

## Searching for Metadata via Commandline

- Dipl.-Inf. Frank Hofmann  
Berlin

# About me. Open Source Involvements and Projects



2000-2007



since 2006



Regional LUG  
Meeting Berlin-  
Brandenburg



since 2009



since 2009

# About me. My work



Linux, Layout & Satz

<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

PARTNER VON:  
**büro2.0**  
Open Source Bürogemeinschaft

<http://www.buero20.org>

- ▼ Berlin open source office community, shared space, 25 companies, 1300m<sup>2</sup>, 60 members

# Table of contents

- ▼ How does the OASIS document format look like?
- ▼ How does a search engine work?
  - ▼ Document processing
  - ▼ Search engine requirements
- ▼ How can you improve the document quality for a better search result?
- ▼ Searching and Retrieval
  - ▼ Searching on a UNIX/Linux system
  - ▼ Searching in a document archive with OASIS documents
- ▼ Links and references

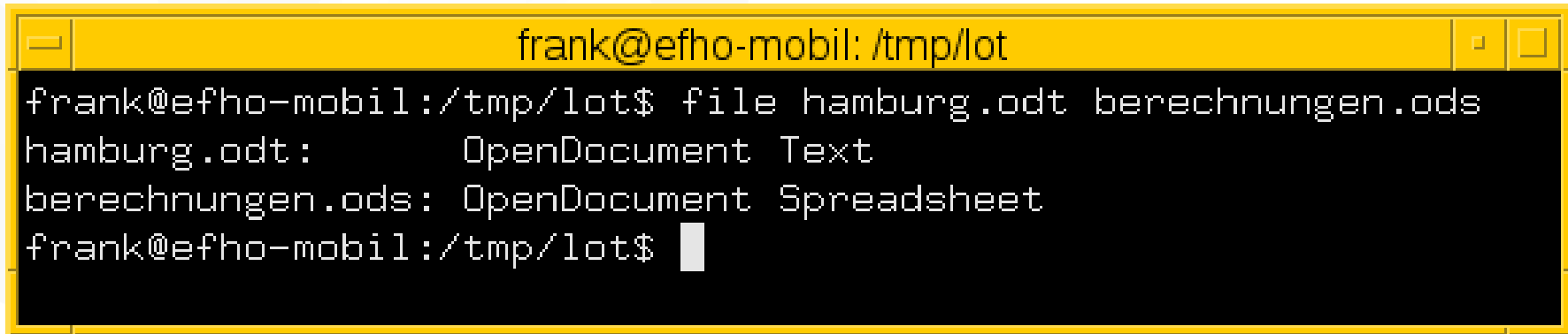
# Structure of the OASIS data format

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

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

# Useful UNIX/Linux commands

- ▼ Display file type  
`file document.odt`

A terminal window with a yellow title bar containing the text 'frank@efho-mobil: /tmp/lot'. The terminal content shows the command 'file hamburg.odt berechnungen.ods' and its output: 'hamburg.odt: OpenDocument Text' and 'berechnungen.ods: OpenDocument Spreadsheet'. The prompt 'frank@efho-mobil: /tmp/lot\$' is visible at the end of the output.

```
frank@efho-mobil: /tmp/lot$ file hamburg.odt berechnungen.ods
hamburg.odt:      OpenDocument Text
berechnungen.ods: OpenDocument Spreadsheet
frank@efho-mobil: /tmp/lot$
```

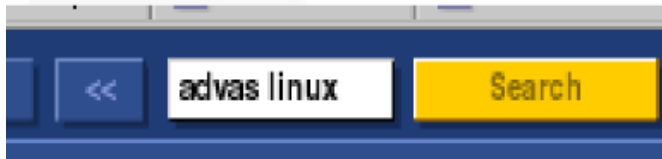
- ▼ Show zip file content  
`unzip -l document.odt`
- ▼ Extract OASIS file content  
`unzip document.odt`

# How does a search engine work?

Document preprocessing

Retrieval process

Display the result



**Clustered Results**

- [advas linux](#) (105)
  - [Debian](#) (22)
  - [RPM, Noarch](#) (13)
  - [Sourceforge](#) (13)
  - [AdvaS Advanced Search](#) (8)
  - [Hofmann, Chemnitzer](#) (7)
  - [Mac](#) (5)
  - [Plus, the python-advas package already had an implementation](#) (3)
  - [Linux Wochen](#) (3)
  - [Software Packages](#) (4)
  - [Gd.Tuwien](#) (2)
- [More](#)



Cited by: [Moc](#)  
Paying Attention to What's Important: Using Focus of Attention... - Foner, Mises (1994) (Context)  
Picky Inference System Learning by Reinforcement Methods - Jouffe (1997) (Context)  
Toward Agent Programs with Circuit Semantics - Nilsson (1982) (Context)

Similar documents (at the sentence level):  
2.9% Reinforcement Learning Architecture - Sutton (Context)

Active bibliography (related documents): [Moc](#) [All](#)  
1.3 Dyna, an Integrated Architecture for Learning, Planning, and... - Sutton (1991) (Context)  
0.4 Planning by Incremental Dynamic Programming - Sutton (1991) (Context)  
0.3 Reinforcement Learning And Its Application To Control - Gillholm (1992) (Context)

Similar documents base d on text: [Moc](#) [All](#)  
0.4 Integrated Architecture for Learning, Planning, and... - Dichow, Mularzka (Context)  
0.3 An Integrated Learning, Planning and Pecking Algorithm Applied... - Aho Wäiser (Context)  
0.3 A Multagent Framework for Planning, Reasoning, and Learning - Wikie (1999) (Context)

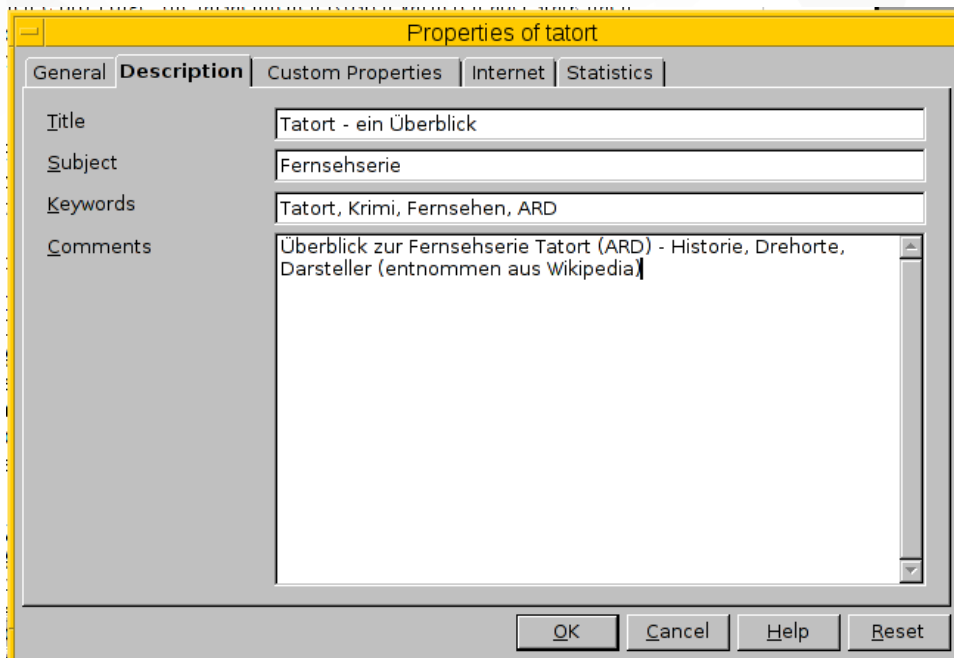
Related documents from re - dRillies: [Moc](#) [All](#)  
4 Learning to predict by the method of temporal differences - Sutton - 1988  
4 Near-optimal adaptive elements that can solve difficult learning control problems (context) - Bertz, Sutton et al. - 1993  
4 Temporal credit assignment in reinforcement learning (context) - Sutton - 1984

# 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`

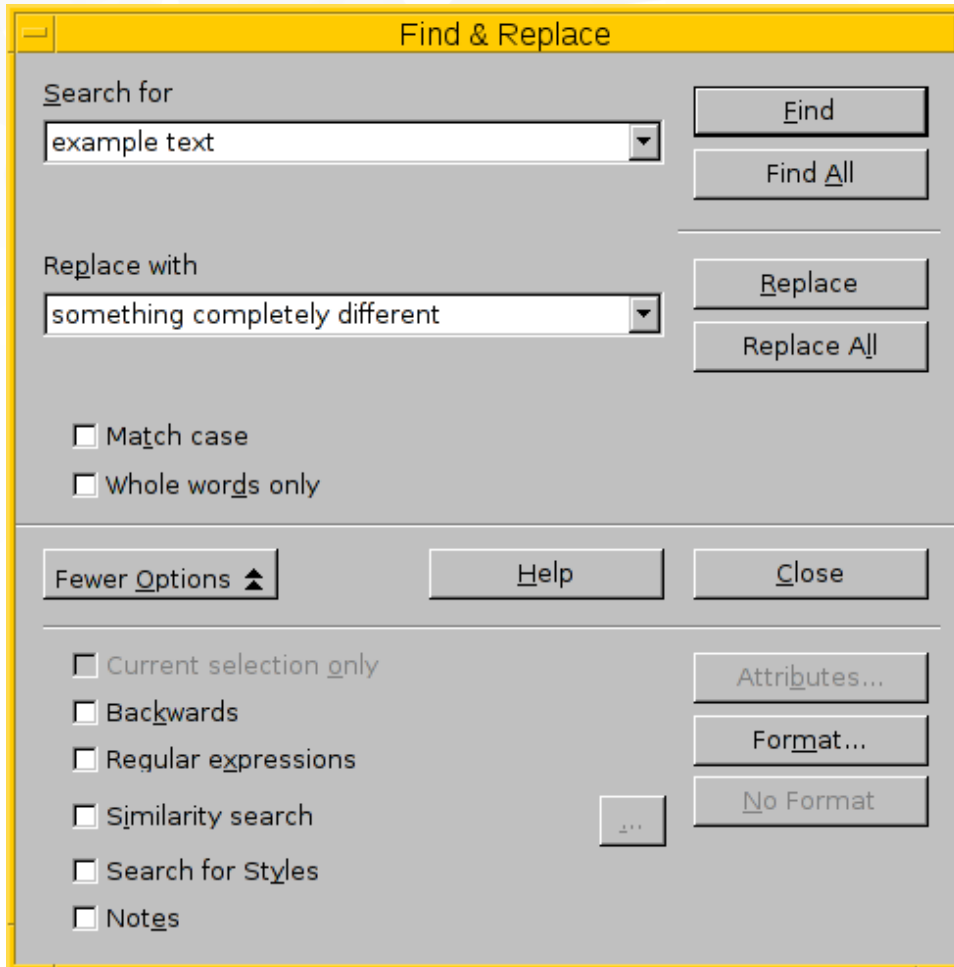
# Searching on a UNIX/Linux system - examples

- ▼ `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 :-)



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](mailto:frank.hofmann@efho.de)  
Web: <http://www.efho.de>



All text and image content in this document is licensed under the [Creative Commons Attribution-Share Alike 3.0 License](https://creativecommons.org/licenses/by-sa/3.0/) (unless otherwise specified). "LibreOffice" and "The Document Foundation" are registered trademarks. Their respective logos and icons are subject to international copyright laws. The use of these therefore is subject to the [trademark policy](#).