

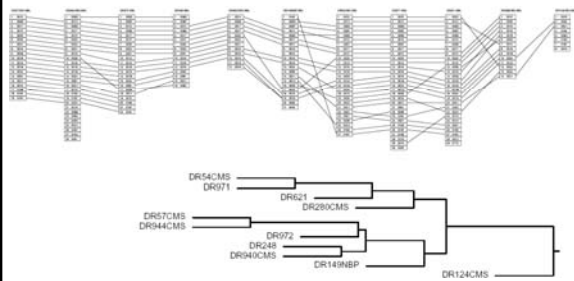
I say XSLT, you say XQuery: Let's call the whole thing off

David J. Birnbaum
Department of Slavic Languages and Literatures
University of Pittsburgh
djbpitt@pitt.edu
<http://clover.slavic.pitt.edu/~djb/>

When it doesn't matter ...

- XSLT or XQuery?
- XSLT push or pull?
- XPath predicate or XSLT `<xsl:if>` / XQuery "where" clause?
- XSLT element and attribute constructors or literal result elements?
 - `<xsl:text>` or plain ol' text
- Element or attribute?
- XML or something else (e.g., RDBMS)?

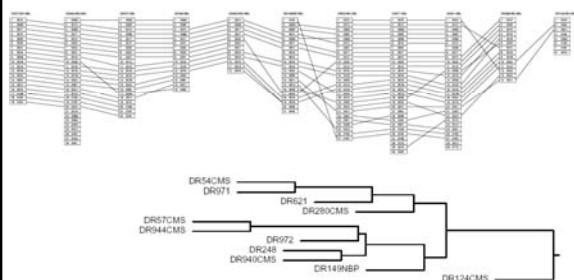
Plectogram visualization



The problem

- Compare the contents of free-form encyclopedic medieval manuscripts
 - Saint's life
 - Prayer
 - Sermon
 - Narrative story
 - Scientific treatise
 - etc.
- Similar contents in similar order suggests shared textual transmission
- Graphic visualization: the plectogram

Plectogram visualization

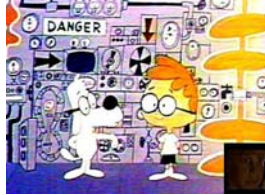


XSLT-based strategy

- XML (extended TEI) manuscript descriptions
 - One XML document per manuscript
 - Descriptions include titles of constituent works
- XSLT to construct SVG plectogram
 - Pass manuscript identifiers (file pointers) as parameters
 - Extract titles of all works in each manuscript (XPath)
 - Map to four-digit hex identifiers
 - Create one SVG column for each manuscript
 - Compare each column to its neighbors and draw lines to connect shared works
 - Animation

Why XSLT instead of XQuery?

- XSLT 1.0 adopted 1999
- Plectogram generator deployed 2002
- XQuery 1.0 adopted 2007



Could now be done with XQuery?

- Should it now be done with XQuery?
- What's wrong with the XSLT approach?
- How would XQuery be different (better)?

What might it mean to be better?

- Greater functionality
- Greater efficiency
- Greater ease of development
 - Simpler technology
 - Greater local expertise for the moment
 - Broader available expertise for the future
- Greater ease of maintenance
 - (See above)
 - Greater maturity or stability of technology

How are these solutions alike?

- Comparable functionality
- Comparable efficiency
- Comparable ease of development
 - Simplicity, local expertise, breadth of expertise
- Comparable ease of maintenance
 - As above
 - Maturity/stability of technology
- Shared XPath addressing

Why does it matter?

- Practically
 - Functionality, efficiency, ease of development, ease of maintenance
- If it doesn't matter practically, does it matter?

Does it matter in some other way?

- Aesthetically?
 - Why is *kludge* a bad word?
- Philosophically?
 - You could do it that way, but it would just be ...
 - ... wrong
- Which approach best fits the task?
- Are there similar questions that involve decisions with comparable practical consequences?
 - Is one solution better as the task grows?
- If you think it doesn't matter ...

Michael Kay 1

- XQuery may be seen as XSLT without ...
 - Template rules
 - Grouping
 - Format dates and times
 - Import modules and selectively override them
- XQuery has compensatory virtues and should not be considered a poor relation of XSLT

Michael Kay 2

- Prefer XSLT for large document publishing
- Prefer XQuery for searching for data in large XML databases
- Other (e.g., message conversion)
 - XQuery for small applications
 - XSLT for large applications (easier to change and reuse)

Priscilla Walmsley 1

- “XQuery and XSLT are both languages designed to query and manipulate XML documents. ... The line between querying and transformation is somewhat blurred.”
- XQuery and XSLT share
 - Data model (sequences, atomic values, nodes, items)
 - XPath 2.0
 - Built-in functions and operators
 - Equivalent components (e.g., <xsl:sort> and order by)

Priscilla Walmsley 2

- XSLT 2.0 provides
 - <xsl:analyze-string>
 - <xsl:result-document>
 - <xsl:for-each-group>
 - <xsl:import>

Priscilla Walmsley 3

- XQuery handles XSLT pull more easily than push
- Different optimizations
 - XSLT implementation may load entire documents into memory
 - XQuery may use indices to retrieve chunks without loading entire documents

The XSLT perspective

- Primarily designed for transforming one XML document into another (first sentence of the specification, cited by Michael Kay)
 - Transforming
 - One XML document
 - Into another
- XSLT is a transformation, not a query, language (Michael Thomas)

Input source(s)

- XSLT prefers one XML document
 - Plectogram generation has no single privileged input source
 - All manuscript files are equivalent input, and are passed as parameters (pointers)
- How does XSLT *prefer* anything?
 - Applications expect a single input
 - dummy.xml
 - <dummy/>

Other (sometimes) arbitrary choices

- XSLT push or pull?
- XML or something else (e.g., RDBMS)?
- XPath predicate or XSLT <xsl:if> / XQuery “where” clause?
- Element or attribute?
 - <xsl:text> or plain ol/ text?
- XSLT element and attribute constructors or literal result elements

Pushmi-pullyu



XSLT push vs pull

- Some XSLT problems can be addressed with either push or pull
- Which is better?
- XSLT push
 - <xsl:template match="gi">
- XSLT pull
 - <xsl:for-each>
 - <xsl:value-of>

Push and pull

- Push
 - Deal with what comes your way
- Pull
 - Ask for what you want when you want it

Push and pull, XSLT and XQuery

- XQuery pulls
 - No template rules
- XSLT can push or pull
 - Which does it prefer?
 - For what type of information?
 - XSLT push deals more easily with heavily mixed content
 - Less explicit testing

XSLT pushers and pullers

- Pushers
 - Bob DuCharme
 - Alex Milowski
 - Uche Ogbuji
- Puller
 - Kevin Williams

Bob DuCharme

- “Push, Pull, Next!”
- “Few stylesheets rely strictly on push or pull”
- XSLT push is *event-driven*
 - Whenever you see X, do Y
- XSLT pull is *procedural*
 - First do Y, then do Z

Alex Milowski

- “There are Monsters in My Closet or How Not to Use XSLT”
- Overuse of: `<xsl:if>`, `<xsl:choose>`, `<xsl:for-each>`, `<xsl:value-of>`
- Use templates instead of `<xsl:value-of>`
- Emphasizing processing efficiency and optimization

Uche Ogbuji

- “Rescuing XSLT from Niche Status” (2001)
- “Push vs pull XSLT” (2005)
- “Pull ... is false simplicity ... Pull is easy when the problem space is simple”

Kevin Williams 1

- “XML for Data: XSL style sheets: push or pull?”
- Push stylesheets
 - “Jump from template to template”
 - Except for simple examples, difficult to read and maintain

Kevin Williams 2

- Pull style sheets
 - “Rely on the coder's ability to know which elements are expected to appear next in the source document.”
 - Er ... or is which elements the user wants to appear next in the *result* document
 - Shorter, easier to read
- Contrasts “data” (use pull) and “narrative” (use push) documents

XML or RDBMS?

- “Data-centric” vs “document-centric”
 - All XML files are documents
 - All documents are data
- Michael Thomas
 - “Highly structured, fine grained”
 - “Loosely structured ... document,” “semi structured ... books, web pages”
- Conceptually
 - RDBMS → Table → Record and field
 - XML → Tree → Ordered hierarchy of content objects
 - XML can represent records and fields
 - How well can an RDBMS represent trees?

For this project we care about ...

- Query and retrieval
- Formatting and delivery

We take for granted or ignore ...

- Storage / persistence
- Modification (insert, delete, update, etc.)
- Caching
- Indexing for optimization
- Constraint checking
- Transmission
- Concurrency
- Locking
- Joins
- Normalization

XML document (tree) or database (table)?



Context

- Ronald Bourret, “XML and Databases”
- Akilah Jackson, “Using Native XML Databases vs. Relational Database Management Systems for Storing, Retrieving, and Querying Denormalized Data”
- Michael Thomas, “XML In An RDBMS World”

Ronald Bourret

- “XML and Databases”
- “... data-centric/document-centric divide is somewhat dated ... many XML documents are not strictly data-centric or document-centric, but somewhere in between”

Akilah Jackson

- “Using Native XML Databases vs. Relational Database Management Systems for Storing, Retrieving, and Querying Denormalized Data”
- “Native XML-databases are ideal for storing, retrieving, and querying denormalized data.”
- “A document-centric approach differs from a data-centric approach in that is the content is not as structured; it is semi-structured ... Content that varies in length and type or empty or missing elements ...”
- SQL isn’t good at addressing documents
 - May not be an issue when you control your data

Michael Thomas

- “XML In An RDBMS World”
- Emphasis on XML as a transactional database, and therefore on concurrency, transactional isolation, locking, etc.
 - Which we don’t care about
- XML data is ordered, RDBMS rows are not
 - Order can be ignored in XML and faked in RDBMS

Dictionaries as edge cases

- The Zaliznjak *Russian Grammatical Dictionary*
 - Lemma (noun)
 - Gender
 - Declension class
 - Accentual paradigm
 - Notes and comments
- Tabular information and hierarchical mixed-content prose
- ввести 6 св 7в/в (-д-) "9", е" (но прич. прош. вве'дший) \$ вводи'ть

XPath predicate or XSLT <xsl:if>?

- Get all the <articleName> elements in Bulgarian
- XPath predicate
 - <xsl:for-each select="\$dr944//articleName[@lang eq 'bg']">
- XSLT <xsl:if>
 - <xsl:for-each select="\$dr944//articleName">
 - <xsl:if test="@lang eq 'bg'">

XPath predicate or XQuery “where” clause?

- for \$i in doc('/db/mss/DR57CMS.XML')//articleName where \$i/@lang eq 'bg' return <hit>{\$i}</hit>
- If you grew up on SQL ...
- eXist optimizes XPath better than “where” clauses
- Or use both together!

Element or attribute?

- Given our information ...
 - XML vs RDBMS (etc.)
- Given XML to process ...
 - XSLT vs XQuery
- Given XSLT ...
 - Push vs pull in XSLT
- Given an XML schema to design ...
 - Element or attribute

In the document: order

- Elements must be ordered
- Attributes cannot be ordered
- Can order be ignored or faked?

In the document: repetition

- Elements may repeat
- Attributes cannot repeat
- Can repetition be ignored or faked?

In the document: substructure

- Elements may contain text, attributes, and subelements
- Attributes may contain only CDATA
- The TEI P5 “War on attributes”
 - Gaiji (<g>) in attribute values

In the schema 1

- Old Church Slavonic Dictionary
 - Noun forms must have gender, verb forms may or may not
 - Personal pronouns must have person, verb forms may or may not have person, nothing else can have person
- The *presence of values* for some grammatical categories depends on the *values* of others

In the schema 2

- The *presence of values* for some grammatical categories depends on the *values* of others
 - In a DTD, the presence of attributes cannot depend on the presence or value of other attributes
 - In a DTD, element hierarchies can enforce these constraints
 - In RelaxNG, the same selectional and co-occurrence constraints may be applied to attributes and elements

<xsl:text> or plain ol’ text

- If your XSLT doesn’t break because of namespaces ...
- ... it breaks because of whitespace handling

Constructors and literal results

- `<xsl:element name="html"> ... </xsl:element>`
- `<html> ... </html>`
- Computed vs static values
- Validate against the XSLT DTD

How does the world look to you?



The ~~philosophy~~ religion of programming languages

- perl
 - There is more than one way to do it
- python
 - There should be one—and preferably only one—obvious way to do it



perl

There is more than one way to do it.



python

There should be one—and preferably only one—obvious way to do it

Programming language philosophy

- perl
 - There is more than one way to do it
 - It doesn't matter which you choose
- python
 - There should be one—and preferably only one—obvious way to do it
 - The other ways are worse

Keep it pure

Tools have purposes

- Does it matter on a general level ...
- ... even when it doesn't on a project-specific level

Or do they?



Where it doesn't make a difference

- XSLT is for one-to-one XML-to-XML transformation
- XSLT push is better than pull
- XQuery is for multiple sources
- RDBMS vs XML
 - If hierarchy or order matters, think XML
 - If you have unordered records with a stable number of fields, think RDBMS
- Element vs attribute
- XSLT constructor vs literal result
- XPath vs <xsl:if> or XQuery "where" clause

Exaptation



Thank you!

David J. Birnbaum
 Department of Slavic Languages and Literatures
 University of Pittsburgh
 djbpitt@pitt.edu
<http://clover.slavic.pitt.edu/~djb/>