
Project presentation:

Michael Ferber
Christoph Hanser

Developing a citation web service with an included logging facility

Knowledge and the Web
- Prof. Dr. Bettina Berendt -

Berlin, 27th of June 2005

Outline: Developing a Citation Web Service

1. Assess CiteSeer
2. Define the Usage Mining Goals
(and then the content of the Logfiles)
3. Specify the Web Service Features
 - a) Development environment
 - b) Interfaces to Clients and TTT/CiteSeer
 - c) Conversion to DIML
4. Perform the Usage Mining

Do you remember?

Outline: Developing a Citation Web Service

- ▶ Choice of an appropriate tool/algorithm
 - ▶ Specify the Web Service
 - ▶ Implement the Web Service
-

Outline: Developing a Citation Web Service

- ▶ Choice of an appropriate tool/algorithm
 - ▶ Specify the Web Service
 - ▶ Implement the Web Service
-

Assess CiteSeer

- It is implemented in Perl
- Consists of a lot of modules like cite.pm, bib.pm, etc.
- The Perl script to get the BibTeX format looks like this:

```
#!/usr/bin/perl -w
# Author: Christoph, christoph.hanser@student.hu-berlin.de
# This invocation of bib creates a BibTeX output of a given citation
use lib './lib';
use cite;
use bib;

my ($citetext) = "Abelson, D., (1990). Preferential cooperative binding
of topoisomerase II to scaffold associated regions. EMBO J. 8 3997-
4006.";

# erstellt den BibTeX Code
print bib::CreateBibTeXFromText ($citetext);
```

And returns the following BibTeX output

```
@misc{ abelson90preferential,
author = "D. Abelson",
title = "Preferential cooperative binding of topoisomerase II to scaffold associated
regions",
text = "Abelson, D., (1990). Preferential cooperative binding of topoisomerase
II to scaffold associated regions. EMBO J. 8 3997-4006.",
year = "1990" }
```

Assess CiteSeer

- ▶ *CiteSeer's citation extraction algorithm* works very successful:
 - ▶ In a test 31 of 35 citations the items were recognized correct.
 - ▶ Drawbacks:
 - ▶ Only authors, title, and publication year are extracted from a given citation
 - ▶ The first name is always abbreviated, a second first name is deleted (e.g. Oliver A. McBryan -> O. McBryan)
 - ▶ If the author is an organization, this is not recognized (e.g. Asian Development Bank -> A. Bank)
- ▶ i.e. It can only be used together with another citation algorithm (or we wait for the next version of CiteSeer's algorithm)

Outline: Developing a Citation Web Service

- ▶ Choice of an appropriate tool/algorithm
- ▶ Specify the Web Service
- ▶ Implement the Web Service

Specify the Web Service

- ▶ **Development environment**
 - ▶ The algorithms TTT, CiteSeer and Paratools are available in Perl and, thus, can be used in Linux easily.
 - ▶ The Web Service could be implemented in PHP or Java/Tomcat among others.
 - ▶ We have decided to use Java because it usually provides more functionality and maintainability.

Specify the Web Service

► Interfaces to Clients and TTT/CiteSeer/Paratool

► Clients (especially the Word macro) will use the following method after invoking the Web Service:

```
+ Citation2DIML (String[] Citation, String ID) : String  
  + CorrectedCitation(String Citation, String ID) : void
```

► The “interface” to the citation extraction algorithms will be an executed linux command line of the service.

```
./getBibTeX.pl  
(using CiteSeers  
bib::CreateBibTeXFromText ...)
```

```
./getCitation.pl  
(using Biblio::Citation::Parser)
```

Thanks to
Daniel Trümper

```
$TTT/SCRIPTS/plain2xml.perl \  
| $TTT/bin/fsgmatch -q ".*TEXT" $TTT/GRAM/char/paras.gr \  
| $TTT/SCRIPTS/openangle.perl \  
| $TTT/bin/fsgmatch -q ".*P|TITLE"  
$TTT/GRAM/char/words.gr \  
| $TTT/SCRIPTS/openangle.perl \  
| $TTT/bin/ltpos -q ".*TEXT" -gs ".*P" -qw ".*W" \  
-std form $TTT/RES/POS-RES/resource.xml \  
| $TTT/bin/fsgmatch -q ".*P|TITLE"  
$TTT/GRAM/xml/numbers.gr \  
| $TTT/bin/fsgmatch -q ".*P|TITLE"  
$TTT/GRAM/xml/numex.gr \  
| $TTT/bin/fsgmatch -q ".*P|TITLE"  
$TTT/GRAM/xml/timex.gr \  
| $TTT/bin/xmlperl $TTT/OUTPUT/SCRIPTS/generaltrans
```

Which should
contain XML
according
to DIML

► In PHP the method `string exec (string command)` seems appropriate,
in Java `Process exec(String command)`

Specify the Web Service

Conversion to DIML

- The algorithms create either BibTeX (CiteSeer), a collection of citation items (TTT), or a user-defined format (ParaTools)
- This has to be converted to DiML (*Dissertation markup language*, do not mix with *Data Injection Markup Language!*)
- The DTD entry:

```
<!ELEMENT citation (#PCDATA | email | url | note |  
workauthor | worktitle | articletitle | serialtitle | address |  
editor | publisher | edition | volume | number | version | pages  
| pubdate | bible | court | law | cut | pagenumber)*>
```

- This looks like

```
<citation worktype="Book">  
<workauthor>Abelson, D.</workauthor>  
<worktitle>Preferential cooperative binding of topoisomerase II to scaffold  
associated regions</worktitle>  
<publisher>EMBO J.</publisher>  
<volume>8</volume>  
<pages>3997-4006</pages>  
<pubdate>1990</pubdate>  
</citation>
```

- This XML string will be returned to the invoking client application

Outline: Developing a Citation Web Service

- ▶ Choice of an appropriate tool/algorithm
- ▶ Specify the Web Service
- ▶ Implement the Web Service

Conclusion

- ▶ The project is on a good way!
- ▶ (as the analysis and design part is check-marked)
- ▶ What is left to do?
- ▶ The implementation of
 - ▶ Web Service,
 - ▶ Invocation of the citation extraction algorithm,
 - ▶ Logging facility.

Outline: Logging Facility

- ▶ Aims and features of the logging facility
 - ▶ Logging: Concept for User Identification
 - ▶ Logging: Status and error codes
 - ▶ Logging: The structure of the log file
 - ▶ Data processing
-

Outline: Logging Facility

- ▶ Aims and features of the logging facility
 - ▶ Logging: Concept for User Identification
 - ▶ Logging: Status and error codes
 - ▶ Logging: The structure of the log file
 - ▶ Data processing
-

The aims of the web service-logging facility

The basic objectives of the web services-logging-facility is are ...

- ▶ **measure the user acceptance**
 - ▶ How many users access the web service?
 - ▶ Is the aim to make more users use the edoc-macro fulfilled?

 - ▶ **determine how the users work with this tool**
 - ▶ How many requests does a user submit over time?
 - ▶ Can the users be grouped?

 - ▶ **evaluate the quality of the web service**
 - ▶ General error logging
 - ▶ Determine the quota of correct/incorrect web service results
 - ▶ Determine the quota of maloperation
 - ▶ Derive requirements for tool improvement
-

Web Services offers less options for mining than websites

Issues we don't want / can't achieve in our logging facility:

- ▶ **Generate recommendations "...People how cited this also cited..."**
 - ▶ The web service does not offer this functionality
 - ▶ Insufficient data pool to generate satisfying recommendations

 - ▶ **Content and navigation-Analysis and Improvement**

 - ▶ **Query Clustering**
 - ▶ There's no use in detecting similar queries in our web service request
-

Features needed in the Web Service facility

- ▶ **...to measure the user acceptance**
 - ▶ All incoming requests have to be logged by the logging facility
 - ▶ Each user has to be identified

 - ▶ **... to determine how the users work with this tool**
 - ▶ Each user has to be identified

 - ▶ **... to evaluate the quality of the web service**
 - ▶ Every outgoing web service answer has to be logged by the logging facility
 - ▶ A error-message-concept has to be developed to determine status of the requests and answers
-

Outline: Logging Facility

- ▶ Aims and features of the logging facility
 - ▶ **Logging: Concept for User Identification**
 - ▶ Logging: Status and error codes
 - ▶ Logging: The structure of the log file
 - ▶ Data processing
-

Pro's and contra's of different user identification concepts

► Identification via IP-address

Pro:

- No initial registration of the user
- Already implemented in standard web-logs

Contra

- No possibility to track user over more than one session

► Identification via initial registration

Pro:

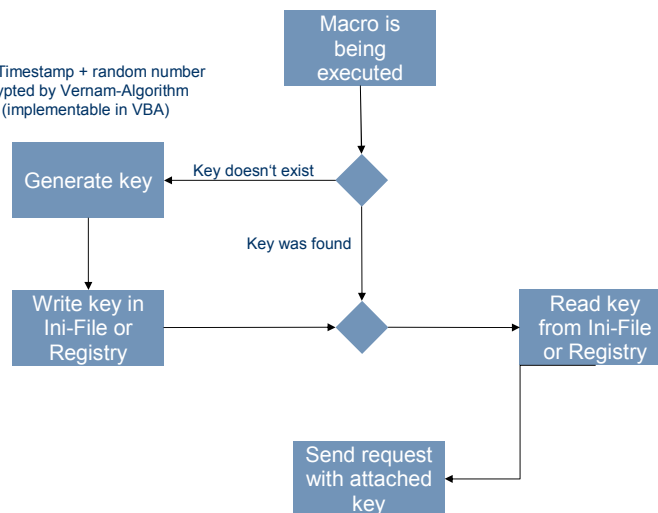
- Exact determination of each user
- Many possibilities to specify groups of user behavior

Contra

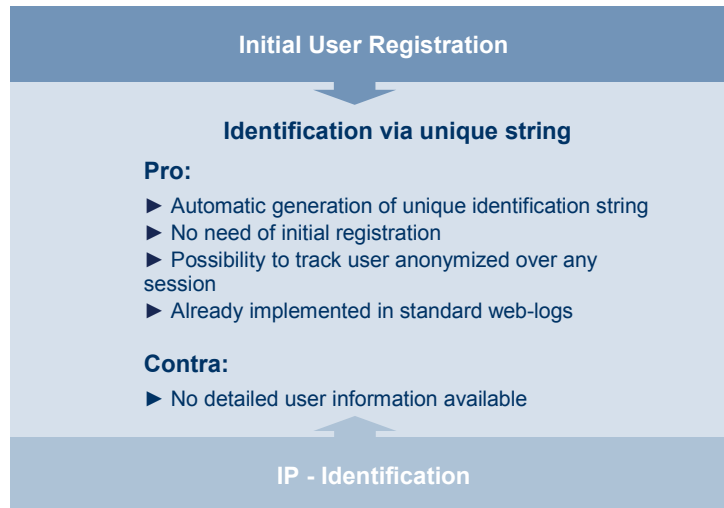
- High entry barrier to use web service and macro
- Problem of using private user data

The process of user identification

i.e. Timestamp + random number
crypted by Vernam-Algorithm
(implementable in VBA)



Pro's and contra's of different user identification concepts cont.



Accessing an Ini-File with VBA using the Windows-API

Structur of an ini-File

```
[Section1]
Key1 = Value1
Key2 = Value2
Key3 = Value3
```

```
' Save a single entry in a section
myIniFile = App.Path & "\" & App.EXENAME & ".ini"
WritePrivateProfileString "Section of Ini-File", „Keyname“,
„Keyvalue“, _ myIniFile

' Deleting a single entry
WritePrivateProfileString "Section of Ini-File", „Keyname“,
vbNullString, _ myIniFile

' Deleting a whole section
WritePrivateProfileString "Section of Ini-File", vbNullString, "",
myIniFile

' Reading a single entry of one section
Dim sValue As String
keyValue = GetIniString("Section of Ini-File", „Keyname“, myIniFile)
```

Accessing the Registry with VBA using the Windows-API

Save a key in the Registry:

```
Private Sub Form_Unload(Cancel As Integer)
    SaveSetting "myProgram", „myKeys", „makroKey", _
    „kjhkjasad982mmkkjk"
End Sub
```

Read a key out of the Registry:

```
Private Sub Form_Load()
    Key = GetSetting("myProgram", "myKeys", "makroKey")
End Sub
```

Outline: Logging Facility

- ▶ Aims and features of the logging facility
 - ▶ Logging: Concept for User Identification
 - ▶ Logging: Status and error codes
 - ▶ Logging: The structure of the log file
 - ▶ Data processing
-

Status and error codes of our web service

	Description
200	Successful
4..	Client errors
499	Undefined client-error
500	Server errors
501	Query could not be transformed
502	Missing query string
503	Incorrect query string
504	Missing identification string
599	Undefined server-error

Outline: Logging Facility

- ▶ Aims and features of the logging facility
- ▶ Logging: Concept for User Identification
- ▶ Logging: Status and error codes
- ▶ Logging: The structure of the log file
- ▶ Data processing

Developing a structure for the log file

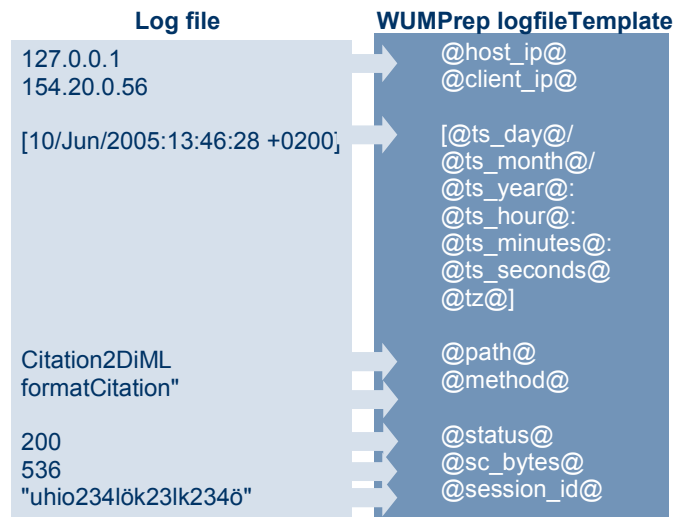
[IP_ADRESS]	IP-address of the current request
[TIME]	Time of request
[SESSION_ID]	Identification key of the macro
[ARGUMENTS]	- Reference string - Called method
[STATUS]	Status of request/answer
[BYTES]	Size of request

```
127.0.0.1 [10/Jun/2005:13:46:28 +0200] "uhio23416k231k2346" `query=Berndt,Bettina?query=formatCitation" 200 536
227.0.0.2 [10/Jun/2005:13:46:28 +0200] "sdf6afskjlasdjfklj" `query=Knopp,Guido?method=formatCitation" 200 232
325.34.0.1 [11/Jun/2005:13:46:28 +0200] "as8uasdjkasdijasdi" `query=?method=formatCitation" 502 445
. . .
```

Outline: Logging Facility

- ▶ Aims and features of the logging facility
- ▶ Logging: Concept for User Identification
- ▶ Logging: Status and error codes
- ▶ Logging: The structure of the log file
- ▶ Data processing

Preparing data processing with WUMPrep



Steps to get a clean web log basis for usage mining

- ▶ **Step 1: Removing request from inhuman agents**
 - ▶ This step can be skipped because no agent requests a web service
- ▶ **Step 2: Filtering of images and unwanted file requests**
 - ▶ This step can be skipped, because there are only "citation requests"
- ▶ **Step 3: Identifying session**
 - ▶ Definition: There are different sessionizing variants. It has to be tested which of these matched best for web services

→ All log entries can be used for mining

- ▶ **Step 4: Complete data preparation by converting log file to ARFF**

**Thank you for
your attention**

**Michael Ferber
Christoph Hanser**