Basic transcription in XML

James Cummings

August 2010
Introduction to the Course

Aims of Course

1. Examine the concept of markup and XML encoding
2. Provide hands-on experience in using TEI XML markup
3. Introduce the some of TEI scheme, its assumptions
4. Provide routes into more detailed information for exploration at your leisure
5. Provide opportunities for questions and discussions relating to your own encoding needs and priorities
Course Structure: Session 1

Session 1: Basic transcription in XML

- Introducing the rules of XML
- Survey of the most common TEI elements
- Introducing the oXygen editor
- Exercise: Basic transcription
Course Structure: Session 2

Session 2: Metadata

- The TEI Header
- Annotating names of people and places
- Optional exercise: Adding metadata
Course Structure: Session 3

Session 3: Detailed transcription with TEI

- Transcription and editorial phenomena
- Exercise: Adding detail to the transcription
- Conclusion: TEI Community and Questions
Course Materials

- **All** course materials including:
  - **All** slides from lectures
  - **All** exercises
  - **All** materials for the exercises

are available on the TEI @ Oxford website.

- The url is: http://tei.oucs.ox.ac.uk/Oxford/2011-10-helsinki/
  and this is where you will need to download your materials for the exercises!

- **All** these materials are licensed with a Creative Commons Attribution license, which means they are freely available for re-use (though do let us know!)
After the workshop...

After the workshop, if you have questions about:

- The workshop materials or teaching other workshops: tei@oucs.ox.ac.uk
- The TEI generally: TEI-L@listserv.brown.edu (join at http://tinyurl.com/teimailinglist)

If you mail the TEI-L mailing list it is better because:

- we’ll still try to answer as well as we would privately
- you get answers not only from us, but TEI experts around the world
- questions from those of all levels of ability stop the list becoming too technical
- everyone benefits from having the answers be public – and you benefit by reading (and sometimes answering!) others’ problems
Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML also now plays an indispensable role in the exchange of a wide variety of data on the Web and elsewhere.
XML: what it is and why you should care

- XML is **structured data** represented as strings of text
- XML looks like HTML, except that:-
  - XML is **extensible**
  - XML must be **well-formed**
  - XML can be **validated**
- XML is application-, platform-, and vendor- independent
- XML empowers the **content provider** and facilitates data integration
XML is an international standard

- XML requires use of ISO 10646 (also known as Unicode)
  - a 31 bit character repertoire including most human writing systems
  - encoded as UTF8 or UTF16
- other encodings may be specified at the document level
- language may be specified on any element level using W3C @xml:lang

The @xml:id attribute is another W3C-defined attribute.
XML terminology

An XML document may contain:-

- elements, possibly bearing attributes
- processing instructions
- comments
- entity references
- marked sections (CDATA, IGNORE, INCLUDE)

An XML document must be well-formed and may be valid
XML terminology Example

```xml
<?xml version="1.0" ?>
<root>
  <element attribute="value"> content </element>
  <!-- comment -->
</root>
```
The rules of the XML Game

- An XML document represents a (kind of) tree
- It has a single root and many nodes
- Each node can be
  - a subtree
  - a single element (possibly bearing some attributes)
  - a string of character data
- Each element has a name or generic identifier
The rules of the XML Game

- An XML document represents a (kind of) tree
- It has a single root and many nodes
- Each node can be
  - a subtree
  - a single element (possibly bearing some attributes)
  - a string of character data
- Each element has a name or generic identifier
The rules of the XML Game

- An XML document represents a (kind of) tree
- It has a single root and many nodes
  - Each node can be
    - a subtree
    - a single element (possibly bearing some attributes)
    - a string of character data
- Each element has a name or generic identifier
The rules of the XML Game

- An XML document represents a (kind of) **tree**
- It has a single **root** and many nodes
- Each node can be
  - a subtree
  - a single **element** (possibly bearing some **attributes**)
  - a string of **character data**
- Each element has a name or **generic identifier**
The rules of the XML Game

- An XML document represents a (kind of) tree
- It has a single root and many nodes
- Each node can be
  - a subtree
    - a single element (possibly bearing some attributes)
    - a string of character data
- Each element has a name or generic identifier
The rules of the XML Game

- An XML document represents a (kind of) tree
- It has a single root and many nodes
- Each node can be
  - a subtree
  - a single element (possibly bearing some attributes)
  - a string of character data
- Each element has a name or generic identifier
The rules of the XML Game

- An XML document represents a (kind of) tree
- It has a single root and many nodes
- Each node can be
  - a subtree
  - a single element (possibly bearing some attributes)
  - a string of character data
- Each element has a name or generic identifier
The rules of the XML Game

- An XML document represents a (kind of) tree
- It has a single root and many nodes
- Each node can be
  - a subtree
  - a single element (possibly bearing some attributes)
  - a string of character data
- Each element has a name or generic identifier
Representing an XML tree

- An XML document is encoded as a linear string of characters
- It begins with a special processing instruction
- Element occurrences are marked by start- and end-tags
- The characters < and & are Magic and must always be "escaped" if you want to use them as themselves
- Comments are delimited by <!-- and -->
- CDATA sections are delimited by <![CDATA[ and ]]> 
- Attribute name/value pairs are supplied on the start-tag and may be given in any order
- Entity references are delimited by & and ;
Parts of an XML document

```xml
<?xml version="1.0"?>
<greetings xmlns="http://www.example.org/greetings">
  <hello xmlns="http://www.example.org/greetings" type="sarcastic">
    hello world!
  </hello>
</greetings>
```

- The XML declaration
- Namespace declarations
- The root element of the document itself
- Other elements and content
- Attribute and value
Parts of an XML document

- The XML declaration
  - Namespace declarations
  - The root element of the document itself
  - Other elements and content
  - Attribute and value

```xml
<?xml version="1.0"?>
<greetings xmlns="http://www.example.org/greetings">
  <hello xmlns="http://www.example.org/greetings"
         type="sarcastic">hello world!</hello>
</greetings>
```
Parts of an XML document

```xml
<?xml version="1.0"?>
<greetings xmlns="http://www.example.org/greetings">
  <hello xmlns="http://www.example.org/greetings" type="sarcastic">hello world!</hello>
</greetings>
```

- The XML declaration
- Namespace declarations
  - The root element of the document itself
  - Other elements and content
  - Attribute and value
Parts of an XML document

```xml
<?xml version="1.0"?>
<greetings xmlns="http://www.example.org/greetings">
  <hello xmlns="http://www.example.org/greetings" type="sarcastic">
    hello world!
  </hello>
</greetings>
```

- The XML declaration
- Namespace declarations
- The root element of the document itself
- Other elements and content
- Attribute and value
Parts of an XML document

```xml
<?xml version="1.0"?>
<greetings xmlns="http://www.example.org/greetings"
>    <hello xmlns="http://www.example.org/greetings"
        type="sarcastic">hello world!</hello>
</greetings>
```

- The XML declaration
- Namespace declarations
- The root element of the document itself
- Other elements and content
- Attribute and value
Parts of an XML document

```xml
<?xml version="1.0"?>
<greetings xmlns="http://www.example.org/greetings"/>
  <hello xmlns="http://www.example.org/greetings"
     type="sarcastic">hello world!</hello>
</greetings>
```

- The XML declaration
- Namespace declarations
- The root element of the document itself
- Other elements and content
- Attribute and value
The XML declaration

An XML document must begin with an XML declaration which does three things:

- specifies that this is an XML document
- specifies which version of the XML standard it follows
- specifies which character encoding the document uses

`<?xml version="1.0" ?>`
`<?xml version="1.0" encoding="iso-8859-1" ?>`

The default, and recommended, encoding is ‘UTF-8’ (Unicode)
Namespace declarations

All TEI documents are declared within the TEI namespace: `<TEI xmlns="http://www.tei-c.org/ns/1.0"> ... </TEI>`

XML documents can include elements declared in different name spaces.

- A namespace declaration associates a namespace prefix with an external URI-like identifier.
- The default namespace *may* be declared using a `xmlns`.
- Other name spaces must all use a specially declared prefix.

```xml
<TEI xmlns="http://www.tei-c.org/ns/1.0"
     xmlns:math="http://www.mathml.org">
  <p>...<math:expr>...</math:expr>...</p>...
</TEI>
```

The `xml` namespace is used by the TEI for global attributes `@xml:id` and `@xml:lang`
<?xml version="1.0" encoding="utf-8" ?>
<div n="1">
  <head>SCENE I. On a ship at sea: a tempestuous noise of thunder and lightning heard.</head>
  <stage>Enter a Master and a Boatswain</stage>
  <sp>
    <speaker>Master</speaker>
    <ab>Boatswain!</ab>
  </sp>
  <sp>
    <speaker>Boatswain</speaker>
    <ab>Here, master: what cheer?</ab>
  </sp>
  <sp>
    <speaker>Master</speaker>
    <ab>Good, speak to the mariners: fall to't, yarely,</ab>
    <ab>or we run ourselves aground: bestir, bestir.</ab>
  </sp>
  <stage>Exit</stage>
</div>
An XML Tree For The Tempest
**XML syntax: the small print**

What does it mean to be **well-formed**?

1. there is a single root node containing the whole of an XML document
2. each subtree is properly nested within the root node
3. names are always case sensitive
4. start-tags and end-tags are always mandatory (except that a combined start-and-end tag may be used for empty nodes)
5. attribute values are always quoted

Note: You can be **valid** in addition to being well-formed. This means you obey the rules of a specified schema, such as the TEI.
Test your XML knowledge

Which are correct?

- `<seg>some text</seg>`
- `<seg><foo>some</foo> <bar>text</bar></seg>`
- `<seg><foo>some <bar></foo> text</bar></seg>`
- `<seg type="text">some text</seg>`
- `<seg type='text'>some text</seg>`
- `<seg type=text>some text</seg>`
- `<seg type="text">some text<seg/>`
- `<seg type="text">some text<gap/></seg>`
- `<seg type="text">some text</Seg>`
Test your XML knowledge

Which are correct?

- `<seg>some text</seg>`
- `<seg><foo>some</foo> <bar>text</bar></seg>`
- `<seg><foo>some <bar></foo> text</bar></seg>`
- `<seg type="text">some text</seg>`
- `<seg type='text'>some text</seg>`
- `<seg type=text>some text</seg>`
- `<seg type="text">some text</seg>`
- `<seg type="text">some text<seg/`>`
- `<seg type="text">some text<gap/></seg>`
- `<seg type="text">some text</Seg>`
Test your XML knowledge

Which are correct?

- `<seg>some text</seg>`
- `<seg><foo>some</foo> <bar>text</bar></seg>`
- `<seg><foo>some <bar></foo> text</bar></seg>`
- `<seg type="text">some text</seg>`
- `<seg type='text'>some text</seg>`
- `<seg type=text>some text</seg>`
- `<seg type="text">some text<seg/>`
- `<seg type="text">some text<gap/></seg>`
- `<seg type="text">some text</Seg>`
Test your XML knowledge

Which are correct?

- `<seg>some text</seg>`
- `<seg><foo>some</foo> <bar>text</bar></seg>`
- `<seg><foo>some <bar></foo> text</bar></seg>`
- `<seg type="text">some text</seg>`
- `<seg type='text'>some text</seg>`
- `<seg type=text>some text</seg>`
- `<seg type="text">some text<seg/>`
- `<seg type="text">some text<gap/></seg>`
- `<seg type="text">some text</Seg>`
Test your XML knowledge

Which are correct?

- `<seg>some text</seg>`
- `<seg><foo>some</foo> <bar>text</bar></seg>`
- `<seg><foo>some <bar></foo> text</bar></seg>`
- `<seg type="text">some text</seg>`
- `<seg type='text'>some text</seg>`
- `<seg type=text>some text</seg>`
- `<seg type="text">some text<seg/>`
- `<seg type="text">some text<gap/></seg>`
- `<seg type="text">some text</Seg>`
Test your XML knowledge

Which are correct?

- `<seg>some text</seg>`
- `<seg><foo>some</foo> <bar>text</bar></seg>`
- `<seg><foo>some <bar></foo> text</bar></seg>`
- `<seg type="text">some text</seg>`
- `<seg type='text'>some text</seg>`
- `<seg type=text>some text</seg>`
- `<seg type=text>some text</seg>`
- `<seg type=text>some text<seg/>`
Test your XML knowledge

Which are correct?

- `<seg>some text</seg>`
- `<seg><foo>some</foo> <bar>text</bar></seg>`
- `<seg><foo>some <bar></foo> text</bar></seg>`
- `<seg type="text">some text</seg>`
- `<seg type='text'>some text</seg>`
- `<seg type=text>some text</seg>`
- `<seg type="text">some text<seg/>`
- `<seg type="text">some text<gap/></seg>`
- `<seg type="text">some text</Seg>`
Test your XML knowledge

Which are correct?

- `<seg>some text</seg>`
- `<seg><foo>some</foo> <bar>text</bar></seg>`
- `<seg><foo>some <bar></foo> text</bar></seg>`
- `<seg type="text">some text</seg>`
- `<seg type='text'>some text</seg>`
- `<seg type=text>some text</seg>`
- `<seg type="text">some text<seg/>`
- `<seg type="text">some text<gap/></seg>`
- `<seg type="text">some text</Seg>`
Test your XML knowledge

Which are correct?

- `<seg>some text</seg>`
- `<seg><foo>some</foo> <bar>text</bar></seg>`
- `<seg><foo>some <bar></foo> text</bar></seg>`
- `<seg type="text">some text</seg>`
- `<seg type='text'>some text</seg>`
- `<seg type=text>some text</seg>`
- `<seg type="text">some text<seg/>`
- `<seg type="text">some text<gap/></seg>`
Test your XML knowledge

Which are correct?

- `<seg>some text</seg>`
- `<seg><foo>some</foo> <bar>text</bar></seg>`
- `<seg><foo>some <bar></foo> text</bar></seg>`
- `<seg type="text">some text</seg>`
- `<seg type='text'>some text</seg>`
- `<seg type=text>some text</seg>`
- `<seg type="text">some text<seg/>`
All TEI documents are structured in a particular manner. This section attempts to describe the different variations on this as briefly as possible.
Structure of a TEI Document

There are two basic structures of a TEI Document:

- `<TEI>` (TEI document) contains a single TEI-conformant document, comprising a TEI header and a text, either in isolation or as part of a `teiCorpus` element.

- `<teiCorpus>` contains the whole of a TEI encoded corpus, comprising a single corpus header and one or more TEI elements, each containing a single text header and a text.
TEI basic structures (1)

```xml
<teiCorpus>
  <teiHeader> <!-- required -->
  </teiHeader>
  <TEI> <!-- required -->
  </TEI>
</teiCorpus>
```
TEI basic structures (2)

```xml
<TEI>
  <teiHeader>
    <!-- required -->
  </teiHeader>
  <facsimile>
    <!-- optional, new in TEI P5 -->
  </facsimile>
  <text>
    <!-- required if no facsimile -->
  </text>
</TEI>
```
What is a text?

- A text may be unitary or composite
  - unitary: forming an organic whole
  - composite: consisting of several components which are in some important sense independent of each other

- A unitary text contains
  - optional front matter
  - `<body>` (required)
  - optional back matter
What is a text?

- A text may be unitary or composite
  - unitary: forming an organic whole
  - composite: consisting of several components which are in some important sense independent of each other

- a unitary text contains
  - optional front matter
  - \texttt{<body>} (required)
  - optional back matter
What is a text?

- A text may be unitary or composite
  - unitary: forming an organic whole
  - composite: consisting of several components which are in some important sense independent of each other

- A unitary text contains
  - optional front matter
  - `<body>` (required)
  - optional back matter
Composite texts

A composite text contains

- optional front matter
- `<group>` (required)
- optional back matter

A corpus is a collection of text and header pairs. It has its own header.
`<group>` tags may self-nest.
TEI text structure (1)

```xml
<text>
  <front>
    <!-- optional -->
  </front>
  <body>
    <!-- required -->
  </body>
  <back>
    <!-- optional -->
  </back>
</text>
```
TEI text structure (2)

```xml
<text>
  <front>
  <!-- ... -->
  </front>
  <group>
    <text>
      <body>
        <p>...</p>
      </body>
    </text>
  </group>
  <back>
  <!-- ... -->
  </back>
</text>
```
Another Grouped Text Example

```xml
<TEI>
  <teiHeader>
    <!-- header information for the whole collection -->
  </teiHeader>
  <text>
    <!-- optional front matter -->
    <group>
      <text>
        <!-- First Body -->
        <body>
          <!-- Second Body-->
          <body>
        </body>
      </text>
    </group>
  </text>
</TEI>
```
Common TEI Elements

The so-called 'Core' module groups together elements which may appear in any kind of text and the tags used to mark them in all TEI documents. This includes:

- paragraphs
- highlighting, emphasis and quotation
- simple editorial changes
- basic names numbers, dates, addresses
- simple links and cross-references
- lists, notes, annotation, indexing
- graphics
- reference systems, bibliographic citations
- simple verse and drama
Paragraphs

<p> (paragraph) marks paragraphs in prose

- Fundamental unit for prose texts
- <p> can contain all the phrase-level elements in the core
- <p> can appear directly inside <body> or inside <div> (divisions)

<p>It was a cottage, the cottage of a dream. And by a cottage I mean, not four plain rooms and a kitchen, but one surprising room opening into another; rooms all on different levels and of different shapes, with delightful places to bump your head on; open fireplaces; a large square hall, oak-beamed, where your guests can hang about after breakfast, while deciding whether to play golf or sit in the garden. Yet all so cunningly disposed that from outside it looks only a cottage or, at most, two cottages persuaded into one.</p>
Highlighting

By *highlighting* we mean the use of any combination of typographic features (font, size, hue, etc.) in a printed or written text in order to distinguish some passage of a text from its surroundings. For words and phrases which are:

- distinct in some way (e.g. foreign, archaic, technical)
- emphatic or stressed when spoken
- not really part of the text (e.g. cross references, titles, headings)
- a distinct narrative stream (e.g. an internal monologue, commentary)
- attributed to some other agency inside or outside the text (e.g. direct speech, quotation)
- set apart in another way (e.g. proverbial phrases, words mentioned but not used)
Highlighting Examples

- `<hi>` (general purpose highlighting)

```
<p>[The rest of this communication is omitted owing to considerations of space.—<hi rend="sc">Ed</hi>.]</p>
```

- `<distinct>` (linguistically distinct)

```
But then I remind myself that the Russian ballet is nothing if not <distinct>bizarre</distinct>.
```

- Other similar elements include: `<emph>`, `<mentioned>`, `<soCalled>`, `<term>` and `<gloss>`
Quotation

Quotation marks can be used to set off text for many reasons, so the TEI has the following elements:

- `<q>` (separated from the surrounding text with quotation marks)
- `<said>` (speech or thought)
- `<quote>` (passage attributed to an external source)
- `<cit>` (groups a quotation and citation)

```xml
<p>
  <said who="#Celia">I know a lovely tin of potted grouse, said Celia, and she went off to cut some sandwiches. By twelve o'clock we were getting out of the train.</said>
</p>
```
Simple Editorial Changes: `<choice>` and Friends

- `<choice>` (groups alternative editorial encodings)
- Errors:
  - `<sic>` (apparent error)
  - `<corr>` (corrected error)
- Regularization:
  - `<orig>` (original form)
  - `<reg>` (regularized form)
- Abbreviation:
  - `<abbr>` (abbreviated form)
  - `<expan>` (expanded form)
I profess not

to know how women's `<choice>`
  `<orig>`heartes`/orig>`
  `<reg>`hearts`/reg>`
`</choice>` are wooed and won. To me they have always been `<choice>`
  `<sic>`maters`/sic>`
  `<corr>`matters`/corr>`
`</choice>` of riddle and `<choice>`
  `<abbr>`admirat'n`/abbr>`
  `<expan>`admiration`/expan>`
`</choice>`.
Additions, Deletions, and Omissions

- `<add>` (addition to the text, e.g. marginal gloss)
- `<del>` (phrase marked as deleted in the text)
- `<gap>` (indicates point where material is omitted)
- `<unclear>` (contains text unable to be transcribed clearly)
Basic Names

- `<name>` (a name in the text, contains a proper noun or noun phrase)
- `<rs>` (a general-purpose name or referencing string)

The `@type` attribute is useful for categorizing these, and they both also have `@key`, `@ref`, and `@nymRef` attributes.
Basic Names Example

<p>The scene opens at a party given by <name nymRef="http://www.meanings-of-name.com/potiphar.html">Potiphar</name> in <name ref="http://en.wikipedia.org/wiki/Venice" type="place">Venice</name>. </p>

<p>It is when the natural end of the story is reached, and <name xml:id="SIMON">Simon</name> has come into his own and has just been wedded to his proper affinity, that the structure seems to me to fall with a crash. I might perhaps, though not without reluctance, have pardoned an impertinent railway accident which leaves <rs corresp="#SIMON">the young man</rs> apparently crippled for life.</p>
Addresses

- `<email>` (an electronic mail address)
- `<address>` (a postal address)
- `<addrLine>` (a non-specific address line)
- `<street>` (a full street address)
- `<postCode>` (a postal (or zip) code)
- `<postBox>` (a postal box number)
- `<name>` can also be used

and the 'namesdates’ module extends this with more geographic names
Basic Address Example

<email>gbs@heaven.com</email>

<address>
  <name>George Bernard Shaw</name>
  <addrLine>Shaw's Corner</addrLine>
  <settlement>Ayot St Lawrence</settlement>
  <district>Hertfordshire</district>
  <postCode>HE 1 XXX</postCode>
  <country>England</country>
</address>
Basic Numbers and Measures

- `<num>` (marks a number of any sort)
- `<measure>` (marks a quantity or commodity)
- `<measureGrp>` (groups specifications relating to a single object)

While `<num>` has simple `@type` and `@value` attributes, `<measure>` has `@type`, `@quantity`, `@unit` and `@commodity` attributes
Number and Measure examples

<l>They went off at a pace I am bound to deplore,</l>
<l>For they did <num value="20">twenty</num> yards in a minute or more</l>
<l>And a yard or <num value="2">two</num> over, a capital score</l>
<l>For Farnaby Fullerton Rigby.</l>

<p>If neither of these values is available, a value of <num>20,35</num> for ash content can be assumed initially and checked, after the sampling has been carried out, using one of the methods described in ISO 13909-7.</p>

It is on these days that we travel to our Castle of Stopes; as the crow flies, <measure quantity="24140" unit="m">fifteen miles</measure> away. Indeed, that is the way we get to it, for it is a castle in the air.
Introduction to the Course

An Introduction to XML

Default Text Structure

Common TEI Elements

Dates

- `<date>` (contains a date in any format and includes a `@when` attribute for a regularised form and a `@calendar` attribute to specify what calendar system)
- `<time>` (contains a time in any format and includes a `@when` attribute for a regularised form)

```xml
<p>At <time when="09:30:00">9.30 o'clock</time>, as the fog lifted somewhat, the rescuing steamer Lyonnesse had sighted the Gothland, fast on the rocks, with a bad list to starboard, and apparently partly filled with pater.</p>
<p>House of Commons, <date when="1914-06-22">Monday, June 22, 1914</date>.</p>
```
Simple Linking

- `<ptr>` (defines a pointer to another location)
- `<ref>` (defines a reference to another location, with optional linking text)

Both elements have:
  - `@target` attribute taking a URI reference
  - `@cRef` attribute for canonical referencing schemes

If the linking text is able to be generated, `<ptr>` and `<ref>` might be used in the same place.
Simple Linking Example

See `<ref target="#Section12">section 12 on page 34</ref>`.

See `<ptr target="#Section12"/>`. 
Lists

- `<list>` (a sequence of items forming a list)
- `<item>` (one component of a list)
- `<label>` (label associated with an item)
- `<headLabel>` (heading for column of labels)
- `<headItem>` (heading for column of items)
Simple List Example

The previous slide contained only:

```xml
<div>
  <head>Lists</head>
  <p>
    <list>
      <item>
        <gi>list</gi> (a sequence of items forming a list)</item>
      <item>
        <gi>item</gi> (one component of a list)</item>
      <item>
        <gi>label</gi> (label associated with an item)</item>
      <item>
        <gi>headLabel</gi> (heading for column of labels)</item>
      <item>
        <gi>headItem</gi> (heading for column of items)
      </list>
    </p>
  </div>
```
Notes

- `<note>` (contains a note or annotation)
- Notes can be those existing in the text, or provided by the editor of the electronic text
- A `@place` attribute can be used to indicate the physical location of the note
- Although notes should usually be encoded where its identifier/mark first appears, notes can also be kept separately and point back to their location with a `@target` attribute
<p>It is not only misfortune that makes strange bedfellows.  
<note place="foot">By-the-by, it is denied that Sir  
&lt;name&gt;Joseph Beecham&lt;/name&gt; was in any way responsible  
for the Government's &lt;title&gt;Pills for  
Earthquakes&lt;/title&gt;, by which it was hoped to avert the  
Irish crisis.&lt;/note&gt;</p>
**Graphics**

- `<graphic>` (indicates the location of an inline graphic, illustration, or figure)
- `<binaryObject>` (encoded binary data embedding a graphic or other object)
- The *figure* module provides `<figure>` and `<figDesc>` for more complex graphics

```
<figure>
  <graphic url="images/014.png"/>
  <head>Garden City Washing-day.</head>
  <p>Our sensitive artist insists on a harmonious colour-scheme.</p>
  <figDesc>A bearded man sits in a deckchair and wags his finger at a woman hanging up washing</figDesc>
</figure>
```
For our exercises we’re going to be using the oXygen XML editor, made by a Romanian company called SynchRo Soft. This has quickly become the market leader in XML editors, but I thought I should explain why we use it. There are other alternatives which you are free to use, but they don’t have the vast array of features that oXygen does.
Editor types

Editing tools cover a wide spectrum:

- Basic text editors
- General programmers’ editors
- XML-aware programmers’ editors
- XML-specific editors
- Word-processors which can export XML
- Data-entry forms
- Image-specific editors

It is likely that people in different roles need different tools.
Things to look for in specialist XML editors

- schema-aware
- constraining element entry
- IDE features
- customizable
- validation, preferably continual
- Multiple display views (as tree, with tags, formatted etc)
- folding structures
- context-sensitive help

For XML editing, oXygen, Emacs, jEdit, XMetaL, XMLSpy, Stylus Studio, Arbortext Adept are all worth a look.
For image markup try University of Victoria Image Markup Tool.
oXygen Features (1)

- Multiple modes for editing XML documents: Author (CSS based), Grid, Text
- TEI Support including: New document templates; Author mode CSS; Transformations to HTML and PDF
- Ability to add/extend/customise for other frameworks
- Available as an Eclipse plugin (Java IDE)
- Java API for developer add-ons
oXygen Features (2)

- Support for all schema languages: such as Relax NG, Schematron, XML Schema, DTDs, NVDL, NRL
- Content completion based on TEI Relax NG schemas
- Tooltip documentation based on TEI Relax NG schemas
- NVDL easily validates TEI documents in multiple namespaces
oXygen Features (3)

- XQuery directly against local/remote XML databases like eXist
- XSLT and FOP support for transformations to XML/HTML/PDF etc.
- WebDAV and FTP support for access to files on remote servers/CMS
- Built-in subversion client for collaborative version control and visual change management
- Spell checking support as you type that is @xml:lang aware
- Included graphical XML Diff to analyse differences between documents
**oXygen Features (4)**

**But maybe most important...**

- Multi-platform: oXygen is available on Windows, Mac OS X, Linux, Solaris, etc.
- They have an enlightened academic pricing policy ($64 USD) (Oxford has a site license)
- The named-user based license allows the same user to use any oXygen distribution on any platform or machine: the same license covers you at work, laptop, and home.
- They are nice enough to give us trial licenses to teach workshops with!
An Introduction to XML

Default Text Structure

Common TEI Elements

Introduction to the Course

TEI Workshop, Helsinki

<oxygen/>

/titleStmt>
<publicationStmt>
<p!-- supply publication information -->
</p>
</publicationStmt>
<sourceDesc>
<p><!-- supply information about the source -->
</p>
</sourceDesc>
</fileDesc>
</teiHeader>
<text>
<front>
<head>THE LEGEND OF SLEEPY HOLLOW.</head>
<head>FOUND AMONG THE PAPERS OF THE LATE DIMRICH KNICKERBOCKER</head>
<br>
</p>
</div>
</body>
<div>
<p>A pleasing land of drowsy head it was,</p>
<p>Of dreams that wave before the half-shut eye;</p>
<p>And of gay castles in the clouds that pass,</p>
<p>For ever flushing round a summer sty.</p>
</div>
</front>

In the bosom of one of those spacious caves which indent the eastern shore of the Hudson, at that broad expansion of the river denominated by the ancient Dutch navigators the Tappan Zee, and where they always prudently shortened sail, and implored the protection of St. Nicholas when they crossed, there lies a small market-town or rural port, which by some is called Greensburgh, but which is more generally and properly known by the name of Tarry Town. This name was given, we are told, in former days, by the good housewives of the adjacent

TEI http://www.tei-c.org
Basic oXygen Editing

```xml
<body>
  <div>
    <head>Summary</head>
    <p>How does a TEI user do the following? <list>
      <item>Data capture</item>
      <item>Editing</item>
      <item>Schema design</item>
      <item>Other forms of validation</item>
    </list>
  </p>
  <div>
    <head>Waltz</head>
    <p>Anchor</p>
    <list>
      <item>Argument</item>
      <item>bibl</item>
      <item>Document transformation tools</item>
      <item>Document delivery tools</item>
      <item>Document storage and management tools</item>
      <item>Programming interfaces</item>
      <item>Specialized applications</item>
    </list>
    <p></p>
  </div>
  <div rend="slide">
    <head>The Two Things To Do</head>
    <list>
      <item>capture the text</item>
      <item>create the markup</item>
    </list>
    <p>Often they occur simultaneously; often not.</p>
    <p>Note that the markup does not necessarily have to be in the same</p>
  </div>
</body>
```
Adding An Element

(encoding description) documents the relationship between an electronic text and the source or sources from which it was derived.
Adding An Attribute

Supplies the date of the change in standard form, i.e. YYYY-MM-DD.
Surround With Element

the dominant dignity and absolute sway with which he lorded it in his little empire, the school, and became wonderfully gentle and ingratiating. He found favor in the eyes of the masters, by petting the children, particularly the youngest; and like the lion hold, which whilst so magnanimously the lamb did hold, he would sit with a child on one knee, and rock a cradle with his foot for whole hours together. </p>

<p>in addition to his other vocations, he was the singing-master of the neighborhood, and picked up many bright skillings by instructing the young folks in psalmody. It was a matter of no little vanity to him, on Sundays, to take his station in front of the church gallery, with a band of chosen singers; where, in his own mind, he completely carried away the palm from the parson. Certain it is, his voice resounded far above all the rest of the congregation; and there are peculiar quevers which the mill-pond, on a still morning, which were descended from the voice of Ichabod Crane. Thus, by that ingenious way which is commonly denominated "by hook and by crook," the worthy pedagogue got on tolerably enough, and was thought, by all who understood nothing of the labor of headwork, to have a wonderfully easy life of it. </p>

<p>The schoolmaster is generally a man of some importance in the female circle of a rural neighborhood; being considered a kind of idle gentlemenlike personage, of vastly superior taste and accomplishments to the rough country swains, and, indeed, inferior in learning only to the parson. His appearance, therefore, is apt to occasion some little stir at the tea-table of a farmhouse, and the addition of a supernumerary dish of cakes or sweetmeats, or, peradventure, the parade of a silver tea-pot. Our man of letters, therefore, was peculiarly happy in the smiles of all the country damsels. Now he would figure among them in the churchyard, between services on Sundays! gathering grapes for them from the wild vines that overrun the surrounding trees; reciting for their amusement all the epitaphs on the tombstones; or sauntering, with a whole bevy of them, along the banks of the adjacent mill-pond; while the more bashful country bumpkins hung backwards and forwards.
the dominant dignity and absolute sway with which he lorded it in his little empire, the school, and became wonderfully gentle and ingratiating. He found favor in the eyes of the mothers, by petting the children, particularly the youngest; and like the lion hold, which so magnanimously the lamb did hold, he would sit with a child on one knee, and rock a craddle with his foot for whole hours together. </p>

<p>in addition to his other vocations, he was the singing-master of the neighborhood, and picked up many bright shillings by instructing the young folks in psalmody. It was a matter of no little vanity to him, on Sundays, to take his station in front of the church gallery, with a band of chosen singers; where, in his own mind, he completely carried away the palm from the parson. Certain it is, his voice resounded far above all the rest of the congregation; and there are peculiar quavers still to be heard in that church, and which may even be heard half a mile off, quite to the opposite side of the mill-pond, on a still Sunday morning, which are said to be legitimately descends from the nose of <name>Ichabod Crane</name>. Thus, by divers little make-shifts in that ingenious way which is commonly denominated "by hook and by crook," the worthy pedagogue got on tolerably enough, and was thought, by all who understood nothing of the labor of headwork, to have a wonderfully easy life of it. </p>

<p>The schoolmaster is generally a man of some importance in the female circle of a rural neighborhood; being considered a kind of idle gentleman-like personage, of vastly superior taste and accomplishments to the rough country swains, and, indeed, inferior in learning only to the parson. His appearance, therefore, is apt to occasion some little stir at the tea-table of a farmhouse, and the addition of a supernumerary dish of cakes or sweetmeats, or, peradventure, the parade of a silver tea-pot. Our man of letters, therefore, was peculiarly happy in the smiles of all the country damsels. How he would figure among them in the churchyard, between services on Sundays! gathering grapes for them from the wild vines that overran the surrounding trees; reciting for their amusement all the epitaphs on the tombstones; or mauntering, with a whole bevy of them, along the
"Another Surround With Element"

```xml
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>XML Example</title>
  </head>
  <body>
    <div type="verse">
      ["Giving evidence recently b<br/>
      Commons, Miss C. E. Collet<br/>
      laundry was killing the small<br/>
      The little crafts! How soon they die! In cottage doors no shuttle clicks; The hand-loom has been ousted by A large concern with lots more sticks. The throb of pistons beats around; Great chimneys rise on Thames's banks; The same phenomena are found In Sheffield, (Yorks) and Oldham (Lancs). No longer now the housewife makes Her rare preserves, for what's the good? The factory round the corner fakes Raspberry jam with chips of wood. 'Tis so with what we eat and wear, Our bread, the boots wherein we splash 'Tis so with what I deemed most fair, Most virginal of all—the Wash. 'Tis this that chiefly, when I chant, Fills my breast with sighs of ruth,
    </div>
  </body>
</html>
```
Я помню чудное мгновенье:
Передо мной явилась ты,
Как мимолетное виение,
Как гений чистой красоты.

В томленьях грусти безнадежной,
В тревогах щемящей суеты,
Звучал мне долго голос нежный
И снискал мне мягкие черты.

Шли годы. Был порой мятежный
Рассеяны прежние мечты,
И я забыл твой голос нежный,
Твои небесные черты.

В глухой, во мраке заточень
Танулись тихо дни мои
Без божества, без вдохновенья,
Без слез, без жизни, без любви.

Душе настало пробужденье;
И вот опять явилась ты,
Как мимолетное виенье,
Как гений чистой красоты.

И сердце бьется в упоенье,
И для него воскресли вновь
И божество, вдохновенье,
И жизнь, и слезы, и любовь.
Or If You Generate Your TEI Schema In Chinese...
XPath Searching Built In

Example XML Document:

```xml
<p>The rest of this communication is omitted owing to considerations of space. — <i>Ed</i> </p>

<p>Greece. "<said>"ISN'T IT TIME WE STARTED FIGHTING</p>

<p>Turkey. "<said>"YES, I DARESAY. HOW SOON COULD YOU BEGIN</p>

<p>Greece. "<said>"OH, IN A FEW WEEKS"</said>"."</p>

<p>Turkey. "<said>"NO GOOD FOR ME. SHAN'T BE READY TILL THE</p>
```

XPath Query:

```
//figure[graphic width="800px"]
```

Results:

<table>
<thead>
<tr>
<th>Resource</th>
<th>System ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1914_july_1.xml</td>
<td>file://home/rahtz/TEI/tei.uocs.ox.ac.uk/Tel/1914_july_1.xml</td>
</tr>
</tbody>
</table>
Tagless editing in oXygen

The Legend of Sleepy Hollow
Washington Irving

THE LEGEND OF SLEEPY HOLLOW, FOUND AMONG THE PAPERS OF THE LATE DIEDRICH KNUCKLEBROCKER.
A pleasing land of dreamy head it was,
Of drearn that wave before the half-shot eye,
And of gay castles in a dream that pass,
For ever flushing red among the sky.
CASTLE OF INDOLENCCE.
Why use oXygen?

1. Is probably the best and most complete XML development IDE available.
2. Ready to use support for a large number of document types (including TEI).
3. Continuous and active development with proactive user community
4. Free support. oXygen provides a very responsive support for all its users free of charge.
5. Huge academic discounts and additional discounts for TEI members.
   There is a huge discount for the Academic licenses of oXygen, that costs $64 with the same set of features as the ‘Enterprise’ license that costs $543. TEI members benefit also from an additional 20% discount.
Exercise: Basic transcription

Now, if we have time, a quick demonstration of the kind of thing you are shortly going to be asked to do in the exercise.