



Processing ODF

Lars Oppermann
Software Engineer
Sun Microsystems



Office Productivity Documents

- Traditionally, office documents are confined to applications.
- We use office documents to express and share important information
- Office documents are an important part of collaboration
- Office documents have been hard to process without the application in which they were created.

Better: Open File Formats and ODF

- Standardized, free to use for anyone
- Easy to access programmatically
- Based on existing standards
 - > Zip, XML
 - > HTML, CSS, SVG, MathML etc...
- Programming platforms support for base technologies
- Well known and understood vocabularies

OpenDocument Format (ODF)

- Specification available at <http://www.oasis-open.org>
- Get “Open Document Essentials” by David Eisenberg!
- Look at documents and experiment
 - > Unzip and use your favorite text editor
 - > Edit doc in OpenOffice and see what happens

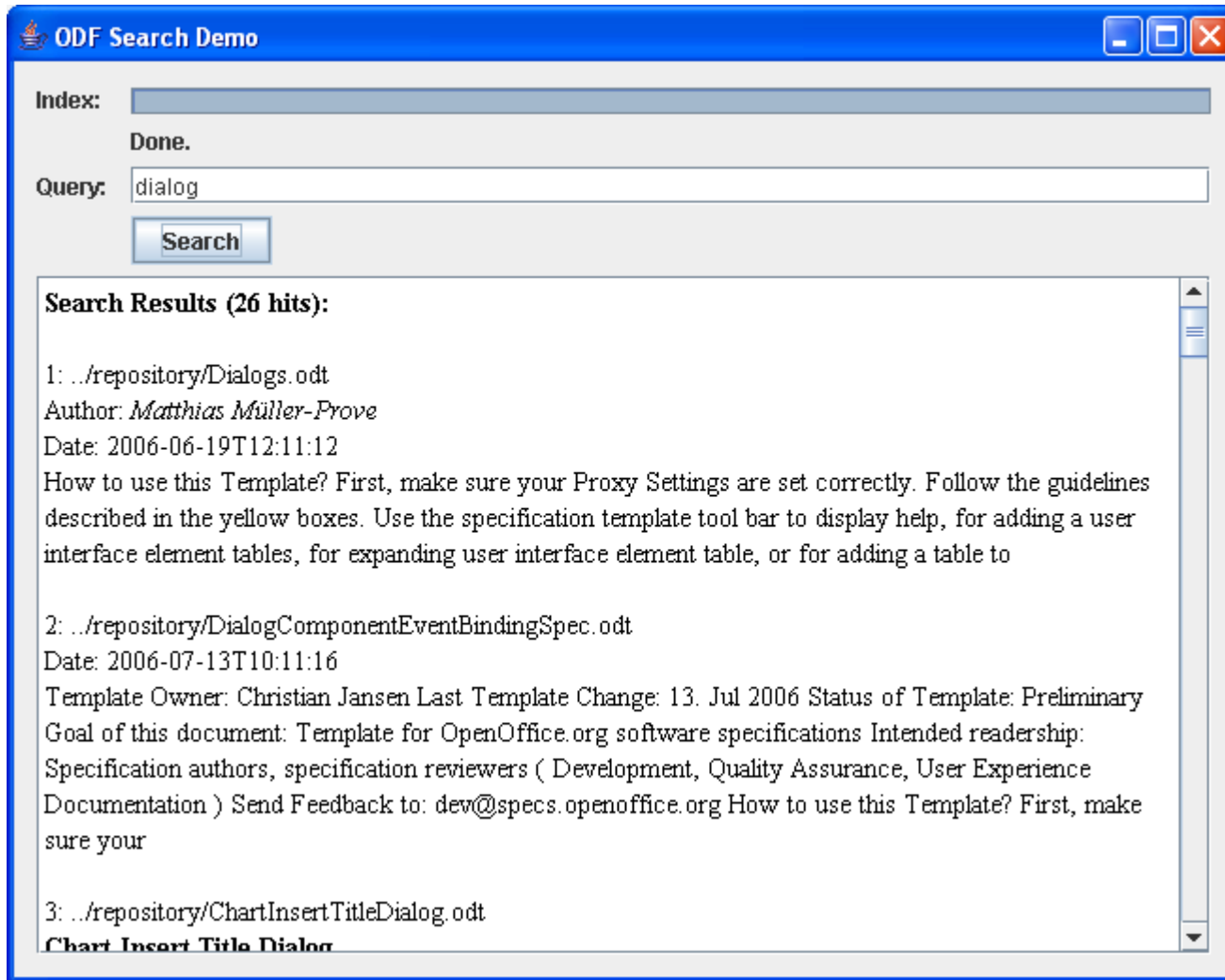
Looking into an ODF file

- Basic anatomy of an ODF file
 - > Zip container
 - Manifest, mimetype and streams
 - content.xml
 - meta.xml
 - styles.xml

Building a search engine

- Adding ODF support to apache Lucene
 - > Wrapping ODF XML as lucene documents
 - > Scanning a directory and build an index
- Search for document content and meta data

ODF search engine (cont.)



ODF Search Demo

Index:

Done.

Query:

Search Results (26 hits):

1: ../repository/Dialogs.odt
 Author: *Matthias Müller-Prove*
 Date: 2006-06-19T12:11:12
 How to use this Template? First, make sure your Proxy Settings are set correctly. Follow the guidelines described in the yellow boxes. Use the specification template tool bar to display help, for adding a user interface element tables, for expanding user interface element table, or for adding a table to

2: ../repository/DialogComponentEventBindingSpec.odt
 Date: 2006-07-13T10:11:16
 Template Owner: Christian Jansen Last Template Change: 13. Jul 2006 Status of Template: Preliminary
 Goal of this document: Template for OpenOffice.org software specifications Intended readership: Specification authors, specification reviewers (Development, Quality Assurance, User Experience Documentation) Send Feedback to: dev@specs.openoffice.org How to use this Template? First, make sure your

3: ../repository/ChartInsertTitleDialog.odt
Chart Insert Title Dialog

Direct XML Processing

- Available in most programming environments
- Widely used and understood
- Flexible
- Low level of abstraction

XSLT

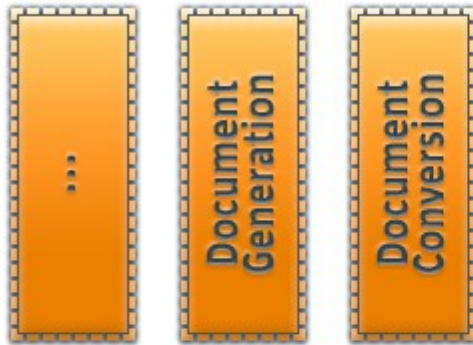
- Good for processing content on the XML level
 - > Conversion
 - > Extraction
 - > Merging
- Complicated if more abstraction is needed
 - > Items that have semantics beyond the XML infoset need special attention. E.g. style inheritance.
- Some infrastructure needed to work on packages
 - > Using “flat” representation of ODF may be an alternative

Frameworks and Toolkits

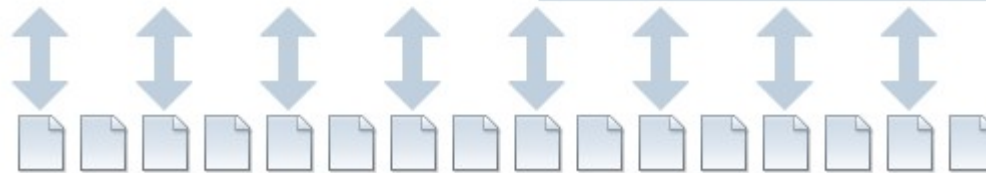
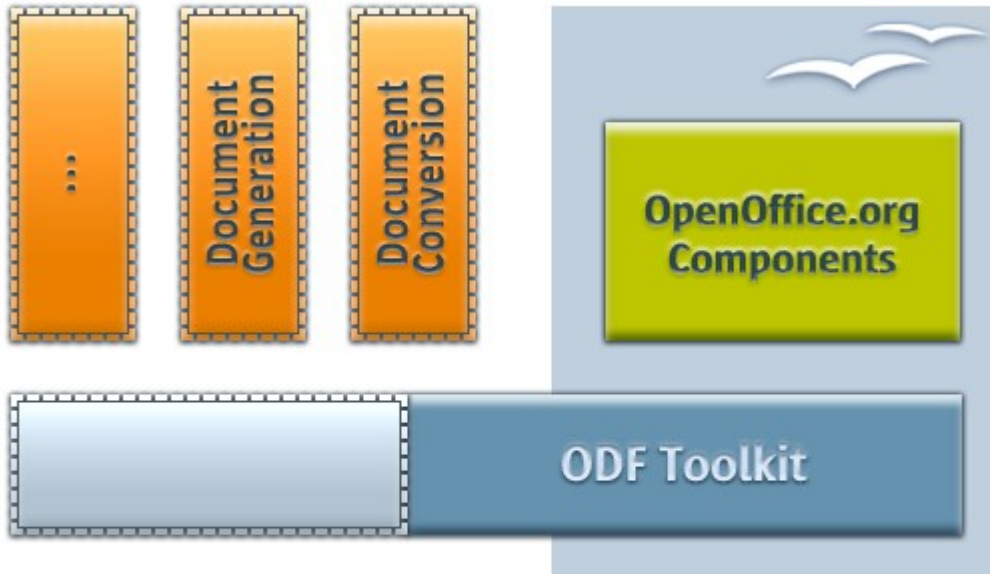
- More abstraction
- Hide XML, expose more ODF semantics
 - > Style inheritance, style management
 - > Page templates
 - > Links, references and footnotes
- Bridge ODF and language paradigms
 - > Use platform specific interface conventions
 - > Platform specific containers and collections

odftoolkit Project

New ODF Applications

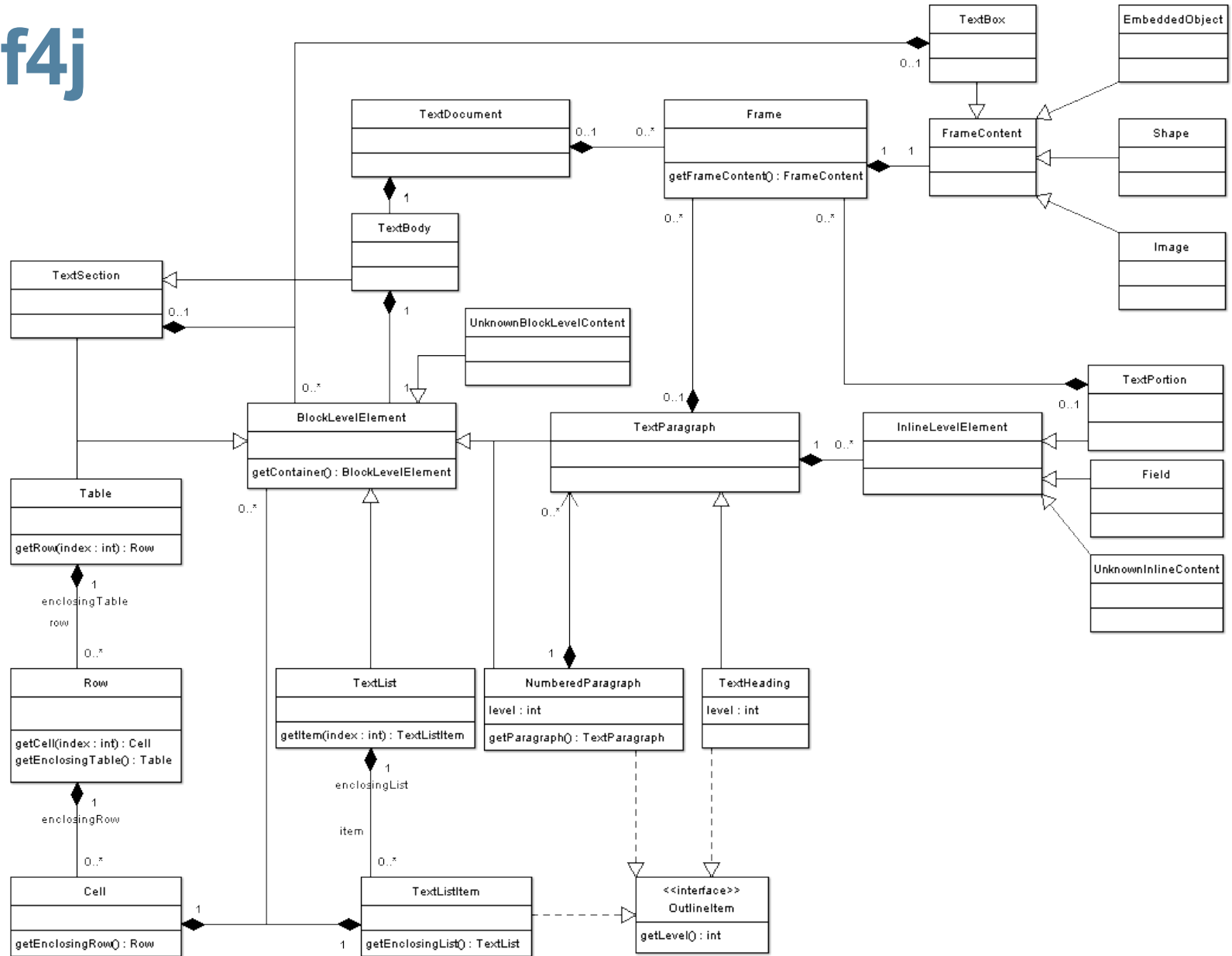


OpenOffice.org Suite



OpenDocument Format (ODF) Files

Odf4j



Odf4j (cont.)

- ODF support for the Java platform
- Supports ODF at various levels of abstraction:
 - > Package
 - > XML
 - > Object Model
- No automatic XML/OO mapping
 - > Rather implement semantics defined in the ODF specification that are not part of the schema
- Try to bridge ODF, XML and platform paradigms

AODL

- ODF for the .NET platform
- Design goals like odf4j
- Implemented in C#
- Offers object model for main ODF document types
- Includes experimental PDF and HTML generators

AODL demo application

Sample Bill Generator

Current Customer

Account ID: Street:

Name: City:

Last name: Phone:

Billing items

	Item	ItemNo	Quantity	Price	PriceTotal
▶	HDD SAMSU	a834723sfd	5	120,9	604,5
	CPU Tray AM	s837428323	3	52,5	157,5

Limitations

- Information that is derived by rendition is not normally available at the file format level
 - > Page/line numbers, list numbering
 - > Computed fields
- Derived information can be persisted by rendering application
- Processing chains should not rely on derived values if document could have been modified by non-rendering application

Future Work

- Harmonize toolkits
- Coherent programming experience for ODF on all platforms
- Create higher level tools on top of frameworks

Future Work (cont.)

- New metadata mechanism
 - > TC subcommittee finished proposal
 - > currently being reviewed
- Enables new ways to extend ODF
 - > based on semantic web technologies
 - > RDF, OWL
 - > integrates content and metadata

Links

- OpenDocument TC
 - > http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office
- OpenDocument Essentials
 - > <http://books.evc-cit.info/>
- odftoolkit project (odf4j, AODL)
 - > <http://odftoolkit.openoffice.org>
- ODF Perl module
 - > <http://search.cpan.org/dist/OpenOffice-OODoc/>



Processing ODF

Lars Oppermann

lars.oppermann@sun.com