Introduction to Course, Markup, XML, and the oXygen XML Editor

James Cummings

April 2009
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 TEI scheme, its assumptions, and its organization
4. Survey the whole landscape of the TEI recommendations
5. Demonstrate how the TEI scheme may be customized to particular needs
6. Demonstrate some real world applications of the TEI scheme
7. Provide routes into more detailed information for exploration at your leisure
8. Provide opportunities for questions and discussions relating to your own encoding needs and priorities
Course Structure

The times of each lecture in the course aren't written in stone and we will go as fast or as slow as students need, adapting as necessary. However the structure we will attempt to follow the structure below:

- 10:00 - 11:00: Lecture 1
- 11:00 - 11:30: Coffee Break
- 11:30 - 12:30: Lecture 2
- 12:30 - 13:00: Practical Exercise
- 13:00 - 13:30: Lecture 3
Day 1: Monday 13 April, 2009 -- Introductions

**Lecture 1:** Introduction to Course, Markup, XML, and the oXygen XML Editor

**Lecture 2:** The TEI, TEI Structure and Core Elements

**Practical:** Exercise: Editing XML in oXygen

**Lecture 3:** Tools for Editing and Publishing TEI Documents
Day 2: Tuesday 14 April, 2009 -- Metadata

Lecture 1: The TEI Header
Lecture 2: Manuscript Description and Facsimile
Practical: Exercise: Describing a Manuscript in TEI XML
Lecture 3: Marking Up Images
Day 3: Wednesday 15 April, 2009 -- Transcription and Pointing

**Lecture 1:** Transcription and Critical Apparatus

**Lecture 2:** Names / Dates / People / Places

**Practical:** Exercise: More TEI Editing

**Lecture 3:** Pointing, Linking, and Stand Off Markup
Day 4: Thursday 16 April, 2009 -- Corpora, Genres, and Glyphs

Lecture 1: Analysis, Speech, and Linguistics
Lecture 2: Verse, Drama, and Dictionaries
Practical: Exercise: Even More TEI Editing
Lecture 3: TEI, Unicode, and Non-standard Characters
Day 5: Friday 17 April, 2009 -- Using the TEI

Lecture 1: Documenting TEI Customisations
Lecture 2: Exploring the TEI Community
Practical: Using Roma
Lecture 3: Conclusions and Group Discussion
Course Materials

- **All** course materials including:
  - **All** slides from lectures (in TEI XML, HTML, and PDF)
  - **All** exercises (in TEI XML, HTML, and PDF)
  - **All** materials for the exercises
  - A PDF booklet combining all these with 'TEI Lite'

are available on the TEI @ Oxford website.

- The url is: http://tei.oucs.ox.ac.uk/Oxford/index.xml

- 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!)

- To save you downloading a huge zip with all the workshop materials, I'll now pass around a USB key or two for you to copy the 'materials' folder from onto your computer
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

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
In order to talk about texts, markup and encoding of texts, we need to understand what we mean by these basic concepts. When we talk about text encoding, what do we mean by a text? What is in a text and what assumptions do we make in reading them?
The Scene: A ship at sea; afterwards an uninhabited island.

ACT ONE

Scene I. On a ship at sea; a tempestuous noise of thunder and lightning heard.

Enter a Shipmaster and a Boatswain.

Master. Boatswain!
Boats. Here, master; what cheer?
Master. Good! Speak to th’ mariners; fall to ’t yarely, or we run ourselves aground; bestir, bestir.

Enter Mariners.

Boats. Heigh, my hearts! cheerly, cheerly, my hearts! yare, yare! Take in the topsail. Tend to th’ master’s whistle. Blow till thou burst thy wind, if room enough.

Enter Alonso, Sebastian, Antonio, Ferdinand, Gonzalo, and Others.

Alon. Good boatswain, have care. Where’s the master? Play the men.
Boats. I pray now, keep below.
Ant. Where is the master, boson?
Boats. Do you not hear him? You mar our labour; keep your cabins; you do

Cheerly, good hearts!—Out of our way, I say. [Exit.

Gon. I have great comfort from this fellow. Methinks he hath no drowning mark upon him; his complexion is perfect gallows. Stand fast, good Fate, to his hanging; make the rope of his destiny our cable, for our own doth little advantage. If he be not born to be hang’d, our case is miserable. [Exeunt.

Re-enter Boatswain.

Boats. Down with the topmast. Yare, lower, lower! Bring her to try wi’ th’ main-course. [A cry within] A plague upon this howling! They are louder than the weather or our office.

Re-enter Sebastian, Antonio, and Gonzalo.

Yet again! What do you here? Shall we give o’er, and drown? Have you a mind to sink?

Seb. A pox o’ your throat, you bawling, blasphemous, incharitable dog!

Boats. Work you, then.

Ant. Hang, cur; hang, you whoreson, inhuman wretch, who wouldst fain be
What's in a text (2)?
What's in a text (3)?

Hwæt wē Gār-Dēna in geār-dagum
bēod-cyninga þrym gefrūnon,
hū ē āþelingas ellen fremedon,
   Oft Scyld Scēfing sceāpēna þrēatum,
   5 monegum mægþum meodo-setla oftēah;
egsode Eorl[æ], syðdan ārest weard
fēasceaft funder; hē þæs frōfre gebād:
wēox under wolcnun, weorō-myndum þāh,
oðþæt him ēghwylc þāra ymb-sittendra
10 ofer hron-rāde ūryan scolde,
The ontology of text

Where is the text?

- in the shape of letters and their layout?
- in the original from which this copy derives?
- in the stories we read into it? or in its author's intentions?

A "text" is an abstraction, created by or for a community of readers. Markup encodes and makes concrete such abstractions.
Encoding of texts

- Texts are more than sequences of encoded glyphs
  - They have **structure** and **content**
  - They also have multiple **readings**
- Encoding, or markup, is a way of making these things explicit

Only that which is explicit can be reliably processed
Styles of markup

- In the beginning there was *procedural* markup
  RED INK ON; print balance; RED INK OFF
- which being generalised became *descriptive* markup `<balance type='overdrawn'>some numbers</balance>`
- also known as *encoding* or *annotation*

Descriptive markup allows for easier re-use of data
Some more definitions

- Markup makes explicit the distinctions we want to make when processing a string of bytes
- Markup is a way of naming and characterizing the parts of a text in a formalized way
- It's (usually) more useful to markup what we think things *are* than what they *look like*
What's the point of markup?

• To make explicit (to a machine) what is implicit (to a person)
• To add value by supplying multiple annotations
• To facilitate re-use of the same material
  • in different formats
  • in different contexts
  • by different users
Separation of form and content

- Presentational markup cares more about fonts and layout than meaning.
- Descriptive markup says what things are, and leaves the rendition of them for a separate step.
- Separating the form of something from its content makes its re-use more flexible.
- It also allows easy changes of presentation across a large number of documents.
Markup as a scholarly activity

• The application of markup to a document can be an intellectual activity

• In deciding what markup to apply, and how this represents the original, one is undertaking the task of an editor

• There is (almost) no such thing as neutral markup -- all of it involves interpretation

• Markup can assist in answering research questions, and the deciding what markup is needed to enable such questions to be answered can be a research activity in itself

• Good textual encoding is never as easy or quick as people would believe

• Detailed document analysis is needed before encoding for the resulting markup to be useful
What does markup capture?

Compare

```xml
<hi rend="dropcap">H</hi>&amp; WYN; ÆT WE GARDE <lb/>na in gear-dagum þeod-cyninga <lb/>þrym gefrunon, hu ða æþelingas <lb/>ellen fremedon. oft scyld scefing sceaph <add>na</add> <lb/>þreatum, moneg<expan>um</expan> mægþum meodo-setl <add>a</add> <lb/>of<damage> <desc>blot</desc> </damage>teah ...
</add>

and

```xml
<lg><l>Hwæt! we Gar-dena in gear-dagum</l><l>þeod-cyninga þrym gefrunon, </l><l>hu ða æþelingas ellen fremedon, </l></lg>

```xml
<lg><l>Oft Scyld Scefing sceaphena þreatum, </l><l>monegum mægþum meodo-setla ofteah; </l><l>egsode Eorle, syððan ærest wearþ </l><l>feasceaf funden...</l></lg>
```
A useful mental exercise

Imagine you are going to markup several thousand pages of complex material....

- Which features are you going to markup?
- Why are you choosing to markup this feature?
- How reliably and consistently can you do this?

Now, imagine your budget has been halved. Repeat the exercise!
Some alphabet soup

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGML</td>
<td>Standard Generalized Markup Language</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
<tr>
<td>DTD</td>
<td>Document Type Definition (or Declaration)</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheet</td>
</tr>
<tr>
<td>Xpath</td>
<td>XML Path Language</td>
</tr>
<tr>
<td>XSLT</td>
<td>eXtensible Stylesheet Language - Transformations</td>
</tr>
<tr>
<td>XQuery</td>
<td>XML Querying</td>
</tr>
<tr>
<td>RELAXNG</td>
<td>Regular Expression Language for XML (New Generation)</td>
</tr>
</tbody>
</table>

Oh, and then there's also **TEI**, the *Text Encoding Initiative*
An Introduction to XML

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 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
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 `;`
<?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
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
<?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`
The Doctype Declaration

You may sometimes find an optional "Document Type" declaration at the start of a document:

```xml
<?xml version="1.0" ?>
<!DOCTYPE greeting SYSTEM "greeting.dtd []">
```

- The DTD is one way of associating the document with its schema (but is not used by W3C or RELAX NG for this purpose)
- The DTD subset is used to provide declarations additional to those in the schema, for example for external files
- The DTD subset may be `internal`, `external`, or both

DTDs are now considered old-fashioned -- RELAX NG schemas are preferred.
The Tempest

<?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>
Example deconstructed: root node

```xml
<?xml version="1.0" encoding="utf-8" ?> <div n="1">
<!-- .... -->
</div>
```
<head>SCENE I. On a ship at sea: a tempestuous noise of thunder and lightning heard. </head>
Example deconstructed: stage direction and speech

```xml
<stage>Enter a Master and a Boatswain</stage>
<sp>
  <speaker>Master</speaker>
  <ab>Boatswain!</ab>
</sp>
```
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>`
  • `<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>`
  - `<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>
  • <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>text</bar></foo></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</text'></seg>`
  • `<seg type="text">some text<gap/></seg>`
  • `<seg type="text">some text< /seg>`
  • `<seg type="text">some text</Seg>`
Introduction to the Course

An Introduction to Textual Markup

An Introduction to XML

Introduction to the oXygen XML editor

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>`
  • `<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>`
  • `<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>`
  • `<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>`
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>`
  • `<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>`
  • `<seg type="text">some text</Seg>`
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 at the element level using @xml:lang

The @xml:id attribute is another W3C-defined attribute.
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.
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 $48 with the same set of features as the Professional license that costs $299. TEI members benefit also of an additional 20% discount.
Basic oXygen Editing
<front>

<head>THE LEGEND OF SLEEPY HOLLOW</head>
<head>FOUND AMONG THE PAPERS OF THE LATE DIEDRICH KNICKERBOCKER</head>

<body>

IN the bosom of one of those spacious coves 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
Surround With Element

1  <?xml version="1.0" encoding="UTF-8"?>
2  <div type="verse">
3    PROGRESS,
4
5   "Giving evidence recently before Commons, Miss C. E. Collet, an
6   official of the National Laundry, said the use of the machine
7   was killing the small
8
9   The little crafts! How soon they die.
10  In cottage doors no shuttle clicks;
11  The hand-loom has been ousted by
12  A large concern with lots more sticks.
13
14  The throbbing of pistons beats around;
15  Great chimneys rise on Thames's banks,
16  The same phenomena are found
17  In Sheffield, (Yorks) and Oldham (Lancs).
18
19  No longer now the housewife makes
20  Her rare preserves, for what's the good?
21  The factory round the corner fakes
22  Raspberry jam with chips of wood.
23
24  'Tis so with what we eat and wear,
25  Our bread, the boots wherein we splash
26  'Tis so with what I deemed most fair,
27  Most virginal of all—the Wash.
28
29  'Tis this that chiefly, when I chant,
30  Fulfils my breast with sighs of ruth,
Or With Russian Text

Я помню чудное мгновенье:
Передо мной явилась ты,
Как мимолетное виденье,
Как гений чистой красоты.

В томлениях грусти безнадежной,
В тревогах шумной суеты,
Звучал мне долго голос нежный,
И снились милые черты.

Шли годы. Буря порывы мятежный,
Рассеял прежние мечты,
И я забыл свой голос нежный,
Твои небесные черты.

В глуби, во мраке заточенья
Тянулись тихо дни мои
Без божества, без вдохновенья,
Без слез, без жизни, без любви.

Душе настало пробужденье:
И вот опять явилась ты,
Как мимолетное виденье,
Как гений чистой красоты.

И сердце бьется в упоенье,
И для него воскресли вновь
И божество, и вдохновенье,
И жизнь, и слезы, и любовь.
Adding An Element

The image shows an XML editor interface with an XML document opened in it. The content of the document is a TEI (Text Encoding Initiative) file, which is a standard for encoding digital texts. The document contains a section of Russian text, which seems to be a literary excerpt.

The XML code snippet shown highlights the `<teiHeader>` and `<fileDesc>` elements. The `<encodingDesc>` attribute is also visible, which documents the relationship between an electronic text and the source or sources from which it was derived.
Adding An Attribute

<fileDesc>
  <titleStmt>
    <title>Pushkin</title>
  </titleStmt>
  <publicationStmt>
    <p>Publication information</p>
  </publicationStmt>
  <sourceDesc>
    <p>Information about the source</p>
  </sourceDesc>
</fileDesc>

<description><change>
  supplies the date of the change in standard form, i.e. YYYY-MM-DD.
</change></description>

<text>
  <div type="poem">
    <head>K***</head>
    Я помню чудное мгновенье,
    Передо мной явилась ты,
    Как мимолетное виденье,
    Как гений чистой красоты.
  </div>
  <div type="poem">
    В тюмени грусти безнадежной,
    В тревогах шумной суеты,
    Звучал мне долго голос нежный,
    Не сны ми ныне сердце черты.
  </div>
</text>
Or If You Generate Your TEI Schema In Chinese...

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?oxygen RNGSchema="file:/TEI/Sourceforge/trunk/P5/test-zh.rnc" type="compact"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
<teiHeader>
  <fileDesc>
    <titleStmt></titleStmt>
  </fileDesc>
</teiHeader>
</TEI>
```

(背景描述) 提供文件非書目性方面的細節描述，更明確地，是所使用的語言及次要語言、在何種情況下製作、參與人員及其環境背景。
If You Really Hate Tags...

The legend of Sleepy Hollow
Washington Irving

THE LEGEND OF SLEEPY HOLLOW. FOUND AMONG THE PAPERS OF THE LATE DIEDRICH KNICKERBOCKER.
A pleasing land of drowsy head it was,
Of dreams that waver before the half-shut eye,
And of gay castles if visions that pass,
For ever fusing with the summer sky.
CASTLE OF INDOLENCE.
XPath Searching Built In