

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

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

June 10, 2014

Abstract

The `smglom` package is part of the `sTeX` collection, a version of `TeX/LATeX` that allows to markup `TeX/LATeX` documents semantically without leaving the document format, essentially turning `TeX/LATeX` into a document format for mathematical knowledge management (MKM).

This package supplies an infrastructure for writing OMDoc glossary entries.

Contents

1	Introduction	2
2	The User Interface	2
2.1	Package and Class Options	2
3	Implementation: The SMGloM Class	3
3.1	Class Options	3
3.2	For Module Definitions	4
3.2.1	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 warnings;
10 use LaTeXML::Package;
11
12 DeclareOption(undef,sub {PassOptions('article','cls',ToString(Digest(T_CS('\CurrentOption'))));
13 ProcessOptions();
14 </ltxml.cls | ltxml.sty>
```

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

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

3.2 For Module Definitions

`gimport` just a shortcut

```

43 <*sty>
44 \newcommand\gimport[2] [] {\def\@test{#1}%
45 \edef\mh@@repos{\mh@currentrepos}%
46 \ifx\@test\@empty\importmhmodule[repos=\mh@@repos,ext=tex,path=#2]{#2}%
47 \else\importmhmodule[repos=#1,ext=tex,path=#2]{#2}\fi
48 \mhcurrentrepos\mh@@repos\ignorespaces}
49 </sty>
50 <*ltxml.sty>
51 DefMacro('gimport[]{}', 'g@import[ext=tex,path=#2]{#1}{#2}');
52 DefConstructor('g@import OptionalKeyVals:importmhmodule {}{}',
53 " <omdoc:imports from='?'&GetKeyVal{#1,'load'}(&canonical_omdoc_path(&GetKeyVal{#1,'load'}))() \
54 afterDigest => \gimportI);

```

To make this work we need a sub that sets the respective values.

```

55 sub gimportI {
56 my ($stomach,$whatsit) = @_;
57 my $keyval = $whatsit->getArg(1);
58 my $repos = ToString($whatsit->getArg(2));
59 my $name = $whatsit->getArg(3);
60 if ($repos) {
61     $keyval->setValue('repos',$repos); }
62 else {
63     $keyval->setValue('repos',LookupValue('current_repos')); }
64 # Mystery: Why does $whatsit->setArgs($keyval,$name) raise a warning for
65 # "odd numbers" in hash assignment? Workaround for now!
66 $$whatsit{args}[1] = $name; # Intention: $whatsit->setArg(2,$name);
67 undef $$whatsit{args}[2]; # Intention: $whatsit->deleteArg(3);
68 importMHmoduleI($stomach,$whatsit);
69 return; }##$
70 </ltxml.sty>

```

`guse` just a shortcut

```

71 <*sty>
72 \newcommand\guse[2] [] {\def\@test{#1}%
73 \edef\mh@@repos{\mh@currentrepos}%
74 \ifx\@test\@empty\usemhmodule[repos=\mh@@repos,ext=tex,path=#2]{#2}%
75 \else\usemhmodule[repos=#1,ext=tex,path=#2]{#2}\fi
76 \mhcurrentrepos\mh@@repos\ignorespaces}
77 </sty>
78 <*ltxml.sty>
79 DefMacro('guse[]{}', 'g@use[ext=tex,path=#2]{#1}{#2}');
80 DefConstructor('g@use OptionalKeyVals:importmhmodule {}{}',
81 " <omdoc:uses from='?'&GetKeyVal{#1,'load'}(&canonical_omdoc_path(&GetKeyVal{#1,'load'}))() \##2
82 afterDigest => \gimportI);
83 </ltxml.sty>

```

`gadopt` just a shortcut

```

84 <*sty>
85 \newcommand\gadopt[2][]{\def\@test{#1}%
86 \edef\mh@repos{\mh@currentrepos}%
87 \ifx\@test\@empty\adoptmhmodule[repos=\mh@repos,ext=tex,path=#2]{#2}%
88 \else\adoptmhmodule[repos=#1,ext=tex,path=#2]{#2}\fi
89 \mhcurrentrepos\mh@repos\ignorespaces}
90 </sty>
91 <*lxml.sty>
92 DefMacro('gadopt[]{}', 'g@adopt[ext=tex,path=#2]{#1}{#2}');
93 DefConstructor('g@adopt OptionalKeyVals:importmhmodule {} {}',
94 " <omdoc:adopts from='?'&GetKeyVal(#1,'load')(&canonical_omdoc_path(&GetKeyVal(#1,'load')))(\#
95 afterDigest => \&gimportI);
96 </lxml.sty>

```

*nym

```

97 <*sty>
98 \newcommand\hypernym[3][]{\if@importing\else\par\noindent #2 is a hypernym of #3\fi}
99 \newcommand\hyponym[3][]{\if@importing\else\par\noindent #2 is a hyponym of #3\fi}
100 \newcommand\meronym[3][]{\if@importing\else\par\noindent #2 is a meronym of #3\fi}
101 </sty>
102 <*lxml.sty>
103 DefConstructor('hypernym [] {}{}', "");
104 DefConstructor('hyponym [] {}{}', "");
105 DefConstructor('meronym [] {}{}', "");
106 </lxml.sty>

```

EdN:1

\MSC to define the Math Subject Classification, ¹

```

107 <*sty>
108 \newcommand\MSC[1]{\if@importing\else MSC: #1\fi}
109 </sty>
110 <*lxml.sty>
111 DefConstructor('MSC{}', "");
112 </lxml.sty>

```

3.2.1 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.

```

113 <lxml.sty>RawTeX(
114 <*sty | lxml.sty>
115 \newenvironment{gviewsig}[4][]{\def\test{#1}\ifx\@test\@empty%
116 \begin{mhviewsig}[frompath=#3,topath=#4]{#2}{#3}{#4}\else
117 \begin{mhviewsig}[frompath=#3,topath=#4,#1]{#2}{#3}{#4}\fi}
118 {\end{mhviewsig}}

```

¹EdNOTE: MK: what to do for the LaTeXML side?

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

```

119 \newenvironment{gviewnl}[5][\def\@test{#1}\ifx\@test\empty%
120 \begin{mhviewnl}[frompath=#4,topath=#5]{#2}{#3}{#4}{#5}\else%
121 \begin{mhviewnl}[#1,frompath=#4,topath=#5]{#2}{#3}{#4}{#5}\fi
122 \smul@select@language{#3}}
123 {\end{mhviewnl}}
124 \</sty | ltxml.sty>
125 \<ltxml.sty>');

```