FunctX:
A Case Study in XML Processing

Priscilla Walmsley
Datypic
XML Prague
March 21, 2009
What is FunctX?

- A set of reusable functions for XQuery and XSLT
- An XML markup language for defining and documenting functions
- An application for documenting and testing function libraries
FunctX: The Function Library
• Over 100 functions
• Useful, reusable functions like substring-after-last-match and distinct-element-paths
• Provided in both XQuery 1.0 and XSLT 2.0
• Open to contributions by anyone
• http://www.functx.com
  – Or xqueryfunctions.com, xsltfuctions.com
• Reuse: help developers create XQuery/XSLT applications quickly
  – pre-tested with multiple processors
  – easy to reuse the whole library (import), or just one function (cut and paste)

• Educational tool
  – Teach about XQuery/XSLT techniques/syntax

• Promote good design
• Goal is *not* standard extensions to processors (à la EXSLT/EXQuery)
  – XSLT/XQuery is not always the language to use for extension functions
  – some FunctX functions would be too obscure to be worth trying to standardize across processors

• But EXSLT/EXQuery can/should peacefully coexist
General Design Philosophy

• Make code clear so it is easily understood

• Try to use XPath 2.0 for function bodies
  – to allow sharing of the definition between XQuery and XSLT

• Create robust functions
  – pay attention to namespaces, types, the empty sequence, etc.

• Adjust functions as necessary to make them work with multiple processors
FunctX: The XML Vocabulary
An XML Markup Language

- An XML vocabulary for defining functions, their associated documentation and example/test cases
- Described by an XML Schema and accompanying validation XSLT
- Namespace is http://www.datypic.com/xmlf
The Function XML

- Audit information
  - history and source of the definition, in tracking and source
- Short and long description
- List of the arguments and return type
- Test and example cases
- The function body itself
- Information on dependencies and related functions
The Library XML

• Defines scope of the library
• Provides general parameters of the library
• Defines categories for organizing the documentation
FunctX: The Application
Purpose

• Creating your own function libraries
  – for a specific XML vocabulary
  – for a specific use case
  – to share across project teams
  – to encourage reuse

• Automatically documenting your library

• Automatically testing your library
1. Validate

2. Generate Function Library (XSLT)

3. Generate Test Cases (XSLT)

4. Run Test Cases (Various Processors)

5. Run Test Reports (XSLT)

6. Generate HTML (XSLT)

7. Generate PDF (XSLT + FOP)

- **Schema (XSD)**
- **validate.xsl**
- **func_library.xsl**
- **func_library.xq**
- **tests.xsl**
- **tests.xq**
- **tests.xml**
- **testoutput.xml**
- **testreport.html**

**Function Definitions (XML)**
FunctX Generation of Documentation

- HTML
  - 1 page per function with code, examples, etc.
  - pages that organize functions into categories from the library XML
  - page that lists all functions alphabetically

- PDF
  - one PDF for the entire library
  - fully hyperlinked and bookmarked
FunctX Testing Capabilities

• Generates a test script containing the sample function calls found in the function XML

• Runs the test library on multiple processors via Ant tasks
  – currently Saxon, eXist, MarkLogic and DataDirect

• Compares actual results to expected results and generates a report on the differences
Version 1.0 of the library has been out for 2 years

Version 1.1 is coming soon

– will include:
  • Revised/expanded function library
  • Application code (XSLT, schemas, Ant script, samples)

– will become an open source project
Future Enhancements

• Handling fatal errors more gracefully
  – Currently, the entire testing process stops if one test crashes

• Supporting more than one signature for a function name

• Taking types into account when comparing test results
  – Currently compares results as either strings or XML
Thoughts and Lessons Learned

• I love XSLT 2.0
  – multiple result documents
  – regular expressions
  – lots of new functions

• XQuery processors vary widely
  – bugs
  – multiple versions
  – differences in the static context
  – levels of static typing