

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

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

March 21, 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 | ltxml.sty>
6 # -*- CPERL -*-
7 package LaTeXML::Package::Pool;
8 use strict;
9 use LaTeXML::Package;
10 ProcessOptions();
11 </ltxml.cls | ltxml.sty>
```

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 \edef\mh@@repos{\mh@currentrepos}%
42 \ifx\@test\@empty\importmhmodule[\mh@@repos]{#2}{#2}%
43 \else\importmhmodule[#1]{#2}{#2}\fi}

guse just a shortcut
44 \newcommand\guse[2] [] {\def\@test{#1}%
45 \edef\mh@@repos{\mh@currentrepos}%
46 \ifx\@test\@empty\usemhmodule[\mh@@repos]{#2}{#2}%
47 \else\usemhmodule[#1]{#2}{#2}\fi}

gadopt just a shortcut
48 \newcommand\gadopt[2] [] {\def\@test{#1}%
49 \edef\mh@@repos{\mh@currentrepos}%
50 \ifx\@test\@empty\adoptmhmodule[\mh@@repos]{#2}{#2}%
51 \else\adoptmhmodule[#1]{#2}{#2}\fi}

gview The gview environment is just a layer over the view environment with the keys
suitably adapted.
52 \newenvironment{gview}[3] []%
53 {\def\@test{#1}\ifx\@test\@empty\begin{view}[from=#2,to=#3]{#2}{#3}\else\begin{view}[from=#2,to
54 {\end{view}}

gviewsketch The gviewsketch environment is just a layer over the viewsketch environment
with the keys suitably adapted.
55 \newenvironment{gviewsketch}[3] []%
56 {\def\@test{#1}\ifx\@test\@empty\begin{viewsketch}[from=#2,to=#3]{#2}{#3}\else\begin{viewsketch
57 {\end{viewsketch}}

gve The gve environment is just a layer over the gviewsketch environment with the
keys and language suitably adapted.
58 \def\@@en{en}\def\@@de{de}
59 \newenvironment{gve}[5] [] {\def\@test{#1}%
60 \ifx\@test\@empty\begin{gviewsketch}[id=#2.#3]{#4}{#5}\else\begin{gviewsketch}[id=#2.#3,#1]{#4}
61 \def\@test{#3}%
62 \ifx\@test\@@en\selectlanguage{english}\fi
63 \ifx\@test\@@de\selectlanguage{ngerman}\fi}
64 {\end{gviewsketch}}
65 </sty | ltxml.sty>
66 <ltxml.sty>');

```

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

```

67 <*sty>
68 \def\symbol{\@ifstar\@gobble\@gobble}
69 </sty>
70 <*ltxml.sty>
71 DefConstructor('\symbol OptionalMatch:* {}',
72     "<mdoc:symbol ?#1(role='primary')(role='secondary') name='#2'/>");
73 </ltxml.sty>

```

`*nym`

```

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

```

EdN:2

`\MSC` to define the Math Subject Classification, ²

```

84 <*cls>
85 \newcommand\MSC{\@gobble}
86 </cls>
87 <*ltxml.cls>
88 DefConstructor('\MSC{}', "");
89 </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.

```

90 <ltxml.sty>RawTeX(
91 <*sty | ltxml.sty>
92 \def\@en{en}\def\@de{de}
93 \newenvironment{gle}[3][]{\def\@test{#1}%
94 \ifx\@test\@empty\begin{module}[id=#2.#3]\else\begin{module}[id=#2.#3,#1]\fi
95 \edef\mh@crepos{\mh@currentrepos}%
96 \gimport[\mh@crepos]{#2}\def\@test{#3}%
97 \ifx\@test\@en\selectlanguage{english}\fi
98 \ifx\@test\@de\selectlanguage{ngerman}\fi
99 {\end{module}}}
100 </sty | ltxml.sty>
101 <ltxml.sty>');

```

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

noun

```
102 <*cls>
103 \newcommand\noun[2]{}
104 </cls>
105 <*ltxml.cls>
106 DefMacro('\noun {}{}','');
107 </ltxml.cls>
```

qualifier

```
108 <*cls>
109 \newcommand\qualifier[3]{}
110 </cls>
111 <*ltxml.cls>
112 DefMacro('\qualifier {}{}{}','');
113 </ltxml.cls>
```