TEI XML : Practical Exercises

Lou Burnard *Date:*

Before you start

These exercises assume you will be using the TEI-Knoppix CD, release 3.6 or later. Unless you are using this TEI-Knoppix CD, you will need access to the net and a web browser such as Firefox or Opera. You will also need an XML-aware editor (in our case, Gnu Emacs).

If you are using the Knoppix CD, you will need to do the following before you start:

- Put the CD into the CD drive of your computer and reboot the computer. All being well, your computer will do the right thing (you may need to adjust the BIOS to make it reboot from the CD rather than the hard disk or network).
- The TEI-Knoppix CD has to check out all the components of your system and load the appropriate drivers before it can start loading the application software bundled with it. This can take five minutes or so, depending on your system specification. Just hit the RETURN key if it pauses during this process. When you see the OSSwatch web page, the startup is complete.
- This version of Knoppix starts up with a UK keyboard. If you want a different one, click on the national flag in the bottom right corner. If the flag you want doesn't appear for selection, right click with your mouse to open the "Add mapping" dialog, which will allow you to add it.

The example files used in these exercises are all available in a folder called samples on your ramdisk, if you are using the Knoppix CD. If not, you can download them from the website associated with this tutorial on the TEI website.

1 Editing an XML file

There are literally dozens of different editors you can choose from to work with XML. For this course we will use the latest incarnation of a long-established favourite: GNU Emacs. The software is an attractive option for those wishing to produce high quality XML-encoded documents on a limited (or nonexistent) budget because it is genuinely ubiquitous, very widely understood, documented in great detail, and comes at a price that every academic project can afford (i.e. zero). It may not be as flashy as some XML editors, but it can still be configured to make entering valid XML text very easy.

One of the distinctive features of emacs is the way it uses keyboard shortcuts for everything. In particular, it sometimes requires you to use the Control (CTRL) key in the way you would normally use the Shift key: you hold it down and hit another key at the same time. In what follows, we represent this by using CTRL to stand for the control key, which is followed by a hyphen if the following key is to be hit simultaneously: for example CTRL-x.

If you want to find out about using emacs the traditional way (from the keyboard) you'll find more help than you can possibly digest on the Help menu. (Try the Tutorial for basics. We will be focussing on XML-specific features in the rest of this tutorial.)

If you find emacs totally unfriendly, don't panic. There are other editors: you could try OxyGen for example, which is also included on the OSS Watch Knoppix CD.

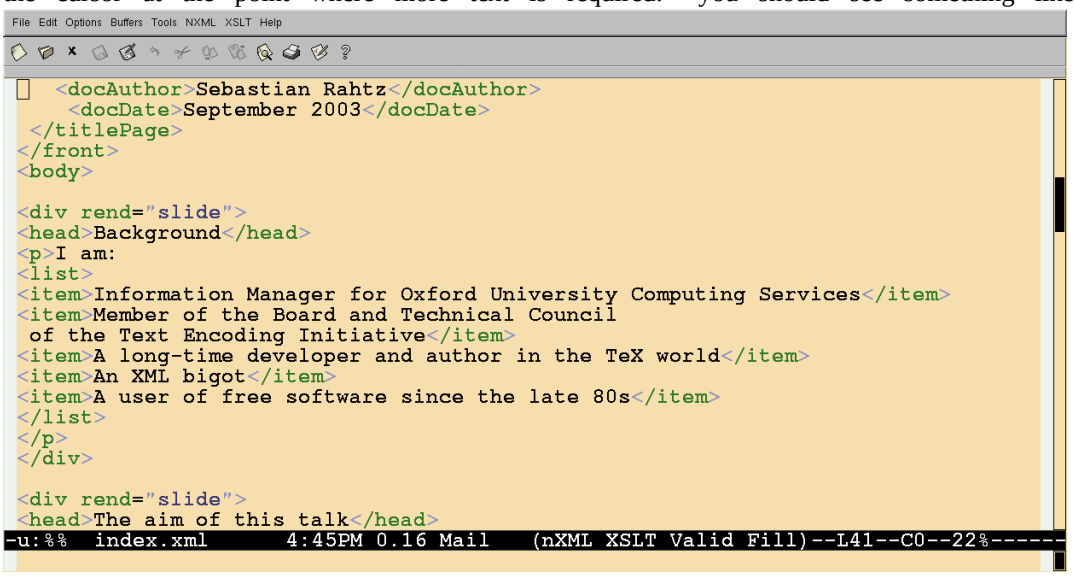
- Start emacs. If there is no Gnu icon on your desktop, you'll find it by clicking the Big K button at bottom left, and selecting Editors from the All Applications menu that opens up.
- The emacs application window opens. Wait for the program to initialize. When you see the emacs splash screen, click on the first button (Open file) on the toolbar (or Select Open File from the File menu).
- The cursor moves to the bottom of the screen, below the status bar, where there is a small area known as the minibuffer. This is the area you and emacs use to send each other messages. At the moment, you need to tell emacs what the name of your new file is to be, and emacs is suggesting a path to it. You must complete the filename and press RETURN before you can continue. Type in a name for your file, making sure that it ends in .xml and then press RETURN

1.1 Building an XML document

TEI-emacs comes with predefined skeleton files which can be used to help you build files conforming to a complex schema. In this exercise, we'll start by encoding some parts of the Punch page.

- The message Using vacuous schema indicates that emacs recognizes that you are trying to create an XML file, but you have not specified a schema for it. Select Insert skeleton TEI files from the TEI menu. Choose TEI P5 from the sub menu that opens.

- Emacs inserts the bare minimum for a TEI conformant document in your window, leaving the cursor at the point where more text is required: you should see something like this:



- This shows the structure of a typical TEI document, with a header at the top, and the body of the document below. XML comments such as 'the main text of the file here' are in italics colours are used to distinguish tags, attributes, and text. But the tags are just as easily edited as the rest of the text, as we will see in a moment.

- The blinking cursor marks the insertion point, and indicates where keyboard input will go. At present, it should be at the point in the document where something more is needed: the red line indicates that the document is invalid. To confirm why it is invalid, use the mouse to click on the red line. The message Required child elements missing appears in the minibuffer. Before fixing that, we suggest you familiarize yourself a bit more with the emacs editing environment.

- Use the mouse or arrow keys to move the insertion point (the cursor position) anywhere in the document, and type or delete a couple of characters. What happens if you do this inside a tag?

- If you make the document invalid, by changing one of the tags, you will see the offending changes are highlighted (underlined in red), the word Invalid will appear in the status line (the line immediately above the minibuffer at the bottom of the screen which includes the name of the file, and your current position within it) and an error message will appear in the minibuffer. In the nXML mode we are using, emacs constantly checks that your document is valid, and warns you when it is not.

- Some parts of the TEI Header have been supplied, using generic text enclosed in square brackets. We suggest you change these to something appropriate: give your text a title such as 'A page from Punch' and an author; supply a publication statement such as 'Unpublished exercise'; and as source description you could specify something like 'emacs exercise for OUCS courses: February 2005'

- To undo the last change you made, select Undo from the Edit menu. If you made more than one change, you can correct it by retyping or deleting (most — but not all — of the keys behave in the way you'd expect). Or you can simply close the file (choose Close on the File menu) and re-open it.

We will now start editing the Punch page in earnest. To make life a bit easier, we've done part of the job for you, but not very well.

- Make sure that the insertion point (cursor) is inside the <body> element. As before, you should see only one red line on the screen.

- Select Insert File from the File menu, type samples/verse.xml into the minibuffer, and press RETURN.

After a short delay, a transcript of the poem on the Punch page appears and the status bar changes to show that the document is now valid. However, validity is not the same as truth! As you will see, if you scroll down a little,we haven't actually done a very good job of tagging this poem: the last 'line' actually contains several lines and stanzas run together. In this part of the exercise we'll try to improve on the tagging.

- Using the mouse or the arrow keys, put the cursor after 'upper lid;' in the third stanza. Type the two characters </. Observe what happens.
- Move the cursor to the start of the next line (before 'So I blew') and type <l>. Observe what happens.
- Now you need to repeat these two steps to add a </l>-tag at the end of this line, and a <l> tag at the start of the next one. Use the short-cut CTRL-e (for end) to move the cursor to the end of the current line. To get to the start of the next line, hit CTRL-f (for forward).
- When you have put the </l> tag at the end of the refrain 'And I still had a fly in my eye', type </ again. Can you explain what happens?
- The sequence </l> closes the current element (the line). If you repeat it, you close the next element up the hierarchy (the line-group). Repeat it again to check you have understood. What tag must you insert at the start of the line 'And then Sir...'?

If you want to see the definition of the tags used in this document, open the file tagging.html using your web browser.

1.2 Defining a macro

Computers are supposed to simplify boring repetitive jobs like typing in tags. Fortunately emacs has a built in macro facility which allows you to reduce the amount of typing needed for repetitive tasks like this. To use it you need to learn some rather arcane emacs keystrokes – but you will find the effort worthwhile.

As noted above, we use CTRL to stand for the control key and the hyphen to show that two keys are to be hit simultaneously rather than one after another. So, for example the sequence CTRL-x (means:

- Hold down the control key
- Hit the x key once; release both keys
- Immediately enter a (on most keyboards this requires you to hold down the shift key, of course.

Defining a simple macro is like making a recording of your keystrokes, which you can then playback as often as you like. In this part of the exercise we'll define a macro which just repeats the sequence you practiced earlier: putting a start-tag at the start of a verse line and an end-tag at its end.

- Put the cursor at the start of the next verse line. ('And then Sir...'). It's important to start in the right place when defining a macro!
- Type CTRL-x (. The message Defining kbd macro... appears in the minibuffer. From now on, whatever you type will be recorded as part of the macro.
- Type <l>. Type CTRL-e to move to the end of the line. Type </ to finish the line, and then CTRL-f to move forward to the start of the next line. It's important that a macro winds up in the right place.
- Type CTRL-x). The message Keyboard macro defined appears in the minibuffer to confirm that your macro is now available.
- Type CTRL-x e to execute your keyboard macro. Did it work as expected? If not, you will need to redefine it, having cleared up any mistakes it introduced...

If your macro works as you expect, you can now use it to help with the rest of the tagging. If typing CTRL-x e once per line seems like a bore, you can tell emacs to repeat the command a specific number of times for you. For example, you could type ESC 4 CTRL-x e to execute your macro 4 times.

1.3 Adding more tags

Some of the verse lines in your example actually take up more than one typographic line. The tag <lb/> could be inserted at the point where a linebreak occurs, if you want to mark this fact. The poem also contains several emphasised words (they are in italic, in the original printed version), which you might want to tag using the <emph> tag.

To find out what tags are valid at any point in the document, you can use the emacs completion feature. Proceed as follows:

- put the cursor in front of the emphatic 'I've' in the refrain of the second stanza and type <
- a message appears in the minibuffer to warn you that <I've> is an Unknown element.

- Hold down the CTRL key and press the RETURN key. A new window opens which (amongst other things) lists the tag names which are valid at this point in your document. The text in the minibuffer changes to read Tag:
- You can select the tag you want by typing its name into the minibuffer. Alternmatively, use the mouse to highlight the one you want (in green) and then click on it. If you click with just the left mouse button you will also need to press RETURN to add the chosen tag to your document; alternatively, click with both left and right mouse buttons simultaneously.
- Don't forget to add the closing > for the tag you have started to add
- Move the cursor to the end of a word with the mouse, or by typing CTRL-right arrow. Type </ to add the appropriate end-tag as before. Now tag the remaining <emph> elements in the document.
- If you want to add linebreaks, remember that this is an empty element: you can type the whole thing in one go: as <lb/>
- If you want to add an attribute to any element, type a space after its name, and press CTRL RETURN as before. If the element has more than one attribute (for example), a list of the available attribute is displayed, for you to select from, as with tags.

1.4 Adding character entities

The poem contains a number of dashes, which have been represented as double hyphens. The dash is a normal Unicode character, but like accented letters and other special punctuation marks is difficult to enter from the keyboard. Emacs has two ways of helping you:

- Put the cursor in front of the two hyphens used to represent a dash in the text. Select the UniChar menu and scan the list of available character names for the one you want: mdash.
- Emacs inserts the correct character for you. You can now delete the two hyphens.
- Now that you know the name for this character, you can also type it in directly. Move to the next two hyphens and this time type the characters —. When you type the semicolon, the required character appears.

The second method is very useful for accented letters. The character names emacs recognizes in this way are the same as the entity names used in HTML and other SGML derivatives: for example, to enter an e with an acute accent, you could type é.

1.5 Global tidying up

While we are looking at semicolons, you may have noticed that this document has redundant whitespace in front of them. We will use this as an excuse to introduce you to one of emacs's more powerful and useful features: the conditional search and replace. Most editors have some kind of search and replace: emacs has several.

- Choose Search from the Edit menu, and then Replace from the submenu which opens from it. Alternatively, type ESC %.
- The cursor moves to the minibuffer, where the prompt Query replace: appears. Type the string you want to replace, in our case ; (space followed by a semicolon), and press RETURN.
- The word with appears in the minibuffer. Type the string you want to see instead, in our case ; (just a plain semicolon please), and press RETURN again.
- If the string you typed in is present in the file, the cursor will move to it, and highlight it. The message in the minibuffer now says Query replacing ; with ; (? for help). If you're sure you want to replace all occurrences in the file, just type !. If you want to replace this one and then move on to check the next, type Y. Type a question mark to see what other options you have.

You may also want to experiment with the cut and paste options on the Edit menu. These work in much the same way as other editors, though few editors have quite so many of them.

1.6 More things to try

Using the tricks you've learned so far to find out what tags are available, you should be able to build up a complete TEI document representing the whole of the Punch page. You will find versions of the other parts of the Punch page called cartoon.xmlplay.xml and paras.xml in the samples directory.

You can add some metadata to your TEI header if you like.

Good luck!

1.7 And finally...

Don’t forget to save your work when you have finished! Use the Save command from the File menu, or the Save As command if you want to save it under a different name. (If you’re working with the Knoppix CD, don’t forget to copy the file from the Ramdisk to some external medium such as a floppy disk or USB key!)