CNXL\LaTeX{}: A \LaTeX{}-based Syntax for Connexions Modules

Michael Kohlhase
Jacobs University, Bremen
http://kwarc.info/kohlhase
January 28, 2012

Abstract

We present CNXL\LaTeX{}, a collection of \LaTeX{} macros that allow to write Connexions modules without leaving the \LaTeX{} workflow. Modules are authored in CNXL\LaTeX{} using only a text editor, transformed to PDF and proofread as usual. In particular, the \LaTeX{} workflow is independent of having access to the Connexions system, which makes CNXL\LaTeX{} attractive for the initial version of single-author modules.

For publication, CNXL\LaTeX{} modules are transformed to CNXML via the \LaTeX{}\XML{} translator and can be uploaded to the Connexions system.
## Contents

1 Introduction ........................................ 3

2 The User Interface .................................. 3
   2.1 Package Options .................................. 3
   2.2 Document Structure ............................... 3
   2.3 Mathematics ...................................... 4
   2.4 Statements ......................................... 4
   2.5 Connexions: Links and Cross-References ......... 5
   2.6 Metadata ........................................... 6
   2.7 Exercises .......................................... 7
   2.8 Graphics, etc. ..................................... 7

3 Limitations ........................................... 7

4 The Implementation .................................. 8
   4.1 Package Options .................................. 8
   4.2 Document Structure ............................... 10
   4.3 Mathematics ...................................... 12
   4.4 Rich Text .......................................... 12
   4.5 Statements ........................................ 15
   4.6 Connexions ........................................ 18
   4.7 Metadata .......................................... 19
1 Introduction

The Connexions project is a\footnote{EdNote: continue; copy from somewhere...}
The CNXML format — in particular the embedded content MathML — is hard to write by hand, so we provide a set of environments that allow to embed the CNXML document model into $\LaTeX$.

2 The User Interface

This document is not a manual for the Connexions XML encoding, or a practical guide how to write Connexions modules. We only document the $\LaTeX$ bindings for CNXML and will presuppose experience with the format or familiarity with\footnote{EdNote: cite the relevant stuff here}.

Note that formatting CNXML documents with the $\LaTeX$ formatter does little to enforce the restrictions imposed by the CNXML document model. You will need to run the $\LaTeX$XML converter for that (it includes DTD validation) and any CNX-specific quality assurance tools after that. \footnote{EdNote: talk about Content MathML and cmathml.sty somewhere}

The $\LaTeX$XML class makes heavy use of the KeyVal package, which is part of your $\LaTeX$ distribution. This allows to add optional information to $\LaTeX$ macros in the form of key-value pairs: A macro \texttt{\textbackslash foo} that takes a KeyVal argument and a regular one, so a call might look like \texttt{\textbackslash foo\{bar\}} (no KeyVal information given) or \texttt{\textbackslash foo\{key1=val1,...,keyn=valn\}\{bar\}}, where \texttt{key1,...,keyn} are predefined keywords and values are $\LaTeX$ token sequences that do not contain comma characters (though they may contain blank characters). If a value needs to contain commas, then it must be enclosed in curly braces, as in \texttt{\textbackslash foo\{args={a, comma, separated, list}\}}. Note that the order the key/value pairs appear in a KeyVal Argument is immaterial.

2.1 Package Options

\texttt{\texttt{\textbackslash showmeta}} The \texttt{cnx} package takes a single option: \texttt{\textbackslash showmeta}. If this is set, then the metadata keys are shown (see [Koh10a] for details and customization options).

2.2 Document Structure

The first set of CNXML$\LaTeX$ environments concern the top-level structure of the modules. The minimal Connexions document in $\LaTeX$ can be seen in Figure 1: we still need the $\LaTeX$ document environment, then the \texttt{\textbackslash cnxmodule} environment contains the module-specific information as a KeyVal argument with the two keys: \texttt{id} for the module identifier supplied by the Connexions system) and \texttt{name} for the title of the module.

\texttt{\texttt{\textbackslash ccontent}} The \texttt{content} environment delineates the module content from the metadata (see Section 2.6). It is needed to make the conversion to CNXML simpler.

\texttt{\texttt{\textbackslash c*section}} CNXML knows three levels of sectioning, so the CNXML$\LaTeX$ class supplies three
\documentclass{cnx}
\begin{document}
\begin{cnxmodule}[name=Hello World,id=m4711]
\begin{ccontent}
\begin{cpara}[id=p01] Hello World\end{cpara}
\end{ccontent}
\end{cnxmodule}
\end{document}

Example 1: A Minimal CNXL\TeX{} Document

as well: csection, csubsection and csubsubsection. In contrast to regular \LaTeX{}, these are environments to keep the tight connection between the formats. These environments take an optional KeyVal argument with key id for the identifier and a regular argument for the title of the section (to be transformed into the CNXML name element).

cpara, cnote

The lowest levels of the document structure are given by paragraphs and notes. The cpara and cnote environment take a KeyVal argument with the id key for identification, the latter also allows a type key for the note type (an unspecified string).  

2.3 Mathematics

Mathematical formulae are integrated into text via the \LaTeX{} math mode, i.e. wrapped in $ characters or between \( and \) for inline mathematics and wrapped in $$ or between \[ and \] for display-style math. Note that CNXML expects Content MathML as the representation format for mathematical formulae, while run-of-the-mill \LaTeX{} only specifies the presentation (i.e. the two-dimensional layout of formulae). The \LaTeX{}XML converter can usually figure out some of the content MathML from regular \LaTeX{}, in other cases, the author has to specify it e.g. using the infrastructure supplied by the cmathml package.

cequation

For numbered equations, CNXML supplies the equation element, for which CNXL\LaTeX{} provides the equation environment. This environment takes a KeyVal argument with the id key for the (required) identifier.

2.4 Statements

CNXML provides special elements that make various types of claims; we collectively call them statements.

cexample

The cexample environment and definition elements take a KeyVal argument with key id for identification.

crule, statement, proof

In CNXML, the rule element is used to represent a general assertion about the state of the world. The CNXL\LaTeX{} rule\(^5\) environment is its CNXL\LaTeX{} coun-

\(^4\)EdNote: what are good values?
\(^5\)EdNote: we have called this “crule”, since “rule” is already used by \TeX{}.
terpart. It takes a KeyVal attribute with the keys id for identification, type to specify the type of the assertion (e.g. “Theorem”, “Lemma” or “Conjecture”), and name, if the assertion has a title. The body of the crule environment contains the statement of assertion in the statement environment and (optionally) a proof in the proof environment. Both take a KeyVal argument with an id key for identification.

\begin{crule}[id=prop1,type=Proposition]
\begin{statement}[id=prop1s]
Sample statement
\end{statement}
\begin{proof}[id=prop1p]
Your favourite proof
\end{proof}
\end{crule}

Example 2: A Basic crule Example

A definition defines a new technical term or concept for later use. The definition environment takes a KeyVal argument with the keys id for identification and term for the concept (definiendum) defined in this form. The definition text is given in the cmeaning environment\(^1\), which takes a KeyVal argument with key id for identification. After the cmeaning environment, a definition can contain arbitrarily many cexample.

\begin{definition}{term=term-to-be-defined, id=termi-def}
\begin{cmeaning}[id=termi-meaning]
{\term{Term-to-be-defined}} is defined as: Sample meaning
\end{cmeaning}
\end{definition}

Example 3: A Basic definition and cmeaning Example

2.5 Connexions: Links and Cross-References

As the name CONNEXIONS already suggests, links and cross-references are very important for CONNEXIONS modules. CNXML provides three kinds of them. Module links, hyperlinks, and concept references.

Module links are specified by the \cnxn macro, which takes a keyval argument with the keys document, target, and strength. The document key allows to specify the module identifier of the desired module in the repository, if it is empty, then the current module is intended. The target key allows to specify the document fragment. Its value is the respective identifier (given by its id attribute in

\(^1\)we have called this cmeaning, since meaning is already taken by T\(\bar{e}\)X
CNXML or the id key of the corresponding environment in \LaTeX). Finally, the strength key allows to specify the relevance of the link.

The regular argument of the \cnxn macro is used to supply the link text. Hyperlinks can be specified by the \link macro in \LaTeX. It takes a KeyVal argument with the key arc to specify the URL of the link. The regular argument of the \link macro is used to supply the link text.

The \term macro can be used to specify the strength key allows to specify the relevance of the link. The regular argument of the \cnxn macro is used to supply the link text. Hyperlinks can be specified by the \link macro in \LaTeX. It takes a KeyVal argument with the key arc to specify the URL of the link. The regular argument of the \link macro is used to supply the link text.

2.6 Metadata

Metadata is mostly managed by the system in Connexions, so we often do not need to care about it. On the other hand, it influences the system, so if we have work on the module extensively before converting it to CNXML, it may be worth-while specify some of the data in advance.

```
\begin{metadata}
  [version=2.19, \\
\begin{authorlist}
  \cnxauthor[id=miko,firstname=Michael,surname=Kohlhase, \\
  email=m.kohlhase@iu-bremen.de]
\end{authorlist}
\begin{keywordlist}
  \keyword{Hello}
\end{keywordlist}
\begin{cnxabstract}
  A Minimal CNXLaTeX Document
\end{cnxabstract}
\end{metadata}
```

Example 4: Typical CNXL\LaTeX Metadata

The metadata environment takes a KeyVal argument with the keys version, created, and revised with the obvious meanings. The latter keys take ISO 8601 norm representations for dates and times. Concretely, the format is CCYY-MM-DDThh:mm:ss where “CC” represents the century, “YY” the year, “MM” the month, and “DD” the day, preceded by an optional leading “-” sign to indicate a negative number. If the sign is omitted, “+” is assumed. The letter “T” is the date/time separator and “hh”, “mm”, “ss” represent hour, minutes, and seconds respectively.

The lists of authors and maintainers can be specified in the authorlist and maintainerlist environments, which take no arguments.

The entries on this lists are specified by the \cnxauthor and \maintainer macros. Which take a KeyVal argument specifying the individual. The id key is the identifier for the person, the honorific, firstname, other, surname, and lineage keys are used to specify the various name parts, and the email key is used to specify the e-mail address of the person.

The keywords are specified with a list of keyword macros, which take the

\footnote{EdNote: continue, pending Chuck’s investigation.}
respective keyword in their only argument, inside a \texttt{keyword} environment. Neither take any KeyVal arguments.

\texttt{cnxabstract}  The abstract of a \texttt{Connexions} module is considered to be part of the metadata. It is specified using the \texttt{cnxabstract} environment. It does not take any arguments.

\subsection*{2.7 Exercises}

\texttt{cexercise, cproblem, csolution} An exercise or problem in \texttt{Connexions} is specified by the \texttt{cexercise} environment, which takes an optional keyval argument with the keys \texttt{id} and \texttt{name}. It must contain a \texttt{cproblem} environment for the problem statement and a (possibly) empty set of \texttt{csolution} environments. Both of these take an optional keyval argument with the key \texttt{id}.

\subsection*{2.8 Graphics, etc.}

\texttt{cfigure} For graphics we will use the \texttt{cfigure}\footnote{\texttt{EdNote: probably better call it \texttt{cgraphics}}} macro, which provides a non-floating environment for including graphics into CNXML files. \texttt{cfigure} takes three arguments first an optional CNXML keys, then the keys of the \texttt{graphicx} package in a regular argument (leave that empty if you don’t have any) and finally a path. So

\begin{verbatim}
\cfigure[id=foo,type=image/jpeg,caption=The first FOO]{width=7cm,height=2cm}{../images/foo}
\end{verbatim}

Would include a graphic from the file at the path 	exttt{../images/foo}, equip this image with a caption, and tell \texttt{BTeXML} that\footnote{\texttt{EdNote: err, exactly what does it tell latexml?}} the original of the images has the MIME type \texttt{image/jpeg}.

\section*{3 Limitations}

In this section we document known limitations. If you want to help alleviate them, please feel free to contact the package author. Some of them are currently discussed in the \texttt{SI\TeX} TRAC \cite{Ste}.

1. none reported yet
4 The Implementation

The cnx package generates to files: the \LaTeX{} package (all the code between ⟨*package⟩ and ⟨/package⟩) and the \LaTeX{}XML 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.

4.1 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). First we have the general options

\begin{verbatim}
\DeclareOption{*package}{\PassOptionsToPackage{\CurrentOption}{metakeys}}
\DeclareOption{*ltxml}{\PassOptionsToPackage{\CurrentOption}{omdoc}}
\end{verbatim}

Finally, we need to declare the end of the option declaration section to \LaTeX{}.

\begin{verbatim}
\ProcessOptions
\end{verbatim}

We first make sure that the sref [Koh10b] and graphicx packages are loaded.

\begin{verbatim}
\RequirePackage{sref}
\RequirePackage{graphicx}
\end{verbatim}

The next step is to declare (a few) class options that handle the paper size; this is useful for printing.

\begin{verbatim}
\DeclareOption{letterpaper}{\setlength{\paperheight}{11in}\setlength{\paperwidth}{8.5in}}
\DeclareOption{a4paper}{\setlength{\paperheight}{297mm}\setlength{\paperwidth}{210mm}}
\ExecuteOptions{letterpaper}
\end{verbatim}

Finally, we input all the usual size settings. There is no sense to use something else, and we initialize the page numbering counter and tell it to output the numbers in arabic numerals (otherwise label and reference do not work).

\begin{verbatim}
\input{size10.clo}
\pagenumbering{roman}
\end{verbatim}

Now comes the equivalent for \LaTeX{}XML: this is something that we will have throughout this document. Every part of the \TeX{}/\LaTeX{} implementation has a \LaTeX{}XML equivalent. We keep them together to ensure that they do not get out of sync.

\begin{verbatim}
# -*- CPERL -*-
package LaTeXML::Package::Pool;
\end{verbatim}
use strict;

use LaTeXML::Package;

RequirePackage('metakeys');

We set up the necessary namespaces, the first one is the default one for CNXML

RegisterNamespace('cnx' => "http://cnx.rice.edu/cnxml);

RegisterNamespace('md' => "http://cnx.rice.edu/mdml/0.4");

RegisterNamespace('bib' => "http://bibtextml.sf.net/");

RegisterNamespace('m' => "http://www.w3.org/1998/Math/MathML");

For LaTeX we also have to set up the correct document type information. The first line gives the root element. The second gives the public identifier for the CNX DTD, then we have its URL, and finally the CNX namespace.

DocType("cnx:document",

"-//CNX//DTD CNXML 0.5 plus LaTeXML//EN",

"../dtd/cnxml+ltxml.dtd",

'&default' => "http://cnx.rice.edu/cnxml",

'md' => "http://cnx.rice.edu/mdml/0.4",

'bib' => "http://bibtextml.sf.net/",

'm' => "http://www.w3.org/1998/Math/MathML",

'ltx' => "http://dlmf.nist.gov/LaTeXML");

And finally, we need to set up the counters for itemization, since we are defining a class file from scratch.

NewCounter('@itemizei', 'document', idprefix=>'I');
NewCounter('@itemizeii', '@itemizei', idprefix=>'I');
NewCounter('@itemizeiii', '@itemizeii', idprefix=>'I');
NewCounter('@itemizeiv', '@itemizeiii', idprefix=>'I');
NewCounter('@itemizev', '@itemizeiv', idprefix=>'I');
NewCounter('@itemizevi', '@itemizev', idprefix=>'I');
NewCounter('enumi', '@itemizei', idprefix=>'i');
NewCounter('enumii', '@itemizeii', idprefix=>'i');
NewCounter('enumiii', '@itemizeiii', idprefix=>'i');
NewCounter('enumiv', '@itemizeiv', idprefix=>'i');
NewCounter('enumv', '@itemizev', idprefix=>'i');
NewCounter('enumvi', '@itemizevi', idprefix=>'i');

DefMacro('\theenumi', '\arabic{enumi}');
DefMacro('\theenumii', '\alph{enumii}');
DefMacro('\theenumiii', '\roman{enumiii}');
DefMacro('\theenumiv', '\Alph{enumiv}');

NewCounter('equation', 'document', idprefix=>'E');
DefMacro('\theequation', '\arabic{equation}');
DefMacro('\textwidth','16cm');

And another thing that is now needed:

EdNote: this will have to change, when Bruce updates to the next version (0.6?)
4.2 Document Structure

Now, we start with the document structure markup. The \texttt{cnxmodule} environment does not add anything to the \LaTeX output, it's attributes only show up in the XML. There we have a slight complication: we have to put an \texttt{id} attribute on the \texttt{document} element in CNXML, but we cannot redefine the \texttt{document} environment in \LaTeX. Therefore we specify the information in the \texttt{cnxmodule} environment. This means however that we have to put in on the \texttt{document} element when we are already past this. The solution here is that when we parse the \texttt{cnxmodule} environment, we store the value and put it on the \texttt{document} element when we leave the \texttt{document} environment (thanks for Ioan Sucan for the code).

\begin{verbatim}
\newenvironment{cnxmodule}{\metasetkeys{cnxmodule}{#1}}{}
\end{verbatim}

\begin{verbatim}
\newenvironment{ccontent}{}{}
\end{verbatim}

The \texttt{ccontent} environment is only used for transformation. Its optional \texttt{id} attribute is not taken up in the \LaTeX bindings.

\begin{verbatim}
\newenvironment{ccontent}{}{}
\end{verbatim}
The sectioning environments employ the obvious nested set of counters.

\newcounter{section}
\srefaddidkey{sectioning}{id}
\newenvironment{csection}[2][]{\stepcounter{section}\strut\[1.5ex]\noindent%
\Large\bfseries\arabic{section}.~{#2}\[1.5ex]
\metasetkeys{sectioning}#{1}]{}
\newcounter{subsection}[section]
\newenvironment{csubsection}[2][]{\refstepcounter{subsection}\strut\[1ex]\noindent%
\large\bfseries\arabic{section}\.\arabic{subsection}.~#2\[1ex]}
\metasetkeys{sectioning}#{1}{}
\newcounter{subsubsection}[subsection]
\newenvironment{csubsubsection}[2][]{\refstepcounter{subsubsection}\strut\[.5ex]\noindent
\bfseries\arabic{section}\.\arabic{subsection}\.\arabic{subsubsection}~#2\[.5ex]}
\metasetkeys{sectioning}#{1}{}  

For the \texttt{<cnx:para>} element we have to do some work, since we want them to be numbered. This handling is adapted from Bruce Miller's \texttt{LaTeX.ltxml} numbered.

\newcounter{para}
\srefaddidkey{para}{id}
\newenvironment{cpara}[1][]{\metasetkeys{para}#{1}\par}  

\begin{Verbatim}
cpara
\texttt{<cnx:section>}
\texttt{\newcounter{section}}
\texttt{\srefaddidkey{sectioning}{id}}
\texttt{\newenvironment{csection}[2][]{\stepcounter{section}\strut\[1.5ex]\noindent%
\Large\bfseries\arabic{section}.~{#2}\[1.5ex]
\metasetkeys{sectioning}#{1}]}{}
\texttt{\newcounter{subsection}[section]}
\texttt{\newenvironment{csubsection}[2][]{\refstepcounter{subsection}\strut\[1ex]\noindent%
\large\bfseries\arabic{section}\.\arabic{subsection}.~#2\[1ex]}}
\texttt{\metasetkeys{sectioning}#{1}{}}
\texttt{\newcounter{subsubsection}[subsection]}
\texttt{\newenvironment{csubsubsection}[2][]{\refstepcounter{subsubsection}\strut\[.5ex]\noindent
\bfseries\arabic{section}\.\arabic{subsection}\.\arabic{subsubsection}~#2\[.5ex]}}
\texttt{\metasetkeys{sectioning}#{1}{}  

\texttt{\newcounter{para}}
\texttt{\srefaddidkey{para}{id}}
\texttt{\newenvironment{cpara}[1][]{\metasetkeys{para}#{1}\par}}

\begin{Verbatim}
cpara
\texttt{\newcounter{section}}
\texttt{\srefaddidkey{sectioning}{id}}
\texttt{\newenvironment{csection}[2][]{\stepcounter{section}\strut\[1.5ex]\noindent%
\Large\bfseries\arabic{section}.~{#2}\[1.5ex]
\metasetkeys{sectioning}#{1}]}{}
\texttt{\newcounter{subsection}[section]}
\texttt{\newenvironment{csubsection}[2][]{\refstepcounter{subsection}\strut\[1ex]\noindent%
\large\bfseries\arabic{section}\.\arabic{subsection}.~#2\[1ex]}}
\texttt{\metasetkeys{sectioning}#{1}{}}
\texttt{\newcounter{subsubsection}[subsection]}
\texttt{\newenvironment{csubsubsection}[2][]{\refstepcounter{subsubsection}\strut\[.5ex]\noindent
\bfseries\arabic{section}\.\arabic{subsection}\.\arabic{subsubsection}~#2\[.5ex]}}
\texttt{\metasetkeys{sectioning}#{1}{}  

\texttt{\newcounter{para}}
\texttt{\srefaddidkey{para}{id}}
\texttt{\newenvironment{cpara}[1][]{\metasetkeys{para}#{1}\par}}
\end{Verbatim}
my(@parents)=$document->findnodes('ancestor::*[@id]',$node); # find 1st id'd parent.

my $prefix=(@parents?$parents[ $#parents]->getAttribute('id').
."":"'');

# Get the previous number within parent; Worried about intervening elements around para's, but...

my(@siblings)=$document->findnodes("preceding-sibling::cnx:para",$node);

my $n=1;

$n = $1+1 if(@siblings && $siblings[ $#siblings]->getAttribute('id')=~/\d+/);

$node->setAttribute(id=>$prefix."p".$n);

Tag('cnx:para',afterOpen=>&number_para);

DefConstructor('\par',sub { $_[0]->maybeCloseElement('cnx:para'); },alias="$\par\n");

Tag('cnx:para', autoClose=>1, autoOpen=>1);

4.3 Mathematics

cenvironment

4.4 Rich Text

In this section, we redefine some \LaTeX\ commands that have their counterparts in CNXML.
We redefine the abbreviate display math environment and the `eqnarray` and `eqnarray*` environments to use the CNXML equation tags, everything else stays the same.

EdNote: Check LaTeX.ltxml frequently and try to keep in sync, it would be good, if the code in LaTeX.ltxml could be modularized, so that the cnx/ltx namespace differences could be relegated to config options.
displaymath We redefine the abbreviate display math environment to use the CNXML equation tags, everything else stays the same.¹¹

¹¹EdNote: check LaTeX.ltxml frequently and try to keep in sync, it would be good, if the code in LaTeX.ltxml could be modularized, so that the cnx/ltx namespace differences could be relegated to config options.
The next set of commands and environments are largely presentational, so we just skip them.

4.5 Statements

cexample

```
\newenvironment{cexample}{\metasetkeys{example}{#1}
\ifx\example@name\@empty\else
\noindent\bfseries{\example@name}\fi}{}
```

```
\def\Semiverbatim{\lookupvalue{LABEL@#1}}
```
The cexercise, cproblem and csolution environments are very simple to set up for \LaTeX. For the LATEXML side, we simplify matters considerably for the moment by restricting the possibilities we have on the CNXML side: We assume that the content is just one <cnx:para> element for the <cnx:problem> and <cnx:solution> elements.\footnote{EdNote: relax this when we have automated the generation of cnx:para elements}

crule

\begin{code}
\begin{verbatim}
\srefaddidkey{cexercise}
\addmetakey{cexercise}{name}
\newenvironment{cexercise}[1][]{
\begin{active}
\metasetkeys{cexercise}{#1}
\end{active}\begin{active}
\ifx\cexercise@name\@empty
\else
\if@twocolumn
\setcounter{cexercise}{0}
\fi
\stepcounter{cexercise}
\noindent\bfseries{\cexercise@name~\arabic{cexercise}}\fi
\end{active}\begin{active}
\end{active}
\srefaddidkey{cproblem}
\newenvironment{cproblem}[1][]{
\begin{active}
\metasetkeys{cproblem}{#1}
\end{active}\begin{active}
\end{active}
\srefaddidkey{csolution}
\newenvironment{csolution}[1][]{
\begin{active}
\metasetkeys{csolution}{#1}
\end{active}\begin{active}
\end{active}
\end{code}
DefKeyVal('rule','id','Semiverbatim');
DefKeyVal('rule','name','Semiverbatim');
DefKeyVal('rule','type','Semiverbatim');
DefEnvironment('{crule}OptionalKeyVals:rule',
  "<cnx:rule ?&defined(&KeyVal(#1,'id'))(id='&KeyVal(#1,'id')')() type='&KeyVal(#1,'type')'>
"&defined(&KeyVal(#1,'name'))(<cnx:name>&KeyVal(#1,'name')</cnx:name>)()
"\n</cnx:rule>\n");

\srefaddidkey{statement}
\newenvironment{statement}[1][]{
\metasetkeys{statement}{#1}}{}
\end{environment}
\srefaddidkey{proof}
\newenvironment{proof}[1][]{
\metasetkeys{proof}{#1}}{}
\end{environment}
\srefaddidkey{definition}
\addmetakey{definition}{term}
\addmetakey{definition}{seealso}
\newenvironment{definition}[1][]{
\metasetkeys{definition}{#1}{\noindent\bfseries{Definition:}}}
\end{environment}
cmeaning
378 ⟨∗cls⟩
379 \srefaddidkey{meaning}
380 \newenvironment{cmeaning}[1][]\{\metasetkeys{meaning}{#1}\}\{}
381 ⟨/cls⟩
382 ⟨/ltxml⟩
383 DefKeyVal('meaning', 'id', 'Semiverbatim');
384 DefEnvironment('{cmeaning}OptionalKeyVals:meaning', '<cnx:meaning %&KeyVals(#1)>#body</cnx:meaning>');
385 ⟨/ltxml⟩

4.6 Conexxions
cnxn
386 ⟨∗cls⟩
387 \addmetakey{cnxn}{document}
388 \addmetakey{cnxn}{target}
389 \addmetakey{cnxn}{strength}
390 \newcommand{\cnxn}[2][]\{#keys, \text{link text}
391 \metasetkeys{cnxn}{#1}\underline{\footnote{\ttfamily\ifx\cnxn\@empty\cnxn\fi\#\cnxn\@target}}\}
392 ⟨/cls⟩
393 ⟨/ltxml⟩
394 DefKeyVal('cnxn', 'document', 'Semiverbatim');
395 DefKeyVal('cnxn', 'target', 'Semiverbatim');
396 DefKeyVal('cnxn', 'strength', 'Semiverbatim');
397 DefConstructor('{\cnxn OptionalKeyVals:cnxn {}','<cnx:cnxn %&KeyVals(#1)\#1</cnx:cnxn>');
398 ⟨/ltxml⟩

link
400 ⟨∗cls⟩
401 \addmetakey{link}{src}
402 \newcommand{\link}[2][]\{#keys, \text{link text}
403 \metasetkeys{link}{#1}\underline{\footnote{\ttfamily\ifx\link\@empty\link\fi\#\link\@target}}\}
404 ⟨/cls⟩
405 ⟨/ltxml⟩
406 DefKeyVal('link', 'src', 'Semiverbatim');
407 DefConstructor('{\link OptionalKeyVals:link {}','<cnx:link %&KeyVals(#1)\#2</cnx:link>');
408 ⟨/ltxml⟩

cfigure The cfigure only gives us one of the possible instances of the <figure> element. In \LaTeX, we just pipe the size information through to \texttt{includegraphics}, in \LaTeXXML, we construct the CNXML structure in CSS?

\footnotesize
\begin{itemize}
\item\texttt{EdNote:13}\extend that
\item\texttt{EdNote:14}\do more about required and optional keys in arguments.
\item\texttt{EdNote:15}\what do we do with the \texttt{graphicx} information about size,... CSS?
\end{itemize}
\begin{center}
\includegraphics{#2}{#3}
\end{center}

\begin{itemize}
\item \textbf{ccite}
\item \textbf{term}
\end{itemize}

\section{4.7 Metadata}
\addmetakey{auth}{surname}
\addmetakey{auth}{lineage}
\addmetakey{auth}{email}
\newcommand{\cnxauthor}[1]\{\metasetkeys{auth}{#1}\auth@first~\auth@sur,\}
\langle\cls\rangle
\langle\ltxml\rangle
\DefKeyVal{\'auth',\'id',\'Semiverbatim'};
\DefKeyVal{\'auth',\'firstname',\'Semiverbatim'};
\DefKeyVal{\'auth',\'surname',\'Semiverbatim'};
\DefKeyVal{\'auth',\'email',\'Semiverbatim'};
\DefConstructor{\cnxauthor OptionalKeyVals:auth',
  "\<md:author id='&KeyVal(#1,\'id\')'>
  . "?&defined(&KeyVal(#1,\'honorific\'))(<md:honorific>&KeyVal(#1,\'honorific\')<md:author>
  . "?&defined(&KeyVal(#1,\'firstname\'))(<md:firstname>&KeyVal(#1,\'firstname\')<md:author>
  . "?&defined(&KeyVal(#1,\'other\'))(<md:other>&KeyVal(#1,\'other\')<md:author>
  . "?&defined(&KeyVal(#1,\'surname\'))(<md:surname>&KeyVal(#1,\'surname\')<md:author>
  . "?&defined(&KeyVal(#1,\'lineage\'))(<md:lineage>&KeyVal(#1,\'lineage\')<md:author>
  . "?&defined(&KeyVal(#1,\'email\'))(<md:email>&KeyVal(#1,\'email\')<md:author>
  . "</md:author>\n";
\langle\ltxml\rangle
\langle\cls\rangle
\langle\ltxml\rangle
\DefConstructor{\maintainer OptionalKeyVals:auth',
  "\<md:maintainer id='&KeyVal(#1,\'id\')'>
  . "?&defined(&KeyVal(#1,\'honorific\'))(<md:honorific>&KeyVal(#1,\'honorific\')<md:maintainer>
  . "?&defined(&KeyVal(#1,\'firstname\'))(<md:firstname>&KeyVal(#1,\'firstname\')<md:maintainer>
  . "?&defined(&KeyVal(#1,\'other\'))(<md:other>&KeyVal(#1,\'other\')<md:maintainer>
  . "?&defined(&KeyVal(#1,\'surname\'))(<md:surname>&KeyVal(#1,\'surname\')<md:maintainer>
  . "?&defined(&KeyVal(#1,\'lineage\'))(<md:lineage>&KeyVal(#1,\'lineage\')<md:maintainer>
  . "?&defined(&KeyVal(#1,\'email\'))(<md:email>&KeyVal(#1,\'email\')<md:maintainer>
  . "</md:maintainer>\n";
\langle\ltxml\rangle
\langle\cls\rangle
\langle\ltxml\rangle
\DefEnvironment{\{keywordlist\}}{"\<md:keywordlist>\#body\</md:keywordlist>"};
\langle\ltxml\rangle
\langle\cls\rangle
\langle\ltxml\rangle
\DefConstructor{\keyword OptionalKeyVals:keyword',
  \newenvironment{keywordlist}{\bfseries{Keywords}:~}{\[1ex\]};
\langle\ltxml\rangle
\langle\cls\rangle
\langle\ltxml\rangle
\DefEnvironment{\{keywordlist\}}{"\<md:keywordlist>\#body\n\</md:keywordlist>"};
\langle\ltxml\rangle
\langle\cls\rangle
536 (*ltxml)
537 DefConstructor('\keyword {}','<md:keyword>#1</md:keyword>");
538 (/ltxml)

\begin{cnxabstract}
\par
\noindent\hfill
\begin{minipage}{10cm}{\bfseries{Abstract}:~}
\end{minipage}\hfill
\end{cnxabstract}

\begin{DefEnvironment}{\{cnxabstract\}}\begin{OptionalKeyVals}\{cnxabstract\}\end{OptionalKeyVals}
\begin{mdabstract}
#body
\end{mdabstract}
\end{cnxabstract}
Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

authorlist, maintainerlist=authorlist, maintainerlist (environment), 6
c*section=c*section (environment), 3
ccontent=ccontent (environment), 3
cequation=cequation (environment), 4
cexample=cexample (environment), 4
cexercise, cproblem, csolution=cexercise, cproblem, csolution (environment), 4
cfigure=\subitem*+\cfigure+, \usage{7}
cnxabstract=cnxabstract (environment), 7
cnxauthor, maintainer=cnxauthor, maintainer (environment), 7
cnxn=\subitem*+\cnxn+, \usage{4}
cpara, cnote=cpara, cnote (environment), 4
crule, statement, proof=crule, statement, 3
cmeaning=definition (environment), 5
definition, cmeaning=definition, cmeaning (environment), 5
environments;\csection=csection (environment), 3
environments;\ccontent=ccontent (environment), 3
environments;\cequation=cequation (environment), 4
environments;\cexample=cexample (environment), 4
environments;\cexercise, \cproblem, \csolution=cexercise, \cproblem, \csolution (environment), 4
environments;\cnxabstract=cnxabstract (environment), 7
environments;\cnxauthor, \maintainer=cnxauthor, maintainer (environment), 4
environments;\cnxn=\subitem*+\cnxn+, \usage{4}
environments;\cpara, \cnote=cpara, cnote (environment), 4
environments;\crule, statement, proof=crule, statement, 3
environments;\cnote=cpara, cnote, 4
environments;\crule, statement, proof=crule, statement, 4
environments;\definition, \cmeaning=definition, \cmeaning, \cdefinition (environment), 4
environments;\keywordlist, \keyword=keywordlist, \keyword (environment), 5
environments;\metadata=metadata (environment), 6
environments;\metadata=metadata (environment), 6
ten=\subitem*+\term+, \usage{6}
References


[Ste] *Semantic Markup for \LaTeX*. Project Homepage. url: \url{http://trac.kwarc.info/sTeX/} (visited on 02/22/2011).