



Introduction 1

The example files used in these exercises are all available in the file tei-workshop.zip on drive T: Before starting work, copy this file to drive D: (your personal directory) and unpack it into a folder called TEI-Workshop.

In this exercise, you will use a tool on the TEI web site to make a XML schema tailored to your needs. You will need access to the net and a browser such as IE, Mozilla, or Opera. Once you have your schema you will also need an XML-aware editor to use it (in our case, Gnu Emacs).

Our goal is to make a very very simple schema, 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. We'll also have to say where it is allowed to appear.

2 Making your own schema

- 1. Open your favourite web browser and go to the URL http://www.tei-c.org.uk/Roma/.
- 2. Choose the modules you want: for this exercise, select the prose base, the figures topping, and the linking topping. (This means you should uncheck one of the three toppings that the system offers you: 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.
- 3. The modules chosen contain many more elements than we need, so we will set things up to ignore most of them. Select Configure elements, excluding them by default from the drop-down menu 'Do you want to change the elements from the additional tagsets in any way?'.
- 4. Tick the check box 'Define some new elements'
- 5. When you have done this, press Submit
- 6. You will see a form allowing you to add a new element, like this:

Roma: add new elements

Base: prose Tagsets: analysis, linking, figures, Output: .rnc Method: P4

New element:

• Name:	
Model Class: class.incl	•
Attribute Class: class.analysis	•
Contents: TEXT	•
My new element ● Description:	

```
Add more elements? 🗌
```

Submit Query

. Unless you feel confident at working out what the possibilities are for yourself, we suggest that that you want:

- (a) an element called 'soundClip'
- (b) as a member of the class 'hqphrase'
- (c) with the attribute class 'xPointer'
- (d) with the content model 'TEXT'
- (e) with some sort of description

When you are done, press Submit

- 7. You will see a list of all the elements now available for inclusion in your schema. 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.
- 8. Now look at the two radio buttons next to each element elements. If the first of these is selected, then that element will be included in your schema; if the second is selected (as they all currently are, except for headers), then that element will be excluded from your schema.
- 9. For this exercise, you need to *include* the following elements:

from the figures module : <figure>, <figDesc>

from the header module : all items are forced to be included, whatever choice you made earlier. Leave these as they are, as the TEI header is a savage beast when cornered

from the structure module <body>, <div>,and <text>

- 10. When you've finished, click on the Submit button. The application will now send you a RelaxNG compact schema. Depending on your browser, you should be asked if you want to save it. Choose the location where you unpacked the samples for today, and save the schema under the name edison.rnc. Look at the result, if you feel strong, or experiment with other options of the web application.
- 11. We've prepared a little test program which you can use to check that you've made your schema correctly: look in your folder, and you will see a file called edison.xml. Saving the schema with the same name will allow Emacs to match it up to the XML file. Open the XML file with Emacs and look at the mode line—does it say Valid? If not, click on the word Invalid, and try to see what went wrong.

3 Making a real file

Your next 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 objects for your use in the sample directory, called edison.jpg and 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 schema, you can now directly reference these objects in your document, using a <figure> and a <soundClip> element respectively. Here is what they might look like:

```
<...<figure url="edison.jpg">
<figDesc>Photograph of Edison
annotated by himself in <date>1887</date></figDesc></figure>
...<soundClip url="edison.mp3">
Edison reminisces about his first
phonograph recording
</soundClip>
```

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 display.xsl. You can invoke this by adding a a stylesheet reference like the following: <?xml-stylesheet type="text/xsl" href="display.xsl"?> at the start of your file. 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 display.xsl edison.xml

at the command prompt (Tools/Shell Command in Emacs). 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.