BERLIN 2012 CONFERENCE

17th-19th October

Searching for Metadata via Commandline

- Dipl.-Inf. Frank Hofmann
  Berlin

About me. Open Source Involvements and Projects

Chemnitzer Linux-Tage

2000-2007

since 2006

BrLUG

Regional LUG Meeting Berlin-Brandenburg

since 2009

PARTNER VON:
büro2.0
Open Source Bürogemeinschaft

since 2009

About me. My work

http://www.efho.de

- distribution of indoor and outdoor wireless devices
- pre-press preparation and print coordination

WIZARDS OF FOSS
Open Source Schulungen

http://www.wizards-of-foss.de

- open source training for experts
  co-founder and trainer

http://www.buero20.org

- Berlin open source office community, shared space, 25 companies, 1300m², 60 members

Table of contents

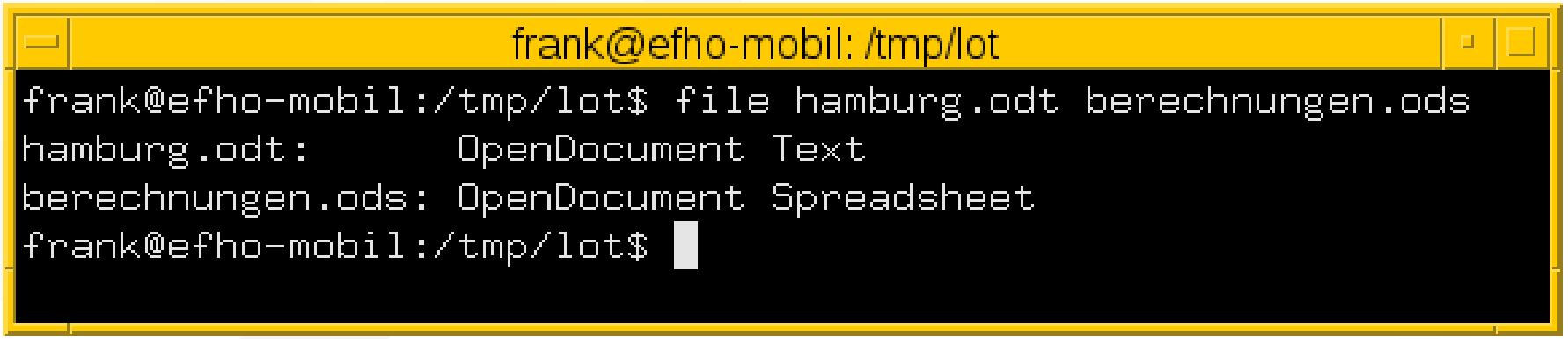
Structure of the OASIS data format

- Compressed file (zip) with a fixed list of files
  xml files with determined elements

META-INF/manifest.xml	List and type of files
Thumbnails/thumbnail.png	Document preview image
Pictures/image.png	Pictures contained in the document
mimetype	Document mimetype information
content.xml	Document content
meta.xml	Document metadata
settings.xml	Document settings
styles.xml	Document style stettings

Useful UNIX/Linux commands

- Display file type
  `file document.odt`

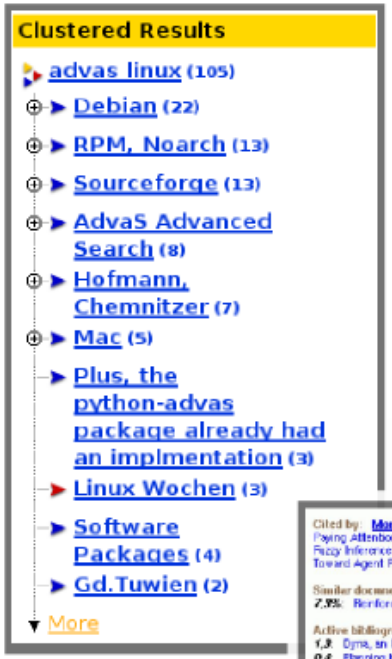


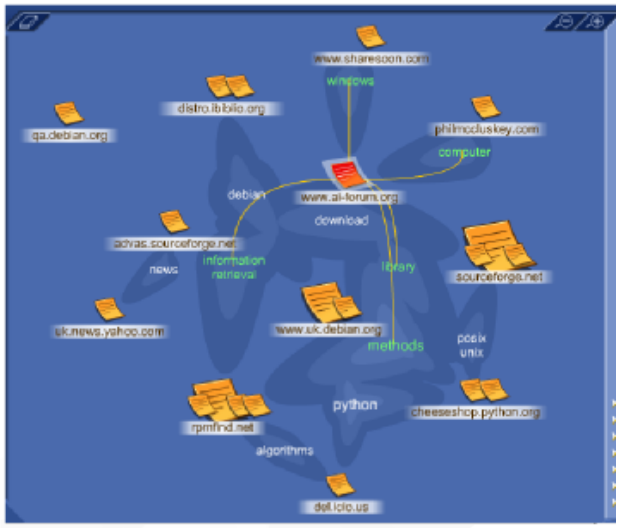
- Show zip file content
  `unzip -l document.odt`

- Extract OASIS file content
  `unzip document.odt`

How does a search engine work?

Document preprocessing > Retrieval process > Display the result

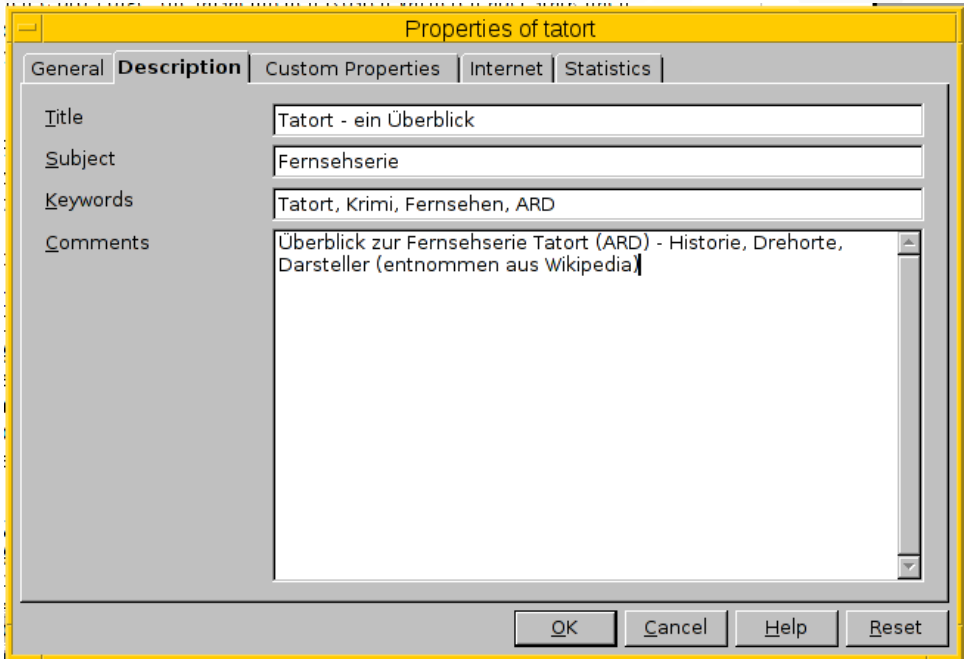




Search engine requirements

- open (or at least well-documented) document format
  search engine has to figure out how to read the document
- complete document metadata
  - mostly empty - nobody does that
  - cannot be set automatically
- structured text using format templates
  - mostly ignored - nobody does that
  - requires strict policies for an organization or company
- document content
  - include text as characters, not as images
  - most images cannot be interpreted by retrieval programs

How can you improve the document quality for a better search result?



- Add metadata to your documents
  see: File → Properties
- Use format templates
  see: File → Templates

Searching on a UNIX/Linux system

- listing and finding documents
  basic UNIX commands: `ls`, `find`
- filtering text files
  legendary UNIX command: `grep`
- filtering xml documents
  not-so-well-known UNIX command: `xml_grep`
- taken from Debian package: xml-twig-tools
  contains: `xml_spellcheck`, `xml_pp`, `xml_grep`, and `xml_split`

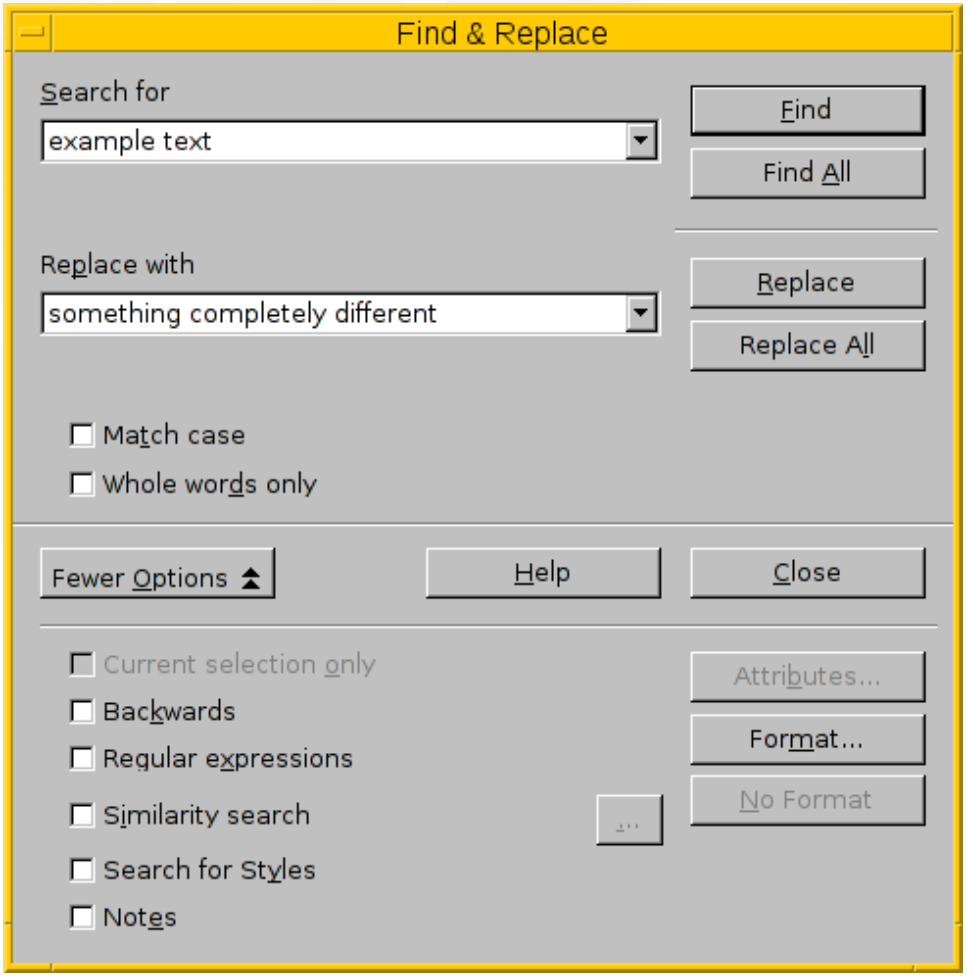
Searching on a UNIX/Linux system - examples

- `xml_grep "dc:description" meta.xml`
  returns a valid xml document with a single node, only

```
$ xml_grep "dc:description" meta.xml
 <?xml version="1.0" ?>
 <xml_grep version="0.7" date="Tue Aug 28 11:29:27 2012">
 <file filename="meta.xml">
   <dc:description>irgendwas</dc:description>
 </file>
 </xml_grep>
$
```

- `xml_grep --text_only "dc:description" meta.xml`
  returns the node value, only

Searching within a single document



- Document content via graphical user interface
  - Keyboard shortcut: CTRL+F
  - Menu item: Edit → Find and Replace
- requires opening the document before searching
- search does not include metadata

Searching within a document archive (#1)

- process automation -- metadata extraction:
  integrate `xml_grep` in a shell script

  example extraction for the document title:

```
unzip -p document.odt meta.xml | xml_grep --text_only
"//office:document-meta/office:meta/dc:title"
```

Searching within a document archive (#2)

- process automation – full-text search:
  - combine `unzip`, `sed` and `grep` in a shell script
  - combine `deepgrep` and `wc` in a shell script

- version #1: includes the node names
  results in false positives
- returns a match if search term is in document content

```
find $1 -name "*.odt" | while read filename
do
    unzip -ca "$filename" content.xml | grep -qli "$2"
    if [ $? -eq 0 ]; then
       echo "search term found in " $filename
    fi
done
```

Searching within a document archive (#3)

- Improved version #1: excludes the node names removes the false positives
- returns a match if search term is in document content

```
find $1 -name "*.odt" | while read filename
do
  unzip -ca "$filename" content.xml | sed 's/<[^>]*>/ /g' | grep -qli "$2"
  if [ $? -eq 0 ]; then
    echo "search term found in " $filename
  fi
done
```

Searching within a document archive (#4)

- simplified version #2 using `deepgrep`

  taken from Debian package: strigi-utils

  - backend of the desktop search engine Strigi
  - totally undocumented, but works perfectly
  - `grep` for archives (`tar.gz`, `zip`, `deb`, `rpm`), `mp3`, `pdf`, `msword`

- example:

  ```
  deepgrep "Berlin" document.odt
  ```
  returns the matches

Searching within a document archive (#5)

- `deepgrep` in a shell script

```
find $1 -name "*.odt" | while read filename
do
  match=`deepgrep "$2" "$filename" | wc -l`
  if [ $match -ne 0 ]; then
    echo "search term found in " $filename
  fi
done
```

- `deepgrep` is 8 to 10 times faster than `xml_grep`

Links and references

- OpenDocument-Format (OASIS): http://de.wikipedia.org/wiki/OpenDocument
- Axel Beckert, Frank Hofmann: Suche in komprimierten Dateien und Archiven, LinuxUser 04/2012
- Axel Beckert, Frank Hofmann: Suche in Datenformaten (Teil 1), LinuxUser 06/2012
- Axel Beckert, Frank Hofmann: Suche in Datenformaten (Teil 2), LinuxUser 07/2012
- Frank Hofmann: Automatisiert in Open/LibreOffice-Dokumenten suchen, LinuxUser 11/2012

Thank you!

BERLIN 2012 CONFERENCE

17th-19th October

Lassen Sie es setzen :-)

LLS

Linux, Layout & Satz

Dipl.-Inf. Frank Hofmann
Hofmann EDV – Linux, Layout und Satz
c/o Büro 2.0
Weigandufer 45 - 12059 Berlin

Mail: `frank.hofmann@efho.de`
Web: `http://www.efho.de`