



Implementation Techniques

Panel Discussion

Moderated by: Hugh Wallis

Director, Technology Standards, XBRL International Inc.

hughwallis@xbrl.org



- Topics
 - Implementation issues around
 - Dimensions
 - Formula
 - Versioning
 - Rendering
 - Open source approaches



- Panellist Introductions
 - Diane Mueller, JustSystems
 - Herm Fischer, UBmatrix
 - Masatomo Goto, Fujitsu
 - Michele Romanelli, Banca d'Italia
 - Philip Allen, Decisionsoft/Corefiling



Business Reporting
for Better Decisions
by Managers, Investors, Governments and Citizens



Diane Mueller



We all agree on the importance of developing a wide user base for XBRL content.

– The Question

- How can we drive this wider adoption?

– The Answer

- By making it easier to develop financial applications that can leverage XBRL



What do Financial Applications Developers need?

- access to the right enabling technology
- development tools and runtime software that can automatically overcome technical issues
- without passing the burden of deep specification knowledge to the developer.



- Today, 4-5 proprietary XBRL processors in use available on the market
 - All have proprietary licenses
 - Costly to develop and maintain
 - All issue different error messages, and leverage different architectures, offer varying interpretations of the XBRL specification and it's modules
- Software Interoperability is a major issue



Open Source Makes it Easy, We* Already Use it!

- XERCES: a validating XML parser
 - Xerces **makes it easy** to give applications the ability to read and write XML data. A shared library is provided for parsing, generating, manipulating, and validating XML documents using the DOM, SAX, and SAX2 APIs.
- DITA Open Toolkit: a implementation of the Darwin Information Typing Architecture (DITA) specification.
 - The Toolkit's transforms **makes it easy** to give applications the ability to manipulate DITA content (topics and maps) into deliverable formats like web (XHTML), print (PDF), and online Help. It is a set of Ant- and Java-based, open source tools that provide a "reference implementation" for processing DITA maps and topical content to multiple output formats.
- Apache Ant: A tool for automating software build processes.
 - Ant uses XML to describe the build process and its dependencies, whereas make has its Makefile format. By default the XML file is named build.xml. Because Ant made it trivial to integrate JUnit tests with the build process, Ant has **made it easy** for developers to to integrate JUnit tests with the build process and adopt test-driven development practices.

*



Open Source Definition

Open source doesn't just mean access to the source code!

1. Free Redistribution
2. Source Code Included
3. Possible to create derived works
4. License must explicitly permit distribution of software built from modified source code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor
7. Rights attached to the program must apply to all to whom the program is redistributed
8. License Must Not Be Specific to a Product
9. License Must Not Restrict Other Software
10. License Must Be Technology-Neutral

Anyone can look at it, modify it, or improve it, provided they agree to share their modifications under the same terms...



Project Type	Other Dependencies	Project's License	XBRL 2.1, FRTA 1.0 & Dimension 1.0 Support	Free & Open?	Link
Viewer	MSFT WinXP , C# Compilation depends on Syncfusion , MSFT .NET Framework ((MS-RSL)	Common Public Attribution License 1.0 (CPAL) (badgeware)	No	((MS-RSL) Licensees may not modify or redistribute the source code.	http://sourceforge.net/projects/rivetdragonview/
Tagger	MSFT WinXP , C# Compilation depends on Syncfusion , MSFT .NET Framework ((MS-RSL)	Common Public Attribution License 1.0 (CPAL) (badgeware)	No	((MS-RSL) Licensees may not modify or redistribute the source code.	http://sourceforge.net/projects/rivetdragontag/
Mapper	Batavia XBRL Java Library 1.x (BXJL) from www.batavia-xbrl.com where it is available for free with a non-production license	AFFERO GENERAL PUBLIC LICENSE	No	permitted to copy and distribute verbatim copies, but changes not allowed .	http://sourceforge.net/projects/batavia-xbrl/
Processor/API	none	GNU General Public License (GPL)	this version does not support new US-GAAP	Modifications & Redistribution allowed	http://sourceforge.net/projects/ubmatrix-xbrl/
Processor/API (read-only)	none	GNU General Public License (GPL)	YES	Modifications & Redistribution allowed	http://sourceforge.net/projects/xbrlapi/



- OS software stacks up well against commercially developed software both in quality and in the level of support that users receive
- Defects are found and fixed very quickly because there are "many eyeballs" looking for the problems
- OS lets application developers focus on their "core competencies" and ensures interoperable use of standard



- What's Next? An Incubator for XBRL OSS?
 - Consortium support for Open Source efforts
 - Resources, Community outreach, funding
 - A first step?
 - RFP for Inline XBRL Extractor XSLT (link)
 - Potential for an Open Tool Kit for XBRL?
 - To Include
 - » A series of Transformational XSLT (Inline XBRL, HTML, XBRL, RDF)
 - » APIs for searching, analyzing and presenting XBRL data
 - » APIs for both reading/writing XBRL



Business Reporting
for Better Decisions
by Managers, Investors, Governments and Citizens



Herm Fischer



Issues which might be relevant

- Suggested points for discussion
 - Open source ramblings
 - Build your own tools
 - Build by yourself from scratch
 - Formula progress



Open source alternatives

- Source Forge UBmatrix XPE 2.5
 - Ancient version of XPE
 - API's lower level (different) than current product
 - Dimensions from history
 - No formulas
 - No web-cache
- Retail XPE 3.5
 - Easy high-level API's, dimensions, formulas, web
 - Fast execution



Build your own tools

- XPE 3.5 works
 - Tree walks and queries optimized
 - Formulas CR2 level
 - Focus on integration and GUI
- Other vendor products available
- DOM works
 - Not really for roll-your-own tools



Build by yourself from scratch?

- You may have the experience
 - UBm TD built by myself, fully compliant but changing to XPE for maintainability and formulas
 - Cliff's product by himself (no formulas)
 - Most others by small very-experienced teams
- DOM is enough to start (plus 1-2 yrs.)
- But, a real, existing, maintained, processor needed for commercially viable product



Formula or {Schematron, NTP}

- Formula is following a process
 - No desire to repeat dimensions mayhem
 - Time is of the essence but not dispensable
- Projects have needs of their constituency
 - Formula spec has found users and does work
 - but --
 - Schematron is an existing product
 - NTP has timing and IP requirements
 - We will not gripe or bicker (have a beer please)



Formula marches onwards

- CR-2 is in preparation and close to issuing
 - Fixes of base spec dimensions errata
 - Simplified dimensions filtering
- Formula use, in or near, production
 - COREP (Bank of Spain)
 - FINREP (Bank of France)
 - SEC (XBRL submission requirements)



Formulas at US SEC

- General Requirements to Process Mandate XBRL Submissions
- Existing system was augmented with new validation rules and restrictions



SEC Edgar Submission Checking

- Short implementation cycle
- US-GAAP submissions with DEI information
- Existence assertions
 - Mandatory DEI concept items
- Value assertions
 - DEI specific concept item requirements
 - Instance document issues (context, namespaces)
 - Taxonomy issues
 - Implemented by functions for the moment
- Test case harness



Why formula approach

- Fast implementation
 - One week turnaround (nearly 24x7, sigh)
 - Hours to first-integration with production system
- No serious Java coding required
- Assertions fit the requirements
- FWG test cases harness utilized



Process of developing

- Formulas first developed in 3 linkbases
 - DEI concept item existence & values linkbase
 - General instance issues assertions linkbase
 - Taxonomy issues assertions linkbase
 - (with functions calls to Java API)
- Quick testing (with real data, hacked up)
- Then
 - formulas broken up into separate linkbases
 - FWG test case harness adopted



Process of executing

- Need to fire selectively
- For each submission instance
 - Load instance
 - Production system does its magic checks
 - Determine applicable formulae (assertions)
 - Dynamic attach of formula linkbases
 - Nice error messages for users
 - Trace information for help desk



Extensions of CR-2

- Labels of messages
 - Adopted VM Wiki feedback was for labels:
text {xpath} more text
- Dynamic attachment of linkbases
- Test case harness usage



Testing, testing, and testing

- Adopt FWG test case approach
 - On desktop, this is obvious
 - On production server, this is a neat solution
 - FWG framework “generates” production submissions
 - Production and unit test versions of each testcase
 - Production has full data and all formulas
 - Unit test strips down data and just one formula



Business Reporting for Better Decisions

by Managers, Investors, Governments and Citizens





Thank You!

Enjoy Washington!

