

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

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

April 27, 2014

## Abstract

The `smglom` package is part of the  $\text{\LaTeX}$  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

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>The User Interface</b>	<b>2</b>
2.1	Package and Class Options . . . . .	2
<b>3</b>	<b>Implementation: The SMGloM Class</b>	<b>3</b>
3.1	Class Options . . . . .	3
3.2	For Module Definitions . . . . .	4
3.3	For Language Bindings . . . . .	5

## **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.

## 3 Implementation: The SMGloM Class

### 3.1 Class Options

To initialize the `smglom` class, we pass on all options to `omdoc.cls`

```
1 <*cls>
2 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{omdoc}}
3 \ProcessOptions
4 </cls>
5 <*ltxml.cls | ltxml.sty>
6 # -*- CPERL -*-
7 package LaTeXML::Package::Pool;
8 use strict;
9 use LaTeXML::Package;
10 DeclareOption(undef,sub {PassOptions('article','cls',ToString(Digest(T_CS('\CurrentOption'))));
11 ProcessOptions();
12 </ltxml.cls | ltxml.sty>
```

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

```
13 <*cls>
14 \LoadClass{omdoc}
15 \RequirePackage{smglom}
16 </cls>
17 <*sty>
18 \RequirePackage{amstext}
19 \RequirePackage{modules}
20 \RequirePackage{dcm}
21 \RequirePackage{statements}
22 \RequirePackage{sproof}
23 \RequirePackage{cmath}
24 \RequirePackage{presentation}
25 \RequirePackage{amsfonts}
26 </sty>
27 <*ltxml.cls>
28 LoadClass('omdoc');
29 RequirePackage('smglom');
30 </ltxml.cls>
31 <*ltxml.sty>
32 RequirePackage('amstext');
33 RequirePackage('modules');
34 RequirePackage('dcm');
35 RequirePackage('statements');
36 RequirePackage('cmath');
37 RequirePackage('presentation');
38 RequirePackage('amsfonts');
39 </ltxml.sty>
```

## 3.2 For Module Definitions

```

gimport just a shortcut
40 <ltxml.sty>RawTeX(
41 <*sty | ltxml.sty>
42 \newcommand\gimport[2] [] {\def\@test{#1}%
43 \edef\mh@@repos{\mh@currentrepos}%
44 \ifx\@test\@empty\importmhmodule[repos=\mh@@repos,ext=tex,path=#2]{#2}%
45 \else\importmhmodule[repos=#1,ext=tex,path=#2]{#2}\fi
46 \mhcurrentrepos\mh@@repos\ignorespaces}

guse just a shortcut
47 \newcommand\guse[2] [] {\def\@test{#1}%
48 \edef\mh@@repos{\mh@currentrepos}%
49 \ifx\@test\@empty\usemhmodule[repos=\mh@@repos,ext=tex,path=#2]{#2}%
50 \else\usemhmodule[repos=#1,ext=tex,path=#2]{#2}\fi
51 \mhcurrentrepos\mh@@repos\ignorespaces}

gadopt just a shortcut
52 \newcommand\gadopt[2] [] {\def\@test{#1}%
53 \edef\mh@@repos{\mh@currentrepos}%
54 \ifx\@test\@empty\adoptmhmodule[repos=\mh@@repos,ext=tex,path=#2]{#2}%
55 \else\adoptmhmodule[repos=#1,ext=tex,path=#2]{#2}\fi
56 \mhcurrentrepos\mh@@repos\ignorespaces}
57 </sty | ltxml.sty>
58 <ltxml.sty>');

*nym
59 <*sty>
60 \newcommand\hypernym[3] [] {\if@importing\else\par\noindent #2 is a hypernym of #3\fi}
61 \newcommand\hyponym[3] [] {\if@importing\else\par\noindent #2 is a hyponym of #3\fi}
62 \newcommand\meronym[3] [] {\if@importing\else\par\noindent #2 is a meronym of #3\fi}
63 </sty>
64 <*ltxml.sty>
65 DefConstructor('\hypernym [] {}{}', "");
66 DefConstructor('\hyponym [] {}{}', "");
67 DefConstructor('\meronym [] {}{}', "");
68 </ltxml.sty>

EdN:1 \MSC to define the Math Subject Classification,1
69 <*sty>
70 \newcommand\MSC[1]{\if@importing\else MSC: #1\fi}
71 </sty>
72 <*ltxml.sty>
73 DefConstructor('\MSC{}', "");
74 </ltxml.sty>

```

---

<sup>1</sup>EdNOTE: MK: what to do for the LaTeXML side?

### 3.3 For Language Bindings

Here we adapt the `smultiling` functionality to the special situation, where the module and file names are identical by design.

**gviewsig** The `gviewsig` environment is just a layer over the `viewsig` environment with the keys suitably adapted.

```
75 <ltxml.sty>RawTeX('
76 <*sty | ltxml.sty>
77 \newenvironment{gviewsig}[4] [] {\def\test{#1}\ifx\@test\@empty%
78 \begin{mhviewsig}[frompath=#3,topath=#4]{#2}{#3}{#4}\else
79 \begin{mhviewsig}[frompath=#3,topath=#4,#1]{#2}{#3}{#4}\fi}
80 {\end{mhviewsig}}}
```

**gviewnl** The `gve` environment is just a layer over the `viewnl` environment with the keys suitably adapted.

```
81 \newenvironment{gviewnl}[5] [] {\def\@test{#1}\ifx\@test\@empty%
82 \begin{mhviewnl}[frompath=#4,topath=#5]{#2}{#3}{#4}{#5}\else%
83 \begin{mhviewnl}[#1,frompath=#4,topath=#5]{#2}{#3}{#4}{#5}\fi
84 \smg@select@language{#3}}
85 {\end{mhviewnl}}
86 </sty | ltxml.sty>
87 <ltxml.sty>');;
```