

Part 2: Introduction to the TEI and its default text structure, header, and core elements

March 8, 2008

Contents

1	Introduction to the TEI	3
1.1	Where did the TEI come from?	3
1.2	What can the TEI do for you?	3
1.3	Goals of the TEI	3
1.4	TEI Deliverables	4
1.5	Legacy of the TEI	4
2	TEI Infrastructure	4
2.1	What is a module?	4
2.2	Modules	4
2.3	Defining a TEI Schema	5
2.4	A Simple Customization	5
2.5	Even more customisation	5
2.6	The TEI Class System	6
2.7	Attribute Classes	6
2.8	att.global	6
2.9	Model Classes	7
2.10	Basic Model Class Structure	7
2.11	Macros	7
2.12	Datatype Macros	7
3	Default Text Structure and Header	8
3.1	Structure of a TEI Document	8
3.2	TEI basic structures (1)	8
3.3	TEI basic structures (1)	8
3.4	text	8
3.5	Composite texts	9
3.6	TEI basic structure (1)	9
3.7	TEI text structure (2)	9
3.8	A text usually has divisions	9
3.9	Divisions may have heads and trailers	9
3.10	Partial and composite divisions	10
3.11	div Example	10
3.12	numbered and unnumbered divs	10
3.13	Classes for divisions	10
3.14	model.divWrapper	11
3.15	model.divTopPart	11
3.16	model.divBottomPart	11

3.17	Grouped and Floating Texts	11
3.18	Grouped texts Example	11
3.19	Floating texts	12
3.20	Floating texts (2)	12
3.21	Floating text Example	12
3.22	Virtual divisions	13
3.23	The TEI Header	13
3.24	The Librarian's Header	13
3.25	Everyman's Header	14
3.26	TEI Header Structure	14
3.27	Example Header: Minimal required header	14
3.28	Example Header: TEI corpus	14
3.29	Types of content in the TEI header	15
3.30	File Description	15
3.31	The File Description	15
3.32	The Source Description	16
3.33	For Example	16
3.34	Association between header and text	16
3.35	Encoding Description	16
3.36	Profile Description	17
3.37	Classification Methods	17
3.38	Example	17
3.39	Detailed characterization of a text	17
3.40	Language and character set usage	18
3.41	Revision Description	18
3.42	Example	18
4	Elements Available in All TEI Documents	18
4.1	Paragraphs	19
4.2	Highlighting	19
4.3	Highlighting Examples	19
4.4	Quotation	20
4.5	Simple Editorial Changes: choice and Friends	20
4.6	Choice Example	20
4.7	Additions, Deletions, and Omissions	21
4.8	Example of add, del, gap, and unclear	21
4.9	Basic Names	21
4.10	Basic Names Example	21
4.11	Addresses	21
4.12	Basic Address Example	22
4.13	Basic Numbers and Measures	22
4.14	Number and Measure Example	22
4.15	Dates	22
4.16	Simple Linking	23
4.17	Simple Linking Example	23
4.18	Lists	23
4.19	Simple List Example	23
4.20	Notes	24
4.21	Note Example	24
4.22	Indexing	24
4.23	Indexing Example	24

4.24 Graphics	25
4.25 Bibliographic Citations	25
4.26 Simple bibl Example	25
4.27 Simple biblStruct Example	25
4.28 Verse	26
4.29 Drama	26
4.30 Sleepy Hollow: The Play	26

1 Introduction to the TEI

This session will look at the Text Encoding Initiative, a project and then international consortium which has been going since 1987 to promote best practice in text encoding.

1.1 Where did the TEI come from?

- Originally, a research project within the humanities
 - Sponsored by three professional associations
 - Funded 1990-1994 by US NEH, EU LE Programme etc.
- Major influences
 - digital libraries and text collections
 - language corpora
 - scholarly datasets
- International consortium established June 1999 (see <http://www.tei-c.org/>)

1.2 What can the TEI do for you?

The TEI provides a framework for the definition of multiple schemas

- it defines and names several hundred useful textual distinctions
- it provides a set of modules that can be used to define schemas making those distinctions
- it provides a customization mechanism for modifying and combining those definitions with new ones using the same conceptual model
- it provides transformation stylesheets for default rendering in HTML or PDF

1.3 Goals of the TEI

- better interchange and integration of scholarly data
- support for all texts, in all languages, from all periods
- guidance for the perplexed: **what** to encode — hence, a user-driven codification of existing best practice
- assistance for the specialist: **how** to encode — hence, a loose framework into which unpredictable extensions can be fitted

These apparently incompatible goals result in a highly flexible, modular, environment

1.4 TEI Deliverables

- A set of recommendations for text encoding, covering both generic text structures and some highly specific areas based on (but not limited by) existing practice
- A very large collection of element **definitions** with associated **declarations** for various schema languages
- A modular system for creating personalized schemas or DTDs from the foregoing
- Software that transforms TEI documents

for the full picture see <http://www.tei-c.org/TEI/Guidelines/>

1.5 Legacy of the TEI

- a way of looking at what 'text' *really* is
- a codification of current scholarly practice
- (crucially) a set of shared assumptions and priorities about the digital agenda:
 - focus on content and function (rather than presentation)
 - identify generic solutions (rather than application-specific ones)

2 TEI Infrastructure

- The TEI encoding scheme consists of a number of modules
- These declare XML elements and their attributes
- An element's declaration assigns it to one (or more) model classes
- Another part declares its possible content and attributes with reference to these classes
- This indirection allows strength and flexibility
- It makes it easy to add/exclude new elements by referencing existing classes

2.1 What is a module?

- A convenient way of grouping together a number of element declarations
- These are usually on a related topic or specific application
- Most chapters focus on elements drawn from a single module, which that chapter then defines
- A TEI Schema is created by selecting modules and add/removing elements from them as needed

2.2 Modules

Module name	Chapter
analysis	Simple Analytic Mechanisms
certainty	Certainty and Responsibility
core	Elements Available in All TEI Documents
corpus	Language Corpora
dictionaries	Dictionaries
drama	Performance Texts
figures	Tables, Formulae, and Graphics

gaiji	Representation of Non-standard Characters and Glyphs
header	The TEI Header
iso-fs	Feature Structures
linking	Linking, Segmentation, and Alignment
msdescription	Manuscript Description
namesdates	Names, Dates, People, and Places
nets	Graphs, Networks, and Trees
spoken	Transcriptions of Speech
tagdocs	Documentation Elements
tei	The TEI Infrastructure
textcrit	Critical Apparatus
textstructure	Default Text Structure
transcr	Representation of Primary Sources
verse	Verse

2.3 Defining a TEI Schema

- A schema helps you know a document is valid in addition to being well-formed
- A TEI schema is a combination of TEI modules, optionally including customizations of the elements/attributes/classes that they contain
- This schema is defined in an application-independent manner with a TEI ODD (one document does it all) file which allows for:
 - creation of a schemas such as DTD, RelaxNG or W3C Schema
 - internationalized documentation which reflects your customization of the TEI
 - documentation of how your schema differs from `tei_all` that is suitable for long-term preservation
- (But we will discuss this in more detail in session 5)

2.4 A Simple Customization

A TEI ODD file can contain as much discursive prose as you want, but needs a `<schemaSpec>` element to define the schema it documents

```
<schemaSpec ident="TEI-minimal" start="TEI">
  <moduleRef key="tei"/>
  <moduleRef key="header"/>
  <moduleRef key="core"/>
  <moduleRef key="textstructure"/>
</schemaSpec>
```

2.5 Even more customisation

```
<schemaSpec ident="Chaucer-MoL" start="TEI">
  <moduleRef key="tei"/>
  <moduleRef key="header"/>
  <moduleRef key="core"/>
  <moduleRef key="textstructure"/>
  <moduleRef key="namesdates"/>
```

```

<moduleRef key="transcr"/>
<!-- We don't need these drama elements: -->
<elementSpec ident="sp" mode="delete" module="core"/>
<elementSpec ident="speaker" mode="delete" module="core"/>
<elementSpec ident="stage" mode="delete" module="core"/>
</schemaSpec>

```

2.6 The TEI Class System

- The TEI distinguishes over 500 elements,
- Having these organised into classes aids comprehension, modularity, and modification.
- Attribute class: the members share common attributes
- Model class: they can appear in the same locations (and often are structurally or semantically related)
- Classes may contain other classes
- Elements inherit the properties from any classes of which they are members

2.7 Attribute Classes

- Attribute classes are given (usually adjectival) names beginning with **att**.
- Members of the **att.naming** class get a key attribute rather than have them define it individually
- If another element needs a key attribute then the easiest way to provide it is to add it to the **att.naming** class
- Classes can be grouped together into a super classes

2.8 att.global

The attributes provided by **att.global** include among others:

xml:id a unique identifier

xml:lang the language of the element content

n a number or name for an element

rend how the element in question was rendered or presented in the source text.

And **att.global** also contains **att.global-linking** so if the 'linking' module is loaded it provides attributes such as:

corresp points to elements that correspond to the current element in some way

copyOf points to an element of which the current element is a copy

next points to the next element of a virtual aggregate of which the current element is part.

prev points to the previous element of a virtual aggregate of which the current element is part

2.9 Model Classes

- Model classes contain groups of elements allowed in the same place
- If you are adding an element which is wanted wherever the `<bibl>` is allowed, we simply add it to the **model.biblLike** class
- Model classes are usually named with a **Like** or **Part** suffix:
 - **model.divLike**: structural class grouping elements for divisions
 - **model.divPart**: structural class grouping elements used inside divisions
 - **model.nameLike**: semantic class grouping name elements
 - **model.persNamePart**: semantic sub-class grouping elements that are part of a personal name

2.10 Basic Model Class Structure

The TEI class system makes a threefold division of elements:

divisions high level major divisions of texts

chunks elements such as paragraphs appearing within texts or divisions, but not other chunks

phrase-level elements elements such as highlighted phrases which can occur only within chunks

The TEI identifies the following groupings from these three:

inter-level elements elements such as lists which can appear either in or between chunks

components elements which can appear directly within texts or text divisions

2.11 Macros

macro.paraContent content of paragraphs and similar elements

macro.limitedContent content of prose elements that are not used for transcription of extant materials

macro.phraseSeq a sequence of character data and phrase-level elements

macro.phraseSeq.limited a sequence of character data and those phrase-level elements that are not typically used for transcribing extant documents

macro.specialPara the content model of elements which either contain a series of component-level elements or else contain a series of phrase-level and inter-level elements

2.12 Datatype Macros

data.key a coded value

data.word a single word or token

data.name an XML Name

data.enumerated a single XML name taken from a documented list

data.duration.w3c a W3C duration

data.temporal.w3c a W3C date

data.truthValue a truth value (true/false)

data.language a language

data.sex human or animal sex

3 Default Text Structure and Header

Two of the major modules used in all TEI documents (the other one is Core)

- Text Structure
- TEI Header

3.1 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.

3.2 TEI basic structures (1)

```
<teiCorpus>
  <teiHeader>
  <!-- required -->
  </teiHeader>
  <TEI>
  <!-- required -->
  </TEI>
</teiCorpus>
```

3.3 TEI basic structures (1)

```
<TEI>
  <teiHeader>
  <!-- required -->
  </teiHeader>
  <facsimile>
  <!-- optional, new in TEI P5 -->
  </facsimile>
  <text>
  <!-- required if no facsimile -->
  </text>
</TEI>
```

3.4 `<text>`

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

3.5 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.

3.6 TEI basic structure (1)

```
<text>
<front>
<!-- optional -->
</front>
<body>
<!-- required -->
</body>
<back>
<!-- optional -->
</back>
</text>
```

3.7 TEI text structure (2)

```
<text>
<front>...</front>
<group>
<front>...</front>
<text>
<!-- required and contains <body> etc. -->
</text>
<back>...</back>
</group>
<back>...</back>
</text>
```

3.8 A text usually has divisions

`<div>`

- generic, hierarchic subdivisions, each incomplete
- the type attribute is used to label a particular level e.g. as 'part' or 'chapter'
- the n attribute gives a particular division a name or number
- the xml:id attribute gives a particular division a unique identifier

3.9 Divisions may have heads and trailers

```
<div>
  <head>Chapter 1</head>
  <!-- content of the div -->
  <trailer>...</trailer>
</div>
```

3.10 Partial and composite divisions

In particular where dealing with unusually large or unusually small texts, encoders may find it convenient to present as textual divisions sequences of text which are incomplete with reference to the original text, or which are in fact an ad hoc agglomeration of tiny texts.

The `org`, `sample` and `part` attributes from `att.divLike` are available to indicate such structural anomalies

- `org` = how the content of the `div` is organized (composite or unitary)
- `sample` = indicates whether this division is a sample of the original source and if so, from which part.
- `part` = whether or not the division is fragmented by some other structural element (for example, a speech divided among one or more verse stanzas)

3.11 <div> Example

```
<div n="2" sample="initial" type="postScript">
  <head>The Legend of Sleepy Hollow</head>
  <head>Postscript</head>
  <p rend="center"> FOUND IN THE HANDWRITING OF MR. KNICKERBOCKER</p>
  <p>THE PRECEDING Tale is given, almost in the precise words in which I
  heard it related at a Corporation meeting of the ancient city of Manhattoes,
  at which were present many of its sagest and most illustrious burghers.
  <gap reason="sampling"/>
  </p>
</div>
```

3.12 numbered and unnumbered divs

The level can be made explicit by using 'numbered' divs (`div1`, `div2`). Opinions vary:

`<div1>` vs. `<div n="1">`

- numbered: the number indicates the depth of this particular division within the hierarchy, the largest such division being 'div1', any subdivision within it being 'div2', etc.
- unnumbered: nest recursively to indicate their hierarchic depth. (And computers can count very well!)

The two styles must not be combined within a single `<front>`, `<body>`, or `<back>` element.

N.B. Divisions always tessellate

3.13 Classes for divisions

The TEI architecture defines five classes, all of which are populated by this module:

- `model.divTop` groups elements appearing at the beginning of a text division.
- `model.divTopPart` groups elements which can occur only at the beginning of a text division.
- `model.divBottom` groups elements appearing at the end of a text division.

- model.divBottomPart groups elements which can occur only at the end of a text division.
- model.divWrapper groups elements which can appear at either top or bottom of a textual division.

3.14 model.divWrapper

<argument> A formal list or prose description of the topics addressed by a subdivision of a text.

<byline> contains the primary statement of responsibility given for a work on its title page or at the head or end of the work.

<dateline> contains a brief description of the place, date, time, etc. of production of a letter, newspaper story, or other work, prefixed or suffixed to it as a kind of heading or trailer.

<docAuthor> (document author) contains the name of the author of the document, as given on the title page (often but not always contained in a byline).

<docDate> (document date) contains the date of a document, as given (usually) on a title page.

<epigraph> contains a quotation, anonymous or attributed, appearing at the start of a section or chapter, or on a title page.

3.15 model.divTopPart

<head> (heading) contains any type of heading, for example the title of a section, or the heading of a list, glossary, manuscript description, etc.

<salute> (salutation) contains a salutation or greeting prefixed to a foreword, dedicatory epistle, or other division of a text, or the salutation in the closing of a letter, preface, etc.

<opener> groups together dateline, byline, salutation, and similar phrases appearing as a preliminary group at the start of a division, especially of a letter.

model.divTop = model.divTopPart + model.divWrapper

3.16 model.divBottomPart

<closer> groups together salutations, datelines, and similar phrases appearing as a final group at the end of a division, especially of a letter.

<signed> (signature) contains the closing salutation, etc., appended to a foreword, dedicatory epistle, or other division of a text.

<trailer> contains a closing title or footer appearing at the end of a division of a text.

<postscript> contains a postscript, e.g. to a letter.

model.divBottom = model.divBottomPart + model.divWrapper

3.17 Grouped and Floating Texts

The **<group>** element should be used to represent a collection of independent texts which is to be regarded as a single unit for processing or other purposes.

<floatingText> contains a single text of any kind, whether unitary or composite, which interrupts the text containing it at any point and after which the surrounding text resumes.

3.18 Grouped texts Example

```
<TEI>
  <teiHeader>
<!-- header information for the whole collection -->
  </teiHeader>
  <text>
<!-- optional front matter -->
```

```

<group>
  <text>
<!-- optional front matter -->
  <body>
<!-- First Body -->
  </body>
</text>
  <text>
<!-- optional front matter -->
  <body>
<!-- Second Body-->
  </body>
</text>
</group>
</text>
</TEI>

```

3.19 Floating texts

As mentioned above, <div>s must tessellate over the entire text

```

<div1>
  <div2>
<!-- content -->
  </div2>
  <div2>
<!-- content -->
  </div2>
</div1>

```

is valid, while

```

<div1>
<!-- content -->
  <div2>
<!-- content -->
  </div2>
<!-- content -->
</div1>

```

is not valid.

3.20 Floating texts (2)

In the second case, div2 is a 'floating' text and its content must be encoded using the <floatingText> element.

The <floatingText> element is a member of the model.divPart class, and can thus appear within any division level element in the same way as a paragraph.

3.21 Floating text Example

```

<p>She was thus ruminating, when a Gentleman enter'd the Room, the Door being
a jar...
calling for a Candle, she beg'd a thousand Pardons, engaged him to sit down,
and let her know, what had so long conceal'd him from her Correspondence.</p>
<pb n="5"/>
<floatingText>
  <body>
    <head>The Story of <hi>Captain Manly</hi>
    </head>
    <p>
<!-- Captain Manly's store here -->
    </p>
  </body>
</floatingText>
<pb n="37"/>
<p>The Gentleman having finish'd his Story ...
<!-- more -->
</p>

```

3.22 Virtual divisions

Where the whole of a division can be automatically generated, for example because it is derived from another part of this or another document, an encoder may prefer not to represent it explicitly but instead simply mark its location by means of a processing instruction, or by using the special purpose `<divGen>` element:

```

<front>
<!-- <titlePage>...</titlePage> -->
  <divGen type="toc"/>
  <div>
    <head>Preface</head>
    <p>...</p>
  </div>
</front>

```

(intended primarily for use in document production or manipulation, rather than in transcription of pre-existing material)

3.23 The TEI Header

The TEI header was designed with two goals in mind

- needs of bibliographers and librarians trying to document ‘electronic books’
- needs of text analysts trying to document ‘coding practices’ within digital resources

The result is that discussion of the header tends to be pulled in two directions...

3.24 The Librarian’s Header

- Conforms to standard bibliographic model, using similar terminology
- Organized as a single source of information for bibliographic description of a digital resource, with established mappings to other such records (e.g. MARC)
- Emerging code of best practice in its use, endorsed by major digital collections
- Pressure for greater and more exact constraints to improve precision of description: preference for structured data over loose prose

3.25 Everyman's Header

- Gives a polite nod to common bibliographic practice, but has a far wider scope
- Supports a (potentially) huge range of very miscellaneous information, organized in fairly ad hoc ways
- Many different codes of practice in different user communities
- Unpredictable combinations of narrowly encoded documentation systems and loose prose descriptions

3.26 TEI Header Structure

The TEI header has four main components:

- `<fileDesc>` (file description) contains a full bibliographic description of an electronic file.
- `<encodingDesc>` (encoding description) documents the relationship between an electronic text and the source or sources from which it was derived.
- `<revisionDesc>` (revision description) summarizes the revision history for a file.
- `<profileDesc>` (text-profile description) provides a detailed description of non-bibliographic aspects of a text, specifically the languages and sublanguages used, the situation in which it was produced, the participants and their setting. (just about everything not covered in the other header elements)

Only `<fileDesc>` is required; the others are optional.

3.27 Example Header: Minimal required header

```
<teiHeader>
<fileDesc>
  <titleStmt>
    <title>A title?</title>
  </titleStmt>
  <publicationStmt>
    <p>Who published?</p>
  </publicationStmt>
  <sourceDesc>
    <p>Where from?</p>
  </sourceDesc>
</fileDesc>
</teiHeader>
```

3.28 Example Header: TEI corpus

```
<teiCorpus>
  <teiHeader type="corpus">
<!-- corpus-level metadata here -->
</teiHeader>
  <TEI>
    <teiHeader type="text">
<!-- metadata specific to this text here -->
</teiHeader>
```

```

<text>
<!-- ... -->
</text>
</TEI>
<TEI>
  <teiHeader type="text">
<!-- metadata specific to this text here -->
  </teiHeader>
  <text>
<!-- ... -->
  </text>
</TEI>
</teiCorpus>

```

3.29 Types of content in the TEI header

- free prose
 - prose description: series of paragraphs
 - phrase: character data, interspersed with phrase-level elements, but not paragraphs
- grouping elements: specialized elements recording some structured information
- declarations: Elements whose names end with the suffix Decl (e.g. `subjectDecl`, `refsDecl`) enclose information about specific encoding practices applied in the electronic text.
- descriptions: Elements whose names end with the suffix Desc (e.g. `<settingDesc>`, `<projectDesc>`) contain a prose description, possibly, but not necessarily, organized under some specific headings by suggested sub-elements.

3.30 File Description

- has some mandatory parts:
 - `<titleStmt>`: provides a title for the resource and any associated statements of responsibility
 - `<sourceDesc>`: documents the sources from which the encoded text derives (if any)
 - `<publicationStmt>`: documents how the encoded text is published or distributed
- and some optional ones:
 - `<editionStmt>`: yes, electronic texts have editions too
 - `<seriesStmt>`: and they also fit into "series".
 - `<extent>`: how many floppy disks, CDs, gigabits?
 - `<notesStmt>`: nuff said

NB A "file" may actually correspond with several operating system files.

3.31 The File Description

- `<titleStmt>`: contains a mandatory `<title>` which identifies the electronic file (not its source!)
- optionally followed by additional titles, and by 'statements of responsibility', as appropriate, using `<author>`, `<editor>`, `<sponsor>`, `<funder>`, `<principal>` or the generic `<respStmt>`

- `<publicationStmt>`: may contain
 - plain text (e.g. to say the text is unpublished)
 - one or more `<publisher>`, `<distributor>`, `<authority>`, each followed by `<pubPlace>`, `<address>`, `<availability>`, `<idno>`

3.32 The Source Description

Most electronic texts were not 'born digital': their source/s need specification in traditional bibliographic style

- `<bibl>`, `<biblStruct>`
- (for texts which were born digital): `<biblFull>` may contain a nested `<fileDesc>`
- `<listBibl>` a list of the foregoing
- prose description
- more specialized elements are available for spoken texts (`<recordingStmt>` etc.) and for manuscripts (`<msDescription>`)

3.33 For Example

```
<sourceDesc>
  <bibl> "The Legend of Sleepy Hollow", published in The Works of Washington
  Irving (New
    York, Putnam, 1861)</bibl>
</sourceDesc>
```

3.34 Association between header and text

By default everything asserted by a header is true of the text to which it is prefixed. This can be overridden:

- as when a text header over-rides or amplifies a corpus-header setting
- when model.declarable elements are selected by means of the `decls` attribute (available on all model.declaring elements)
- using special purpose selection/definition elements e.g. `<catRef>` and `<taxonomy>` (see below)

Most components of the encoding description are declarable.

3.35 Encoding Description

`<encodingDesc>` groups notes about the procedures used when the text was encoded, either summarized in prose or within specific elements such as

- `<projectDesc>`: goals of the project
- `<samplingDecl>`: sampling principles
- `<editorialDecl>`: editorial principals, e.g. `<correction>`, `<normalization>`, `<quotation>`, `<hyphenation>`, `<segmentation>`, `<interpretation>`
- `<classDecl>`: classification system/s used
- `<tagUsage>`: specifics about usage of particular elements

The `<encodingDesc>` can replace the user manual, or facilitate semi-automatic document management, given agreed codes of practice.

3.36 Profile Description

An extensible rag-bag of descriptions, categorized only as ‘non-bibliographic’. Default members of the `model.profileDescPart` class include:

- `<creation>`: information about the origination of the intellectual content of the text, e.g. time and place
- `<langUsage>`: information about languages, registers, writing systems etc used in the text
- `<textDesc>` and `<textClass>`: classifications applied to the text by means of a list of specified criteria or by means of a collection of pointers, respectively
- `<particDesc>` and `<settingDesc>`: information about the ‘participants’, either real or depicted, in the text
- `<handList>`: information about the hands identified in a manuscript

3.37 Classification Methods

`<textClass>` provides a classification (by domain, medium, topic...) for the whole of a text expressed in one or more of the following ways:

- direct reference to a locally defined category (using `<catRef>`)
- reference to an externally defined category (using `<classCode>`)
- documented by `<keywords>`

3.38 Example

```
<textClass>
  <catRef target="#X123"/>
  <classCode scheme="DD12">001.9</classCode>
  <keywords>
    <term>End of the World</term>
    <term>Day of Judgment</term>
    <term>Apocalypse</term>
  </keywords>
</textClass>
<classDecl>
  <taxonomy>
    <category xml:id="X1">
      <catDesc>Homiletic writing</catDesc>
    <category xml:id="X123">
      <catDesc>Day of Judgment</catDesc>
    </category>
  </category>
</taxonomy>
</classDecl>
```

3.39 Detailed characterization of a text

`<textDesc>` provides a description of a text in terms of its ‘Situational parameters’

```

<textDesc n="novel">
  <channel mode="w">print; part issues</channel>
  <constitution type="single"/>
  <derivation type="original"/>
  <domain type="art"/>
  <factuality type="fiction"/>
  <interaction type="none"/>
  <preparedness type="prepared"/>
  <purpose type="entertain" degree="high"/>
  <purpose type="inform" degree="medium"/>
</textDesc>
<!-- These subelements constitute the class model.textDescPart: redefine
that to roll your own. -->

```

3.40 Language and character set usage

The <langUsage> element is provided to document usage of languages in the text. Languages are identified by their ISO codes:

```

<langUsage>
  <language ident="en">English</language>
  <language ident="bg-cy">Bulgarian in Cyrillic characters </language>
  <language ident="bg">Romanized Bulgarian</language>
</langUsage>

```

3.41 Revision Description

A list of <change> elements, each with a date and who attributes, indicating significant stages in the evolution of a document. Most recent first.

3.42 Example

```

<revisionDesc>
  <change date="2006-08-09" resp="#LB">handedits following newhrd-
gen.xml</change>
  <change date="2000-10-11" resp="#OUCS">Final manual corrections for BNC-
W</change>
  <change date="2000-10-18" resp="#OUCS">Further manual corrections for BNC-
W</change>
  <change date="2000-01-08" resp="#OUCS">Manually changed catdescriptions
etc. for BNC-W</change>
  <change date="1994-11-30" resp="#OUCS">First release for BNC-1</change>
</revisionDesc>

```

4 Elements Available in All TEI Documents

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

4.1 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>From the listless repose of the place, and the peculiar character of
its inhabitants, who are descendants from the original Dutch settlers, this
sequestered glen has long been known by the name of <name type="place">SLEEPY
HOLLOW</name>, and its rustic lads are called the Sleepy Hollow Boys
throughout all the neighboring country.</p>
```

4.2 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)

4.3 Highlighting Examples

- `<hi>` (general purpose highlighting)

```
Ichabod prided himself upon his
<hi rend="RedUnderline">dancing</hi>
as much as upon his vocal powers.
```

- `<distinct>` (linguistically distinct)

a worthy wight of the name of Ichabod Crane; who sojourned, or, as he expressed it, **<distinct>**tarried**</distinct>**, in Sleepy Hollow

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

4.4 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)

Summoning up, therefore, a show of courage, he demanded in stammering accents, **<said who="#ICrane">**Who are you?**</said>** He received no reply.

4.5 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)

4.6 Choice Example

```
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>.
```

4.7 Additions, Deletions, and Omissions

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

4.8 Example of `<add>`, ``, `<gap>`, and `<unclear>`

```
<add place="left">The Cause</add> The immediate
cause, however, of the prevalence of supernatural
<del>tales</del>
<add place="supra">stories</add>
in these parts, was doubtless owing to the<unclear reason="blood splat-
ter">vicinity</unclear>
of Sleepy Hollow.
<gap reason="illegible">
  <desc>The rest of this paragraph is covered
    in dried blood.</desc>
</gap>
```

4.9 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.

4.10 Basic Names Example

```
<p> Such is the general purport of this legendary
superstition, which has furnished materials for many a wild story in that
region of shadows; and<rs corresp="#HSH">the spectre</rs> is known,
at all the country firesides, by the name of
the<name xml:id="HSH" ref="myths.xml#HeadlessHorseman" type="spirit">Headless
Horseman of
  <name key="SleepyH001" type="place">Sleepy
    Hollow</name>
</name>.</p>
```

4.11 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

4.12 Basic Address Example

```
<email>HeadlessHorseman@example.com</email>
```

```
<address>
  <name>Ichabod Crane, Schoolmaster</name>
  <addrLine>The Schoolhouse</addrLine>
  <street>5 Headless Horseman Way</street>
  <addrLine>Sleepy Hollow, Tarrytown, NY</addrLine>
  <postCode>10591</postCode>
</address>
```

4.13 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

4.14 Number and Measure Example

```
He had<num value="3">three</num> or
<num value="4">four</num>boon companions.
```

```
This road leads through a sandy hollow,
shaded by trees for about<measure quantity="402.34" unit="m">a quarter of a
mile</measure>, where it
crosses the bridge famous in goblin story, and just beyond swells the green
knoll on which stands the whitewashed church.
```

4.15 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)

```
<p>
  <date when="1792-11-01">The next morning</date>
  the old horse was found without his saddle, and with the bridle under his
  feet, soberly cropping the grass at his master's gate. Ichabod did not make
  his appearance at<time when="1792-11-01T07:30:00">breakfast</time> --
  <time when="18:30:00">dinner-hour</time>
  came, but no Ichabod.
</p>
```

4.16 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.

4.17 Simple Linking Example

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

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

4.18 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)

4.19 Simple List Example

```
<div>
  <head>Lists</head>
  <p>
    <list type="unordered">
      <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)</item>
</list>
</p>
</div>

```

4.20 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

4.21 Note Example

Then, as he wended his way, by swamp and stream and awful woodland, every sound of nature, at that witching hour, fluttered his excited imagination: the moan of the whip-poor-will<note place="foot" type="auth">The whip-poor-will is a bird which is only heard at night. It receives its name from its note, which is thought to resemble those words.</note> from the hill-side; the boding cry of the tree-toad, that harbinger of storm;

4.22 Indexing

- If converting an existing index, use nested lists. For auto-generated indexes:
- `<index>` (marks an index entry) with optional `indexName` attribute
- The `<term>` element is used to mark a term inside an `<index>` element
- The `<index>` element can self-nest for hierarchical index entries

4.23 Indexing Example

```

<p> As Ichabod jogged slowly on his way,
his eye, ever open to every symptom of culinary abundance, ranged with delight
over the treasures of jolly autumn<index>
  <term>Seasons, Autumn</term>
</index>. On all sides he beheld vast store

```

```
of apples<index>
  <term>Fruit, Apples</term>
</index>; some hanging in oppressive
opulence on the trees; some gathered into baskets and barrels for the market;
others heaped up in rich piles for the cider-press<index>
  <term>Machinery</term>
<index>
  <term>Cider Presses</term>
</index>
</index>.</p>
```

4.24 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 would provide `<figure>` and `<figDesc>` for more complex graphics

```
<p>The following engraving shows Brom Bones chasing
Ichabod Crane:<graphic url="BromBonesAndIchabod.png"/>
</p>
```

4.25 Bibliographic Citations

- `<bibl>` (loosely structured bibliographic citation)
- `<biblStruct>` (structured bibliographic citation)
- `<listBibl>` (a list of bibliographic citations such as a bibliography)
- The 'header' module also includes `<biblFull>` (fully-structured bibliographic citation based on the TEI `fileDesc` element)

4.26 Simple `<bibl>` Example

```
He was, moreover, esteemed by the women as a man of great erudition, for he
had read several books quite through, and was a perfect master of<bibl>
  <author>Cotton Mather's</author>
  <title>History of New England Witchcraft</title>
</bibl>, in which, by the way, he most firmly
and potently believed.
```

4.27 Simple `<biblStruct>` Example

```
<biblStruct>
  <monogr>
    <title>Magnalia Christi Americana: or, The
      ecclesiastical history of New-England, ...</title>
    <author>Mather, Cotton (1663-1728)</author>
    <imprint>
      <publisher>Printed for Thomas Parkhurst, at the
```

```

    Bible and Three Crowns in Cheapside.</publisher>
    <pubPlace>London</pubPlace>
    <date when="1702">MDCCII</date>
  </imprint>
</monogr>
</biblStruct>

```

4.28 Verse

- <l> (a line of verse)
- <lg>(a line group such as stanza or paragraph)

```

<lg>
  <l>A pleasing land of drowsy head it was,</l>
  <l> Of dreams that wave before the half-shut eye;</l>
  <l>And of gay castles in the clouds that pass,</l>
  <l> For ever flushing round a summer sky.</l>
  <trailer>CASTLE OF INDOLENCE.</trailer>
</lg>

```

4.29 Drama

- <sp> (an individual speech in a performance text)
- <speaker> (the name of the speaker(s) as given in the performance text)
- <stage> (a stage direction of any sort within a dramatic text)

4.30 Sleepy Hollow: The Play

```

<sp>
  <speaker>Brom Bones</speaker>
  <stage>Said boastfully whilst crossing downstage-right to his horse</stage>
  <p>I will double the schoolmaster up, and lay him on a shelf of his own
  school-house;</p>
</sp>
<stage>Brom mounts horse and rides away</stage>

```