Before we begin

- Launch <oXygen/>
  - If you haven’t yet installed <oXygen/>:
    1. Go to http://www.oxygenxml.com
    2. Download and install
    3. Follow instructions to request free 30-day evaluation license
- Workshop web site
  - http://varna.obdurodon.org

Session 1: Introduction to XML

El’Manuscript, Varna, 2014-09-15

David J. Birnbaum
djborn@gmail.com
http://www.obdurodon.org

Andrei Bojadžiev
aboj@uni-sofia.bg

Outline

- Sample projects
  - Manuscript description: Repertorium
  - Manuscript transcription: Codex Suprasliensis
- XML and text
  - Overview
  - Pseudo-markup and markup
  - Elements
  - Attributes
  - Well-formedness
- The creation of a digital text
- Editing XML in <oXygen/>
- [Hands-on practice editing XML]

Sample projects

- Repertorium of Old Bulgarian literature and letters
  - Institute of Literature, BAS
  - http://repertorium.obdurodon.org
- Codex Suprasliensis
  - Institute of Literature, BAS
  - http://suprasliensis.obdurodon.org (edition)
  - http://csup.ilit.bas.bg (project)

Overview

- OHCO: ordered hierarchy of content objects
- Three views of XML
  - Tree
  - Serialization
  - Boxes

Three views of a document

```xml
<book>
  <introduction>…</introduction>
  <chapter>…</chapter>
  <section>…</section>
  <subsection>…</subsection>
  …</section>
  …</subsection>
  …</chapter>
  …</chapter>
  …</book>
```
**Hamlet, First quarto, 1603**

![Image of Hamlet's first quarto]

**Pseudo-markup**

- *Hamlet*
  - Stage directions
  - Speeches
  - Speakers
  - Other characters
  - Metrical lines
- **General**
  - Paragraph spacing and indentation
  - Centering and bolding of titles
  - Hanging indentation for bibliographic lists
  - Italics for emphasis, foreign words, book titles, etc.

**The XML view of content and markup**

- Content is the textual data
  - Transcribed from source (e.g., a manuscript)
  - Created by the editor (e.g., manuscript catalogue)
- Markup describes the role of different data components
- No *pseudo markup* in your content
  - No editorial parentheses, square brackets, angle brackets, slashes and backslashes, italics, etc.

**XML building blocks**

- Textual (character data) content
- Elements
  - Structural components of the document
- Attributes
  - Properties of elements

**Elements**

- Elements have matching start and end tags
  - `<title>` ...
  - `</title>`
- (Some elements are empty and self-closing)
  - `<bookmark/>
- Element names must begin with a letter and may contain letters, digits, and underscores (no spaces; no other punctuation)
  - Underscore: `<personal_name>`
  - Camel case: `<personalName>`

**The “X” in XML**

- eXtensible Markup Language
  - The user determines the tag set
  - Pro: you determine how to characterize your data
  - Con: you are responsible for determining how to characterize your data
- You decide
  - What to tag
  - How to tag it (what to call it)
Three types of markup

- Descriptive: what the object is (emphasized)
  - `<em>yes</em>`
- Presentational: what the object looks like (italicized)
  - `<i>yes</i>`
- Procedural
  - `[instructions to the machine]`

Why DH projects use descriptive markup

- Italics: emphasis, foreign, book title, etc.
- Emphasis: italic, bold, loud (audio device), etc.
- Separation of levels: content and presentation
  - Encode descriptively
  - Transform to presentational final form for rendering (HTML, PDF, etc.)
- Multipurposing: format the same content objects different way for different purposes

Texts and trees

- Why XML looks at texts as trees
  - Computers can traverse trees quickly
  - Documents are hierarchical, right?
- Hierarchical challenges
  - Multiple, overlapping hierarchies
    - Physical hierarchy: folios, lines
    - Intellectual hierarchy: texts (with subelements: chapters, sections, paragraphs, etc.)
  - Relationships at a distance
    - Cross-references and other pointers
    - References and pointers to other documents

Attributes

- Qualifying information about elements
- Encoded inside the start tag, after the element name
  - Attribute name="value" pair
    - `<place xml:lang="fr">Paris</place>`
  - `<title type="journal">Scripta & e-Scripta</title>`
  - Attribute names are subject to the same rules as element names
  - Attribute values must be quoted (matching single or double straight quotation marks)

An XML document must be well-formed

- Single root element
- Proper nesting (no overlapping tags)
  - Good: `<em><foreign oui</em><em>!</foreign></em>`
  - Bad: `<em><foreign oui</em></em><em>!</foreign>`
- Name and name start characters for element and attribute names
  - Attribute values must be quoted (single or double)
  - Reserved characters must be encoded as entities
    - `&` & `&amp;`;
    - `<` & `&gt;`
  - Indentation is for human convenience

Sample

```xml
<book>
  <author>Michael Kay</author>
  <title>"4"XSLT 2.0 and XPath 2.0 Programmer's Reference</title>
  <publisher>John Wiley &amp; Sons, Inc.</publisher>
  <pubPlace>10475 Crosspoint Boulevard, Indianapolis, IN 46256</pubPlace>
</book>
```
What’s wrong?

Creating a digital text

- In theory
  1. Document analysis, then ...
  2. Schema development, then ...
  3. Markup
- In practice
  - The preceding is a cycle, and not a sequence
  - Markup is part of the process of document analysis
- Nonetheless
  - Start with document analysis, not with angle brackets

Why use an XML editor?

- oXygen/ (http://www.oxygenxml.com)
- XML-aware
  - Real-time and on-demand validation
  - Completion hinting
  - Multiple views
  - (Schema-aware ... stay tuned)
- IDE (integrated development environment)
  - XSLT (extensible stylesheet language transformations)
  - Debugger
  - Other development tools

Editing XML in <oXygen/>

- Create a new file
  - File → New → New document → XML document
  - Shortcuts: Ctrl+N, leftmost icon at top of screen
- Create an element
  - Type a start tag (in angle brackets)
  - oXygen/ automatically creates the matching end tag
- Change an element
  - Change the start tag; the end tag changes automatically to match
- Wrap text in an element
  - Select the text, type Ctrl+I (for 'element'), type the element name
  - To use the same wrapper as last time, select and type Ctrl+/
- Split an element
  - Put the cursor at the split point and type Shift+Alt+d
- Pretty-print (wrap) the text
  - Shift + Ctrl+y, pretty-print (indentation) icon

Hands on

- Choose a document with a regular structure
  - Google a recipe for your favorite food
  - Find a menu from your favorite restaurant
  - Encode a letter by Oscar Wilde
    http://law2.umkc.edu/faculty/projects/urials/wilde/letter/complete.html
  - Encode a sonnet by William Shakespeare
  - ... or choose your own text
- Copy into a new XML document in <oXygen/>
- Mark it up in XML
  - Imagine a research or other context where you're marking up your text for a reason
  - Identify and tag major structural components
  - Identify and tag small, in-line items that might be useful
  - Explore tagging options at http://www.tei-c.org

Credits

- Three views of a document
  - http://www.wwp.northeastern.edu/outreach/seminars/uvic_xslt_2012/presentations/xslt/xml_and_xpath_01.xhtml