TEI XML : Using Roma

Lou Burnard *Date:*

1 Making your own schema

Our goal here 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 what kind of element it is and where it is allowed to appear.

1. Open the Roma application. If you are online, you can point your favourite web browser at http://tei.oucs.ox.ac.uk/Roma/, or, if you are working from the CD, just click on the Roma link on the OSS Watch page (or, equivalently, point your web browser at http://localhost/Roma/).
2. The Roma start screen allows you to create a new customization, or to upload an existing customization for further work. We will start from scratch, which is the default option. Press the `Submit` button at bottom right of the screen to continue.

The next and subsequent screens show you a row of buttons for acting on your customization (`Save`, `Customize`, `New`, and `Help`), and a row of buttons for each of the major stages or tasks making up a customization (`Modules`, `Add elements`, `Change classes`, `Language`, `Schema`, and `Documentation`. We won't explore all of these in this exercise. By default the Customize your Customization screen is displayed. This allows you to specify a file name and other details for the schema you're about to make. It also allows you to change the interface language if you wish (you can change this to French if you like). For now, we will accept the defaults. Press the `Modules` button to proceed.

The modules screen shows two lists: on the left are all available TEI modules; on the right are the modules currently selected for your schema. You can add modules from the list on the left, and remove modules from the list on the right, by clicking the appropriate word next to the module you wish to operate on.

1. For this exercise, we will add the `figures` module to the default modules already selected. Click the word `add` next to "Tables Formulae Figures".
2. The modules chosen contain many more elements than we need, so we now need to remove some of them. Click the name of a module in the list of modules in your schema to see a list of the elements this module defines. Start by clicking on the `core` module, then look at the textstructure, and figures modules (Leave the others alone for now).

Each element listed has a name, a radio button indicating whether it is to be included or excluded, a tag name, a description, and a link to a further screen where its attributes are specified. You can toggle inclusion or exclusion of all elements in the list by clicking the appropriate column heading. Click on `Exclude` to remove all elements from the module.

Now work down the list clicking the `Include` button to restore the elements needed for this exercise. When you have finished with each module, press the `Submit` button to update the schema specification you are building. Then press the `back` link to go back to the list of modules to select the next module for customization.

from the core module `<TEI>`, `<date>`, `<graphic>`, `<head>`, `<item>`, `<list>`, `<name>`, `<note>`, `<p>`, `<respStmt>`, `<title>`

from the textstructure module `<body>`, `<div><text>`

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

We now need to define our new element. Click the `Add Elements` button at the top of the screen. The Defining a new element form is displayed. which allows you to enter a name for your element, to specify the classes of which it is a member, its content model, and its description. Unless you feel confident about working out what the possibilities are for yourself, we suggest that you proceed as follows:

1. name the element ‘soundClip’
2. make it a member of the class ‘tei.Incl’
3. make it a member of the attribute class ‘tei.pointer’
4. give it the content model ‘TEXT’
5. supply a description such as ‘references an external audio resource’

Remember to press `Submit` when you have finished (you may have to scroll the screen to the right to see the button).

We are now ready to generate a schema. Click the *Schema* button. You can generate a schema in any of several formats, including RelaxNG, W3C schema, and XML DTD language. Use the default (RelaxNG compact syntax), and press Submit. Your browser may ask whether you want to save or open the generated file: you should save it into your `Work` directory. Look at the result, if you feel strong, or experiment with other options of the web application.

We’ve prepared a little test file which you can use to check that you’ve made your schema correctly. When you open it, however, you will have to tell emacs to use your newly made schema. Proceed as follows.

- Open the file `samples/edison.xml` with emacs. As this is an XML file emacs will open it in NXML mode, by default.
- Choose Set Schema from the XML menu, and select `File...`, the last option on the submenu this opens.
- Supply the name and location of your schema file (`myTei.rnc`, probably) in the minibuffer at the bottom of the screen. Press Return.
- Emacs will ask whether it should save this information to the `schemas.xml` file. Type yes, and press return.

Now check the mode line. Does it say Valid? If it does, try introducing an error in the document. If it does not, click on the word Invalid (or the red line on the screen), and try to see what went wrong (hint: this is not a TEI P5 document: change the root element from `<TEI.2>` to `<TEI xmlns="http://www.tei-c.org/ns/1.0">`).

2 Making a real file

Your next challenge is to transform the test document into one 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. It’s up to you what you actually write, but here is some suggestion for what your page might look like:

```
<div><head>Edison in sound and vision</head><p>Thomas Alva Edison was an inordinately se
 <figure>
    <graphic url="edison.jpg"/>
    <figDesc>Photograph of Edison annotated by himself in <date>1887</date>
     </figDesc>
 </figure>
</p>
<p>Edison’s original phonograph of 1877 worked by incising metallic foil, and was of only
 <soundClip  url="edison.mp3">
      Edison reminisces about his first
      phonograph recording
 </soundClip>
     </p>
</div>
```

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`. We have also prepared a simple CSS stylesheet (`display.css`) which can be used to display your file without first converting it.

To see what CSS can do, add a stylesheet reference like the following: `<?xml-stylesheet type="text/css" href="display.css"?>` at the start of your file, and then try opening the XML file with Firefox.

CSS won't do much for you when you need to transform the tagging of your document: it won't convert the `<figure>` or `<soundClip>` elements into the `<a>` elements you need to display these resources on a web page. To convert elements, you need to use a more powerful stylesheet language, i.e. XSLT.

To show what XSLT can do, we've provided a sample stylesheet in the file `display.xsl`. Open this file with your editor to see how it works: you will see that this too needs changing to make it TEI P5 compatible.

You need to make the following changes:

- Make it reference the stylesheet library at `http://www.tei-c.org.uk/Stylesheets/P5/html/teihtml.x`
- Make the template for `soundClip` namespace-aware. (You learned how to do this last week!)

Once you've made these changes, use this stylesheet with an XSLT processor such as 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.

Note that the stylesheet we supply makes use of the TEI online library of such stylesheets, so you must be online to complete this part of the exercise.