The Intermediate CNXML*

Ricardo Radaelli-Sanchez
Connexions

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License †

Abstract

This is the second installment of the CNXML language tutorials. It is designed to give a more comprehensive look at the CNXML tags and explain some more advanced uses of the language. Includes helpful examples of CNXML tags, extending the module created in The Basic CNXML tutorial.

1 Example

As is often the case in textbooks, authors will include examples in the middle of a chapter or section. For this reason CNXML provides a tag that allows an author to include examples in a document. The example tag has a unique `id` attribute and can contain most tags as children, the first being an optional title[1]. For specifics you should always consult the CNXML Spec[2] .

Example 1

```
<example id='tboneexam'>
  <figure id='tbonefig'>
    <title>T-Bone Steak</title>
     <media id="image-example" display="block" alt="A T-bone Steak.">
       <image type='image/jpg' src='tbone.jpg'/>
     </media>
  </figure>
</example>
```

*Version 2.22: Apr 10, 2009 2:04 pm -0500
†http://creativecommons.org/licenses/by/1.0
[1]"The Basic CNXML": Section Title <http://cnx.org/content/m9000/latest/#name>
[2]http://cnx.rice.edu/technology/cnxml/0.5/spec

2 Figure

The `figure` tag provides the structure for creating a figure within a document. They can contain either two or more subfigure (Section 3: Subfigure) tags, or a single media (p. 2), table[3], or code[4] tag.

The `figure` tag has two attributes:

- `id` - a unique ID, required
- `orient` - defines how multiple subfigure (Section 3: Subfigure)s are to be displayed. It takes two values, `vertical` or `horizontal`, and will default to `horizontal`.

The optional first tag of the `figure` tag is title[5] which is used to title a figure.

The `title` tag is followed by any of the tags listed above; however, the most commonly used tag is media, which is used to include any sort of media such as images, video, music, or java applets. The media object tags have two required attributes:

- `src` - the location of the displayed media
- `mime-type` - defines the type of media being displayed, which can be any valid MIME[6] type.
  - audio - audio/mp3, audio/wav, etc.
  - video - video/qt, video/mov, etc.
  - image - image/png, image/gif, etc.
  - application - application/PostScript, application/x-java-applet, etc.

The final tag is the optional `caption` which is used to add a small caption to the figure.

Example 2

```
<figure id='tbone'>
  <title>T-Bone Steak</title>
  <media id="image-example" display="block" alt="A T-bone Steak.">
     <image mime-type='image/jpeg' src='tbone.jpg'/>
  </media>
  <caption>
    Upon successful completion of these documents, you should be able
    to grill a steak that looks just as good!
  </caption>
</figure>
```

3 Subfigure

The `subfigure` tag is used when you want to include more than one media (p. 2), code[7] or table[8] within the same figure (Section 2: Figure).

[3] "The Advanced CNXML" <http://cnx.org/content/m9007/latest/#table>
[4] "The Advanced CNXML": Section Code <http://cnx.org/content/m9007/latest/#code>
[5] "The Basic CNXML": Section Title <http://cnx.org/content/m9000/latest/#name>
[6] http://www.ietf.org/rfc/rfc1341.txt?number=1341
[7] "The Advanced CNXML": Section Code <http://cnx.org/content/m9007/latest/#code>
[8] "The Advanced CNXML" <http://cnx.org/content/m9007/latest/#table>

The usage of the `subfigure` tag is similar to that of figure (Section 2: Figure). It has an optional `id` attribute, an optional first child title[9] tag, a single media (p. 2), code[10] or table[11], followed by an optional caption (p. 2).

Now the `orient` attribute for figure (Section 2: Figure) becomes very important. `orient` lets you specify whether the subfigures should be displayed side-to-side or one on top of the other.

Example 3

```
<figure orient='horizontal' id='horfig'>
  <title>Steaks</title>
  <subfigure id='subfigtbone1'>
    <title>T-Bone</title>
    <media id="image-example" display="block" alt="A T-bone Steak.">
      <image type='image/jpeg' src='tbone.jpg'/>
    </media>
  </subfigure>
  <subfigure id='subfingnystrip1'>
    <title>New York Strip</title>
    <media id="image-example" display="block" alt="A NY Strip.">
      <image mime-type='image/jpeg' src='ny_strip.gif'/>
    </media>
  </subfigure>
  <caption>
      Upon successful completion of these documents, you
      should be able to grill a steak that looks just as good!
  </caption>
</figure>
```

Or

```
<figure orient='vertical' id='verfig'>
  <title>Steaks</title>
  <subfigure id='subfigtbone2'>
    <title>T-Bone</title>
    <media id="image-example" display="block" alt="A T-bone Steak.">
      <image mime-type='image/jpeg' src='tbone.jpg'/>
    </media>
  </subfigure>
  <subfigure id='subfig2'>
    <title>New York Strip</title>
    <media id="image-example" display="block" alt="A NY Strip.">
      <image mime-type='image/jpeg' src='ny_strip.jpg'/>
    </media>
  </subfigure>
  <caption>
```

[9] "The Basic CNXML": Section Title <http://cnx.org/content/m9000/latest/#name>
[10] "The Advanced CNXML": Section Code <http://cnx.org/content/m9007/latest/#code>
[11] "The Advanced CNXML" <http://cnx.org/content/m9007/latest/#table>

```
    Upon successful completion of these documents, you
    should be able to grill a steak that looks just as good!
  </caption>
</figure>
```

4 List

The `list` tag is used to make lists. It has two attributes:

- `id` - a unique ID, required
- `list-type` - defines the formatting of the list. `list-type` takes the values `bulleted` (default), `enumerated`, `named-item` or `inline`

The `list` tag has two children: title (p. 2), which is optional, and `item`, which is where the list information is stored.

Example 4: Example List

```
<list id='marinade' list-type='enumerated'>
  <title>Beer Marinade</title>
  <item>pour beer into large bowl</item>
  <item>add chili powder to taste</item>
  <item>squeeze half lime into beer marinade</item>
  <item>place steak in beer, let soak for 30 minutes</item>
</list>
```

The resulting list will look like:

Beer Marinade

1. pour beer into large bowl
2. add chili powder to taste
3. squeeze half lime into beer marinade
4. place steak in beer, let soak for 30 minutes

Example 5: New List Types Example

CNXML 0.6 gives you much more control over the list environment. Now you will be able to choose from eight preset bullet styles as well as an option that allows you to choose your own literal text to serve as the bullet style. The enumerated list type now offers several styles, including Arabic numerals, upper and lower case alphabet characters, and also upper and lower case Roman numerals. In addition to these changes, you can now also select to have your lists follow a stepwise progression. In CNXML 0.6 the named-item list has been slightly altered, and is now called a labeled-item list. As you may have guessed, the change is quite intuitive. Instead of using <name> to specify the item's label, you use <label>.

Here is an example of a stepwise enumerate list:

```
<list id="eip-165" list-type="enumerated" number-style="arabic" class="stepwise">
<title>Beer Marinade</title>
<item>pour beer into large bowl</item>
<item>add chili powder to taste</item>
<item>squeeze half lime into beer marinade</item>
<item>place steak in beer, let soak for 30 minutes</item>
</list>
```

The resulting list will look like:

Beer Marinade

Step 1: pour beer into large bowl
Step 2: add chili powder to taste
Step 3: squeeze half lime into beer marinade
Step 4: place steak in beer, let soak for 30 minutes

5 Equation

The `equation` tag is used to set off and number equations in CNXML documents by using ASCII text, MathML and embedded media (p. 2) to display math.

NOTE: It is strongly encouraged, however, to use equation with MathML[12] tags when displaying math.

5.1 ASCII Text and Images

The first child of equation is an optional title[13] followed by any number of media (p. 2) tags.

Example 6

```
<equation id="eqn14">
  <title>Euler's Relation</title>
  <media id="equation-example" display="block" alt="Euler's Relation.">
     <image mime-type='image/gif' src='euler.gif' />
  </media>
</equation>
```

```
<equation id='eqn15'>
  <title>Simple Arithmetic</title>
    11+27=38
```

[12] "Content MathML" <http://cnx.org/content/m9008/latest/>
[13] "The Basic CNXML": Section Title <http://cnx.org/content/m9000/latest/#name>

```
</equation>
```

This equation will display as:

Simple Arithmetic

$$11+27=38 \tag{1}$$

You could also write this equation using MathML:

```
<equation id="eqn22">
<m:math>
   <m:mn>11</m:mn>
   <m:mo>+</m:mo>
   <m:mn>27</m:mn>
   <m:mo>=</m:mo>
   <m:mn>38</m:mn>
</m:math>
</equation>
```

6 Definition

The `definition` tag is used to define a word in a CNXML document. It has a required `id` attribute and three children: term[14], `meaning` and example (Section 1: Example). How to use definition is a little confusing, so don't forget to check out Example 7.

The first child tag should be term[15] which contains the word/phrase being defined. It is then followed by a `meaning` tag which is followed by any number of example (Section 1: Example)s. This process repeats for all meanings.

Example 7

```
<definition id='tbonedef'>
  <term>T-Bone</term>
  <meaning>
    "The T-bone steak is cut between 1 and 3 inches thick and comes
     from the center section of the short loin. This steak is
     characterized by its T-shape bone, has a fine-grained shell and a
     small tenderloin eye,"
     <cite>http://www.chophousecalgary.com/steak.html</cite>.
  </meaning>
  <example id='tboneexam'>
    <figure id='tbonefig'>
      <title>T-Bone Steak</title>
        <media id="image-example" display="block" alt="A T-bone Steak.">
          <image mime-type='image/jpeg' src='tbone.jpg'/>
```

[14] "The Basic CNXML": Section Term <http://cnx.org/content/m9000/latest/#term>
[15] "The Basic CNXML": Section Term <http://cnx.org/content/m9000/latest/#term>

```
        </media>
      </media>
    </figure>
  </example>
</definition>
```

Definition 1: T-Bone
"The T-bone steak is cut between 1 and 3 inches thick and comes from the center section of the short loin. This steak is characterized by its T-shape bone, has a fine-grained shell and a small tenderloin eye," *http://www.chophousecalgary.com/steak.html*.
Example

T-Bone Steak

Figure 1

7 Rule

The `rule` tag is used to insert a rule, such as a theorem, axiom, or rule of thumb, into a cnxml document. It has two attributes:

- `id` - required, unique ID

- `type` - required, specificies the type of rule (e.g. theorem, axiom, rule of thumb, etc.)

It may also have an optional title[16] and it must have one or more statement (Section 7.1: Statement) tags and zero or more proof (Section 7.2: Proof) or example (Section 1: Example) tags.

7.1 Statement

The `statement` tag is used inside a `rule` tag and defines the statement of the rule. It has an optional `id` attribute, which, like all IDs, must be unique. It also has two children, para[17] and equation (Section 5: Equation).

7.2 Proof

The `proof` tag is used inside the `rule` tag and marks the proof of the rule. It has an optional `id` attribute and may contain another rule, para[18], equation (Section 5: Equation), figure (Section 2: Figure), or list tag.

7.3 Using rule

Example 8

```
<rule id='murph' type='law'>
  <title>Murphy's Law</title>
  <statement>
    <para id='murphp1'>
      If there are two or more ways to do something, and one of those
      ways can result in a catastrophe, then someone will do it.
    </para>
  </statement>
  <proof>
    <para id='murphp2'>
      Edward A. Murphy, Jr. was one of the engineers on the
      rocket-sled experiments that were done by the U.S. Air Force in
      1949 to test human acceleration tolerances (USAF project
      MX981). One experiment involved a set of 16 accelerometers
      mounted to different parts of the subject's body. There were two
      ways each sensor could be glued to its mount, and somebody
      methodically installed all 16 the wrong way around. Murphy then
      made the original form of his pronouncement, which the test
      subject (Major John Paul Stapp) quoted at a news conference a
      few days later
      <cite>http://www.lylemariam.com/murphy.htm</cite>.
    </para>
  </proof>
</rule>
```

[16] "The Basic CNXML": Section Title <http://cnx.org/content/m9000/latest/#name>
[17] "The Basic CNXML": Section Para <http://cnx.org/content/m9000/latest/#para>
[18] "The Basic CNXML": Section Para <http://cnx.org/content/m9000/latest/#para>

Law 1: Murphy's Law
If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.
Proof:
Edward A. Murphy, Jr. was one of the engineers on the rocket-sled experiments that were done by the U.S. Air Force in 1949 to test human acceleration tolerances (USAF project MX981). One experiment involved a set of 16 accelerometers mounted to different parts of the subject's body. There were two ways each sensor could be glued to its mount, and somebody methodically installed all 16 the wrong way around. Murphy then made the original form of his pronouncement, which the test subject (Major John Paul Stapp) quoted at a news conference a few days later *http://www.lylemariam.com/murphy.htm*.

8 Finishing Remarks

Thanks for making it through another tutorial. I'm sure that you still want to know more so here's a link to The Advanced CNXML[19].

[19] "The Advanced CNXML" <http://cnx.org/content/m9007/latest/>