`omd.sty`: A generic framework for extensible Metadata in LaTeX*

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

July 20, 2010

Abstract

The `omd` 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 the infrastructure for extending sTeX macros with OMDoc metadata. This package is mainly intended for authors of sTeX extension packages.

Contents

*Version v0.9 (last revised 2010/06/25)

1 Introduction

The `omd` package supplies the infrastructure for extending sTEX macros with ontology-based metadata. The `omd` infrastructure is intended to support the new metadata infrastructure for the OMDoc format [Koh06] introduced in OMDoc1.3 [Koh10][1]

EdNote(1)

2 The User Interface

`\omdaddkey` The `\omdaddkey` command takes two arguments, a metadata group ⟨*group*⟩ and a metadata keyword name ⟨*key*⟩. It registers ⟨*key*⟩ in the metadata group ⟨*group*⟩. The keys registered for a metadata group can be used for defining `\omdsetkeys` macros with a key/value arguments via the `\omdsetkeys` macro, see for instance the the definition in Figure 1. With these definitions in a used package[1] `\foo[type=bar,id=f4711]` is formatted to

I have seen a *foo* of type `bar` with identifier `f4711`!

```
\omdaddkey{foo}{id}
\omdaddkey{foo}{type}
\newcommand\foo[1][]{\omdsetkeys{foo}{#1}
I have seen a \emph{foo} of type \texttt{\foo@type} with identifier
\texttt\foo@id!}
```

Example 1: Defining a macro with metadata

3 The Implementation

We build on the `keyval` package which we first need to load.

```
⟨*package⟩
\RequirePackage{keyval}[1997/11/10]
⟨*ltxml⟩
⟨/ltxml⟩
```

`\omdaddkey` An invocation of `\omdaddkey{`⟨*group*⟩`}{`⟨*key*⟩`}` macro first extends the clearkeys macro and then defines the key ⟨*key*⟩ with the `\define@key` macro from the `keyval` package. This stores the key value given in the local macro `\`⟨*group*⟩`@`⟨*key*⟩.

```
⟨*package⟩
\newcommand\omdaddkey[3][]{\omd@ext@clear@keys{#2}{#3}{#1}%
\define@key{#2}{#3}[#1]{\expandafter\gdef\csname #2@#3\endcsname{##1}}}
⟨/package⟩
⟨*ltxml⟩
⟨/ltxml⟩
```

[1] EdNote: continue

[1] The character is only allowed in packages.

\omdsetkeys

```
⟨*package⟩
\newcommand\omdsetkeys[2]{\csname clear@#1@keys\endcsname\setkeys{#1}{#2}}
⟨/package⟩
⟨*ltxml⟩
⟨/ltxml⟩
```

\omd@ext@clear@keys `\omd@ext@clear@keys{`⟨*group*⟩`}{`⟨*key*⟩`}{`⟨*default*⟩`}` extends (or sets up if this is the first `\omdaddkey` for ⟨*group*⟩) the `\clear@`⟨*group*⟩`@keys` macro to set the default value ⟨*default*⟩ for ⟨*key*⟩. The `\clear@`⟨*group*⟩`@keys` macro is used in the generic `\omdsetkeys` macro below.

```
⟨*package⟩
\newcommand\omd@ext@clear@keys[3]{\@omd@ext@clear@keys{#1}{#1@#2}{#3}}
\newcommand\@omd@ext@clear@keys[3]{\@ifundefined{clear@#1@keys}%
{\expandafter\def\csname clear@#1@keys\endcsname%
{\expandafter\gdef\csname #2\endcsname{#3}}}%
{\expandafter\g@addto@macro\csname clear@#1@keys\endcsname%
{\expandafter\gdef\csname #2\endcsname{#3}}}}
⟨/package⟩
⟨*ltxml⟩
⟨/ltxml⟩
```

3.1 Using better defaults than empty

\omdaddkeynew `\omdaddkeynew` is an experimental version of `\omdaddkey` which gives `\omd@unspecified` as an optional argument, so that it is used as the default value here and then test for it in `\omfidus`. But unfortunately, this does not work yet.

```
⟨*package⟩
\newcommand\omdaddkeynew[3][]{\omd@ext@clear@keys{#2}{#3}{#1}%
\define@key{#2}{#3}{\expandafter\gdef\csname #2@#3\endcsname{##1}}}
⟨/package⟩
⟨*ltxml⟩
⟨/ltxml⟩
```

EdNote(2) \omd@unspecified Ain internal macro for unspecified values. It is used to initialize keys.[2]

```
⟨*package⟩
\newcommand\omd@unspecified{an omd-defined key left unspecified}
⟨/package⟩
⟨*ltxml⟩
⟨/ltxml⟩
```

\omdifus This just tests for equality of the first arg with `\omd@unspecified`

```
⟨*package⟩
\newcommand\omdifus[4]{\message{testing #1@#2=\csname#1@#2\endcsname}\expandafter\ifx\csname #1
⟨/package⟩
⟨*ltxml⟩
⟨/ltxml⟩
```

[2] EdNote: MK: we could probably embed an package error or warning in here

3.2 Finale

Finally, we need to terminate the file with a success mark for perl.

42 ⟨ltxml⟩*1;*

References

[Koh06] Michael Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. LNAI 4180. Springer Verlag, Aug. 2006. URL: http://omdoc.org/pubs/omdoc1.2.pdf.

[Koh10] Michael Kohlhase. "An Open Markup Format for Mathematical Documents OMDoc [Version 1.3]". Draft Specification. 2010. URL: https://svn.omdoc.org/repos/omdoc/branches/omdoc-1.3/doc/spec/main.pdf.