What is a tree really?

Patrick Durusau
Society of Biblical Literature
TEI 2003 Nancy, France
Descriptive versus Procedural Markup

• Separation of concerns
  – How Text is Processed from
  – How Text is Described

• Allows decisions about processing to be deferred

• Added advantage of portability between processing systems

• Describes the structure of texts
Separation sounds Great!

• Great Divide Begins! (or does it?)
  – GML/SGML adopts angle bang syntax for descriptive markup
  – Encodes the structures in texts
  – But not how to process or presentation

• On the other hand:
  – Instead of traditional presentation
  – We now have markup trees
Bear fruit that befits repentance,

and do not presume to say to yourselves, ‘We have Abraham as our father’; for I tell you, God is able from these stones to raise up children of Abraham.
Trees as Presentation

<?xml version="1.0"?>
<text>
<verse id="Matt.3.8">
   <sentence>
   Bear fruit that befits repentance,
   </sentence>
</verse>
<verse="Matt.3.9">
   and do not presume to say to yourselves, ‘We have Abraham as our father’; for I tell you, God is able from these stones to raise up children of Abraham.
   </sentence>
</verse>
</text>
Which Tree to Follow?

- Traditional XML says either:
  - text/verse, or
  - text/sentence
- But both cannot be present
- Why?
- Predetermined that all markup in a file must be recognized as markup and presented as a well-formed tree
Choosing A Tree

• Recognize all markup
  – Odd requirement, history of parsing files that are not SGML/XML with selective recognition of markup
  – Can even selectively recognize SGML/XML markup so long as it is already well formed
  – Why limit markup options with the recognize all option?
  – Simplicity of parsing!
Simplicity of Parsing

• Simplicity harmful to markup!
  – Well-formedness contrary to:
    • Known features of texts
    • Needs of scholars
  – Well-formedness may make sense for documents without DTDs or Schemas
  – But what scholarly encoded document will exist without a DTD or Schema?
  – Markup limited by ease of parsing?
Simplicity of Parsing II

• Validating SAX based parsers
  – Recognize the GI anyway
  – Order of processing is the problem
  – Fires on any “<“
  – Only to then discover it is not in the DTD or schema
  – What if the ordering were reversed?
  – That is: Build the tree to recognize, then parse for markup that matches?
Simplicity of Parsing III

• But what of the other “markup?”
• Can you say “string?”
• If markup recognition is conditional:
  – Can impose unlimited layers of markup inline on a text
  – Can search for structures in any tree, and match against strings that are markup in another tree
  – Divorces markup from a particular presentation
Is Selective Recognition Possible?

• XPath/XQuery
  – Efficient Filtering of XML Documents with XPath Expressions, Chee-Yong Chan, Pascal Felber, Minos Garofalakis, Rajeev Rastogi
  – YFilter: Efficient and Scalable Filtering of XML Documents Yanlei Diao, Peter Fischer, Michael J. Franklin, Raymond To
  – Efficient Filtering of XML Documents for Selective Dissemination of Information, Mehmet Altinel, Michael J. Franklin
Is Selective Recognition Likely?

- SC34/WG1 Document Schema and Description Languages (includes, RELAX-NG)
- Part 1: Overview of ISO/IEC 19575
  - Path based addressing (role of relationships that are not hierarchical)
  - JITTs (Just-In-Time-Trees) has been suggested as one approach to consider
Simplistic Markup or Simplistic Parsing

• The choice is fairly simple:
  – Simplistic markup, or
  – Simplistic parsing

• Latter may have been appropriate, Sun workstations had 128K RAM, 100 MHz processors

• Laptops now routinely have 1 GB RAM, and over 1 GHz processors
Workarounds or a Solution?

• All of the current options for overlapping markup compensate for simplistic parsing
• Parsing research has advanced but markup parsing has remained static
• Workarounds are not solutions!
• Our texts need a solution
• Our users deserve a solution
What Can TEI Do?

• Develop compelling use cases for overlapping markup
• Demonstrate the advantages of non-simplistic parsing for markup (sigh, yes the commercial side of things)
• Press our needs in forums such as SC34 WG3
Conclusion

• Simplistic parsing will continue so long as no one makes the case for better parsing of markup

• The “someone” to make the case is the academic markup community

• Why? We should not dumb down our texts for the convenience of avoiding further development of markup parsers!