

Authoring with the TEI

Sebastian Rahtz

Oxford University

The problem

Authoring with
the TEI

Sebastian
Rahtz

- The TEI provides a wealth of elements arranged in classes and modules many of them are useful in writing new documents.
- Many elements are redundant, and a few new ones are needed.
- How do we decide what is needed, what extra can we do to make life easy for writers, and how do we write a TEI extension to describe the result?

Surveying current usage

Authoring with
the TEI

Sebastian
Rahtz

Looking at the XML files comprising
<http://www.oucs.ox.ac.uk> and
<http://www.oss-watch.ac.uk>:

Documents

1698

Elements

233840

Unique elements

122

Elements occurring more than 10 times

94

Note that TEI Lite defines 151 elements

Most popular elements

- 1 <ref>: 1714
- 2 <title>: 1766
- 3 <Filespec>: 1861
- 4 <figure>: 2178
- 5 <emph>: 3412
- 6 <date>: 3451
- 7 <lb>: 3678
- 8 <list>: 4369
- 9 <head>: 5910
- 10 <div>: 6240
- 11 <hi>: 7489
- 12 <row>: 10579
- 13 <p>: 24893
- 14 <cell>: 33342
- 15 <xref>: 37013
- 16 <item>: 37940

What do people want to do?

Authoring with
the TEI

Sebastian
Rahtz

- 1 Make lists
- 2 Refer to other resources
- 3 Make paragraphs
- 4 Make tables
- 5 Include graphics
- 6 Emphasize things
- 7 Refer to file names
- 8 Force line breaks (!)

What might be lacking?

Authoring with
the TEI

Sebastian
Rahtz

- Good facilities for code quoting
- Structured licence and copyright information
- Simplified lists
- Identifiers for computer-related things
- Simplified content models

Key decisions

- no numbered `<div>` elements
- no elements for marking-up transcribed text (`<gap>`, `<reg>` etc)
- simplified models for `<front>`, `<body>`, `<back>`, `<div>`, `<list>`
- new high-level `<uList>`, `<oList>` and `<glossList>` to replace old `<list>`
- add special elements for code, file paths etc
- some header extensions
- no more than 100 TEI elements

Elements in teiauth

97 elements available:

structural

`<back>`, `<body>`, `<div>`, `<divGen>`,
`<front>`, `<head>`, `<group>`, `<TEI>`,
`<text>`, `<titlePage>`, `<titlePart>`

header-related

`<authority>`, `<availability>`,
`<change>`, `<copyright>`, `<distributor>`,
`<editionStmt>`, `<fileDesc>`,
`<keywords>`, `<language>`, `<licence>`,
`<notesStmt>`, `<profileDesc>`,
`<projectDesc>`, `<publicationStmt>`,
`<quotation>`, `<reason>`, `<resp>`,
`<respStmt>`, `<revisionDesc>`,
`<seriesStmt>`, `<taxonomy>`,
`<teiHeader>`, `<titleStmt>`

Elements in teiauth (2)

block-level

`<ab>`, `<cell>`, `<figDesc>`, `<figure>`,
`<label>``<note>`, `<p>`, `<row>`, `<table>`,
`<verbatim>`

inline

`<abbr>`, `<anchor>`, `<binaryObject>`,
`<choice>`, `<command>`, `<date>`, `<emph>`,
`<equiv>`, `<foreign>`, `<formula>`, `<gi>`,
`<graphic>`, `<hi>`, `<index>`, `<indexTerm>`,
`<lb>`, `<measure>`, `<name>`, `<path>`, `<ptr>`,
`<q>`, `<ref>`, `<seg>`, `<software>`, `<term>`,
`<time>`, `<url>`

Elements in teiauth (3)

list-related

<gloss>, <glossList>, <item>, <oList>,
<uList>

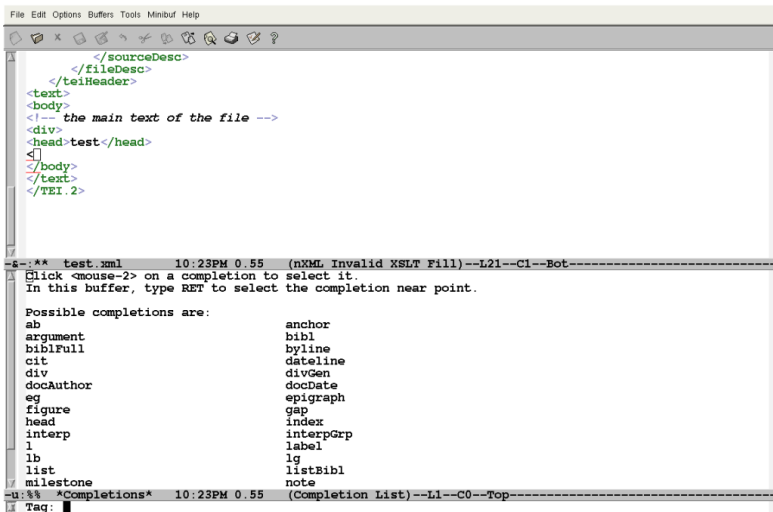
bibliography- and place-related

<addrLine>, <address>, <author>,
<bibl>, <biblItem>, <biblScope>,
<biblStruct>, <edition>, <editor>,
<extent>, <idno>, <listBibl>,
<monogr>, <postBox>, <postCode>,
<principal>, <pubPlace>, <publisher>,
<street>, <title>

Before: block-level choice

Authoring with
the TEI

Sebastian
Rahtz



The screenshot shows an XML editor window with a menu bar (File, Edit, Options, Buffers, Tools, Minibuf, Help) and a toolbar. The main text area contains the following XML code:

```
</sourceDesc>
</fileDesc>
</teiHeader>
<text>
<body>
<!-- the main text of the file -->
<div>
<head>test</head>
<
</body>
</text>
</TEI.2>
```

Below the text area is a status bar showing: `g-: ** test.xml 10:23PM 0.55 (nXML Invalid XSLT Fill)--L21--C1--Bot`. Below the status bar is a completion list window with the following text:

Click <mouse-2> on a completion to select it.
In this buffer, type RET to select the completion near point.

Possible completions are:

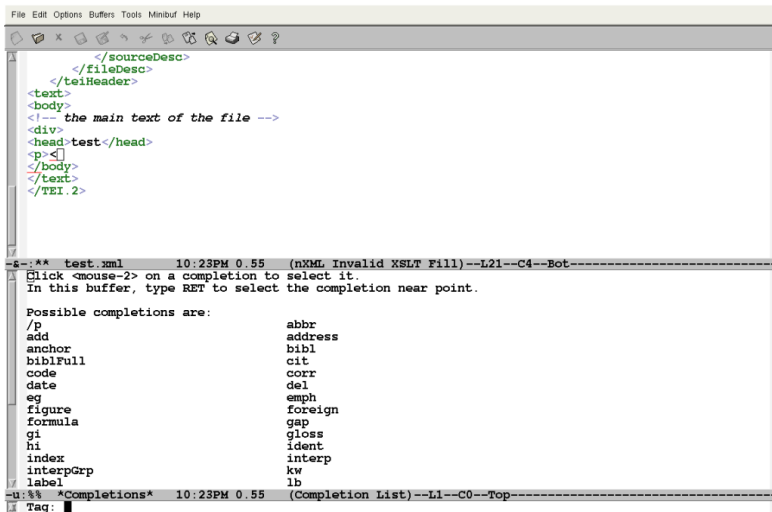
ab	anchor
argument	bibl
biblFull	byline
cit	dateline
div	divGen
docAuthor	docDate
eg	epigraph
figure	gap
head	index
interp	interpGrp
l	label
lb	lg
list	listBibl
milestone	note

At the bottom of the completion list window is a status bar showing: `-u: %% *Completions* 10:23PM 0.55 (Completion List)--L1--C0--Top`. Below the completion list window is a status bar showing: `Tag: █`.

Before: inline choice

Authoring with
the TEI

Sebastian
Rahtz



The screenshot shows a TEI editor window with a menu bar (File, Edit, Options, Buffers, Tools, Minibuf, Help) and a toolbar. The main text area contains the following XML code:

```
</sourceDesc>
</fileDesc>
</teiHeader>
<text>
<body>
<!-- the main text of the file -->
<div>
<head>test</head>
<p><input type="text" value=""/>
</body>
</text>
</TEI.2>
```

The status bar at the bottom indicates: `g--:** test.xml 10:23PM 0.55 (nXML Invalid XSLT Fill)--L21--C4--Bot`. Below the status bar, a message box says: "Click <mouse-2> on a completion to select it. In this buffer, type RET to select the completion near point." Below this, a list of possible completions is shown:

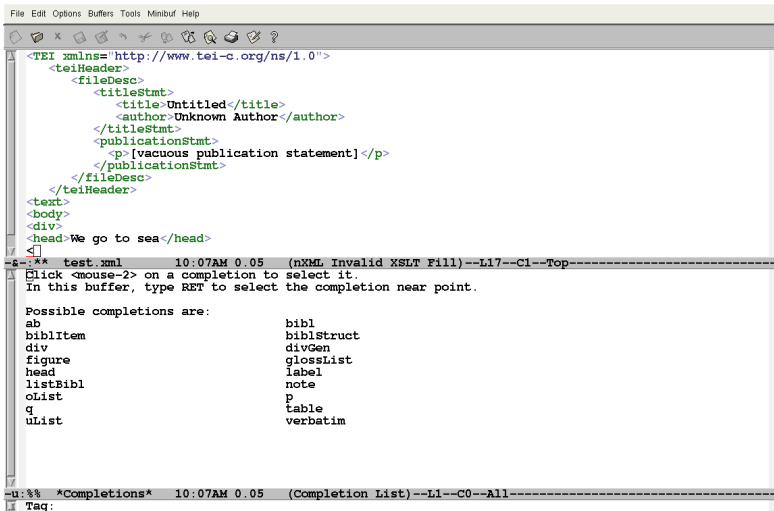
Possible completions are:	
/p	abbr
add	address
anchor	bibl
biblFull	cit
code	corr
date	del
eg	emph
figure	foreign
formula	gap
gi	gloss
hi	ident
index	interp
interpGrp	kw
label	lb

The status bar at the bottom indicates: `-u:%% *Completions* 10:23PM 0.55 (Completion List)--L1--C0--Top`. Below the status bar, a message box says: "Tag: []".

After: block-level choice

Authoring with
the TEI

Sebastian
Rahtz



The screenshot shows a text editor window with a menu bar (File, Edit, Options, Buffers, Tools, Minibuf, Help) and a toolbar. The editor contains the following XML code:

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>Untitled</title>
        <author>Unknown Author</author>
      </titleStmt>
      <publicationStmt>
        <p>[vacuous publication statement]</p>
      </publicationStmt>
    </fileDesc>
  </teiHeader>
  <text>
    <body>
      <div>
        <head>We go to sea</head>
      </div>
    </body>
  </text>
</TEI>
```

Below the code, a status bar shows: `test.xml 10:07AM 0.05 (nXML Invalid XSLT Fill)--L17--C1--Top`. A message box is displayed with the text: "Click <mouse-2> on a completion to select it. In this buffer, type RET to select the completion near point." Below the message, a list of possible completions is shown:

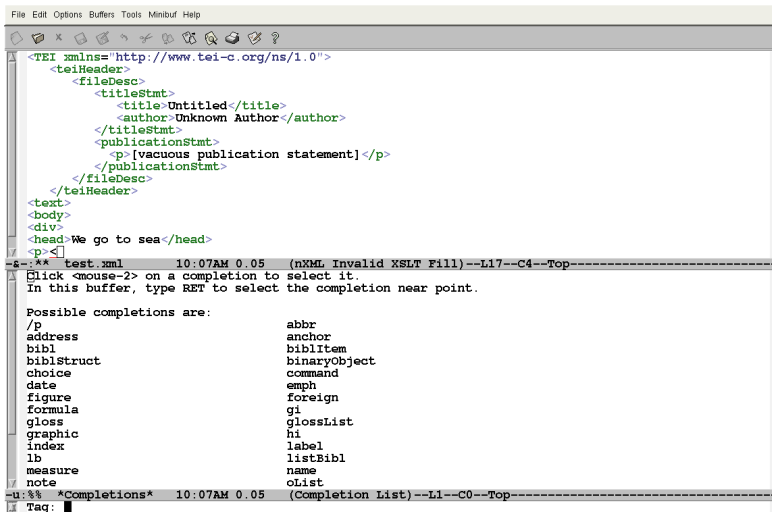
Possible completions are:	
ab	bibl
biblItem	biblStruct
div	divGen
figure	glossList
head	label
listBibl	note
oList	p
q	table
uList	verbatim

At the bottom, another status bar shows: `*Completions* 10:07AM 0.05 (Completion List)--L1--C0--All`. The word "Tag:" is visible at the very bottom left.

After: inline choice

Authoring with
the TEI

Sebastian
Rahtz



The screenshot shows a text editor window titled "test.xml" with a menu bar (File, Edit, Options, Buffers, Tools, Minibuf, Help) and a toolbar. The editor contains XML code for a TEI header and a text body. The code is as follows:

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>Untitled</title>
        <author>Unknown Author</author>
      </titleStmt>
      <publicationStmt>
        <p>[vacuous publication statement]</p>
      </publicationStmt>
    </fileDesc>
  </teiHeader>
  <text>
    <body>
      <div>
        <head>We go to sea</head>
        <p><|>
```

Below the code, a status bar shows "s-: ** test.xml 10:07AM 0.05 (nXML Invalid XSLT Fill)--L17--C4--Top". A tooltip is visible over the cursor, displaying the following text:

Click <mouse-2> on a completion to select it.
In this buffer, type RET to select the completion near point.

Possible completions are:

/p	abbr
address	anchor
bibl	bibliItem
bibliStruct	binaryObject
choice	command
date	emph
figure	foreign
formula	gi
gloss	glossList
graphic	hi
index	label
lb	listBibl
measure	name
note	oList

At the bottom, another status bar shows "u: %% *Completions* 10:07AM 0.05 (Completion List)--L1--C0--Top". The text "Tag: " is visible at the very bottom of the editor window.

The header, part 1

Authoring with
the TEI

Sebastian
Rahtz

```
<fileDesc>
  <titleStmt>
    <title>A beginners guide to TEI authoring</title>
  </titleStmt>
  <publicationStmt>
    <publisher>TEI project</publisher>
    <authority>Oxford University</authority>
    <availability>
      <licence>
        http://creativecommons.org/licenses/by-sa/2.0/uk/
      </licence>
    <copyright>
      <name>Sebastian Rahtz</name>
    </copyright>
  </availability>
  <date>2005-09-18</date>
</publicationStmt>
</fileDesc>
```

The header, part 2

Authoring with
the TEI

Sebastian
Rahtz

```
<revisionDesc vcdte="$Date: 2005/10/12 $" vcwho="$Author:
rahtz $" vcrevision="$Revision$">
  <change>
    <date>2005-09-18</date>
    <respStmt>
      <resp>author</resp>
      <name>Sebastian Rahtz</name>
    </respStmt>
    <reason>creation</reason>
  </change>
</revisionDesc>
```


Using SVG

Authoring with
the TEI

Sebastian
Rahtz

The `<graphic>` element has been extended to allow `<svg>` as a child element.

```
<figure>
  <head>An example of a drawing using SVG</head>
  <graphic>
    <svg width="11.487cm" height="3.564cm" viewBox="0.610
2.150 12.097 5.714">
      <polygon fill="#FFFFFF" stroke="none" stroke-width="0.100" p
4.900,2.200 2.800,5.550 "/>
      <polygon fill="none" stroke="#FF0035" stroke-width="0.100" p
4.900,2.200 2.800,5.550 "/>
      <polygon fill="#FFFFFF" stroke="none" stroke-width="0.100" p
12.000,5.664 9.725,2.450 "/>
      <polygon fill="none" stroke="#FF0035" stroke-width="0.100" p
12.000,5.664 9.725,2.450 "/>
    </svg>
  </graphic>
</figure>
```

Showing SVG

Authoring with
the TEI

Sebastian
Rahtz

oh dear ... no SVG implementation

So instead

Authoring with
the TEI

Sebastian
Rahtz

My holiday photograph instead



Simple example of ODD: delete elements

Authoring with
the TEI

Sebastian
Rahtz

```
<elementSpec ident="add" module="core" mode="delete"/>
<elementSpec ident="addName" module="namesdates" mode="delete"/>
<elementSpec ident="alt" mode="delete" module="linking"/>
<elementSpec ident="altGrp" mode="delete" module="linking"/>
<elementSpec ident="altIdent" module="core" mode="delete"/>
<elementSpec ident="analytic" mode="delete" module="core"/>
<elementSpec ident="bloc" module="namesdates" mode="delete"/>
<elementSpec ident="broadcast" mode="delete" module="header"/>
```

More complex ODD: simplify elements

Authoring with
the TEI

Sebastian
Rahtz

```
<elementSpec module="textstructure" mode="change" ident="back">
  <content>
    <rng:group>
      <rng:zeroOrMore>
        <rng:ref name="model.divWrapper"/>
      </rng:zeroOrMore>
      <rng:zeroOrMore>
        <rng:ref name="div"/>
      </rng:zeroOrMore>
      <rng:zeroOrMore>
        <rng:ref name="model.divWrapper.bottom"/>
      </rng:zeroOrMore>
    </rng:group>
  </content>
</elementSpec>
```

More complex ODD: new element

Authoring with
the TEI

Sebastian
Rahtz

```
<elementSpec ident="path">
  <equiv mimeType="text/xsl" filter="equivs.xsl" name="ident"/>
  <desc>A directory or file specification</desc>
  <classes>
    <memberOf key="model.pPart.data"/>
  </classes>
  <content>
    <rng:text/>
  </content>
  <exemplum>
    <egXML>
      <p>The grammar for Xaira command files
        is installed at <path>/usr/share/xaira</path>.</p>
    </egXML>
  </exemplum>
</elementSpec>
```

Tricksy ODD: syntactic sugar

Authoring with
the TEI

Sebastian
Rahtz

```
<elementSpec ident="uList">
  <equiv name="list" mime="text/xml" filter="equivs.xml"/>
  <gloss/>
  <desc>A sequence of items organized as an
    unordered list.</desc>
  <classes>
    <memberOf key="model.listLike"/>
  </classes>
  <content>
    <rng:zeroOrMore>
      <rng:ref name="item"/>
    </rng:zeroOrMore>
  </content>
</elementSpec>
```

What was that `<equiv>` thing?

Authoring with
the TEI

Sebastian
Rahtz

It pointed to a template in an XSL file:

```
<xsl:template name="list">
  <list>
    <xsl:copy-of select="@xml:id|@n"/>
    <xsl:attribute name="type">
      <xsl:choose>
        <xsl:when test="local-name()='oList'">ordered</xsl:when>
        <xsl:when test="local-name()='uList'">unordered</xsl:when>
        <xsl:when test="local-name()='glossList'">gloss</xsl:when>
        <xsl:otherwise>
          <xsl:message terminate="yes">
            <xsl:value-of select="local-name(.)"/> is mapped to
            "list", but I do not know what to do to with
it</xsl:message>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:attribute>
    </list>
  </xsl:template>
```


Resources for this talk

Authoring with
the TEI

Sebastian
Rahtz

- teauth.odd: ODD source of schema
- teiauth.xml: This talk
- teiauth.rng: Schema (Relax NG)
- teiauth.rnc: Schema (Relax NG compact)
- teiauth.xsd: Schema (W3C)
- xml.xsd: Support XSD file
- svg.xsd: Support XSD file
- xlink.xsd: Support XSD file
- oddtofilter.xsl: Creating transform to canonicalize
- equivs.xsl: Canonicalisation rules