

Standards Update: XSLT 3.0

Michael H. Kay

Main Themes of XSLT 3.0

- Streaming
- Packages (separate compilation)
- Richer data types and function library

Streaming

- **xsl:stream** reads a document and processes it in streaming mode
- Streaming modes and templates allow streamed rule-based processing
- Language restrictions to ensure streamability, plus extensions to ensure that the use cases can be satisfied
- **xsl:iterate** provides “iteration with memory”
- **xsl:merge** for multiple streamed inputs
- **xsl:fork** for multiple streamed outputs

Streaming Status

- Spec largely stable since my talk at XML Prague 2010
- Many features implemented in Saxon-EE 9.3
- Demonstrated at Balisage 2010
 - 15Gb transformation in 30 minutes
 - 45Gb input, 15Gb output in 6 hours

Packages

- Objectives:
 - efficiency benefits through separate compilation
 - software engineering benefits through improved modularity and reuse
 - IP protection of source code
- Features:
 - ability to mark code public/private/final/abstract
 - interfaces between packages
 - control over customizability

Packages Status

- Outline spec agreed during XSL WG meeting last week
- Needs to be written up by the editor!
- No implementation activity yet.

New data types and functions

- Higher order functions
- Maps
- Trigonometry
- Parse/Serialize XML
- Parse/Serialize JSON

Higher-order Functions in XPath 3.0

- **Inline functions**
 - `my:sort(customer, function($c) {$c/zipcode})`
- **Function literals**
 - `my:sort(customer/name, lower-case#1)`
- **Curried functions**
 - `my:sort(employee/salary, substring-after(?, '$'))`
- **Starter set of higher-order functions**
 - `map()`, `filter()`, `fold-left()`, `fold-right()`, `map-pairs()`

JSON support in XSLT 3.0

- `parse-json()` converts JSON to XDM maps
- Lossless conversion
- Can then use template rules to convert to custom XML vocabulary
- Validating against popular XML representations of JSON
- `serialize-json()`
 - handles `parse-json()` results “out of the box”
 - customizable using callback function

Other XSLT 3.0 Functionality

- **xsl:evaluate**
 - dynamic XPath evaluation
- **xsl:try/xsl:catch**
- extended syntax for patterns
- composite keys (for grouping and **key()**)
- expanded names
 - “uri”:local
 - allowed anywhere a QName can appear
- declare context item type
 - for the stylesheet
 - for a template