MAX-PLANCK-INSTITUT FÜR WISSENSCHAFTSGESCHICHTE
Max Planck Institute for the History of Science

Max Planck Digital Library (MPDL)

Subproject: Content based web access

Josef Willenborg

Technical Specification

Version	0.2
Author	Josef Willenborg
Created	17.09.2008
Last modified	17.09.2008
Last modified by	Josef Willenborg

Content

1. Architecture and design

has to be done

2. Software

For fulfilling the requirements the following software will be used:

Maintenance of document bases
ESciDoc could be used as the central document base system (intermediate service could be used: ???). All Create/Update/Delete-Operations of XML-documents have to trigger the same operation to the indexing software in some way (???).

Indexing and querying (basic system)
Functional comparision of Lucene, eXist and Oracle (other software such as Tamino, DB2, Postgres could be examined later if needed):

Requirement	**Lucene**	**eXist**	**Oracle Standard Edition One**
Open software	+	+	-
Price	free	free	+ (development and production: 654 Euro for 2 years)
Customizable (Sources extensible for specific needs)	++	-	-
Easy maintenance and usage	++	+	+
Powerful in functionality	+	+	++
Scalable, platforms	+	+	++
Fulltext querying for XML-docs (general)	+	+ (slower regular expressions)	++ (but no regular expressions)
Logical operators: And, or, andNot	+	+	+
Wildcard querying: * (left, middle,right in the word)	+	+	++
Wildcard querying: _ (one character), % (some characters)	-	+	+
Stemming in different languages	+	-	++
Stemming extensible with language specific dictionary	+ (Java programmable	-	++ (build in)

Date range queries	+	+	+
Writing similarity (fuzzy)	+	+ (Ngrams)	+
Phonetic similarity (soundex)	-	-	+
Near operator	+	+	+
Case sensitive querying	+ (with filter)	+	+
Thesaurus searching	-	-	++
Searching for XML-documents by attributes (e.g. author, title)	+	+	+ (within-operator)
Structural querying of XML-documents	-	++ (XPath, XQuery)	++ (XPath, XQuery)
Structural querying in one XML-document (also without index)	Programmable with Java and Xquery	++ (XPath, XQuery)	++ (XPath, XQuery)
Content encoding in UTF8 (query and content)	+	+	+
Query results are customizable	++ (Java)	++ (XPath, XQuery)	++ (Java, SQL, XPath, XQuery)
Query results could be sorted by relevance	+	-	+
Query results could be sorted alphabtically by fields (author, publication year)	+	+ (?)	+
Datastore for specific languages	+	+	+
Datastore content as file, URL and database cell	+ (file)	+ (file)	++ (file, url, cell)
Storing of relational data (for specific needs)	-	+	++ (complex)
Web interfaces	+ (JSP)	++ (JSP, SOAP with XPath/XQuery, REST, WebDAV, XMLRPC, Atom Publishing Protocol)	+ (JSP, SOAP with XPath/XQuery, REST)
Build in WebServer with Servlet-Container for dynamic SQL and XQuery execution (also WebServices)	-	++	++
Maximal size of XML-document	> 2 GB, efficiency is no problem	theoretically as big as the maximum size of a file on the file system; but system hangs if it is too big (> 100 MB)	theoretically as big as the maximum size of a file on the file system; ? has to be tested ?
Maximal number of XML-documents	> 1.000.000	2^{31}	?
...			

...			

Summary:
Lucene could be the system for fulltext querying (with stemming, etc.).
eXist could be the system for structural queries (with XQuery).
Oracle could be the system for both (fulltext querying and structurural querying) and also has the web engine.

Web based querying and presenting XML-documents

Functionality:
- query GUI for searching XML-documents:
- rendering of the document with XSLT
- enrichment of the document with inline images and links to external resources(to e.g. Pollux, geospatial data, etc.)

Lucene:
Web engine: Tomcat as JSP-Container, WAR-application files, static XML-files

eXist:
Webengine with Servlet-Container already contained in database system (has to be activated), access with XQuery or other techniques

Oracle:
Webengine with Servlet-Container already contained in database system (has to be activated), access with XQuery, dynamic SQL or other techniques

3. Hardware

1. Developer-Client-Machines: Mac OS X computer or notebook with relevant developer software

2. Developer-Server: Linux server (small) for document bases and server software (Indexing and querying software, web engine, backup-software).

3. Productional-Server (at the end of the project): Linux server (as big as the user needs) for document bases and server software (Indexing and querying software, web engine).

4. Project organisation

All project descriptions (requirements, specifications, project plans, etc.) are available in Trac-Wiki:
https://itgroup.mpiwg-berlin.mpg.de:8080/tracs/mpdl-project-software

All produced project software with documentation are maintained by Subversion and could be browsed in Trac-Wiki:

https://itgroup.mpiwg-berlin.mpg.de:8080/tracs/mpdl-project-software/browser