RelaxNG with Son of ODD or, What the TEI Did Next

Lou Burnard

Birmingham, Feb 2006

TEI, a new phase

The P5 release of the TEI Guidelines has three aims:

Interoperability taking advantage of the work done by others

Expansion addressing areas as yet untamed

Internal audit cleaning up the accretions of a decade

... all without losing touch with its core constituency

Interoperability

A lot of other people have been working in this area since 1987!
TEI P5 must fit into a joined-up digital world, along with

- W3C standards (XLink, schema, etc)
- Unicode character encoding
- Specialized markup vocabularies (MathML, SVG, DocBook, etc)
- Other metadata schemas (METS, EAD, etc)
- Other conceptual models and ontologies
- .... and TEI P4

Expansion: why?

- TEI P4 did not (could not) cover everything!
- The TEI has always been ahead of the pack in promoting evolutionary change:
  - Some parts of TEI P4 were successfully experimental (e.g. the extended pointer syntax, corpus metadata)...
  - ... some were influentially experimental and have become FaQs (‘frequently answered questions’) e.g. synchronization and standoff
  - ... others were just experimental, and have been overtaken by events (e.g. writing system declaration, feature structures, terminology...)
- A key deliverable: better tools for customization and integration

Internal audit

- P1 to P4 were drafted by dozens, but edited (mostly) by just two people on a variety of platforms, and processed with a pile of exotic SGML text processing utilities, mainly home grown;
- The TEI source of P5 needs to be made accessible and shareable by many, using today's rich variety of XML text processing tools
- a proper change control/document management system is indispensable

Are we nearly there yet?

- Infrastructural developments
- What's new so far?
- Customization and Modularity
- Internationalization

Infrastructural developments

- The TEI editors' toolkit:
  - more than one XML editor
  - a library of XSLT scripts
  - a version control system
  - a test suite
  - self-validating source and examples
- Working practices:
  - the workgroup model
  - role of the council
  - snapshot releases
    - Feb 2005
    - Aug 2005
    - Oct 2005
    - Feb 2006
- Opening the TEI: moving the source to sourceForge

What's new so far?

- New modules for gaiji, msDescription
- The war on attributes
- Linking mechanisms
- Attribute datatypes
- The class struggle

But first... what's in the draft?

New and forthcoming content in TEI P5

New

- schema documentation and generation
- manuscript description
- new infrastructure chapter
- `<choice>`, `<index>`, `<graphic>` etc.
- feature structures (now ISO 24610)
- standoff annotation, Xlink, Xptr, Xinclude &c.

Forthcoming

- "personography"
- handling of overlap
- dictionaries and terminologies
- physical bibliography
- relation of header to other metadata standards
- FAND and xText

Gaiji: is your journey really necessary?

- Getting rid of `&wibble;` in favour of the actual character (or `&#xxxx;`) is highly recommended
- If you *really* need to use non-Unicode characters...
  - wherever text is possible as content, <g> can be used, either as a pointer, or to hold any convenient representation
  - nonstandard characters and glyphs can now be defined in the header
- we now use xml:lang (just as we now use xml:id)

Documenting your use of the private use area


```
<charDesc>
<glyph xml:id="z103">
<glyphName>LATIN LETTER Z WITH TWO STROKES</glyphName>
<mapping type="standardized">Z</mapping>
<mapping type="PUA">&#E304;</mapping>
</glyph>
</charDesc>
```

We may now refer to

```
<g ref="#z103"/>
```

and expect the processing application to work out what to do

Character documentation for glyph variants


```
<charDesc>
  <glyph xml:id="r1">
  <glyphName>LATIN R WITH ONE FUNNY STROKE</glyphName>
   <charProp>
      <localName>entity</localName>
       <value>r1</value>
   </charProp>
   <graphic url="r1img.png"/>
 </glyph>
  <glyph xml:id="r2">
  <glyphName>LATIN R WITH TWO FUNNY STROKES</glyphName>
   <charProp>
      <localName>entity</localName>
       <value>r2</value>
   </charProp>
   <graphic url="r2img.png"/>
 </glyph>
</charDesc>
```

New module on Manuscript Description

(Actually not so new...)

- Supports obsessively detailed description of manuscripts (or brief characterization)
  - in the header of a digital edition or facsimile
  - in the body of a catalogue
- Seems to have become de facto standard
- Scope for expansion e.g. binding descriptions

The war on attributes

- an attribute value cannot contain markup
- the language of an element's content and its attributes must be the same

Work started with the `<choice>` element to replace "mirror" tags; now complete:

```
<sic corr="what!?">whaaa</sic>
<choice><sic>whaaa</sic><corr>what!?</corr></choice>
```

```
<gap desc="transcriber dozes off"/>
<gap><desc>transcriber dozes off</desc>
 <desc lang="fr">transcripteur s'endort</desc></gap>
```

Linking mechanisms

- P4 had two different ways of linking:

  internal `<ptr>`: using ID/IDREF

  external `<xptr>` : using TEI-invented syntax

- But the world has moved on!
- In P5, all pointing is done in the same way, using URI
- A URI may be absolute...

```
<ptr target="http://www.tei-c.org/P5/Guidelines/SA.html"/>
```

- .. relative (the base is value of xml:base)...

```
<list xml:base="http://www.tei-c.org/Members/">
  <item><ref target="2005-Sofia">this meeting</ref></item>
  <item><ref target="2004-Baltimore">last year's</ref></item>
</list>
```

- .. or you may use a "bare name"

```
<sp who="#Macbeth"><speaker>Mac.</speaker> ...
```

- and other XPointer framework schemes may be used

Other XPointer framework schemes

Six new XPointer schemes defined:

- xpath()
- left(), right()
- range()
- string-range()
- match()

```
<ref xml:base="http://www.tei-c.org/Talks/2005/Sofia/"
  target="p5report.xml#range(xpath(//div[12]/list/item[1]),
          xpath(//div[12]]/list/item[5])">
the six added schemes
</ref>
```

Attribute datatypes

- attribute values are now declared by referring to a TEI datatype
- each TEI datatype maps to a W3C XML Schema datatype, and can therefore be validated by regular XML software
- the indirection makes it easier for users to make customizations (and editors to make changes!)
- Currently defined TEI datatypes:

  normalized expressions of quantity certainty, probability, numeric, count

  other normalized values duration, temporal, truthValue, language, sex

  specialized pointers outputMeasurement, namespace, pattern, pointer, pointers

  symbolic names key, word, words, name, names, enumerated, code

Customization

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.
5. ODDs are TEI documents, like any other

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 key="core"/>
  <moduleRef key="tei"/>
  <moduleRef key="textstructure"/>
  <moduleRef key="header"/>
  <moduleRef key="linking"/>
  <elementSpec ident="head" mode="change">
    <content><rng:ref name="model.text"/></content>
  </elementSpec>
</schemaSpec>
```

produces the schema for TEI Lite, with a slight change

ODD processors

- We maintain a library of XSLT scripts that can generate
  - The TEI Guidelines in canonical TEI XML format
  - The Guidelines in HTML or PDF
  - RelaxNG, DTD, or W3C schema fragments
- The same library is used by the customization layer to generate
  - project-specific documentation
  - project-specific schemas
  - translations into other (human) languages
- We use eXist as a database for extracting material from the P5 sources

The TEI abstract model

- Each element declares the module it belongs to: elements cannot appear in more than one module.
- A markup system (a schema) consists of a number of discrete modules, which can be combined more or less as required.
- A schema is made by combining references to modules with other declarations.
- Each module extends the range of elements and attributes available by adding new members to existing classes of elements.

The rise of the class system (1)

- Class membership can do two distinct things for an element:
  1. attribute classes, named att.xxxx, give its members some attributes:
  2. model classes, named model.xxxx, allow its members to join a ‘club’
- Content models reference ‘clubs’ rather than specific elements (wherever possible)
- There are two ways of naming a club:

model.xxxLike elements which are semantically like an xxxx (but fraternize with others)

model.xxxPart sibling elements which constitute an xxxxx

The class struggle

Consider

```
foo (bar|baz|bam|zip)*
```

We could say both

- <foo> contains barLike elements
- <bar> etc. are members of the fooPart class

Either way, we redefine the content model:

```
foo (model.barLike)*
```

The P4 content models offer *a lot* of scope for simplification of this kind...

The rise of the class system (2)

- Classes are easier to understand and remember than elements
- Adding a new element becomes a matter of deciding what it is ‘like’, or what it is a ‘part’ of
- Specialization of the TEI generic structure for specific needs becomes a simple declarative matter

Why the stress on customization?

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

We can do all that in an ODD

More interestingly..


```
<schema>
 <moduleRef key="header"/>
 <moduleRef key="verse"/>
 <elementSpec ident="soundClip">
  <classes>
   <memberOf key="model.data"/>
  </classes>
  <attList>
   <attDef ident="location">
    <desc>supplies the location of the clip</desc>
    <datatype>
     <rng:ref name="data.pointer"/>
    </datatype>
   </attDef>
  </attList>
  <desc>includes an audio object in a document.</desc>
 </elementSpec>
 <elementSpec ident="head" mode="change">
  <content>
   <rng:text/>
  </content>
 </elementSpec>
```

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 `att.type` class gives you a generic type attribute, which can be over-ridden for specific class members

Overriding a value-list


```
<elementSpec ident="list" module="core">
 <classes>
  <memberOf key="att.typed"/>
 </classes>
 <attDef ident="type" mode="replace">
  <valList type="closed">
   <valItem ident="ordered">
    <gloss>Items are ordered</gloss>
   </valItem>
   <valItem ident="bulleted">
    <gloss>Items are bulleted</gloss>
   </valItem>
   <valItem ident="frabjous">
    <gloss>Items are frabjous</gloss>
   </valItem>
  </valList>
 </attDef>

</elementSpec>
```

Our gestures towards ontological mapping

The `<equiv>` element can supply 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

It can also be used to specify a stylesheet transformation where syntactic sugar has been applied, for example to specify formally that `<placeName>` is equivalent to `<name type="place">`

deleta est carthago

Roma is the first of a new generation of TEI tools

- currently available as web app only
- closely coupled with TEI P5 source
- generates customised schemas in DTD, W3C, or RelaxNG
- also generates documentation
- development plans:
  - re-implement as standalone Java app
  - build in more intelligence

You don’t have to write XML: Roma (1)

Roma: generating validators for the TEI

Modules

New | Customize | Modules | Add Elements | Change Classes | Language | Schema | Save | Documentation | Help

List of TEI Modules

	Module name	A short description	Changes
add	analysis	Simple analytic mechanisms	
add	certainty	Certainty and uncertainty	
add	core	Elements common to all forms of the TEI	
add	corpus	Header extensions for corpus texts	
add	declarefs	Feature system declarations	
add	dictionaries	Printed dictionaries	
add	drama	Performance texts	
add	figures	Tables, formulae, and 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	Names and dates	
add	nets	Graphs, networks and trees	
add	spoken	Transcribed Speech	
add	tagdocs	Documentation of TEI modules	
add	tei	Structural declarations for the TEI	

List of selected Modules

remove	core
remove	tei
remove	header
remove	textstructure

Roma (2)

Roma: generating validators for the TEI

Change attribute classes

New | Customize | Modules | Add Elements | Change Classes | Language | Schema | Save | Documentation | Help

List of attribute classes

Class name	Description	Attributes
att.TEIform	defines an attribute (TEIform) common to all tags in the TEI scheme, and recommended for all user-defined extensions.	changeAttributes
att.analytic	defines a set of attributes for associating specific analyses or interpretations with appropriate portions of a text, which are enabled for all elements when the additional tag set for simple analysis is selected.	changeAttributes
att.ascribed	elements representing speech ascribed to a speaker.	changeAttributes
att.datable	defines the set of attributes common to all elements that contain datable events.	changeAttributes
att.datePart	attributes for component elements of temporal expressions involving dates and time	changeAttributes
att.declarable	groups elements which may be independently selected (using the special purpose decls attribute) from a candidate list of declarations within a TEI header.	changeAttributes
att.declaring	groups elements which may be independently associated with a particular declarable element within the header, thus overriding the inherited default for that element.	changeAttributes
att.divLike	defines a set of attributes common to all elements which behave in the same way as divisions.	changeAttributes
att.editLike	elements which carry attributes describing editorial interventions.	changeAttributes
att.enjamb	groups elements bearing the enjamb attribute.	changeAttributes
att.entryLike	groups the different styles of dictionary entries.	changeAttributes
att.global	defines a set of attributes common to all elements in the TEI encoding scheme.	changeAttributes
	defines a set of attributes for hypertext and other linking, which are enabled for	

Roma (3)

Name

Model classes

model.addrPart	model.dateLike	model.editorialDeclPart	model.frontPart.drama
model.biblLike	model.datePart	model.encodingPart	model.gLike
model.biblPart	model.divPart	model.entryLike	model.global
model.blockLike	model.divPart.spoken	model.entryParts	model.global.edit
model.catDescPart	model.divPart.stage	model.entryParts.top	model.global.meta
model.choicePart	model.divPart.verse	model.featureVal	model.gramPart
model.common	model.divWrapper	model.formPart	model.headerPart
model.complexVal			model.hiLike

groups elements which can occur at the start of any division class element.

Attribute classes

att.TEIform	att.datePart	att.editLike	att.global.linking	att.measured	att.pointing
att.analytic	att.declarable	att.enjamb	att.identified	att.metrical	att.pointing.group
att.ascribed	att.declaring	att.entryLike	att.interpLike	att.naming	att.ptrLike.form
att.datable	att.divLike	att.global	att.lexicographic	att.personal	att.rdgPart

Contents

Text

```
<content
xmlns:rng="http://relaxng.org/ns/structure/1.0">
</content>
```

Description

Submit Query

Open TEI

- The TEI consortium now releases the Guidelines under a GNU Public license
- All development now takes place in public using CVS on Sourceforge
- Feature requests and bug tracking are also on Sourceforge
- TEI components are available as Debian Linux packages

However, the name TEI remains a trademark, and technical work continues to be authorized by TEI Technical Council, elected by members of the Consortium.

Open TEI: what does it mean?

- The TEI remains a community initiative, driven by the needs of its members and users
- To encourage more devolved development we need to build a larger community of developers
- This means both making entry level development easier and peer approval more visible
- Which means we need more participation from all potential TEI users, as members of SIGs, Workgroups, and Council ...

The TEI needs You