MAX PLANCK
digital library

MetaLib als WERKZEUG für die Metasuche

MetaLib als geeignetes TOOL für die eigene Kommunikation mit Such-Schnittstellen

- vs. -

MetaLib (CKB) als gehosteter SERVICE, mit zentraler Pflege und Verteilung der Funktionalität

Erik Altmann

Max Planck Digital Library

Konfigurations-Optionen

- Z39.50
  - Term Transformation, Format-Conversion, (Feld-)Conversion
- WEBCONFIG_XML/EXTERNAL (XML)
  - Erweiterte TT-Maske, URL-Maske, Parsen der Trefferzahl
  - external_conf (xml-Verarbeitung – eingeschränkt?)
- EXTERNAL
  - FIND – 1 Skript für Suche/Trefferzahl
  - PRESENT – 1 Skript für Ergebnisliste
  - PRESENT_SINGLE – 1 Skript für Vollanzeige

WEBCONFIG (XML), erweiterte Query-Transformation

Query Transformation

In Variable	Regular Expression
CODE1	s/WYR/y1H/
CODE2	s/WYR/y2H/
CODE*	s/WRD/tH/
CODE*	s/WTI/tH/
CODE*	s/WSU/kH/
CODE*	s/WAU/aH/
TERM*	s/\?//g
TERM*	s/\"//g

URL Mask Creation

http://wals.info/refdb/search?journal=wals&tMH=substring&format=rss&_action_export=Export&l=10

Condition	URL Mask
CODE1=CODE2	&CODE1=TERM1 TERM2
CODE1!=CODE2	&CODE1=TERM1&CODE2=TERM2

SRU - CQL?

URL Patterns

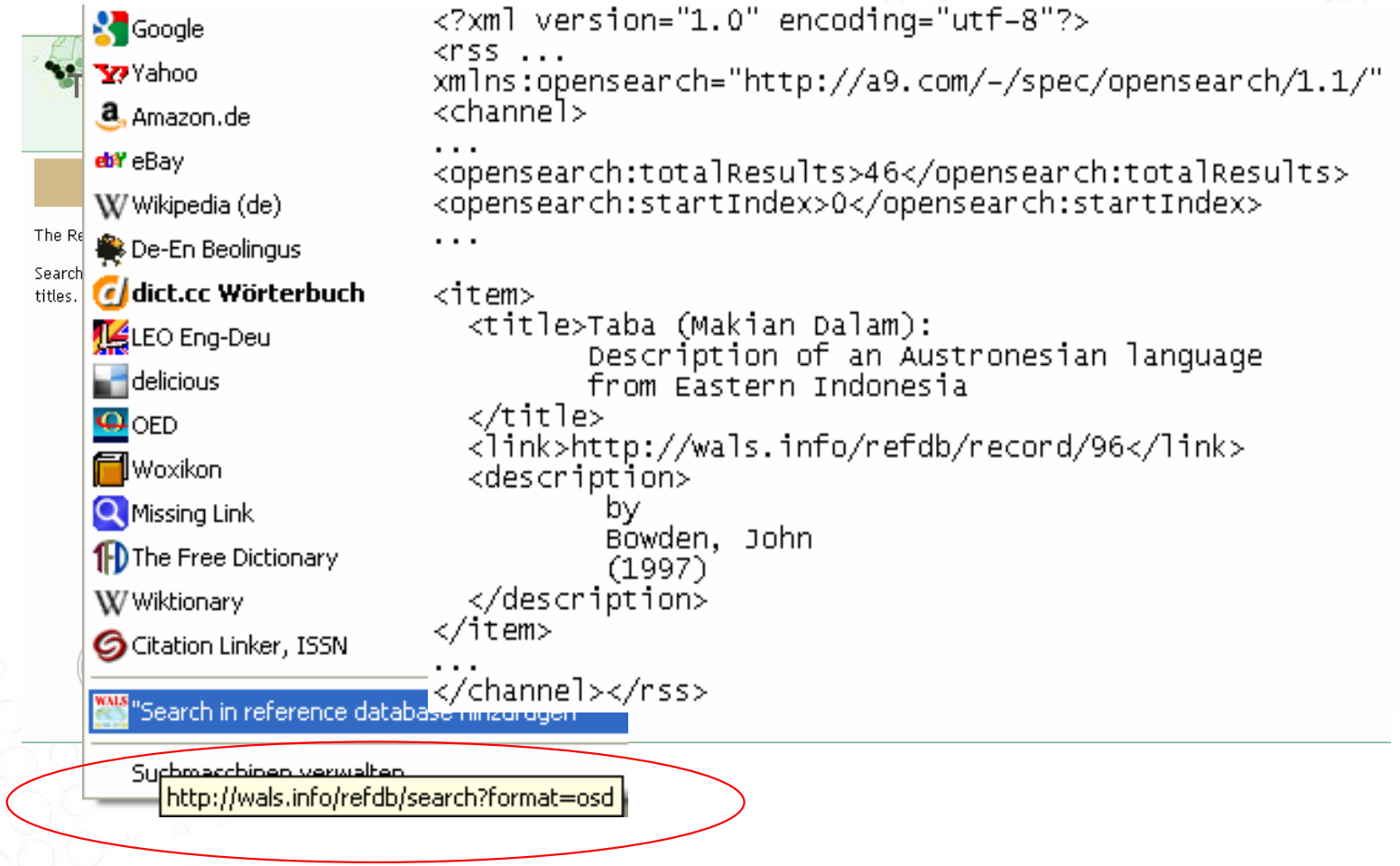
<opensearch:totalResults>(\d+)</opensearch:totalResult:
<opensearch:totalResults>0</opensearch:totalResults>

XML - XPATH?

Neue EXTERNAL (XML) - <CONFIG>.conf

```
<externalConf>
  <find>
    <mlSearchRequest>
      <constructQuery>
        <expr pid="termTransformation" num="01">
          <condition></condition>
          <varNameIn>CODE1</varNameIn>
          <regExp>s/WRD/any/</regExp>
          <varNameOut></varNameOut>
          <comment></comment>
        </expr>
        <expr pid="maskLayout" num="11">
          <condition></condition>
          <urlMask>TERM1=CODE1</urlMask>
          <comment></comment>
        </expr>
      </constructQuery>
    </mlSearchRequest>
    <hitPattern>
      <exact>&lt;no_of_hits&gt;(\d+)&lt;/no_of_hits&gt;</exact>
      <zero></zero>
    </hitPattern>
  </find>
</externalConf>
```

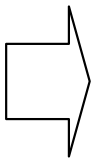
OpenSearch, RSS 2.0




```
<?xml version="1.0" encoding="utf-8"?>
<rss ...
xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/" ...>
<channel>
...
<opensearch:totalResults>46</opensearch:totalResults>
<opensearch:startIndex>0</opensearch:startIndex>
...

<item>
  <title>Taba (Makian Dalam):
         Description of an Austronesian language
         from Eastern Indonesia
  </title>
  <link>http://wals.info/refdb/record/96</link>
  <description>
          by
          Bowden, John
          (1997)
  </description>
</item>
...
</channel></rss>
```

WEBCONFIG, external_conf

```
<externalConf>
<present>
 <navigate>
            <requestMethod>GET</requestMethod>
            <requestParaml>offset=#ML_SET_ENTRY#</requestParaml>
 </navigate>
 <xmlParser>
   <basicTags>
     <item>
       <title>$245||a</title>
       <link>$856||u</link>
       <description>$500||a</description>
     </item>
   </basicTags>
 </xmlParser>
</present>
</externalConf>
```



Your search query: austronesian
In current set: 46 - WALS Online

Table View Brief View

1 of 46

Database:	WALS Online
Title:	Taba (Makian Dalam): Description
Author:	Bowden, John
Year:	1997
Note:	by Bowden, John (1997)
External:	http://wals.info/refdb/record/96

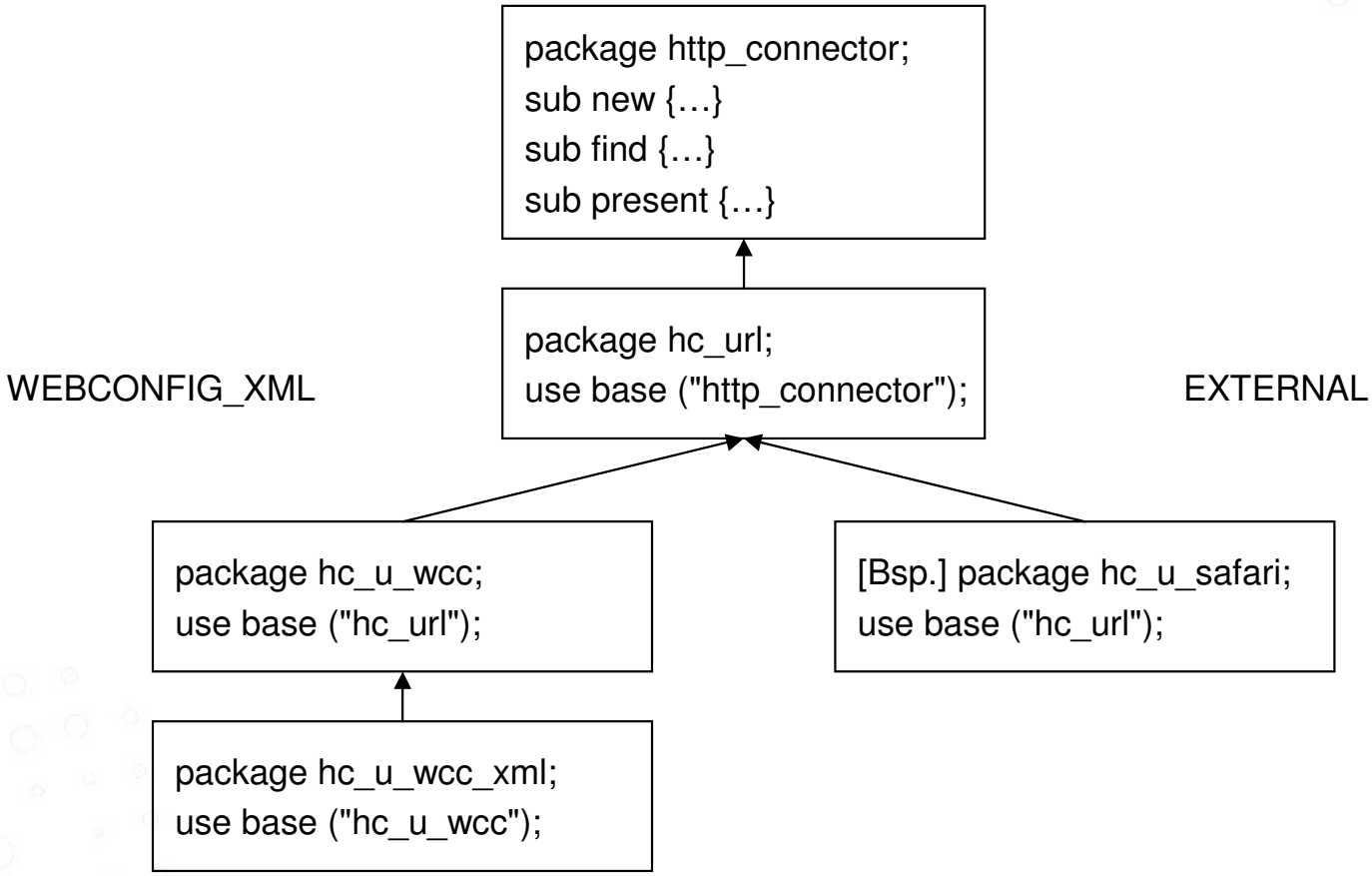
Komplexeres xml

```
<externalConf>
   <present>
      ...
      <xmlParser>
            <basicTags xmlns:refdb="wals">
               <refdb:record>
                  <refdb:bibData refdb:field="##BibCond|">##BibValue|</refdb:bibData>
                     ...
               </refdb:record>
            </basicTags>
            <virtualTags>
               ...
            <vTag>
              <tag>$245||a</tag>
              <condition>##BibCond|=title</condition>
              <layout>##BibValue|</layout>
            </vTag>
               ...
            </virtualTags>
      </xmlParser>
   </present>
</externalConf>
```

Probleme mit external_conf, xmlParser

```
<xmlParser>
  <basicTags>|
    <record>
      <recordData>
        <search-result:search-result-record>
          <escidocItem:item>
            <escidocMetadataRecords:md-records>
              <escidocMetadataRecords:md-record>
                <escidocMetadataProfile:publication>
                  <dc:title>$245||a</dc:title>
                </escidocMetadataProfile:publication>
              </escidocMetadataRecords:md-record>
            </escidocMetadataRecords:md-records>
          </escidocItem:item>
        </search-result:search-result-record>
      </recordData>
    </record>
  </basicTags>
</xmlParser>
```

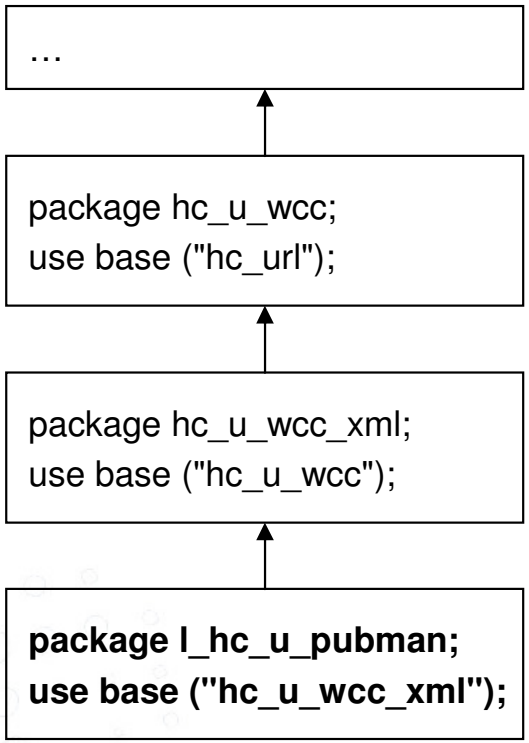
MetaLib Perl-Klassen in $aleph_ext - ?



perldoc http_connector.pm

```
http_connector
     The New External Object programs model.  http_connector is
     the Root for this object.  The basic object has a seperate
     find method and present method each comprised of the
     folowing functions:
         1.createFindRequest
         2.execFindQuery
         3.parseFindResults
         4.printFindResults See seperate documentation for full
     details

     ...

     find

     Each Inherited object will only need to run this method to
     execute find. It consists of 4 virtual methods that
     inheriting objects must implement

     present

     ...

     presentSingle

     ...
```

Experiment: spezielle WEBCONFIG_XML ($aleph_ext/hc_u_wcc_xml.pm)



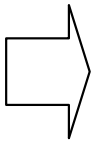

```
package l_hc_u_pubman;

use strict;
# inherit from WEBCONFIG_XML
use base ("hc_u_wcc_xml");

# use local xslt processing package
use lib $ENV{'aleph_dev'} . "/lib/MPG";
use XSLTPROC;


...
```

Methode printPresentResults überschreiben

```
perldoc hc_u_wcc_xml.pm
     parsePresentResults
     does nothing The XML parser will be called in Print
...
# implement custom xml parsing method
sub printPresentResults  {

    my ($self) = @_;

    my $xslt = new XSLTPROC;
    my $xslfile = $ENV{'aleph_ext'} . "/l_pubman.xsl";
    my $xml = '';

    # http response string
    my $response = $self->{response};

    # get xml
    if($response =~ /(<\s*\?\s*xml.*>)/s) {
       $xml = $1;

       # transform xml
       my ($flag, $records) = $xslt->transform_xml($xml, $xslfile);
```

MetaLib-Datenstrom erzeugen

```
if($flag) {

## format of $record string ##
#
# RECORD-FORMAT="PLAIN"
# RECORD
# 245 $$a<TITLE>
# END-RECORD
# END-OF-DATA
#
#####################################

print $records;
```



```
<?xml version="1.0"?>
<xsl:stylesheet ...>

<xsl:output method="text" />

<xsl:template match="/">RECORD-FORMAT="PLAIN"
<xsl:apply-templates select="..." />
END-OF-DATA
</xsl:template>

<xsl:template match="...">RECORD
<xsl:apply-templates select="..." />
END-RECORD
</xsl:template>

<xsl:template match="...">
...
<!-- Title -->
<xsl:for-each select="dc:title">
<xsl:if test=". != ''">
245 $$a<xsl:value-of select="normalize-space(.)" />
</xsl:if>
</xsl:for-each>
...
<!-- Language -->
<xsl:for-each select="dc:language">
<xsl:if test=". != ''">
546 $$a<xsl:value-of select="normalize-space(.)" />
</xsl:if>
</xsl:for-each>
...
</xsl:template>
</xsl:stylesheet>
```

Der „offizielle“ (?) Weg

- Resource Management Guide Version 4.3 – EXTERNAL_XML --- This access method is used in CKB configurations only. Resources using this configuration are Search & View. (Previously known as WEBCONFIG_XML.)
- http://www.exlibrisgroup.org/display/MetaLibOI/Writing+External+Programs – FIND, PRESENT, PRESENT_SINGLE
- Vorteil: standalone, funktioniert unabhängig
- Nachteil: Umfangreich, alle „Tasks“ (STDIN lesen, http-Anfrage, Daten parsen u. ausgeben) müssen selbst ausgeführt werden

Alternativen, Hilfen

- Templates
  - $aleph_ext/sru_find, sru_present
    - use ML_REQ_PARSER2;
    - require "call_httpd_timeout_new";
  - $aleph_dev/dat01/external_conf/sru_dc_template.conf
- Nelli, External Search Programs for MetaLib – SRU: http://wiki.helsinki.fi/display/Nelli/MetaLib+External+Search+Programs#MetaLibExternalSearchPrograms-SRU –> srwu-<Version>.tar – aktuelle Verteilung unter MetaLib?

Schlussfolgerungen und Fragen

- Es gibt viele gute Ansätze und Lösungen, aber keinen zufriedenstellenden und „gültigen“ Standard?
- xml-Ressourcen eher im Kontext von (nicht-standardisierten) http-Targets
- Ist MetaLib/MetaSearch Next Gen noch als Werkzeug für die Einbindung mit eigenen Targets vorgesehen (oder Anfrage an CKB bzw. „Me Too“)?
- Wie groß ist der Bedarf auf Anwenderseite?
- Wie ist die Weiterentwicklung/die Dokumentation der vorhandenen Komponenten geplant?
- In welchem Stadium befindet sich die Entwicklung?
- Welchen Stellenwert hat die Weiterentwicklung der Metasuch-Komponente?

Quellen

- http://www.exlibrisgroup.com/docportal
- http://www.exlibrisgroup.org/display/MetaLibOI/Home
- http://wiki.helsinki.fi/display/Nelli/MetaLib+External+Search+Programs
- WEBCONFIG_XML - Mr. Tal Ayalon, "Advanced MetaLib KB Tools Training", Stockholm, Sweden, September 2006

Vielen Dank - Fragen? – altmann@mpdl.mpg.de