

# 1. Baking your own pizza

In this exercise, you will use the TEI pizza chef in order to make a special purpose XML dtd. You will need access to an internet browser such as IE5, Netscape, or Opera, and some experience of using Windows.

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.

- Go to the URL <http://www.hcu.ox.ac.uk/TEI/pizza.html> and read the general discussion of how the Pizza Chef works.
- Choose the tagsets 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 tagset 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 tagsets 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 tags now available for inclusion in your DTD. Click on any element name to see full information about it. Check the first of the three tick boxes to the right if you want to include the element in your dtd. For this exercise, we will need only the following elements: `<body>`, `<date>`, `<div>`, `<head>`, `<item>`, `<list>`, `<note>`, `<p>`, `<respStmt>`, `<TEI.2>`, `<text>`, and `<title>`. In addition, we will need the `<figDesc>` element from the tagset for figures and tables.
- 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 settings, you may see this displayed, or the browser may ask if you want to save it. Either way, make sure you do NOT save it as an HTML file. (If you do, you will have to edit the HTML header and footer out of the file before you can use it). If possible, save it on your local disk with the name `exercise.ent`, though some browsers may insist on naming the file `exercise_ent.txt`. Use the BACK button to return to the browser window (if it has expired from the cache you may have to reload it)
- Although we want to use the `<figure>` element in this exercise, we want to simplify its content model and add an attribute. Find this element in the list and 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 `exercise.dtd`, in the same way as before
- Minimize the web browser window, and open the second of the two files you have just received from the pizzachef using Wordpad, or the editor of your choice, i.e. emacs. You will see that it contains the standard TEI definition for the `<figure>`

element, which currently reads

```
<!ELEMENT figure - - (head?, p*, figDesc?, text?)>
<!ATTLIST figure %a.global entity ENTITY #IMPLIED>
```

Using the editor, change this declaration to read

```
<!ELEMENT figure - - (head?, figDesc)>
<!ATTLIST figure %a.global entity ENTITY #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 - o (date) >
<!ATTLIST soundClip %a.global;
                entity ENTITY #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.ent file. Open the `exercise.ent` file with your editor.
- As you see, this file is mostly composed of "IGNORE" declarations for the elements we are not using from our chosen tagsets. 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
- Once you're happy with your extensions file, it's time to send them to the pizzachef. Open the browser window again. Press the Back button to go back to the main pizzachef page, and scroll down to the point where you are invited to specify the names of your extension files. Use the browse button to locate first the extensions.ent file and next the extensions.dtd file.
- Scroll down to Step 5, near the end of the page. Select option `b XML with no tag omissibility information`. 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!
- Start emacs and open a new XML file. Type in the following two lines at the start:

```
<?xml version="1.0"?>
<!DOCTYPE TEI.2 SYSTEM "myPizza.dtd">
```

Go to the DTD menu and select the first option `compile DTD`. You should now find that the Markup menu will offer you only elements from your new DTD.

- You may wonder why your DTD contains more elements than you specified above. Because the TEI Header is a mandatory component of every TEI document, the current version of the pizza chef does not allow you to modify

it in any way. However, you should find that once you start entering material into the <body> of your document, the number of elements available to you is drastically reduced. Welcome to the wonderful world of document standards!