# Instructions for the text encoding activities in the courses on *Digitising Cultural Heritage Material*

Mikael Gunnarsson
Högskolan i Borås,
Sektionen för biblioteks- och informationsvetenskap.

February 18, 2022

These instructions for the course on digitisation within the *Master's programme on Library and Information Science, Digital Library and Information Services*, and *Masterprogram i Biblioteks- och informationsvetenskap, distansutbildning* include advices on how to use the exercises offered and additional exercises for learning to work with TEI. They are not normative in any way and any suggested strategies or similar things may be overridden by instructions given in the context of course activities and by instructions in the learning platform.

## 1 Introduction

The Text Encoding Initiative (TEI) framework is an overarching standard for the encoding of digitised text, encompassing a wide range of plausible requirements that may vary a lot according to

particular project needs.

In practice, TEI always needs to be tailored to such particular needs and all its components are not intended to be used in all projects. It defines a wide range of elements, attributes and strategies in order for international harmonization of encoding practices to be accomplished. Thus, deviations from its intentions are not recommended and need to be firmly motivated.

The TEI is documented in the *Guidelines* (`http://www.tei-c.org/Guidelines/P5/`) and the TEI website as a whole (`http://www.tei-c.org`) is a valuable resource for many things, such as giving access to supporting tools for working with the TEI. The *Guidelines* must always be consulted when deciding upon the use of a particular element or attribute.

## 2 Editing tools

Using TEI nearly always implies the use of XML for encoding and serialization. This is because XML encodings are amenable for transformations into several different target formats, such as HTML, EPUB or PDF, making it possible to publish and distribute material in any suitable way.

Therefore, you need some experience in working with XML and access to some XML editor. There are many XML editing tools to choose from, ranging from freely available standalone text editors to complex development off-the-shelf frameworks. The latter ones are often fairly expensive.

It is important to remember that any text editor, such as Microsofts NOTEPAD or Mac OS Xs TEXTEDIT can be used for

text editing as long as you see to it that the text is saved in the appropriate character encoding scheme (i.e. mostly UTF-8).

However, though these simple tools can be used for corrections of small errors in already encoded texts, they soon become inconvenient for more encompassing tasks. In addition, you most certainly need access to real-time validation (that checks if the encoding is correct with respect to the TEI scheme, at the same time as you add the markup) and tools for transforming the encodings into humanly readable formats.

For this purpose we strongly recommend the oXygen framework (`https://www.oxygenxml.com/`), its editor in particular, since it has extended support for TEI. In fact it is developed in close connection with the TEI community. We will be using this framework for illustration, so you will get acquainted with it through exercises and recordings.

Unfortunately, it is not altogether free. There is an evaluation license that lasts for 30 days, for which you can order a license key and use it for free during part of the course. If you want to continue using this tool after 30 days you must order an academic license for approx. $ 99.

You may already be acquainted with the EditiX editor (`http://www.editix.com/`) from previous courses. You can use this one as well, it is similar to oXygen. It is also similar to oXygen in that you can get an evaluation license for 30 days for free. Compared to oXygen the handy real-time validation and the tramsformation possibilities are there in EditiX as well but a bit more tricky to use and not that powerful.

If you really cannot afford spending money on an XML editor there is a last option, not recommended, but plausible. The

XML Spear (http://www.donkeydevelopment.com/) editor is free and has real-time validation and support for transformation, but lacks the tailored support for TEI that oXygen has. The same goes for jEdit.

All four tools run on Mac OS X, Microsoft Windows and – as far as I know – on all Linux implementations.

# 3 Exercises

What follows in the remainder of this text is guidance for a set of exercises intended for the workshop(s). If you do not participate in the workshop(s) you should be particularly meticulous in doing these on your own.

The exercises cover three parts

1. encoding the body of the text and any front or back matter of the source object

2. encoding the so called TEI header and provide necessary metadata for the source and target objects as well as the decisions taken and the procedures surrounding the digitisation task

3. transformation of the encoded text into humanly readable formats

where *source object* means the physical object and/or the image object digitised[1], and *target object* means the TEI/XML version of the digitised object.

The exercises for the two first parts are bundled together in a zip file together with source material and suggestions for solution ("spoilers"), and are distributed from within the learning platform. The TEI exercises are created by James Cummings previously at the University of Oxford, now at University of Newcastle (`https://www.ncl.ac.uk/elll/staff/profile/jamescummings.html` ), who is, among other things, involved in the development of the TEI framework.

## 3.1 Encoding the text

Download the zipped file (`IntroducingTEI.zip`) from the learning platform to a place on your work station where you can find it. This means you should notice the path to where you put it. Use the file explorer to locate the zipped file and unpack it. Open `exercise01.pdf` and `exercise02.pdf`.

**Exercises 1** and **2** take you through the basic procedures and also give you a short recapitulation of basic XML encoding, if you happen to have forgotten it from previous courses.

The sample used for these exercises is a letter for which you have two scanned jpeg images and the raw text of the contents,

[1] In some cases you do not have access to the physical source and need to take a digital image as a starting point. In such cases you have two objects where one is considered primary, but need to describe both.

because the images are quite hard to read (in the subfolder `material`).Please note that letters are just one type of cultural heritage objects that can be encoded by TEI.

## 3.2 Encoding the TEI header

**Exercise 3** treats the elaboration of the TEI header, which is something you should be particularly careful about, since it regards the metadata, the resource description, of both the source and target object, without which the texts could be unusable.

## 3.3 Transformations

The result of your work this far is an XML file. It can in fact be a powerful thing, more powerful than is discernable by just looking at it. This depends on the fact that it is structured in a predictable way and amenable for processing in many different ways.

The first you may do is to explore the facilities that come with OXYGEN. These facilities comprise a large set of XSL files tailored for TEI and a built-in XSL processor that can take the source and apply transformations on it, thus producing a derivative file of some kind.

Choose from the `Document` menue `Apply transformation scenario` from the option `Transformation`. You will be presented by a set of options that offers you the possibility to choose different target formats.

Tick the option `TEI P5 XHTML` and click the button for `Apply associated`. This will run the transformation process ("sce-

nario"). Depending on your local settings, different things may happen. Hopefully a web browser window will open and display an HTML version of your XML encoding.

Study the output and compare it with the source XML file.

1. Which source data are present in the output and which are not?

2. Study the HTML encoding and identify mappings between TEI elements and HTML elements

Now, try another scenario. Choose `Configure transformation scenario` (instead of `Apply transformation scenario` ) and untick the `TEI P5 XHTML` option. Choose `TEI P5 PDF` instead. Run the scenario as before. Hopefully `Adobe Acrobat` will launch your created PDF file — otherwise try to locate where OXYGEN has stored the output. Normally it should be in your working directory, i.e. where you have your source XML file.

Compare the XHTML and PDF outputs. What are the differences?

The results of the transformation scenarios in the way you have applied them hitherto are fixed and depend on how they are predefined by the author of the XSL stylesheets (i.e. Sebastian Rahtz, who sadly passed away in 2016). The authoring of these scenarios have been developed on a generic basis, trying to foresee variations of encoding practices to some degree but still keeping it general to have a working solution that fits all variations to some degree. With such an approach you need to be satisfied with the end result, unless you delve into the mysteries of XSL and redefine the processing and presentations of each element and attribute in your *own* project.

The transformation scenarios are documented at
`https://wiki.tei-c.org/index.php/Tei-xsl`

On the learning platform there is also a short exercise on creating XSL from scratch.