physml.sty: An Infrastructure for Marking Up PhysML in TEX/LATEX[*]

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

August 15, 2010

Abstract

The physml package allows mark up PhysML structures in LATEX documents that can be harvested by automated tools or exported to PDF, while at the same time generating conventional title information.

Contents

[*] Version ? (last revised ?)

1 Introduction

The physml package is part of the sTeX project (see [Ste; Koh08]) and extends sTeX with an infrastructure for creating PhysML markup [Phy; HKS06] from LaTeX sources.

2 The User Interface

EdNote(1) [1]

2.1 Package Options

showmeta The physml package takes a single option: showmeta. If this is set, then the metadata keys are shown (see [**Kohlhase:metakeys:ctan**] for details and customization options).

2.2 Observables

2.3 Systems

2.4 Values

2.5 Experiments

[1] EdNote: document the functionality here

3 The Implementation

The sref package generates two files: the LaTeX package (all the code between ⟨*package⟩ and ⟨/package⟩) and the LaTeXML bindings (between ⟨*ltxml⟩ and ⟨/ltxml⟩). We keep the corresponding code fragments together, since the documentation applies to both of them and to prevent them from getting out of sync.

We first set up header information for the LaTeXML binding file.

```
⟨*ltxml⟩
package LaTeXML::Package::Pool;
use strict;
use LaTeXML::Package;
⟨/ltxml⟩
```

3.1 Package Options

We declare some switches which will modify the behavior according to the package options. Generally, an option xxx will just set the appropriate switches to true (otherwise they stay false).[2]

EdNote(2)

```
⟨*package⟩
\DeclareOption{showmeta}{\PassOptionsToPackage{\CurrentOption}{metakeys}}
\ProcessOptions
⟨/package⟩
```

```
⟨*package⟩
\RequirePackage{sref}
⟨/package⟩
```

physml

```
⟨package⟩\newcommand{\physml}{PhysML}
```

3.2 Observables

observable

```
⟨*package⟩
\srefaddidkey{obs}
\addmetakey{obs}{algebra}
\newenvironment{observable}[1][]% keyword args
{\metasetkeys{obs}{#1}{\bf{Observable (\obs@id):}}\begin{description}}
{\end{description}}
⟨/package⟩
⟨*ltxml⟩
DefKeyVal('obs','name','Semiverbatim');
DefKeyVal('obs','algebra','Semiverbatim');
DefEnvironment('{observable} OptionalKeyVals:obs',
    "<omdoc:observable name='&KeyVal(#1,'name')'"
   .            "?&defined(&KeyVal(#1,'algebra'))(algebra='&KeyVal(#1,'algebra')')()>
   .  "#body"
```

[2] EdNote: need an implementation for LaTeXML

```
     . "</omdoc:observable>\n");
⟨/ltxml⟩
```

```
⟨*package⟩
\newcommand{\obsref}[1]{ref: #1}
⟨/package⟩
⟨*ltxml⟩
DefConstructor('\obsref{}',"<omdoc:observable xref='#1'/>");
⟨/ltxml⟩
```

refinement

```
⟨*package⟩
\srefaddidkey{refinement}
\newenvironment{refinement}[1][]{\item[Refinement]}{}
⟨/package⟩
⟨*ltxml⟩
DefKeyVal('refinement','id','Semiverbatim');
DefEnvironment('{refinement} OptionalKeyVals:refinement',
       "<omdoc:refinement ?&defined(&KeyVal(#1,'id))(xml:id='&KeyVal(#1,'id')')()>"
     . "#body"
     . "</omdoc:refinement>\n");
⟨/ltxml⟩
```

opdef

```
⟨*package⟩
\newenvironment{opdef}[1][]{\item[Opdef:]}{}
⟨/package⟩
⟨*ltxml⟩
DefKeyVal('opdef,'id','Semiverbatim');
DefEnvironment('{opdef} OptionalKeyVals:opdef',
       "<omdoc:opdef ?&defined(&KeyVal(#1,'id))(xml:id='&KeyVal(#1,'id')')()>"
     . "#body"
     . "</omdoc:opdef>\n");
⟨/ltxml⟩
```

3.3 Systems

system

```
⟨*package⟩
\srefaddidkey{system}
\newenvironment{system}[1][]% keyword args
{\metasetkeys{system}{#1}{\bf{System (\system@id):}}\begin{description}}
{\end{description}}
⟨/package⟩
⟨*ltxml⟩
DefKeyVal('system,'id','Semiverbatim');
DefEnvironment('{system} OptionalKeyVals:id',
       "<omdoc:system ?&defined(&KeyVal(#1,'id))(xml:id='&KeyVal(#1,'id')')()>"
     . "#body"
```

```
      . "</omdoc:system>\n");
⟨/ltxml⟩
```

EdNote(3) [3]

```
⟨*package⟩
\newcommand{\sysref}[1]{ref: #1}
⟨/package⟩
⟨*ltxml⟩
⟨/ltxml⟩
```

realization

```
⟨*package⟩
\newenvironment{realization}[1][]{\item[Realization]}{}
⟨/package⟩
⟨*ltxml⟩
DefKeyVal(’realization,’id’,’Semiverbatim’);
DefEnvironment(’{realization} OptionalKeyVals:realization,
        "<omdoc:realization ?&defined(&KeyVal(#1,’id’))(xml:id=’&KeyVal(#1,’id’)’)()>"
      . "#body"
      . "</omdoc:realization>\n");
⟨/ltxml⟩
```

preparation

```
⟨*package⟩
\newenvironment{preparation}[1][]{\item[Preparation]}{}
⟨/package⟩
⟨*ltxml⟩
DefKeyVal(’preparation,’id’,’Semiverbatim’);
DefEnvironment(’{preparation} OptionalKeyVals:preparation’,
        "<omdoc:preparation ?&defined(&KeyVal(#1,’id’))(xml:id=’&KeyVal(#1,’id’)’)()>"
      . "#body"
      . "</omdoc:preparation>\n");
⟨/ltxml⟩
```

state

```
⟨*package⟩
\srefaddidkey{state}
\addmetakey{state}{of}
\newenvironment{state}[1][]% keyword args
{{\bf{State:}}}{\par\noindent}
⟨/package⟩
⟨*ltxml⟩
DefKeyVal(’state’,’id’,’Semiverbatim’);
DefKeyVal(’state’,’of’,’Semiverbatim’);
DefEnvironment(’{state} OptionalKeyVals:state’,
        "<omdoc:state ?&defined(&KeyVal(#1,’id’))(xml:id=’&KeyVal(#1,’id’)’)()"
      .               "?&defined(&KeyVal(#1,’of’))(of=’&KeyVal(#1,’of’)’)()>"
      .   "#body"
```

[3] EdNote: need to implement this in LaTeXML

```
    . "</omdoc:state>\n");
⟨/ltxml⟩
```

3.4 Values

statevalue

```
⟨*package⟩
\newcommand{\statevalue}[2]{{\tt{#1}}$\rightarrow${#2}}
% \srefaddidkey{value}
% \addmetakey{value}{for}
% \newenvironment{value}[1][]% keyword args
% {{\bf{Value:}}}{\par\noindent}
⟨/package⟩
⟨*ltxml⟩
DefKeyVal('value','id','Semiverbatim');
DefKeyVal('value','for','Semiverbatim');
DefEnvironment('{state} OptionalKeyVals:value',
      "<omdoc:value ?&defined(&KeyVal(#1,'id'))(xml:id='&KeyVal(#1,'id)')()"
    .             "?&defined(&KeyVal(#1,'for'))(for='&KeyVal(#1,'of)')()>"
    .   "#body"
    . "</omdoc:value>\n");
⟨/ltxml⟩
```

3.5 Experiments

measurement

```
⟨*package⟩
\srefaddidkey{measurement}
\newenvironment{measurement}[1][]% keyword args
{{\bf{Measurement:}}}{\par\noindent}
⟨/package⟩
⟨*ltxml⟩
DefKeyVal('measurement,'id','Semiverbatim');
DefEnvironment('{measurement} OptionalKeyVals:measurement,
      "<omdoc:measurement ?&defined(&KeyVal(#1,'id'))(xml:id='&KeyVal(#1,'id)')()>"
    .   "#body"
    . "</omdoc:measurement>\n");
⟨/ltxml⟩
```

experiment

```
⟨*package⟩
\srefaddidkey{experiment}
\newenvironment{experiment}[1][]% keyword args
{{\bf{Experiment:}}}{\par\noindent}
⟨/package⟩
⟨*ltxml⟩
DefKeyVal('experiment,'id','Semiverbatim');
DefEnvironment('{experiment} OptionalKeyVals:experiment,
      "<omdoc:experiment ?&defined(&KeyVal(#1,'id'))(xml:id='&KeyVal(#1,'id)')()>"
      . "#body"
      . "</omdoc:experiment>\n");
⟨/ltxml⟩
```

evidence

```
⟨*package⟩
\srefaddidkey{evidence}
\newenvironment{evidence}[1][]% keyword args
{{\bf{Evidence:}}}{\par\noindent}
⟨/package⟩
⟨*ltxml⟩
DefKeyVal(’evidence,’id’,’Semiverbatim’);
DefEnvironment(’{evidence} OptionalKeyVals:evidence,
        "<omdoc:evidence ?&defined(&KeyVal(#1,’id’))(xml:id=’&KeyVal(#1,’id)’)()>"
      . "#body"
      . "</omdoc:evidence>\n");
⟨/ltxml⟩
```

interpretation

```
⟨*package⟩
\srefaddidkey{interpretation}
\newenvironment{interpretation}[1][]% keyword args
{{\bf{Interpretation:}}}{\par\noindent}
⟨/package⟩
⟨*ltxml⟩
DefKeyVal(’interpretation,’id’,’Semiverbatim’);
DefEnvironment(’{interpretation} OptionalKeyVals:interpretation,
        "<omdoc:interpretation ?&defined(&KeyVal(#1,’id’))(xml:id=’&KeyVal(#1,’id)’)()>"
      . "#body"
      . "</omdoc:interpretation>\n");
⟨/ltxml⟩
```

3.6 Bookkeeping

We declare all tags as receiving `xml:id` and `stex:sref` attributes, pinpointing a unique identifier and their location in the LaTeX source.

```
⟨*ltxml⟩
Tag(’omdoc:observable’,afterOpen=>\&numberIt,afterClose=>\&locateIt);
Tag(’omdoc:realization’,afterOpen=>\&numberIt,afterClose=>\&locateIt);
Tag(’omdoc:preparation’,afterOpen=>\&numberIt,afterClose=>\&locateIt);
Tag(’omdoc:state’,afterOpen=>\&numberIt,afterClose=>\&locateIt);
Tag(’omdoc:value’,afterOpen=>\&numberIt,afterClose=>\&locateIt);
Tag(’omdoc:measurement’,afterOpen=>\&numberIt,afterClose=>\&locateIt);
Tag(’omdoc:experiment’,afterOpen=>\&numberIt,afterClose=>\&locateIt);
Tag(’omdoc:evidence’,afterOpen=>\&numberIt,afterClose=>\&locateIt);
Tag(’omdoc:interpretation’,afterOpen=>\&numberIt,afterClose=>\&locateIt);
Tag(’omdoc:refinement’,afterOpen=>\&numberIt,afterClose=>\&locateIt);
Tag(’omdoc:opdef’,afterOpen=>\&numberIt,afterClose=>\&locateIt);
Tag(’omdoc:system’,afterOpen=>\&numberIt,afterClose=>\&locateIt);
```

```
⟨/ltxml⟩
```

3.7 Finale

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

```
⟨ltxml⟩1;
```

References

[HKS06] Eberhard Hilf, Michael Kohlhase, and Heinrich Stamerjohanns. "Capturing the Content of Physics: Systems, Observables, and Experiments". In: *Mathematical Knowledge Management, MKM'06*. Ed. by Jon Borwein and William M. Farmer. LNAI 4108. Springer Verlag, 2006, pp. 165–178. URL: `http://kwarc.info/kohlhase/papers/mkm06physml.pdf`.

[Koh08] Michael Kohlhase. "Using LaTeX as a Semantic Markup Format". In: *Mathematics in Computer Science* (2008), pp. 279–304. URL: `https://svn.kwarc.info/repos/stex/doc/mcs08/stex.pdf`.

[Phy] *PhysML - Capturing the Content of Physics*. URL: `http://www.omdoc.org/projects/physml/`.

[Ste] *Semantic Markup for LaTeX*. URL: `https://trac.kwarc.info/sTeX/` (visited on 12/02/2009).