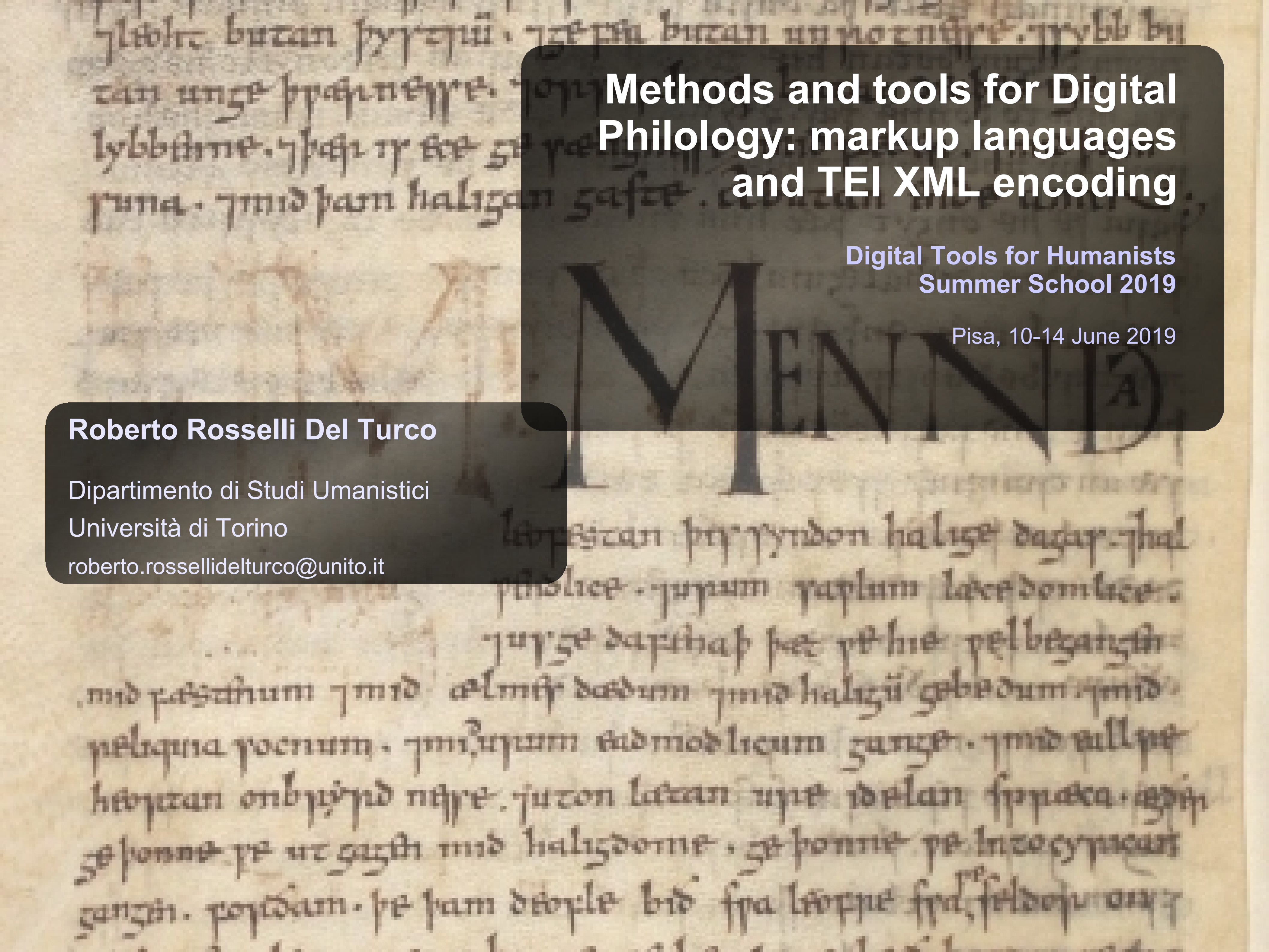


Methods and tools for Digital Philology: markup languages and TEI XML encoding

Digital Tools for Humanists
Summer School 2019

Pisa, 10-14 June 2019

Roberto Rosselli Del Turco

Dipartimento di Studi Umanistici
Università di Torino
roberto.rossellidelturco@unito.it

Markup languages

- there are many markup languages, which differ greatly
- fundamental distinction: **procedural** markup vs. **descriptive** markup
- procedural markup is typical of word processors:
  - instructions for specifying where the characters should appear on the page, their appearance, etc.
  - WYSIWYG approach, but also see LaTeX
  - the user doesn’t see or modify the markup directly (but again see LaTeX)
- descriptive markup describes text
- this distinction isn’t as neat as one would love to think, see for instance the structural aspect of text

Descriptive markup

- allows the scholar to do a semantic annotation of text
- the current standard is the XML language (← SGML)
  - in spite of the multiple hierarchies problem
- XML has been used to produce many different encoding schemas:
  - **TEI** schemas for all types of texts
  - TEI-derived schemas: EpiDoc, MEI, CEI, etc.
  - other schemas: DOCBOOK, MML – Music Markup Language, MathML, SVG, etc.
- it is also possible to create a personal encoding schema, but you would need a very good reason not to use TEI XML

Markup languages: XML

- SGML is the “father” of XML (*eXtensible Markup Language*)
- XML was created to replace both SGML, offering similar characteristics but a much lower complexity, and also HTML, going beyond the intrinsic limits of the latter
- XML was born as a “simplification” of SGML, of which it constitutes a subset, to be used on the WWW and other areas
- la **flessibilità** di questo linguaggio è tale da renderlo adatto a molti altri usi

XML general feature

- XML is a **standard** regulated by the W3C consortium (http://www.w3.org/), first version dates back to 1998 (updated in 2000)
- XML is completely **indipendent** from operating system, applications and hardware platform used
- XML is **extensible** by definition, there is no fixed number of elements as for HTML
- XML allows you to create, store and disseminate digital documents with the guarantee of long term durability since XML documents are just text documents
- XML is used for a wide range of purposes: document publishing, data transmission via the Internet, user interface description, etc.

XML markup: basic notions

- see above: text encoding is a method, the XML markup language the technical solution
- an **element** consists of two **tags** where the element name is delimited by angular brackets: **< ... >**
- the first is the *opening* tag, the second the *closing* one
- the closing tag is easily recognizable thanks to a slash immediately after the first bracket: **</ ... >**
- `<title>La Divina Commedia</title>`

opening tag **text** **closing tag**

XML markup: basic notions

- an element can include one or more **attributes**:

```
<line n="1">Nel mezzo del cammin di nostra vita</line>
```

- the `<line>` element includes an **n** attribute which has a value of one in this example
- attribute values have to be delimited by **"..."**
- besides elements, a marked up document can include other objects, such as comments:

```
<!-- Need to finish this part... -->
```

XML markup: semantic annotation

- who are going to encode a text for? even if XML is human readable, the final recipient will be some text processing **software**
- we use semantic annotation to make **explicit**, and thence **processable**, what is implicit for us
- in other words, we are using markup to annotate text in a **formal language**
- NLP has other goals, completely different approach

XML encoding

XML markup: semantic annotation

- markup languages can be used to perform a **descriptive** annotation of a text
- designed to be **inline** markup, but also stand-off
- annotation (= encoding) records important features of the text:
  - the position of the annotated fragment within the text body (**structural** markup)
  - features of the text fragment (**semantic** markup)
- **text is described for what it is, not for what it looks**

XML encoding

XML markup: semantic annotation

- see f.i. this sentence:

The *Guidelines for Electronic Text Encoding and Interchange* are *really* exhaustive. They allow you to understand how to prepare an *ad hoc* encoding schema for your project.

- same typographic style used for
  - a title
  - an emphasized word
  - a couple words in a foreign language

XML encoding

XML markup: semantic annotation

- using HTML we could encode it this way:

```
The <i>Guidelines for Electronic Text Encoding and
Interchange</i> are <em>really</em> exhaustive.
They allow you to understand how to prepare an
<i>ad hoc</i> encoding schema for your project.
```

- the <i>, <b> etc. HTML markups are real procedural instructions to indicate that those text strings should be rendered in italics, bold, etc.
- the <em> element, vice versa, has a semantic basis (it is an abbreviation for emphasis)

XML encoding

XML markup: semantic annotation

- using XML we could encode it this way:

```
The <title>Guidelines for Electronic Text
Encoding and Interchange</title> are
<emphasis>really</emphasis> exhaustive. They
allow you to understand how to prepare an
<foreignlanguage>ad hoc</foreignlanguage>
encoding schema for your project.
```

- XML tags describe text based on its **meaning**, not its appearance
- separation of content and form → style sheets

XML markup: semantic annotation

- descriptive markup must correspond to specific needs related to the study of a text
- the first step is to define an **encoding model** for the intended project:
  - what you are interested = want to mark in a text
  - what are the goals: research, visualization, manipulation and extraction of data, etc.?
- in the next step the encoding model will be “translated” into an **encoding schema**: the tools (elements etc.) to achieve the objectives

XML markup: basic notions

- XML elements can hold different kinds of content:
    - **structural** content: the element can contain other elements only, not text
    - **mixed** content: the element can contain both other elements and text
    - **textual** content: the element can contain only text, not other elements
- there can also be **empty** elements, devoid of any content: in the which case the opening tag also include closing sign, f.i. <pb/>, <gap/> etc.
- element content is defined in the encoding schema

Markup and web browsers

- if you try to load an XML document in a browser, the latter does not know how to display it, you need a style sheet
- but this is also intentional: content and presentation are meant to be **separate**
  - descriptive markup for processing, search, analysis, etc. functions.
  - style sheets for graphic rendering
- the markup, however, does not automatically establish an ontology of the text
- merely provides the scholar with the tools to do so

XML syntax

- an XML document ***must*** be **well formed**, i.e. it has to comply with the basic XML syntax rules:
  - only one root element
  - all branches go inside the root
  - always close open tags (exception: empty elements)
  - tag names are *case sensitive*
  - attribute values always go between quotes (' or ")
  - no tag overlap
- not well formed documents can't be processed

The encoding schema

- an encoding schema specifies:
  - the name of the root element
  - the names of all available elements
  - the names (and possible predefined values) of all attributes
  - which elements have which attributes
  - rules concerning the element hierarchy: which elements can go inside which ones
- it constitutes a sort of **grammar** for your documents
- checking an XML document against a schema is called **validation**

XML markup: basic notions

- an XML document starts with this line:

  ```
  <?xml version="1.0" encoding="utf-8"?>
  ```

  - a simple processing instruction declaring “I am an XML document!”
- after that you can have more processing instructions
  - usually the encoding schema declaration
- after which you find the root element
  - all other content fits into the root element
  - hierarchy more visible thanks to text indentation

XML encoding

```
<?xml version="1.0" encoding="UTF-8"?>

<bibliography>

  <entry type="book">
    <author>Research Library Group - Digital Library
          Federation</author>
    <title>Guides to Quality in Visual Resource Imaging</title>
    <pubplace>Washington, DC</pubplace>
    <publisher>Council on Library and Information
          Resources</publisher>
    <date>2000</date>
  </entry>

  <entry type="book">
    <author>Segre, C.</author>
    <title>Introduction to the Analysis of Literary Text</title>
    <pubplace>Bloomington</pubplace>
    <publisher>Indiana University Press</publisher>
    <date>1988</date>
  </entry>

</bibliography>
```

The Text Encoding Initiative

TEI – Text Encoding Initiative

- international consortium (http://www.tei-c.org/)
- motto: *TEI: Yesterday's information tomorrow*
- “an international and interdisciplinary standard that enables libraries, museums, publishers, and individual scholars to represent a variety of literary and linguistic texts for online research, teaching, and preservation”
- *Guidelines for Electronic Text Encoding and Interchange* (http://www.tei-c.org/Guidelines/)

TEI schemas

- highly modular infrastructure: hundreds of elements grouped in separate modules → pick and choose
- TEI P5 version (2007) introduces several improvements with regard to transcription of primary sources
- in particular new “Digital facsimiles” section
- other important changes:
  - new manuscript description module
  - new <choice> element
- still to be rewritten: Critical Apparatus module

TEI schemas

- a minimal TEI P5 document consists of:
  - XML declaration ed eventuali *processing instructions*
  - TEI schema declaration
  - structural elements
  - semantic elements
- even if you only use the basic modules, you have a powerful and flexible schema at your disposal
- recommended “light” schema: TEI Lite available on the TEI website (using Rome) and with Oxygen
  - “shortcut”: validate using TEI All on the Web → new document with XML Copy Editor and Oxygen

TEI basic modules

- the basic modules needed for (almost) any schema:

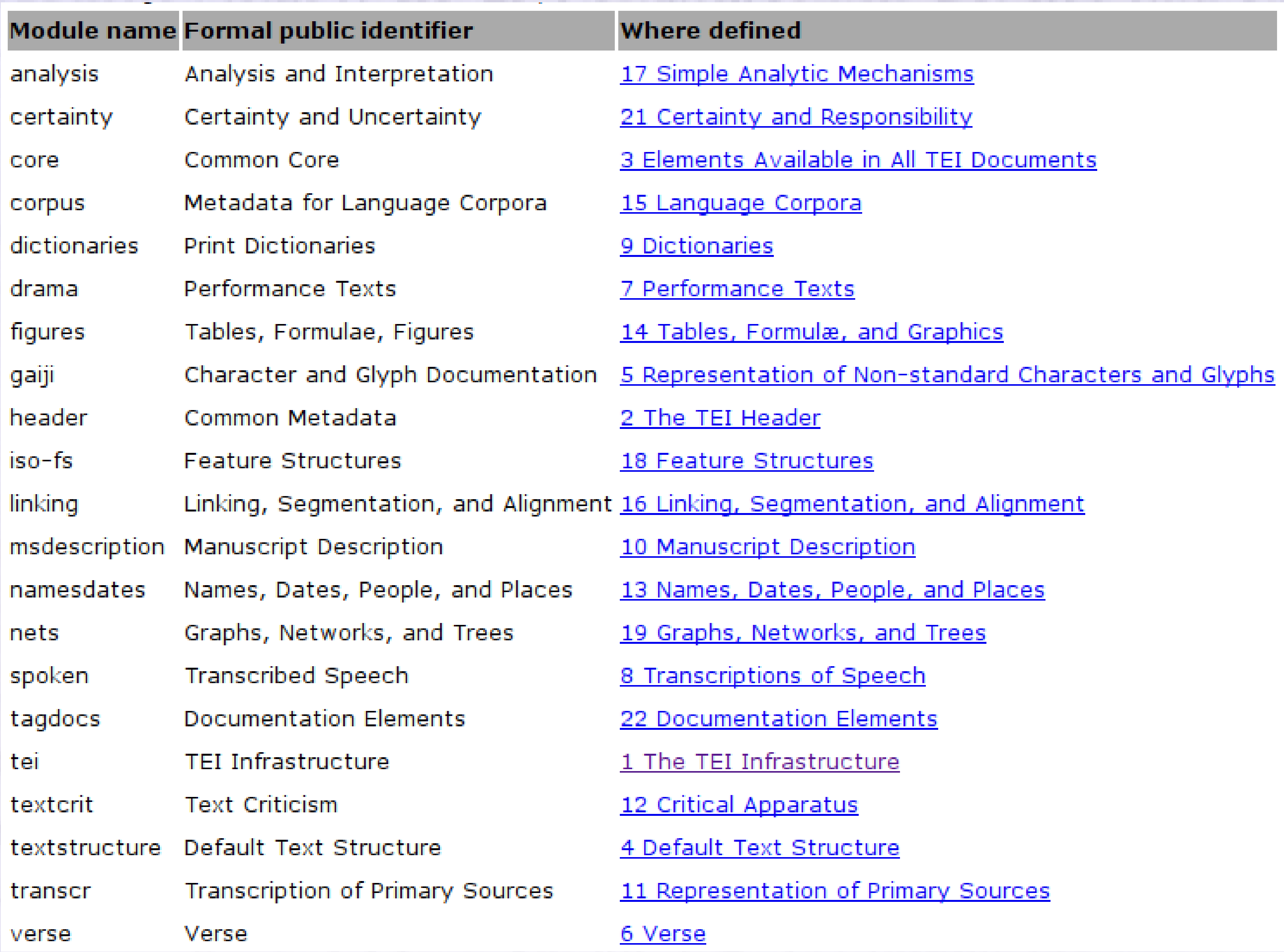
tei	element classes, datatypes, macros
header	metadata for the TEI document
textstructure	structural elements
core	elements for most types of document

- more details in chapter 1 of the Guidelines (*The TEI Infrastructure*) http://www.tei-c.org/release/doc/tei-p5-doc/en/html/ST.html)
- you can reduce the number of elements, and create even a smaller schema → careful!

TEI modules for digital philology

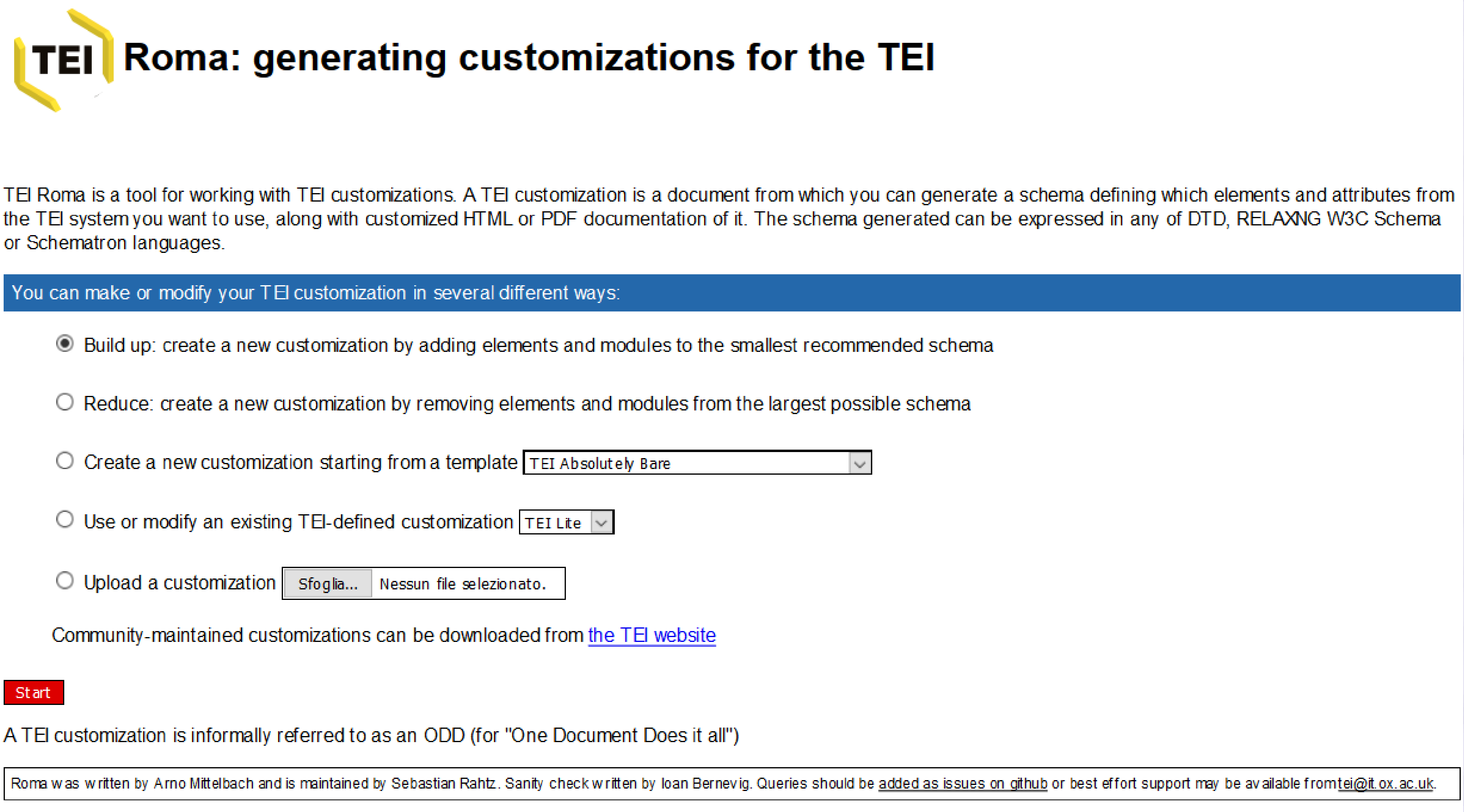
- manuscript description
  - http://www.tei-c.org/release/doc/tei-p5-doc/en/html/MS.html
- primary sources transcription
  - http://www.tei-c.org/release/doc/tei-p5-doc/en/html/PH.html
- critical apparatus
  - http://www.tei-c.org/release/doc/tei-p5-doc/en/html/TC.html
- non standard characters and glyphs
  - http://www.tei-c.org/release/doc/tei-p5-doc/en/html/WD.html
- editorial intervention elements of the core module
  - http://www.tei-c.org/release/doc/tei-p5-doc/en/html/CO.html#COED
- more modules: analysis, linking, namesdates

The Text Encoding Initiative



Module name	Formal public identifier	Where defined
analysis	Analysis and Interpretation	17 Simple Analytic Mechanisms
certainty	Certainty and Uncertainty	21 Certainty and Responsibility
core	Common Core	3 Elements Available in All TEI Documents
corpus	Metadata for Language Corpora	15 Language Corpora
dictionaries	Print Dictionaries	9 Dictionaries
drama	Performance Texts	7 Performance Texts
figures	Tables, Formulae, Figures	14 Tables, Formulæ, and Graphics
gaiji	Character and Glyph Documentation	5 Representation of Non-standard Characters and Glyphs
header	Common Metadata	2 The TEI Header
iso-fs	Feature Structures	18 Feature Structures
linking	Linking, Segmentation, and Alignment	16 Linking, Segmentation, and Alignment
msdescription	Manuscript Description	10 Manuscript Description
namesdates	Names, Dates, People, and Places	13 Names, Dates, People, and Places
nets	Graphs, Networks, and Trees	19 Graphs, Networks, and Trees
spoken	Transcribed Speech	8 Transcriptions of Speech
tagdocs	Documentation Elements	22 Documentation Elements
tei	TEI Infrastructure	1 The TEI Infrastructure
textcrit	Text Criticism	12 Critical Apparatus
textstructure	Default Text Structure	4 Default Text Structure
transcr	Transcription of Primary Sources	11 Representation of Primary Sources
verse	Verse	6 Verse

The Text Encoding Initiative



Roma: generating customizations for the TEI

TEI Roma is a tool for working with TEI customizations. A TEI customization is a document from which you can generate a schema defining which elements and attributes from the TEI system you want to use, along with customized HTML or PDF documentation of it. The schema generated can be expressed in any of DTD, RELAXNG W3C Schema or Schematron languages.

You can make or modify your TEI customization in several different ways:

- ● Build up: create a new customization by adding elements and modules to the smallest recommended schema
- ○ Reduce: create a new customization by removing elements and modules from the largest possible schema
- ○ Create a new customization starting from a template TEI Absolutely Bare
- ○ Use or modify an existing TEI-defined customization TEI Lite
- ○ Upload a customization Sfoglia... Nessun file selezionato.

Community-maintained customizations can be downloaded from the TEI website

A TEI customization is informally referred to as an ODD (for "One Document Does it all")

Roma was written by Arno Mittelbach and is maintained by Sebastian Rahtz. Sanity check written by Ioan Bernevig. Queries should be added as issues on github or best effort support may be available fromtei@it.ox.ac.uk.

TEI document structure

- ‘root’ element (**<TEI>**) holding:
  - metadata (**<teiHeader>**)
  - one or more text(s) (**<text>**) or a digital facsimile
- TEI header:
  - different types of document metadata
  - file description (**<fileDesc>**)
  - other information with regard to the encoding, document content, work revisions, and more
- it may also include introductory materials and other information to make it easier document interchange with other TEI projects

The TEI header

- minimal TEI header:

```
<teiHeader>
  <fileDesc>
   <titleStmt>...</titleStmt>
   <publicationStmt>...</publicationStmt>
   <sourceDesc>...</sourceDesc>
  </fileDesc>
</teiHeader>
```

- essential metadata holding title, publishing details and the original source (if any) of a document
- they allow to archive and handle documents on a bibliographic level

The Text Encoding Initiative

TEI header example

```
<teiHeader>
  <fileDesc>
   <titleStmt>
     <title>La Divina Commedia: versione elettronica</title>
     <respStmt>
       <resp>Conversione TEI P5 a cura di</resp><name>M. Rossi</name>
     </respStmt>
   </titleStmt>
   <publicationStmt>
     <publisher>Università di Pisa</publisher><date>2002-11-07</date>
     <availability status="restricted"><p>Contattare il responsabile
       del progetto, vietata la riproduzione.</p></availability>
   </publicationStmt>
   <sourceDesc>
     <bibl><title>La Divina Commedia</title><author>Dante Alighieri
     </author><publisher>Mondadori</publisher><date>1988</date></bibl>
   </sourceDesc>
 </fileDesc>
</teiHeader>
```

Structural elements

- **<text>** a single text, any kind of text
  - starting point for actual content hierarchy
  - can be preceded or replaced by a **<facsimile>**
- within <text> we can find four elements:
- **<front>** stuff preceding the text (if present)
- **<body>** the actual text
- **<back>** stuff following the text (if present)
- but a more complex structure is possible:
- **<group>** alternative to <body>, it containts different texts therefore holding a series of <text>s

TEI unitary document

- simple example of a TEI document with one text:

```
<?xml version="1.0" encoding="utf-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader> [metadata] </teiHeader>
  <text>
    <front> [title page, preface, ...] </front>
    <body>  [text body ...] </body>
    <back>  [commentary, appendix ...]</back>
  </text>
</TEI>
```

TEI composite document

```
<TEI>
 <teiHeader> [ header of the composite text ] </teiHeader>
  <text>
   <front> [ front matter of the composite text ] </front>
    <group>
     <text>
       <front> [ front matter of the first text ] </front>
       <body>  [ body of the first text  ]        </body>
       <back>  [ back matter of the first text ]  </back>
     </text>
     <text>
       <front> [ front matter of the second text]  </front>
       <body>  [ body of the second text  ]        </body>
       <back>  [ back matter of the second text ]  </back>
     </text>
         ...   [ more texts or groups of texts ]   ...
    </group>
   <back>      [ back matter of the composite text  ] </back>
  </text>
</TEI>
```

Other basic structural elements

- general text divisions: **<div>**
  - no nesting limit
  - different types can be specified
- paragraphs: **<p>**
- quotations: **<q>, <quote>** (direct speech, citations, etc.)
- poetry: stanzas **<lg>** and single lines **<l>**
- dramatic texts: speeches **<sp>** which can include paragraphs <p> or lines <l>, and stage directions **<stage>**
- *milestone tags*: **<pb/>**, **<lb/>**, **<cb/>**, **<milestone/>**
- note that a <div> may contain a **<floatingText>**: possibility to add complex hierarchies

Global attributes

- some attributes can be used by **any** element (see the *att.global* attribute class), in particular:
  - **n** a number or a string of characters to identify an element
  - **rend** information about original text rendering
  - **rendition** similar to @rend, but it points to <rendition> elements in the <encodingDesc> section
  - **xml:lang** language of the text inside an element
  - **xml:id** a identifier for an element
- **NB**: depending on the modules composing a TEI schema more global attributes may be available

Use of global attributes 1

```
<text>
  <body>
    <div n="ch1" type="chapter">
      <pb n="1"/>[...]
      <p>[...] I wonder if you ever read <title
         rend="underline" xml:lang="fra">Les fleurs du
         mal</title>
           [...]</p>
      <p>[...] a remarkable example <foreign
         xml:lang="fra">savoir faire</foreign>
         [...]</p>
      [...]
    </div>
    [ more divs ... ]
  </body>
</text>
```

Use of global attributes 2

```
<text>
  <body>
    <div n="ch1" type="chapter"> <pb n="1"/> [...]
      <p n="1">[...] described elsewhere (see for
       instance <ref target="#Rossi94">Rossi 1994</ref>)
      </p> [...]
    </div>
                   [ more divs ... ]
    <div n="bib" type="bibliography">
       [...]
       <bibl xml:id="Rossi94">
         <author>Rossi, M.</author>[...]</bibl>
       [...]
    </div>
  </body>
</text>
```

Direct speech and quotations 1

- **<q>** text quoted from external sources: direct speech, quotations, etc.

```
La mia maestra della prima superiore mi salutò di sulla porta della classe e mi disse: <q rend="PRE mdash">Enrico, tu vai al piano di sopra, quest'anno; non ti vedrò nemmen più passare!</q>
```

- **<quote>** phrase attributed to external sources

```
<p>E allora disse: <q rend="PRE lsquo POST rsquo">Ecco come comincia la Divina Commedia: <quote>Nel mezzo del cammin di nostra vita / Mi ritrovai per una selva oscura</quote>.</q></p>
```

Direct speech and quotations 2

- **<said>** text thought or pronounced aloud
- **<cit>** a bibliographic citation

```
Lexicography has shown little sign of being
affected by the work of followers of J.R. Firth,
probably best summarized in his slogan, <cit>
  <quote>You shall know a word by the company it
  keeps.</quote>
  <ref target="firth1957">(Firth, 1957)</ref>
</cit>
```

- using the **@target** attribute it is possible to link the <ref> to a bibliographic entry (see above)

Frequent mistakes 1

- careful when compilining the <fileDesc> element
  - it must be used to provide information about the actual document
  - title and author may be the same as those of an original document, but other information differs
  - it must **not** include parts of the text
- titles are encoded with <title> **only** when they are bibliographic titles
  - titles and headings in the text are encoded using the **<head>** element

Frequent mistakes 2

- <div>s can’t be used alternating it with <p>s at the same hierarchical level:

```
<div> […] </div>
<p> […] </p>            ← INVALID!!!
<div> […] </div>
```

  - you can have a <p> before <div>s though

- <div> and all other structural elements cannot contain text:

```
<div>some text</div>    ← INVALID!!!
<person>Mary</person>  ← INVALID!!!
```

A note about possible mistakes

- while encoding a text possible mistakes fall into three categories:
  - **syntactical** errors: an element put in the wrong place in the hierarchy, text inside a structural element, etc.
  - **markup** errors: using an element unfit for the purpose, e.g. <emph> instead of a <title>
  - text **interpretation** errors: semantic markup is wrong or actually lacking
- first type errors are the "best ones" because are easy to spot and correct, last type are the most difficult to find out
  - need competences with regard to content, not simply technical skill about text encoding

Associating an encoding schema

- associating a pattern to a document is essential to validate it
- association of a schema to the TEI XML document:
  - XML Copy Editor: XML → Associate → System DTD
  - Oxygen: Document → Schema → Associate Schema...
  - by manually entering the <!DOCTYPE> if you use a DTD, or the processing instruction **<?xml-model>** described in the chapter A Gentle Introduction to XML (https://www.tei-c.org/release/doc/tei-p5-doc/en/html/SG.html)

```
<?xml-model href="tei-lite.rng"?>
```

- the **third** method is the best one wrt compatibility

Associating an encoding schema

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TEI SYSTEM "tei-lite.dtd">

<TEI xmlns="http://www.tei-c.org/ns/1.0">

   <teiHeader> ... </teiHeader>

   <text>
        <body>
            <p></p>
        </body>
   </text>

</TEI>
```

Associating an encoding schema

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-model href="tei-lite.rng"?>

<TEI xmlns="http://www.tei-c.org/ns/1.0">

   <teiHeader> ... </teiHeader>

   <text>
       <body>
            <p></p>
       </body>
   </text>

</TEI>
```