

`smglom.cls/sty`: Semantic Multilingual Glossary for Math

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

February 14, 2014

Abstract

The `omdoc` 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).

This package supplies an infrastructure for writing OMDoc glossary entries.

Contents

1	Introduction	3
2	The User Interface	3
2.1	Package and Class Options	3
3	Implementation: The OMDoc Class	4
3.1	Class Options	4
3.2	Input	4
3.3	For Module Definitions	5
3.4	For Language Bindings	6

1 Introduction

2 The User Interface

2.1 Package and Class Options

`smglom.cls` accepts all options of the `omdoc.cls` and `article.cls` and just passes them on to these.¹

EdN:1

¹EdNOTE: describe them

3 Implementation: The OMDoc Class

3.1 Class Options

To initialize the `omdoc` class, we declare and process the necessary options.

```
1 <*cls>
2 \DeclareOption{showmeta}{\PassOptionsToPackage{\CurrentOption}{metakeys}}
3 \ProcessOptions
4 </cls>
5 <*ltxml.cls>
6 # -*- CPERL -*-
7 package LaTeXML::Package::Pool;
8 use strict;
9 use LaTeXML::Package;
10 ProcessOptions();
11 </ltxml.cls>
```

We load `omdoc.cls`, and the desired packages. For the \LaTeX ML bindings, we make sure the right packages are loaded.

```
12 <*cls>
13 \LoadClass{omdoc}
14 \RequirePackage{amstext}
15 \RequirePackage{modules}
16 \RequirePackage{dcm}
17 \RequirePackage{statements}
18 \RequirePackage{sproof}
19 \RequirePackage{cmath}
20 \RequirePackage{presentation}
21 \RequirePackage{amsfonts}
22 \RequirePackage[english,ngerman]{babel}
23 \RequirePackage{smglom}
24 </cls>
25 <*ltxml.cls>
26 LoadClass('omdoc');
27 RequirePackage('amstext');
28 RequirePackage('modules');
29 RequirePackage('dcm');
30 RequirePackage('statements');
31 RequirePackage('cmath');
32 RequirePackage('presentation');
33 RequirePackage('amsfonts');
34 RequirePackage('babel',options=>['english','ngerman']);
35 RequirePackage('smglom');
36 </ltxml.cls>
```

3.2 Input

`ginput` iterates over the language bindings.

```

37 <ltxml.sty>RawTeX(
38 <*sty | ltxml.sty>
39 \newcommand\ginput[2] [] {\input{#2}\@for\@I:=#1\do{\input{#2.\@I}}}
```

3.3 For Module Definitions

gimport just a shortcut

```

40 \newcommand\gimport[2] [] {\def\@test{#1}%
41 \ifx\@test\@empty\importmhmodule[smglom/smglom]{#2}{#2}%
42 \else\importmhmodule[smglom/#1]{#2}{#2}\fi}
```

guse just a shortcut

```

43 \newcommand\guse[2] [] {\def\@test{#1}%
44 \ifx\@test\@empty\usemhmodule[smglom/smglom]{#2}{#2}%
45 \else\usemhmodule[smglom/#1]{#2}{#2}\fi}
```

gadopt just a shortcut

```

46 \newcommand\gadopt[2] [] {\def\@test{#1}%
47 \ifx\@test\@empty\adoptmhmodule[smglom/smglom]{#2}{#2}%
48 \else\adoptmhmodule[smglom/#1]{#2}{#2}\fi}
```

gview The **gview** environment is just a layer over the **view** environment with the keys suitably adapted.

```

49 \newenvironment{gview}[3] []%
50 {\def\@test{#1}\ifx\@test\@empty\begin{view}[from=#2,to=#3]{#2}{#3}\else\begin{view}[from=#2,to=
51 \end{view}}}
```

gviewsketch The **gviewsketch** environment is just a layer over the **viewsketch** environment with the keys suitably adapted.

```

52 \newenvironment{gviewsketch}[3] []%
53 {\def\@test{#1}\ifx\@test\@empty\begin{viewsketch}[from=#2,to=#3]{#2}{#3}\else\begin{viewsketch
54 \end{viewsketch}}}
```

gve The **gve** environment is just a layer over the **gviewsketch** environment with the keys and language suitably adapted.

```

55 \def\@@en{en}\def\@@de{de}
56 \newenvironment{gve}[5] [] {\def\@test{#1}%
57 \ifx\@test\@empty\begin{gviewsketch}[id=#2.#3]{#4}{#5}\else\begin{gviewsketch}[id=#2.#3,#1]{#4}
58 \def\@test{#3}%
59 \ifx\@test\@@en\selectlanguage{english}\fi
60 \ifx\@test\@@de\selectlanguage{ngerman}\fi
61 \end{gviewsketch}}
62 </sty | ltxml.sty>
63 <ltxml.sty>');;
```

symbol has a starred form for primary symbols. Both do nothing.

```

64 <*sty>
```

```

65 \def\symbol{\@ifstar\@gobble\@gobble}
66 \</sty>
67 \<*ltxml.sty>
68 DefConstructor('\symbol OptionalMatch:* {}',"<omdoc:symbol name='#1'>");
69 \</ltxml.sty>

```

*nym

```

70 \<*cls>
71 \newcommand\hypernym[3][]{#2 is a hypernym of #3}
72 \newcommand\hyponym[3][]{#2 is a hyponym of #3}
73 \newcommand\meronym[3][]{#2 is a meronym of #3}
74 \</cls>
75 \<*ltxml.cls>
76 DefConstructor('\hypernym [] {}{}',"");
77 DefConstructor('\hyponym [] {}{}',"");
78 DefConstructor('\meronym [] {}{}',"");
79 \</ltxml.cls>

```

EdN:2

\MSC to define the Math Subject Classification,²

```

80 \<*cls>
81 \newcommand\MSC{\@gobble}
82 \</cls>
83 \<*ltxml.cls>
84 DefConstructor('\MSC{}',"");
85 \</ltxml.cls>

```

3.4 For Language Bindings

gle The gle environment is just a layer over the module environment with the keys and language suitably adapted.

```

86 \<ltxml.sty>RawTeX(
87 \<*sty | ltxml.sty>
88 \def\@en{en}\def\@de{de}
89 \newenvironment{gle}[3][]{\def\@test{#1}%
90 \ifx\@test\@empty\begin{module}[id=#2.#3]\else\begin{module}[id=#2.#3,#1]\fi
91 \gimport{#2}\def\@test{#3}%
92 \ifx\@test\@en\selectlanguage{english}\fi
93 \ifx\@test\@de\selectlanguage{ngerman}\fi}
94 {\end{module}}
95 \</sty | ltxml.sty>
96 \<ltxml.sty>');

```

noun

```

97 \<*cls>
98 \newcommand\noun[2]{}
99 \</cls>

```

²EDNOTE: MK: what to do for the LaTeXML side?

```

100 <*ltxml.cls>
101 DefMacro('noun {}{}','');
102 </ltxml.cls>

```

qualifier

```

103 <*cls>
104 \newcommand\qualifier[3]{}
105 </cls>
106 <*ltxml.cls>
107 DefMacro('qualifier {}{}{}','');
108 </ltxml.cls>

```