ODF Interoperability: The Price of Success

Robert Weir
Software Architect
IBM Software Group
robert_weir@us.ibm.com
http://www.robweir.com/blog
What is Interoperability?

“Interoperability means the ability of information and communication technology (ICT) systems and of the business processes they support to exchange data and to enable the sharing of information and knowledge.”

IDABC's “European Interoperability Framework”
Legos – the intuitive example

Interoperable since 1958.

0.002mm tolerances.
Many ODF Implementations

With $N$ editors, there are $N^*(N-1)$ interoperability paths: 2, 6, 12, 20, 30, 42, 56, 72, 90
And don't forget the non-editors

Before:

Paper

Now:

Search Engine

Web

Web Service

Database

A single document can easily be touched by a dozen different applications from different vendors during its lifetime.

The ultimate destination of your document is unknown to you and likely unknowable.
The Interoperability Tax

Search Engine

Web

Web Service

Database

= processing step with loss caused by poor interoperability

Losses may be:

- Fidelity
- Data
- Performance
- User frustration
- Reputation
- Opportunity
Perfect Interoperability is Easy*

* But expensive

Cost per transaction

Level of fidelity

Redo whatever automation fails to handle

Manual rework

\[ \text{total\_cost} = \%\text{automatic}\times\text{cost\_automatic} + \%\text{manual}\times\text{cost\_manual} \]
The Goal

- But expensive

Level of fidelity

Cost per transaction

Automation

Manual rework

Starting Point

Improve the level of interoperability within the ecosystem

* But expensive
A range of available editors

- emacs
- wiki editor
- HTML editor
- OpenOffice
- Illustrator
- Photoshop

Structure vs. Visual Specificity
And a range of formats
And in terms of control...

User-to-User fidelity is high here

Modern WYSIWYG Editors are caught in the middle

interoperability with business processes is high here

Control of the Author

Control of the Receiver

JPEG

PDF

ODF

HTML

DITA/DocBook
So what do you emphasize?

- Modern word processor has evolved into a multi-paradigm tool that supports different styles of use:
  - Highly structured data oriented use
  - Ad-hoc, visually-oriented layout

- Users have expectations that OpenOffice is suited for both uses. Until the last person who ever used a typewriter is dead, this will continue.
Traditional Trade-offs

1. Visual Richness of authoring environment
2. Power
3. Ability to say anything
4. Pixel Perfection
5. High Fidelity

1. Accessibility
2. Universality
3. Ability of everyone to understand
4. Structure
5. Semantic richness

Not a Law of Nature, but a tendency. The glory goes to those who can solve both problems at once.
Vigorous writing is concise. A sentence should contain no unnecessary words, a paragraph no unnecessary sentences, for the same reason that a drawing should have no unnecessary lines and a machine no unnecessary parts. This requires not that the writer make all his sentences short, or that he avoid all detail and treat his subjects only in outline, but that every word tell.

--William Strunk Jr. in The Elements of Style
Things that cause problems

• Application issues
  – Implementation defects
  – Functional subsets
  – Functional supersets (extensions)

• Standard issues
  – Specification errors
  – Undefined behaviors
  – Implementation-defined behaviors
The Conundrum

OpenOffice

KOffice

ODF Standard

Google Docs & Spreadsheets

What is the effective overlap?
Solution Patterns

• Standards-development
  – Multi-vendor, multi-stakeholder participation
  – Expert review
  – Implementation concurrent with standards development

• Standards
  – Detailed conformance clauses
  – Deep schemas, allowing deep validation
  – Reference implementations

• Post-standardization activities
  – Translation of standard
  – Development of conformance assessment/certification
  – Multiple implementations
A powerful pattern

Standard

Reference Implementation

Test Suite
A powerful pattern

- The standard contains the definition of a conformant document
  - (but the standard may have errors or ambiguities)
- The test suite exercises and validates each feature of the standard
  - (but the test suite may have errors or omissions)
- The reference implementation is written to the standard, and tested with the test suite
  - (but the implementation may have errors or missing functionality)
Checks and Balances

• A test case fails. What is the cause?
  – An error in the application?
  – Is it an error in the test suite?
  – An error in the standard?
• Identify the cause of the failure
• Fix
• Continue until you have a complete test suite and a reference implementation that passes all of the test cases.
A Reference Implementation

- Should implement 100% of the standard, including all optional requirements.
- It should be the first one, or one of the first applications to implement any new feature in the standard.
- For any implementation-defined behaviors, it should document how it behaves.
- Although it may extend the standard, it should have a mode of operation where it is strictly conformant.

- Let's work to make OpenOffice.org be the full reference implementation for ODF!
A Test Suite: A rough estimate

- ~ 700 page ODF specification
- ~ 5 testable statements per page
- ~ 4 test cases per statement to test limits, positive and negative test cases, etc.

- So, on the order of 10,000 test cases, or 2 PY of effort.
That takes care of OpenOffice

• But what about everyone else?

• Having a single ODF Reference Implementation, by itself, doesn't really solve the interoperability problem.

• But by having a good standard, a good test suite, and a good reference implementation, we allow other implementations to benefit as well.

• Think of it as a “public health” issue. We are only healthy if we ensure that others can be healthy as well, otherwise the system falls apart.
This can help move us from...

With \( N \) editors, there are \( N^*(N-1) \) interoperability tests
With $N$ editors, there are $N$ interoperability tests
Things that foster interoperability

- **In applications:**
  - use of interoperable data formats
  - a strictly conforming mode of operation
  - guidance to the user on how to use the product in an interoperable way
  - inclusion of document templates and defaults that encourage interoperability
  - allowing validation of documents

- **In data formats:**
  - clean separation of content, attributes, behavior and metadata
  - reuse of existing, established standards
  - thorough review
  - standardization
Things that foster interoperability

- In organizations:
  - adoption of a single standard document format
  - adoption of applications with proven conformance to that document standard
  - training of users on how to create interoperable documents

- In users:
  - capture information at the highest level possible
  - adding metadata
  - providing annotations for accessibility
  - using named styles
Progress in Interoperability

- Test Suites
- Validators
- Translators
ODF Test Suite

OpenDocument Fellowship
Developers Corner

Table of Contents
- Home
  - General (4 + 29=33)
    - Document Structure (4)
    - MetaData (4)
      - Predefined Metadata
      - User defined Metadata
      - Multiple Document Contents
    - SingleDocumentContents
      - Document types (0 + 6=6)
      - Document metadata (0 + 2=2)
      - Application settings (0 + 4=4)
      - Scripts (0 + 1=1)
      - Font declaration and styles (0 + 4=4)
      - Page Layout (0 + 6=6)
      - Change tracking (0 + 6=6)
    - Formatting Properties (130)
    - Text Contents (13 + 22=35)
    - Text Frames (2)
    - Drawing Content (33 + 11=44)
    - Presentation Content (55 + 5=60)
    - Spreadsheet Content
    - Chart content (0 + 30=30)
    - Images Content
    - International
- Book
- Test suite
- Search
- ODF Tools

OpenDocument Sample Documents

We are working towards developing a comprehensive set of sample documents for the OpenDocument specifications developed by the Oasis consortium.

Most documents has associated rendering samples created with:
- OpenOffice
- KOffice on Linux

The results are listed in the summary

The set of sample documents was developed at the Networking and Mobile Computing Laboratory at the School of Electrical Engineering and Computer Science of the University of Central Florida.

Developed by: Yi Luo and Majid A. Khan under the supervision of Lotzi Bölöni.

KOffice samples and automation contributed by Thomas Zander.

The development was sponsored by Intel corporation and released under the Creative Commons Attribution 2.5 License.

Contact us at: opendocument-sample mailing list.

http://develop.opendocumentfellowship.org/testsuite
ODF Validator

ODF Validator - OpenDocument Validation Service

This is the Fellowship’s ODF Validation Service, a free service that checks OpenDocument files for conformance with the ODF specification.

Select file
Browse... validate

Privacy
We take your privacy seriously. Neither the Fellowship nor Cyclone3 will ever sell or distribute any document you upload. Uploaded documents are not used for any purpose other than validation. All documents are destroyed as soon as validation is complete.

Guidance
ERROR: Document not conform with the ODF specification.
WARNING: Not a violation of the specification but may indicate problems.
For example: If the file does not contain a mimetype the validator will produce a warning since a mimetype is a SHOULD in the ODF specification. But an undefined mimetype is an error as it violates the ODF spec.

Acknowledgements
ODF validator written by Alex Hudson as part of the Fellowship’s ODF Tools project. Web service provided by Cyclone3 and maintained by Roman Fordinal. Learn more about Cyclone3 ODFValidator

http://opendocumentfellowship.org/validator
ODF Add-in for Microsoft Word

**Overview**

The goal for this project is to provide an Add-in to Microsoft Word XP/2003/2007 to allow opening and saving OpenDocument format (ODF) files.

The converter is based on XSL transformations between two XML formats, along with some pre- and post-processing to manage the packaging (zip / unzip), schema incompatibility processings and the integration into Microsoft Word. We chose to use an Open Source development model that allows developers from all around the world to participate & contribute to the project.

Along with the Add-in for Microsoft Word, we also provide a command line translator that allows doing batch conversions. This translator could also be run on the server side for certain scenarios.

http://odf-converter.sourceforge.net/
Sun Microsystems Announces OpenDocument Format (ODF) Plug-in Application for Microsoft Office

Users of accessibility devices now fully able to participate in organizations switching to ODF

MENLO PARK, Calif. February 7, 2007 Sun Microsystems, Inc. (NASDAQ: SUNW), the largest code contributor to free and open source communities, today announced the upcoming availability of the StarOffice 8 Conversion Technology Preview plug-in application for Microsoft Office 2003. The early access version of the OpenDocument Format (ODF) plug-in, available as a free download, will allow seamless two-way conversion of Microsoft Office documents to ODF.

"Organizations can now consider switching to ISO/IEC 26300 OpenDocument Format while protecting employees needing assistive devices only supported by legacy Microsoft software," said Rich Green, executive vice president, Software at Sun Microsystems. "ODF is important because it ensures documents will still be readable long into the future while allowing a wide choice of proprietary and open source software choices to work with the documents."

The StarOffice 8 Conversion Technology Preview is primarily based on the OpenOffice.org platform, the open-source office productivity suite developed by the OpenOffice.org community including the founder and main contributor Sun Microsystems. Sun offers distributions and configurations of and support for OpenOffice.org under the StarOffice brand. The initial plug-in application will support the conversion of text documents (.doc/.odt) only, but full support of spreadsheet and presentation documents is expected in April. The conversion is absolutely transparent to the user and the additional memory footprint is minimal.

http://www.sun.com/software/star/openoffice
# Thursday 20th - ODF Camp

## Timetable

**Location:** Same as OOoCon

<table>
<thead>
<tr>
<th>Time</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:00 - 09:15</td>
<td>Introductory Statement</td>
</tr>
</tbody>
</table>
| 09:15 - 09:45 | Keynote: Peter Vandenabeele  
ODF plug-ins and other solutions to implement the Belgian open standards directive |
| 09:45 - 10:00 | How does the ODF Boot Camp Work?                                                |
| 10:00 - 10:45 | Session 1: Review of documents                                                     |
|              | Break                                                                             |
| 11:15-13:45  | Session 2: Review and analysis of documents                                       |
|              | Lunch Break                                                                       |
| 15:00-16:45  | Session 3: Coding                                                                |
|              | Break                                                                             |
| 17:00-18:30  | Session 4: Coding, solutions                                                      |
| 18:40-19:00  | Wrap up and next steps                                                           |
One standard
One test
Accepted everywhere

World Standards Day 14 October 2002