Baking your own pizza

Lou Burnard
Date:

In this exercise, you will use the TEI pizza chef in order to make a special purpose XML DTD. You will need access to the internet and a browser such as IE5, Netscape, or Opera. Once you have your DTD you will also need an XML-aware editor to use it.

Our goal in this exercise is to make a very very simple DTD, which we can use to mark up a multimedia document. We don't need anything like the full complexity of TEI Lite, much less the full TEI. We just want to mark up headings, dates, lists, paragraphs, figures and ... sound clips. Unfortunately, the TEI Guidelines don't seem to have an element specifically for marking up sound clips, so we need to invent it. While we are being imaginative, we will also add a scale attribute to the existing TEI `<figure>` element, which will be used to resize images for display, and an attribute url to simplify the process of embedding images in webpages.

1 Baking your own pizza

- Go to the URL http://www.tei-c.org.uk/pizza.html and read the general discussion of how the Pizza Chef works.
- Choose the modules you want: for this exercise, select the prose base and the figures topping. (This means you should uncheck two of the three toppings that the chef offers you: linking and analysis) If you want to read about a module in detail, you can click on its name to browse the full text of the relevant part of the TEI Guidelines.
- The next section of the pizzachef web page concerns character entity sets; you can safely ignore these for the moment.
- The two modules chosen contain many more elements than we need, so we will create an extension file which ignores most of them. Click on the button which reads `Generate Local Modifications Files (B)` - by default we want to ignore all the elements.
- You will see a list of all the elements now available for inclusion in your DTD. Click on any element name to see full information about it. This will take you to the formal definition for that element within the TEI Guidelines. Use the Back button of your browser to return to the list of available tags. (If the page has expired from the cache, you may need to reload it). Explore the meanings and usage of any elements you are curious about.
- Now look at the three radio buttons to the right of the list of available elements. If the first of these is selected, then that element will be included in your dtd; if the second is selected (as they all currently are), then that element will be excluded from your DTD. The third radio button should be selected for any element which you want to include in your DTD but with some modification.
- For this exercise, you need to *include* the following elements:

  from the Figures module : `<figDesc>`

  from the Core module `<body>`, `<date>`, `<div>`, `<head>`, `<item>`, `<list>`, `<name>`, `<note>`, `<p>`, `<respStmt>`, `<TEI.2>`, `<text>`, and `<title>` (Yes, we know that `<name>` and `<note>` appear twice. It's a bug. Choose the first one in each case.)

  You also need to *modify* the `<figure>` element, as further discussed below.

- When you've finished, click on the Send me the TEI.extensions.ent file button. The pizzaChef will now send you a TEI extension entity file. Depending on your browser, you may see this displayed, or the browser may instead ask if you want to save or execute it. If you see it displayed, choose "Source" from the "View" menu; this will open the file in your default editor from which you can save the file in the usual Windows dialog box.
- Save the file on your local disk with the name edison-extns.ent. (Make sure you specify the file type as 'All files' to prevent Windows helpfully adding ".txt" to the filename. Or enter the filename with quote marks round it).
- You want to use the `<figure>` element in this exercise, but with some extra attributes. You also need to add the declaration for the soundClip element. Go back to the list of available elements (you may need to reload this page) and find the `<figure>` element; click on the third of the three check boxes to indicate that you plan to modify it.
- Click on the Send me the TEI.extensions.dtd file button. The pizzaChef will now send you a TEI extension DTD file, which you should save on your local disk with the name edison-extns.dtd, in the same way as before.
- Open this file using Notepad, or the editor of your choice, i.e. emacs (the easiest way to do this is by using the View Source option of your web browser). You will see that it contains the standard TEI definition for the `<figure>` element, which currently reads

```
<!ELEMENT figure %om.RR; ((%m.Incl;)*, (head,
(%m.Incl;)*)?, (p, (%m.Incl;)*)*, (figDesc, (%m.Incl;)*)?,
(text, (%m.Incl;)*)?)>
 <!ATTLIST figure %a.global; entity
ENTITY #IMPLIED TEIform CDATA 'figure' >
```

  Don't worry about the rebarbative SGML syntax. If you want to simplify the content model, feel free to do so at your own risk. For the purposes of this exercise, all you need to do is to add a couple of lines to the line beginning `<!ATTLIST figure %a.global;` . Using the editor, change this declaration to read

```
<!ATTLIST figure %a.global;
                entity ENTITY #IMPLIED
                url CDATA #IMPLIED
                scale CDATA "100">
```

  and save the file.
- Since we also want to add a new element to the DTD, not defined by the TEI, its definition must be added to this same extensions.dtd file. Type in the following declaration for the new element, and save the DTD file again

```
<!ELEMENT soundClip %om.RR; (#PCDATA|date|name)* >
<!ATTLIST soundClip %a.global;
                   entity ENTITY #IMPLIED
                   url CDATA #IMPLIED
                   duration CDATA #REQUIRED >
```

- We are not quite finished. Simply defining a new element does not include it in the content model of any other element: the easiest way of doing that is to include it in one of the TEI model classes, which involves modifying the corresponding parameter entity in the extensions entity file. Open the edison-extns.ent file with your editor.
- At present, this file is composed of `"IGNORE"` declarations for the elements we are not using from our chosen modules. To add the `<soundClip>` element to the data class, simply type

```
<!ENTITY % x.data  "soundClip|">
```

  at the start of the file. (The space after the percent sign, and the vertical bar are both essential!). Save the file
- We've prepared a little test program which you can use to check that you've made your edits correctly: look in your folder, and you will see a file called edison-test.xml. Open this file with emacs and select VALIDATE from the SGML/XML menu (or type CTRL-C CTRL-V). Press RETURN twice. All being well, you should see only the following error message: `no document element` (since you have not included anything but the DTD in this file). If you get any other message, your extensions files are incorrect. The error message will at least tell you the line number where you went wrong, if you don't undertand anything else about it!

- Once you're happy with your modification files, it's time to send them to the pizzachef. Open the browser window again. Press the Back button twice to go back to the main pizzachef page, and scroll down to the point where you are invited to specify the names of your modification files. Use the browse button to locate first the TEI.extensions.ent file (edison-extn.ent) and next the TEI.extensions.dtd file (edison-extn.dtd).
- Now press the `Generate full DTD button`. The pizzachef will now send you another file, which contains your compiled XML dtd. Save it under the name myPizza.dtd. Congratulations! You have made a valid XML dtd! (If you saw Zombies, then you missed out some vital elements from your edison-extn.ent)
- You may wonder why your DTD contains more elements than you specified above. The extra elements are needed for the TEI Header which is a mandatory component of every TEI document. The current version of the pizza chef does not allow you to modify the Header in any way.

2 Using your new DTD

You can use your new DTD with any XML editor. In this tutorial you will create an XML file using xMetal, and then edit it with emacs (or the other way round if you prefer!).

- Locate the Xmetal icon on the desktop and click on it
- Depending on what was happening last time it was used, Xmetal will open either some other file, or a brief tutorial file. In either case, choose New from the File menu.
- Choose Create Blank XML document from the dialog box
- A dialogue box opens for you to select a Rules file or DTD. Navigate to the desktop (or wherever you saved your DTD file) and select it. If the DTD is opened successfully, you will be asked about "Preserve Whitespace Options": press RETURN to continue.
- From the Insert menu, select Element. A list of available elements in your DTD appears in the righthand window. Choose TEI.2 (this should be the default)
- Xmetal displays your document in one of four different formats: Plain Text, Tags On, Normal, or Page Preview. You can choose which format to use from the View menu, or by clicking on the icons at the bottom left of the text panel. Select Tags On for now.
- As you see, several of the tags for a TEI document have already been inserted for you. Supply a title, a publication statement, and a source description.
- Note how, as you move the cursor to different positions in the document, different elements are available for insertion. Insert enough elements for your document to be valid and complete.
- If you want to go on working with Xmetal, you may like to set formatting properties for some of the elements, using the Quick Styles option on the Format menu.
- When you're ready, save the file as exercise.xml; if you have introduced XML errors, Xmetal will tell you so. You need to fix these before you can proceed.

Alternatively, you can use emacs to edit your new file, using this DTD. We don't give a full description of using emacs here; just enough to get you started with the file you created.

- Locate the emacs icon on the desktop and double-click on it
- Select Open from the File menu. Using the "Find file" dialog, navigate to the location where you saved your exercise.xml, and open it.
- When you saved the file from Xmetal, it specified the location of the DTD in the `Doctype` statement at the start of the file. For emacs to use this information however, you need to parse the DTD. This is accomplished by selecting Parse DTD from the DTD menu.
- Now, when you position the cursor at any point in your document and select Insert Element from the Markup menu, you will see only the elements which are legal at that point, as with Xmetal.
- To validate your file inside emacs, choose Validate from the SGML/XML menu.

There is a fuller description of how to use emacs in this way for the Master DTD at http://www.tei-c.org.uk/Master/Workshops/master-emacs.html

3 And now...

Your final challenge is to make an XML document which includes a picture and a soundclip, and then to transform it to HTML for display on a website.

We have provided a couple of example entities for your use at the URLs http://www.tei-c.org.uk/Talks/edison.jpg and http://www.tei-c.org.uk/Talks/edison.mp3 respectively. One shows a photograph of Edison annotated by the great man himself: the story goes that this was found only slightly charred after a fire which destroyed Edison's original factory in New Jersey. The sound clip is the famous *Mary had a little lamb* recording, as recounted by Edison in a recording made in 1927.

With your new DTD, you can now directly reference these entities in your document, using a `<figure>` and a `<soundClip>` element respectively:

```
<p>...<figure
      url="http://www.tei-c.org.uk/Talks/edison.jpg">
    <figDesc>Photograph of Edison
     annotated by himself</figDesc></figure>
...<soundClip
     url="http://www.tei-c.org.uk/Talks/edison.mp3" duration="20">
      Edison reminisces about his first
      phonograph recording, <date>1887</date></soundClip>
     </p>
```

Once your document is valid, you can transform it into an HTML web page by using an XSLT stylesheet. We have prepared a suitable stylesheet for this purpose in the file edison.xsl. You can invoke this by adding a a stylesheet reference like the following: `<?xml-stylesheet type="text/xsl" href="edison.xsl"?>` at the start of your file, after the DOCTYPE declaration. Some browsers (e.g. IE6) may render this XML directly, but a more reliable method is to run a free-standing XSLT processor such as saxon or xsltproc to generate a static HTML page from your document. Try typing

```
xsltproc -o edison.html edison.xsl edison.xml
```

at the command prompt. This will generate an HTML file called edison.html which any web browser should be able to display. What actually happens when the user clicks on the sound clip link will, of course, depend on how their browser is configured ... but that is another story.