced by the |beamer| class
347 %     \begin{macrocode}
348 \ifnotes%
349    \renewenvironment{columns}{%
350      \par\noindent%
351      \begin{minipage}%
352      \slidewidth\centering\leavevmode%
353    }{%
354      \end{minipage}\par\noindent%
355    }%
356    \newsavebox\columnbox%
357    \renewenvironment{column}[1]{%
358      \begin{lrbox}{\columnbox}\begin{minipage}{#1}%
359    }{%
360      \end{minipage}\end{lrbox}\usebox\columnbox%
361    }%
362 \fi%
363 ⟨/package⟩
364 ⟨*ltxml.sty⟩
365 DefEnvironment('{columns}',"#body");
366 DefEnvironment('{column}{}',"#body");
```

   We also need to deal with overlay specifications introduced by the beamer
class.[6]
     [7]

---

[6]EDNOTE: this is just to keep latexml quiet, no real functionality here.
[7]EDNOTE: Deyan: We reuse the CMP itemizations defined in the omdoc.cls.ltxml binding, adjusting the parameters to be overlay-sensitive

13

```
367 DefConstructor('\uncover','#1');
368 #Define a Beamer Overlay Parameter type
369  DefParameterType('BeamerOverlay', sub {
370    my ($gullet) = @_;
371    my $tok = $gullet->readXToken;
372    if (ref $tok && ToString($tok) eq '<') {
373      $gullet->readUntil(T_OTHER('>'));
374    } else {
375      $gullet->unread($tok) if ref $tok;
376      undef; }},
377      reversion=> sub {
378  (T_OTHER('<'), $_[0]->revert, T_OTHER('>'));
379        });
380
381 #Take the "from" field of the overlay range
382 sub overlayFrom {
383   return "" unless defined $_[0];
384   my $overlay=ToString($_[0]); $overlay =~ /^(\d+)/; $1;}
385
386 #Reuse the CMP itemizations, only adjust the \item constructors.
387 DefMacro('\beamer@group@item[] OptionalBeamerOverlay IfBeginFollows', sub {
388   my($gullet,$tag,$overlay,$needwrapper)=@_;
389   $overlay=$overlay||T_OTHER("");
390   ( T_CS('\group@item@maybe@unwrap'),
391     ($needwrapper ? (Invocation(T_CS('\beamer@group@item@wrap'),$tag,$overlay)->unlist) : ()) )
392 DefConstructor('\beamer@group@item@wrap {} OptionalBeamerOverlay',
393       "<omdoc:omtext ?#2(overlay='&overlayFrom(#2)')()>"
394       . "?#1(<dc:title>#1</dc:title>)()"
395             . "<omdoc:CMP>",
396       beforeDigest=>sub {
397  Let('\group@item@maybe@unwrap','\group@item@unwrap');
398 #$_[0]->bgroup;
399 return; },
400       properties=>sub{ RefStepItemCounter(); });
401 #DefConstructor('\beamer@itemize@item[] OptionalBeamerOverlay',
402 #       "<omdoc:li ?#2(overlay='&overlayFrom(#2)')() >"
403 #     . "?#1(<dc:title>#1</dc:title>)()",
404 #        properties=>sub{ RefStepItemCounter(); });
405 DefConstructor('\beamer@enumerate@item[] OptionalBeamerOverlay',
406       "<omdoc:li ?#2(overlay='&overlayFrom(#2)')() >"
407     . "?#1(<dc:title>#1</dc:title>)()",
408       properties=>sub{ RefStepItemCounter(); });
409 DefConstructor('\beamer@description@item[] OptionalBeamerOverlay',
410       "<omdoc:di ?#2(overlay='&overlayFrom(#2)')() >"
411     . "?#1(<omdoc:dt>#1</omdoc:dt>)()<omdoc:dd>", # trust di and dt to autoclose
412       properties=>sub{ RefStepItemCounter(); });
413 ⟨/ltxml.sty⟩#$
```

Now, some things that are imported from the pgf and beamer packages:

```
414 ⟨∗ltxml.sty⟩
```

```
415 DefMacro('\putgraphicsat{}{}{}','\mygraphics[#2]{#3}');
416 DefMacro('\putat{}{}','#2');
417 ⟨/ltxml.sty⟩

418 ⟨∗package⟩
419 \ifproblems%
420    \newenvironment{problems}{}{}%
421 \else%
422    \excludecomment{problems}%
423 \fi%
424 ⟨/package⟩
425 ⟨∗ltxml.sty⟩
426 DefEnvironment('{problems}','#body');
427 ⟨/ltxml.sty⟩
```

4.8   Finale

Finally, we set the slide body font to the sans serif, and we terminate the LaTeXML bindings file with a success mark for perl.

```
428 ⟨package⟩\ifnotes\else\sf\fi
429 ⟨ltxml.sty | ltxml.cls⟩1;
```

References

[CR99]   David Carlisle and Sebastian Rathz. *The graphicxl package*. Part of the TEX distribution. The Comprehensive TEX Archive Network. 1999. URL: `https://www.tug.org/texlive/devsrc/Master/texmf-dist/doc/latex/graphics/graphicx.pdf`.

[Koh15] Michael Kohlhase. `metakeys.sty`: *A generic framework for extensible Metadata in LATEX*. Tech. rep. Comprehensive TEX 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).

[Tana]   Till Tantau. *beamer – A LaTeX class for producing presentations and slides*. URL: `http://www.ctan.org/pkg/beamer` (visited on 01/07/2014).

[Tanb]   Till Tantau. *User Guide to the Beamer Class*. URL: `http://www.ctan.org/tex-archive/macros/latex/contrib/beamer/doc/beameruserguide.pdf`.