Odt2DAISY: Authoring Full DAISY Books with OpenOffice.org

Vincent Spiewak, Christophe Strobbe, Jan Engelen
Katholieke Universiteit Leuven
Kasteelpark Arenberg 10 – bus 2442
3001 Heverlee
Belgium

Abstract
This article presents the architecture of Odt2daisy, an extension for OpenOffice.org. It enables users to export DAISY 3 XML and Full DAISY (XML+Audio) from OpenOffice.org Writer. Odt2DAISY works on Microsoft Windows, Mac OS X, Linux and OpenSolaris. It relies on the operating system's text-to-speech engine(s) to generate audio, so the supported languages depend on the user's system. Audio languages available at the time of writing are:

- Windows: English / default TTS voice
- Mac OS X: English with a high quality voice / default TTS voice
- Linux / OpenSolaris: the 27 languages supported by espeak, including English, French, German and Spanish.

1 State of Art

1.1 OpenOffice.org

OpenOffice.org (OOo) 3 is the leading free and open-source office suite for word processing, spreadsheets, presentations, graphics and databases. It is available in more than 80 languages and works on all common computer platforms.
It is not only used by individuals but also by public administrations, schools, universities and commercial organisations[1]. OpenOffice.org version 3.0 was downloaded more than 59 million times[2] and is included in many Linux distributions. It has native support for ISO/IEC 26300, the standard that defines the OpenDocument Format (ODF).

1.2 DAISY

DAISY is the standard for Digital Talking Books, developed and maintained by the DAISY Consortium and adopted by NISO as ANSI/NISO Z39.86. It is the world's most widely used format for talking books. The DAISY format has been adopted by a large number of digital libraries for the blind, not only for talking books, but also as a storage format. Users include TPB – Swedish Library for the Blind, the Library of Congress in the US, BrailleNet in France.

[1] OpenOffice.org maintains a list of major users at http://wiki.services.openoffice.org/wiki/Major_OpenOffice.org_Deployments.
[2] See http://marketing.openoffice.org/marketing_bouncer.html.

1.3 Odt2Daisy

OpenOffice.org with odt2daisy is a cross-platform, accessible, free and open source authoring environment for DAISY books. It can be used by anyone who needs to convert documents from any format supported by OpenOffice.org (including ODT, RTF, and Microsoft Word) to DAISY books.
For example, it can be used by teachers who need to provide accessible documents for disabled students and by resource centres that produce accessible documents for the print impaired in schools and universities.

2 Analysis

2.1 OpenDocument Format

The OpenDocument Format is an open-source format used principally by the office suite OpenOffice.org. The specification was originally developed by SUN Microsystems, but the standard is developed and maintained by the OASIS ODF Technical Committee. The format was published as ISO/IEC 26300:2006 Open Document Format for Office Applications (OpenDocument) v1.0.

2.1.1 OpenDocument Text (ODT)

The OpenDocument Text format is one of the formats defined by the OpenDocument specification. It consists of a ZIP file that contains a set of files and folders (XML, pictures and binaries).

- meta.xml : document metadata
- settings.xml : preferences and configuration of OpenOffice.org
- styles.xml : document style definitions
- content.xml : document content
- Pictures/ : document pictures
- Object N/ : MathML formulas can be include in Object N/content.xml.

2.1.2 “Flat XML ODT” Format

The “Flat XML ODT” format is basically used by “XML Based Filters”. This single XML file contains all the content of the ODT file, and is based on the concatenation of the XML files previously mentioned. Image and objects are included in a <office:binary-data> tag using base64 encoding.

2.2 Universal Network Object (UNO)

UNO is the base component technology for OpenOffice.org. It allows developers to utilize and write components that interact across languages, component technologies, computer platforms, and networks.
The UNO API allows programmers to develop:

- plugins (add-ons)
- components (add-ins)
- client softwares connecting to OpenOffice.org instance (locally or remote)

2.3 Filters

OpenOffice.org represents a document internally as a model. The UNO component that allows the transformation of the model into a file (and vice versa) is called a Filter[3].
If a developer wants to add a filter for a new file format (enabling import, export, or both) to OpenOffice.org, there are three approaches:

- linking against the application core
- use the document API
- "XML Based Filter"

This first approach is hard to maintain because it requires updates each time there is a change in the core data structure or the interface. Using the document UNO API is safer since API interfaces are more stable than the core interfaces, and provide an abstraction from the core applications. The "XML Based Filter" approach is probably the simplest because one only needs to provide an XSLT style sheet. OpenOffice.org 3.0 offers tools to package, install and remove them under "Tools" > "XML Filter Settings...". But this method is also the less flexible. For example, it doesn't support the addition of parameters to the filter[4]

3 Design

The design of Odt2daisy is guided by the following rules:

- the add-on must be cross-platform,
- the add-on must be simple to install, remove and update (only one oxt[5]),
- the add-on must be easy to use,
- the add-on must show some dialogues to allow parameter input (e.g. UID, Title, Author, …),
- the business logic must be reusable by a third-party application.

3.1 Components of Odt2daisy

The figure in Annex 1 illustrates how the different components of Odt2daisy fit together. Each of these components is described in the sections below.

3.1.1 Java OpenDocument Library (JODL)

JODL[6] allows the conversion of an ODT file into a "Flat XML ODT" file. It implements the following types of manipulation of or in the Document Object Model (DOM):

- merge : concatenation of all XML files into a single one,
- images : extraction and filename normalization of all pictures,
- page numbering : add a <pagenum> tag to facilitate page support,
- clean-up : cleaning the XML file (remove empty headings, etcetera).

Like many other Java libraries, JODL can be reused as a command-line tool.

---

[3] See Interface com.sun.star.document.XFilter.
[4] See http://www.mail-archive.com/dev@xml.openoffice.org/msg00832.html.
[5] OXT stands for OpenOffice.org Extension.
[6] JODL is not to be confused with the Java library jOpenDocument at http://www.jopendocument.org/.

3.1.2 Odt2daisy (Library)

The Odt2daisy Library enables the conversion of an ODT file into DAISY 3.0 XML format using JODL and an XSLT style sheet. A DAISY 3.0 DTD validation process is included in order to check if the DAISY XML is valid. It can be also reused as a command-line tool.

3.1.3 Odt2daisy (Extension)

The Odt2daisy Extension acts as a wrapper since it uses:

- the UNO API to save the current document into a temporary ODT file
- the Odt2daisy Library to convert it into DAISY 3.0 XML,
- the DAISY Pipeline Lite to convert it into Full DAISY 3.0.

3.1.4 DAISY Templates

The project also includes a set of localised templates using the "Non Code Extensions" architecture[7]. These templates include DAISY Custom Styles to write "rich" DAISY books.

3.1.5 DAISY Pipeline Lite

The Odt2daisy extension includes a version of the Pipeline Lite for which LAME and SOX (SOX for Mac OS X) were compiled statically. Static compilation of these components is necessary in order to have a standalone Pipeline Lite.
Pipeline Lite is extracted from the extension into the user's system directory[8] at the first export as full DAISY. An update mechanism was also developed in order to be able to remove or re-extract the Pipeline Lite in the future.

3.2 Unit Testing

JODL and the Odt2daisy library are tested using JUnit 4 and XMLUnit[9]. A set of 148 ODT and DAISY XML files was created and reviewed manually. The files attempt to cover the whole ODT specification. The goal of this Unit Testing process is to avoid non-regression bugs. It is also a good way of getting an overview of covered cases

4 Results

4.1 DAISY 3.0 XML

Odt2daisy enables authoring of "rich" DAISY XML files with multilangual content. All Western languages in OpenOffice.org are supported. The user interface and the templates are also localised.[10]

---

[7] See http://wiki.services.openoffice.org/wiki/Non-code_extensions.
[8] Or a temporary directory if the user's home folder does not have enough disk space.
[9] See http://xml.openoffice.org/filter/.
[10] See http://odt2daisy.sf.net/l10n.

4.2 Full DAISY

Odt2daisy enables export of Full DAISY (XML+Audio). Supported languages for audio generation depend on the text-to-speech engines installed on the user's system.

4.2.1 Windows & Mac OS X

Odt2daisy enables export of Full DAISY English documents out of the box. Users can also use any other text-to-speech engine (free or commercial), but they first need to set it as default TTS in the operating system settings. Multilingual documents are not yet supported.

4.2.2 Linux & OpenSolaris

Odt2daisy enables export of Full DAISY documents in any of the 27 languages supported by the open-source TTS engine espeak, including English, French, German and Spanish. Multilingual documents are also supported.

5 Issues

Odt2daisy does not successfully convert all ODT files. The remaining issues can be classified in four categories: DAISY structure issues, ODT structure issues, unsupported features, and bugs.

5.1 DAISY Structure Issues

DAISY books created with odt2daisy must be well structured according to the DAISY format. For example, a document with a “Heading 1” followed by a “Heading 3” will result in a DTD validation error. Another example is the use of GIF files: DAISY accepts only JPEG, PNG and SVG.

5.2 ODF Structure Issues

ODT files that serve as a basis for authoring DAISY book must be well structured according to OpenOffice.org. For example, if the user adds a numbered heading using the OOo Numbering List style, the resulting structure in the ODT will be a list with an item containing the heading.

5.3 Bugs and Unsupported Features

This type of issues should be reported to the odt2daisy author at vspiewak@sourceforge.net.

6 Future Work

6.1 Improvements

Odt2daisy tries to cover the whole ODT specification. However, odt2daisy does not yet cover every feature, for example Bibliography. This type of unsupported features will be implemented in the future.

6.2 Accessibility Checker

DAISY and ODT structure issues suggest the need of an accessibility checker: a wizard which provides help for authoring accessible documents. For example, the accessibility checker could ensure that the user specifies a text alternative for every picture in the document. This accessibility checker would not only benefit authors of DAISY files but also authors who wish to export other formats from OpenOffice.org, for example, HTML.

6.3 Acronym and Abbreviation Support

Acronyms and abbreviations are not covered by OpenOffice.org and the ODF specifications. This makes it impossible to mark up acronyms and abbreviations appropriately and link them with an appropriate definition. Since OOo cannot handle acronyms and abbreviations, Odt2daisy cannot handle them either, so audio generated by the add-on cannot properly render them. The OpenOffice.org HTML exporter cannot properly render them either.

7 Acknowledgements

The work of Vincent Spiewak, author of odt2daisy and odt2dtbook, was supervised by Jan Engelen and Christophe Strobbe. Work on odt2dtbook was initially supervised by Dominique Archambault.
Work on odt2daisy was undertaken in the framework of the integrated project AEGIS, funded by the IST Program of the European Commission — project reference 224348.
Vincent Spiewak would like to express his gratitude to those who gave him the opportunity to complete this work, in particular:

- Peter Korn (Sun Microsystems; AEGIS project technical manager),
- Gilles Blain (Head of Master Sub-Discipline “Technologies Applicatives”, Université Pierre et Marie Curie (UPMC), France).

8 Annex A: Component Diagram

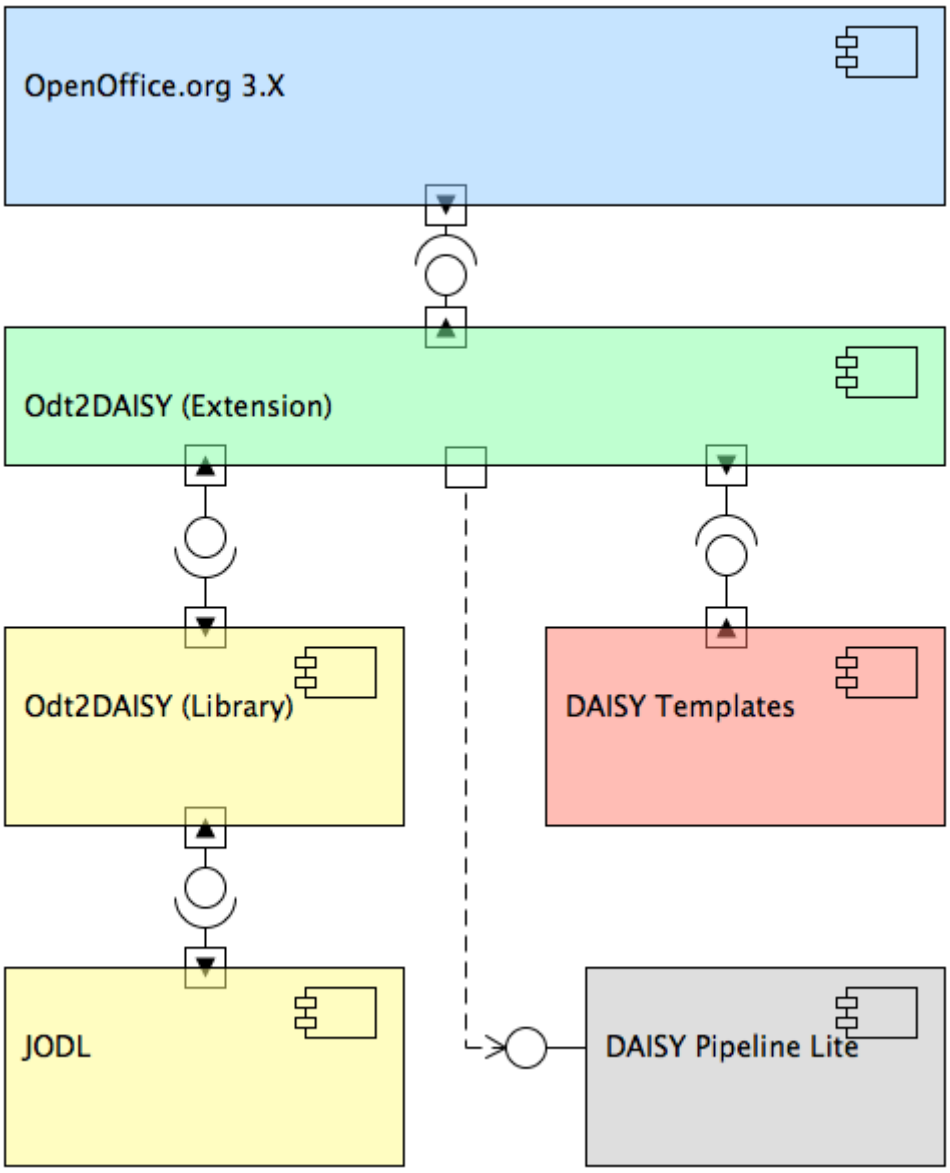

Illustration 1: Component diagram of Odt2daisy