Exercise: Using Roma

January 2010

1 Before you start

In this exercise, you will use **Roma**, a web tool available from the TEI web site and usable with any web browser: Firefox or Internet Explorer or Opera for example. Once you have created your schema, you will also need an XML-aware editor: oXygen in our case.

Our goal is to make a schema which we can use to mark up the sample page. We don't need all of of TEI Lite, much less the full TEI, but we do need bits of various modules. We'll also have to tinker with some of the modules, to make our schema more helpful with daily editing.

2 Making your own schema

1. Open the Roma application, by pointing your favourite web browser at `http://tei.oucs.ox.ac.uk/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 means ticking the first radio button ("Build schema (Create a new customisation by adding elements and modules to the smallest recommended schema)"). Press the `Submit` button at bottom left of the screen to continue.

3. The next and subsequent screens show you a row of tabs for acting on your customization (`Save`, `Customize`, `New`, and `Help`), and a row of tabs 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, and also to change the interface language if you wish. For now, accept the defaults. Go to the `Modules` tab to proceed.

4. 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.

5. For this exercise, we will need the following extra module:

   - `figures`

   Click the word `add` next to the name of the module.

6. The modules chosen contain many more elements than we need, so we will now remove some of them, simplying the view in the XML editor. Click the name of a module in the *List of selected modules* (the right-most column) to see a list of the elements this module defines.

7. 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. You can click on `Exclude` to remove all elements from the module.

8. Now work down the list clicking the radio button to restore or add the elements needed for this exercise. Remember to press the `Submit` button when you have finished with each module. Press the `Modules` or `back` links to go back to the list of modules.

   from the core module delete `<add>`, `<analytic>`, `<cb>`, `<cit>`, `<corr>`, `<del>`, `<distinct>`, `<expan>`, `<gap>`, `<gloss>`, `<headItem>`, `<headLabel>`, `<index>`, `<meeting>`, `<mentioned>`, `<milestone>`, `<orig>`, `<pb>`, `<postBox>`, `<postCode>`, `<reg>`, `<series>`, `<sic>`, `<street>`, `<time>`, and `<unclear>`

   from the textstructure module delete `<argument>`, `<byline>`, `<closer>`, `<dateline>`, `<epigraph>`, `<imprimatur>`, `<opener>`, `<salute>`, `<titlePage>`, and `<trailer>`.

   from the figures module delete `<cell>`, `<formula>`, `<row>` and `<table>`.

9. We are now ready to generate a schema. Click the *Schema* tab, and then press Submit, taking the default option of a RELAXNG compact schema. Your browser will ask whether you want to save or open the generated file: you should save it into your Desktop directory (the default). Look at the result, if you feel strong, or experiment with other options of the web application.

10. Complete this stage by going to the *Save* tab and saving your work. Do not close the web browser, we'll use it again shortly.

3 Using your schema in oXygen

You can use oXygen and the file you made earlier to check that you've made your schema correctly. Proceed as follows:

1. Open the file you made earlier in oXygen.

2. Go to the menu `Document`, then `XML Document` and then `Associate Schema`. Choose the `RELAXNG` tab, and locate your schema file (using the middle folder icon on the right to browse).

3. If all goes well, oXygen will insert some code to mark the schema location, and attempt to validate the file. Try inserting some new elements, and you should see a much-reduced collection.

4 Enhancing your schema

Now let us return to the Roma, and make the schema more constrained, to make sure we get just what we want in our documents. The example we have chosen is to constrain the allowed values of the `type` attribute on `<div>`, and to make it compulsory.

1. Go back to Roma. If you have closed the browser, you can restart Roma and load the session you saved earlier.

2. Go to the *Modules* tab and click on `textstructure` in the right-hand column.

3. Find `<div>` and click on `Change attributes` on the right-hand side.

4. This will show you all the attributes of `<div>`. Click on `type`, and you will be able to change its properties:

   - Change the `Is it optional` radio button to make it compulsory
   - Change the radio button for `Closed list?` to make it a closed list
   - In the box for `List of values`, type

```
cartoon,verse,drama
```

(ie a list of possible values, separated by commas).

- Click on Submit Query

5. Now save the schema as before and reload the file in oXygen. There should be a validation error, because the last `<div>` in the Punch file has no type attribute.

5 Documenting your schema with Roma

One of the major benefits of Roma, is that after you have customized your schema it can produce two ways of documenting the changes you have made. One of these is the ODD file that you saved earlier (and can save again if you want) which indicates how your schema differs from full TEI, what modules you have included, what elements you may have added and changed, amongst other things. This is a good file to keep with your generated schema, in case you need to generate a new schema with additional elements or constrain it further. It also means that others can generate the schema for your documents in different schema languages if needed.

You may wish to open the ODD file you saved earlier and see how this format works.

You don't need to use the web version of Roma to create ODD files, you can just author them by hand if you are feeling geeky. There is a command-line Roma script which you can use to generate any of the outputs which the web version produces.

However, since the TEI Guidelines themselves are made up of ODD documents, Roma also allows you to generate a set documentation for your particular customization of the TEI. You don't need to worry about creating the documentation for the existing elements you have included, Roma already knows about them. If you've added new elements you are able to provide descriptions and information about the element at that point, and Roma will use that. Moreover, if you've changed the names of the elements or their descriptions (perhaps for reasons of internationalization) it will use the new names you have provided in the documentation it generates.

You may wish to generate this from the `Documentation` tab.

This concludes the brief exercises on **Roma** for customizing the TEI schema. If you have time you may wish to experiment with making some other customizations to your schema! Some ideas are below.

6 Other things to try with Roma

1. How do you go about renaming an existing element? What happens in the ODD when you do?

2. How do you add a new element? What namespace does it end up in when you do? How can you control this?

3. Once you have an ODD, you can generate project-specific documentation. Where would you put this prose in the ODD file? Try generating some test documentation with additions you have made.

4. Experiment with starting with different exemplar customization Roma offers. Which do you think would be best for your project?

5. Save the ODD for the TEI ALL Plus schema -- how does this add in schemas from other namespaces?

6. How do you go about changing the content model of an element? What implications does this have for your schema? How is it expressed in ODD?

7. Try modifying an ODD 'manually' in oXygen to include/exclude other elements and then submitting it to Roma to generate a schema.