1 Taming the TEI Tiger

He's really friendly when you get to know him

2 The TEI Guidelines

- are Guidelines: not prescriptive
- summarize a consensus about the significant particularities of a huge range of (mostly) textual materials
- are expressed as two fat volumes of prose (with examples) and (inextricably mixed with it) a set of formal definitions
- the definitions can be expressed in a variety of schema languages:
  - in TEI P1-P3 (1991-1999) as a modular SGML DTD
  - in TEI P4 (2000) as an SGML or XML DTD
  - in TEI P5 (2005-) as XML DTD, W3C Schema, or RelaxNG

3 Basic concepts

- The TEI is a *modular* system: you use it to build an encoding scheme appropriate to your needs, by selecting specific modules
- Each module defines a group of elements and attributes
- Elements are classified structurally and semantically
  - semantic classes group elements which have similar meanings — elements like names, or like editorial interventions for example
  - structural classes group elements which behave similarly in the structure — elements like paragraphs, or like phrases for example
  - we also talk of attribute classes: these group elements which all have the same attribute definitions

4 Mandatory (ish) modules

- teistructure
  - defines all named element classes and macros
  - and basic " book-like" structure for prose, verse, drama
- Core
  - the TEI header
  - 'core' elements "common to all kinds of text"

5 Optional modules

- Alternative structures
  - eg transcribed speech, dictionaries ...
- Specialist applications
  - linking and alignment; analysis; non-standard characters and glyphs; feature structures; certainty; physical transcription; textual criticism; names and dates; language corpora; manuscript description. . .
  - and not forgetting the ODD system

6 There is NO SUCH THING as “the TEI dtd”

TEI Lite (`http://www.tei-c.org/Lite/`)

- is our guess at what most people want, most of the time
- realistic for existing texts, and even for new document production, e.g. TEI technical documentation

At P5 the task of making your own TEI schema is much simplified.

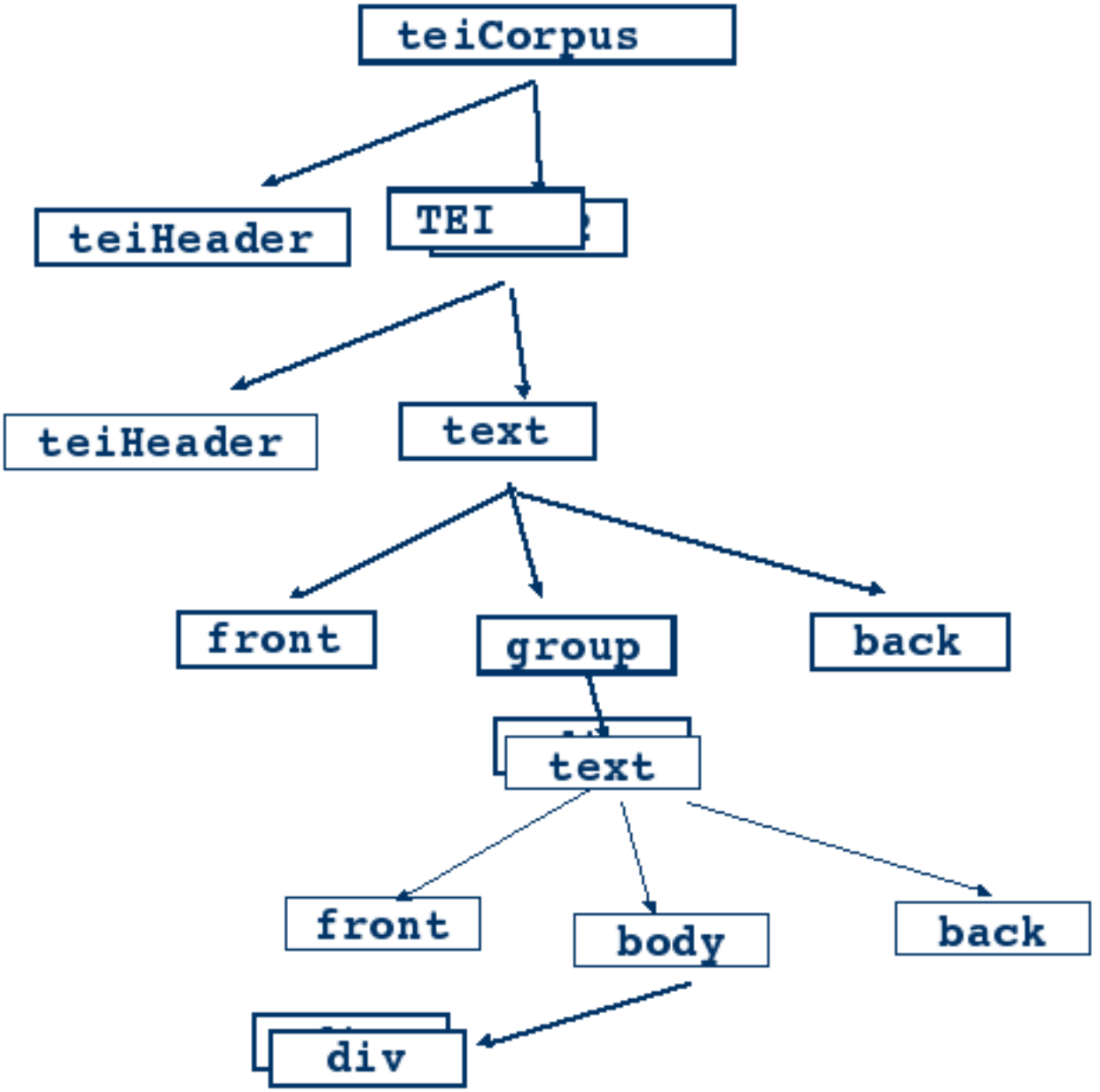
7 Basic structure(s)

- Every TEI-conformant document comprises a *header* followed by (at least one) *text*
- the header contains:
  - mandatory file description
  - optional encoding, profile and revision descriptions
- the header is essential for:
  - bibliographic control and identification
  - resource documentation and processing

8 Structure of a TEI text

- A text may be unitary or composite
- a unitary text contains
  - optional front matter
  - optional back matter
  - a body
- in a composite text, the body is replaced by a group of texts (or nested groups)
- A corpus is a collection of text and header pairs, which also has its own header.

9 TEI basic structure



10 A text usually has divisions

- generic, hierarchic subdivisions, each incomplete
- the type attribute is used to label a particular level e.g. as "part" or "chapter"
- vanilla or numbered tags may be used to identify level explicitly
- the n attribute gives a particular division a name or number
- the xml:id attribute gives a particular division a unique identifier
- associated `<head>` and `<trailer>` elements (from the divtop class) may also be supplied

11 For example...

```
<text>
<front>
<!-- titlepage, etc here -->
</front>
<body>
<div type='book' n='I' xml:id='JA0100'>
<head>Book I.</head>
<div type='chapter' n='1' xml:id='JA0101'>
<head>Of writing lives in general,...
```

```
<!-- remainder of chapter 1 here -->
</div>
<div n='2' xml:id='JA0102'>
<!-- chapter 2 here -->
</div>
<!-- remainder of book 1 here -->
</div>
<div type='book' n='II' xml:id='JA0200'>
<!-- book 2 here -->
</div>
<!-- remaining books here -->
</body></text>
```

12 TEI global attributes

- Defined in the core module, available for all elements:
  - xml:id supplies a unique identifier
  - n supplies a (non-unique) name or number
  - rend gives a suggestion about rendition (appearance)
  - xml:lang identifies the language using an ISO standard code
- Defined in the linking module
  - corresp, synch, ana for specific association types
  - next, prev for aggregating fragmented elements

13 Text components

What are divisions composed of?

- prose is mostly paragraphs (`<p>`)
- verse is mostly lines (`<l>`), sometimes in hierarchic groups (`<lg>`)
- drama is mostly speeches (`<sp>`) containing `<p>` or `<l>` elements interspersed with stage directions (`<stage>`)

These may be mixed, and may also appear directly within undivided texts.

14 For example

```
<div type="book">
<l>Of Man's first disobedience, and the fruit</l>
<l>Of that forbidden tree whose mortal taste</l>
<l>Brought death into the World, and all our woe,</l>
<l>With loss of Eden...</l>
....
</div>
```

```
<lg type="haiku">
<l n="1">Summer grass --</l>
<l n="2">all that's left</l>
<l n="3">of warriors' dreams</l>
</lg>
```

15 For example

```
<stage>Enter Barnardo and Francisco,
two Sentinels,at several doors</stage>
<sp who="Barnardo"><l part="f">Who's there? </l></sp>
<sp who="Francisco"><l>Nay, answer me.  Stand and unfold yourself. </l></sp>
<sp who="Barnardo"><l part="i">Long live the king! </l></sp>
<sp who="Francisco"><l part="m">Barnardo? </l></sp>
<sp who="Barnardo"><l part="f">He. </l></sp>
```

16 Core phrase level elements include...

- phrases that are conventionally typographically distinct
- “data-like” (names, numbers, dates, times, addresses)
- editorial intervention (corrections, regularizations, additions, omissions ...)
- cross references and links

17 for example...

```
<head>
Of writing lives in general,and particularly of
<title>Pamela </title>, with a word by the bye of
<name>Colley Cibber</name> and others.</head>
<p>It is a trite but true observation, that
<q>examples work more forcibly on the mind than precepts</q>
<p><name>Mr. Joseph Andrews</name>,
<rs>the hero of our ensuing history</rs>, was esteemed to be ...
```

18 Direct speech

- Use the who attribute to show speakers
- Speeches can be nested in other speeches

```
<q who="Wilson"> Spaulding, he came down into
the office just this day eight weeks with
this very paper in his hand, and he
says:-- <q who="Spaulding">I wish to
the Lord, Mr. Wilson, that I was a
red-headed man.</q></q>
```

19 Foreign language phrases

- The xml:lang attribute may be attached to any element
- Use `<foreign>` if nothing else is available
- Use ISO 639-2 code to identify language

```
Have you read
<title xml:lang="deu">Die Dreigroschenoper</title>?
```

```
<mentioned xml:lang="fra">
Savoir-faire </mentioned>
 is French for know-how.
```

```
John has real <foreign xml:lang="fra">
savoir-faire </foreign>.
```

20 Names and other referring strings

- The `<rs>` (referring string) element is used for any kind of name or reference

```
<q>My dear <rs type="person" key="BENM1">Mr. Bennet</rs>,</q>
said <rs type="person" key="BENM2"> his lady</rs> to him
one day,<q>have you heard that
<rs type="place" key="NETP1">Netherfield Park</rs>
is let at last?</q>
```

21 Correction and Regularization

- <corr> marks a correction
- <sic> marks a (deliberate) non-correction
- <reg> marks a regularization
- <orig> marks something deliberately un-normalized
- Use <choice> to indicate a combination of possible encodings

22 For his nose was as sharp as a pen and a table of green feelds

```
... and <reg>he</reg
<corr resp="Theobald">babbl'd</corr> ...
```

```
... and <choice>
  <orig>a</orig>
  <reg>he</reg>
</choice>
<choice>
  <sic>table</sic>
  <corr resp="Theobald">babbl'd</corr>
</choice>of green
<choice>
  <orig>feelds</orig>
  <reg>fields</reg>
</choice>
```

23 'Inter' class elements

- <list> lists of all kinds
- <note> notes (authorial or editorial)
- <figure> pictures or figures
- <table> tables
- <bibl> bibliographic descriptions

24 Lists

- use <list> for lists of any kind (use type attribute to distinguish)
- use <label> in two-column lists as alternative to n attribute
- may be nested as necessary

25 for example...

```
<list type="xmas">
  <label>For my true love</label>
  <item>
    <list type="bullets">
      <item>three calling birds></item>
      <item>two french hens</item>
      <item>a partridge in a pear tree</item>
    </list>
  </item>
  <label>For Uncle Joe</label>
  <item>socks as usual</item>
</list>
```

26 Figures and graphics

The presence of a graphic is indicated by the <graphic> element, usually contained within a <figure> element which groups together:

- The title of the graphic (<head>)
- A description of the graphic (<figDesc>) for use by software unable to render the graphic
- The graphic resource itself is pointed to by a URL attribute on the <graphic> element, and may also have attributes scale, height, width
- <figure>s may self-nest, and may also contain other display class items such as. <formula>s

27 Example

28 Example

```
<figure>
<head>Mr Fezziwig's Ball</head>
<figDesc>A Cruikshank engraving showing Mr Fezziwig leading a
group of revellers.</figDesc>
<graphic url="fezz.gif"/></figure>
```

29 Tables

- a <table> element contains <row>s of <cell>s
- spanning is indicated by rows and cols attributes
- role attribute indicates whether row or column holds data or a label
- embedded tables are permitted

30 for example...

A three column table		
Row1	123	4567
Row2	abc	defgh

```
<table>
<row>
 <cell  cols="3" role="label">A three column table</cell>
</row>
<row>
 <cell role="label">Row1</cell><cell>123</cell><cell>4567</cell>
</row>
<row>
 <cell role="label">Row2</cell><cell>abc</cell><cell>defgh</cell>
</row>
</table>
```

31 Bibliography

- The <listBibl> element lists bibliographic citations

- Individual citations may be represented loosely as `<bibl>` elements, or in a more structured way as `<biblStruct>` elements
- In either case, elements from the tei.biblpart class are used, e.g.
  - `<author>`, `<editor>`, (generic) `<respStmt>` etc.
  - `<title>` with optional level attribute to distinguish monographic, analytic etc.
  - `<imprint>` groups publication info (publisher, date etc.)
  - `<biblScope>` adds page references etc.
- Individual citations may be linked to in the usual way

32 Example

```
<p>See for example <ref target="#REG92">Regis (1992)</ref>...
```

```
<div><head>Bibliography</head>
<listBibl>
 <bibl xml:id="REG92">
  <author>Ed Regis</author>
  <title level="m">Great Mambo Chicken and the Trans-Human Experience</title>
  <pubPlace>London </pubPlace>
  <publisher>Penguin Books</publisher>
  <date>1992</date>
  <biblScope>pp 144 ff</biblScope>
 </bibl>
</listBibl>
</div>
```

33 Notes

- Use `<note>` for notes of any kind (editorial or authorial)
- if in-line, use place attribute to specify location
- if out of line, either use
  - target attribute to specify attachment point
  - or mark attachment point as a `<ref>`

34 for example...

```
<lg><l>The self-same moment I could pray></l>
<l>And from my neck so free</l>
<l>The albatross fell off, and sank</l>
<l>Like lead into the sea.
<note type="auth" place="margin">
The spell begins to break.</note> </l>
</lg>
```

or

```
...
<l>The albatross fell off, and sank</l>
<l xml:id="L213">Like lead into the sea. </l>
</lg>
...
<note type="auth" place="margin" target="#L213">
The spell begins to break.</note>
```

35 Exercise 1

The Punch exercise files contain prose, verse, drama. Use your preferred XML editor and the supplied teilite schema to tag them as as you think appropriate.

Or, if you prefer, choose one of the other texts supplied and start thinking about what TEI tags you would use to mark it up.

36 Other Modules

Your choice from:

- Transcription of spoken texts
- Dictionaries and lexica
- Varieties of linguistic annotation
- Nonstandard characters and glyphs
- Linking, alignment, non-hierarchic structures
- Detailed metadata (the TEI Header)
- Manuscript Description
- Text-critical apparatus
- Physical description
- Onomastics and ontologies
- The ODD system

37 Literate programming ODD-style

The TEI Guidelines, its DTD, and its schema fragments, are all produced from a single XML resource containing:

1. Descriptive prose (lots of it)
2. Examples of usage (plenty)
3. Formal declarations for components of the TEI Abstract Model:
   - elements and attributes
   - modules
   - classes and macros
4. We call this resource an **ODD** (One Document Does it all) although the master source is instantiated as a gazillion XML mini-documents.

38 So what?

The TEI scheme can only be used by customizing it.

Customizations are also expressed in the ODD language

For example:

```
<schemaSpec ident="myTEIlite">
<desc>This is TEI Lite with simplified heads</desc>
  <moduleRef name="teistructure"/>
  <moduleRef name="linking"/>
  <moduleRef name="core"/>
  <moduleRef name="teiheader"/>
  <elementSpec ident="head" mode="change">
    <content><rng:text/></content>
  </elementSpec>
</schemaSpec>
```

produces the schema for TEI Lite, with a slight change

39 ODD processors

- We supply a library of XSLT scripts that can generate
  - The book in canonical TEI XML format
  - The book in HTML or PDF
  - RelaxNG, DTD, or W3C schema fragments
- The same library is used by Roma: the new customization layer to generate
  - project-specific documentation
  - project-specific schemas
  - translations into other (human) languages

Roma: generating validators for the TEI | Search TEI

Save	Customize	New	Help		
Modules	Add Elements	Change Classes	Language	Schema	Docum

Warning! this version of Roma uses a pre-release draft

Modules

List of TEI Modules

	Module name	A short description
add	analysis	Simple analytic mechanisms
add	certainty	Certainty and uncertainty
add	core	Elements available in all forms of the TEI main D
add	corpus	Header extensions for Corpus Texts
add	declarefs	Feature System Declaration
add	dictionaries	Base tag set for printed dictionaries
add	drama	Base tag set for Performance texts
add	figures	Tables, Formulae, Figures
add	gaiji	Character and Glyph documentation
add	header	The TEI Header
add	iso-fs	Feature Structures
add	linking	Linking, Segmentation and Alignment
add	msdescription	Manuscript Description
add	namesdates	Additional classes for names and dates
add	nets	Graphs, networks and trees
add	sharedheader	Auxiliary DTD for Independent Header
add	spoken	Base tag set for Transcribed Speech
add	tagdocs	Declares the elements making up the module do
add	tei	Main document type declaration file
add	textcrit	Tags for text criticism

41 The TEI abstract model

- The TEI abstract model sees a markup scheme (a schema) as consisting of a number of discrete modules, which can be combined more or less as required.
- A schema is made by combining references to modules and optional element over-rides.
- Each element declares the module it belongs to: elements cannot appear in more than one module.
- Each module extends the range of elements and attributes available by adding new members to existing classes of elements, or by defining new classes.

42 The TEI class system

- Class membership can do two distinct things for an element:
  1. give it some attributes
  2. allow it to join a 'club'
- Content models reference 'clubs' rather than specific elements (wherever possible)
- Content models are named patterns, distinct from element names
- (There are also special named patterns for common content models such as `macro.phraseSeq`)

This information is all presented formally in an ODD document

43 What does an ODD look like?

```
<elementSpec module="spoken" ident="pause">
 <classes>
   <memberOf key="tei.comp.spoken"/>
   <memberOf key="tei.timed"/>
   <memberOf key="tei.typed"/>
 </classes>
 <content>
   <rng:empty xmlns:rng="..."/>
 </content>
  <attList>
    <attDef ident="who" usage="opt">
    <datatype><rng:data type="IDREF"/></datatype>
      <valDesc>A unique identifier</valDesc>
      <desc>supplies the identifier of the
      person or group pausing.
      Its value is the identifier of a <gi>person</gi>
      or <gi>persGrp</gi> element in the TEI header.
      </desc>
    </attDef>
 </attList>
 <desc>a pause either between or within utterances.</desc>
</elementSpec>
```

44 ... from which we generate

```
pause = element pause { pause.content }
pause.content =
  empty,
  tei.global.attributes,
  tei.comp.spoken.attributes,
  tei.timed.attributes,
  tei.typed.attributes,
  pause.attributes.who,
  pause.newattributes,
  [ a:defaultValue = "pause" ] attribute TEIform { text }?
pause.newattributes |= empty
tei.comp.spoken |= pause
tei.timed |= pause
```

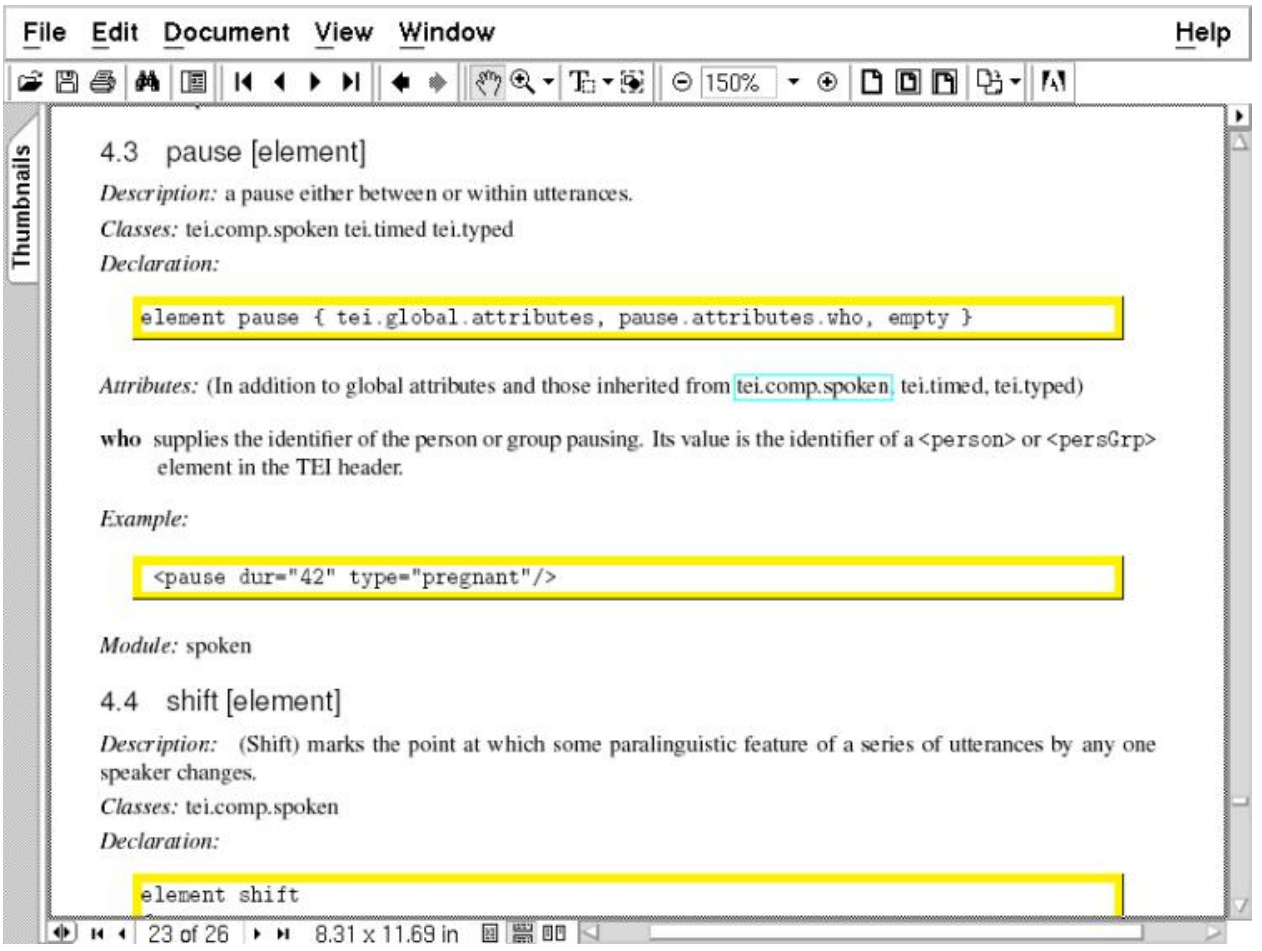
```
pause.attributes.who =
   attribute who { pause.attributes.who.content }?
pause.attributes.who.content = xsd:IDREF
```

45 .. which translates to

```
<!ENTITY % pause 'INCLUDE' >
<![ %pause; [
<!ELEMENT %n.pause; %om.RR; EMPTY>
<!ATTLIST %n.pause;
   %tei.global.attributes;
   %tei.timed.attributes;
   %tei.typed.attributes;
   who   IDREF #IMPLIED
   TEIform CDATA 'pause'   >

<!ENTITY % tei.comp.spoken "%x.tei.comp.spoken;
%n.event; | %n.kinesic; | %n.pause; | %n.shift;
| %n.u; | %n.vocal; | %n.writing;">
```

46 ... and, indeed, to



47 Customizing the TEI

The TEI has over 20 modules. A working project will:

- Choose the modules they need
- Probably narrow the set of elements within a module
- Probably add local datatype constraints
- Possibly add new elements
- Possibly localize the names of elements

48 We also do all that in an ODD

```
<schema>
<moduleRef name="tei"/>
<moduleRef name="header""/>
<moduleRef name="textstructure"/>
<moduleref name="linking"/>
</schema>
```

49 From which we can generate...

```
<grammar ns="http://www.tei-c.org/P5/"
 xmlns="http://relaxng.org/ns/structure/1.0"
 datatypeLibrary=
   "http://www.w3.org/2001/XMLSchema-datatypes">
<include href="Schema/tei.rng"/>
<include href="Schema/header.rng"/>
<include href="Schema/textstructure.rng"/>
<include href="Schema/linking.rng"/>
</grammar>
```

50 More interestingly..

```
<schema>
<moduleRef name="teiheader"/>
<moduleref name="verse"/>
<!-- add a new element -->
<elementSpec ident="soundClip">
<classes memberOf="tei.data"/>
 <attList>
  <attDef ident="url">
  <datatype><rng:data type="URI"/></datatype>
  <valDesc>A location path</valDesc>
  <desc>supplies the location of the clip</desc>
  </attDef>
 </attList>
 <desc>includes an audio object in a document.</desc>
</elementSpec>
<!-- change an existing element -->
<elementSpec ident="head" mode="change">
<content><rng:text/></content>
</elementSpec>
</schema>
```

51 Overriding a value-list

```
<elementDecl ident="list" module="core">
<classes>
 <memberOf key="tei.typed"/>
</classes>
<!--... -->
<attDef ident="type" mode="replace">
<valList>
<valItem ident="ordered">numerically ordered</valItem>
<valItem ident="bulleted">un-numbered but bulleted</valItem>
<valItem ident="frabjous">simply frabjous</valItem>
</valList>
</attDef>
</elementDecl>
```

52 Uniformity of description

- modules, elements, attributes, value-lists are treated uniformly
- each has an identifier, a gloss, a description, and one or more equivalents
- each can be added, changed, replaced, deleted within a given context
- for example, membership in the `tei.typed` class gives you a generic TYPE, which can be over-riden for specific class members

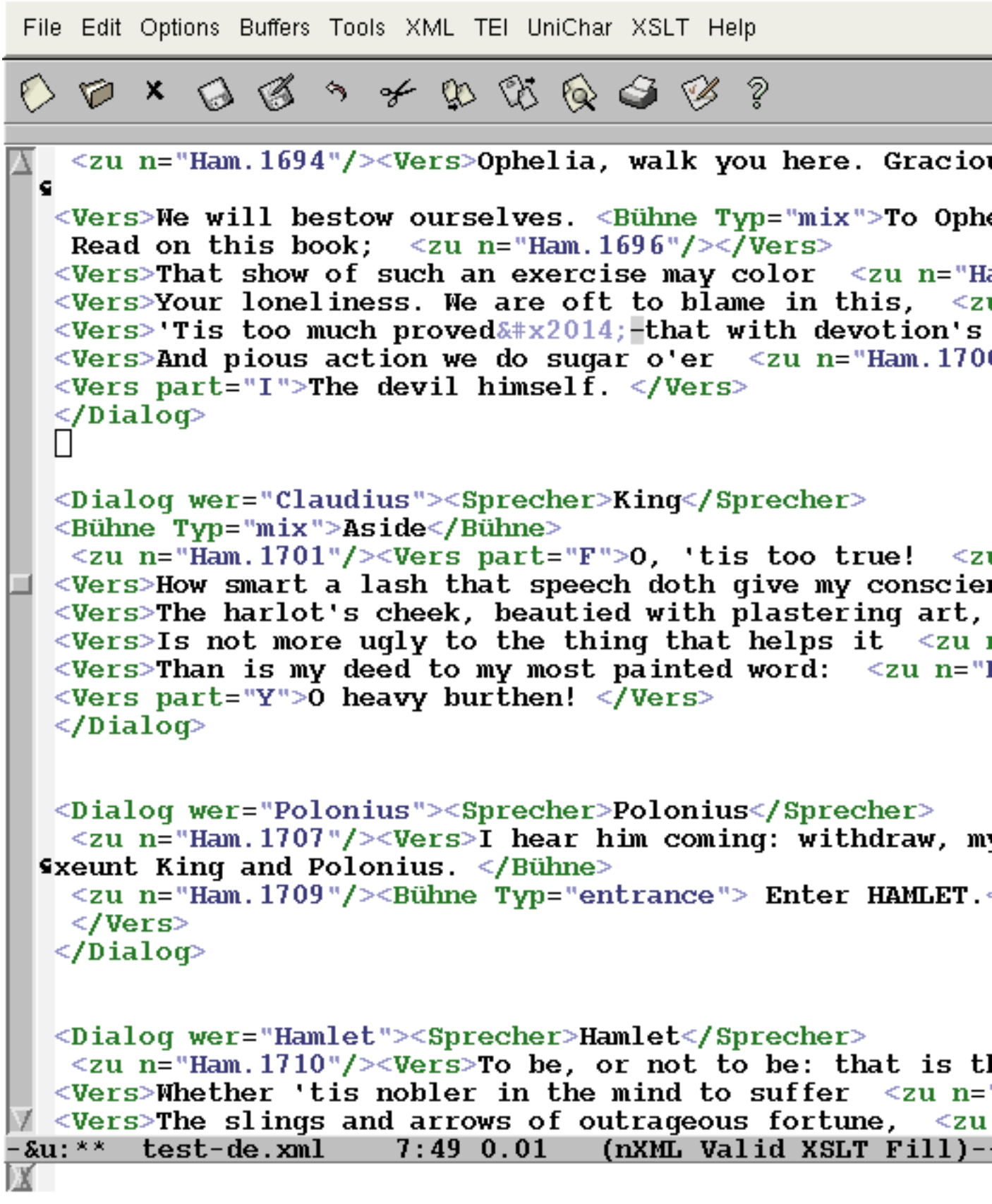
53 Internationalization

All TEI elements are surrounded by a naming layer, which allows their user-visible names to be changed. This covers:

- element names
- attribute names
- attribute values
- short descriptions

The translation database is maintained separately, so attribute names and values are translated once only

54 Hamlet with German tags



55 Our gesture towards ontological mapping

The `<equiv>` element supplies a URI which identifies an equivalent concept (*not* a name) in some externally-defined ontology, e.g.

- ISO data category registry
- CIDOC conceptual reference model
- Wordnet