

Interactive search results for Digital Libraries

John Papadakis¹, John Andreou¹ and Vassilios Chrissikopoulos²

¹ Department of Informatics, University of Piraeus
80 Karaoli & Dimitriou str, 18534 Piraeus, Greece
{jpap, gandreou}@unipi.gr

² Department of Archives and Library Sciences, University of Ionio,
Plateia Eleytherias, 49100 Corfu, Greece
vchris@ionio.gr

Abstract. In this paper, we propose an architecture that enables interactive ranking and filtering of search results in a digital library by taking under consideration their importance. The importance of a document is measured according to its popularity among the users of the digital library. Further search results manipulation can be performed after the creation of the corresponding list from the search engine and preferably in a different context, without any interaction with the server. A prototype implementation is also presented that demonstrates the functionality of the proposed architecture.

Keywords. Information retrieval, digital libraries, filtering, ranking, importance

1 Introduction

Traditionally, information retrieval modules in the Web and especially in the field of Web-based digital libraries which could be described as a more structured fraction of the Web, are evaluated according to their precision and recall [1]. Precision measures how well the retrieved documents match the referred query and recall reveals the percentage of relevant documents that are retrieved by the search engine [2].

Moreover, in a search results list of many items, documents are usually ranked according to their relevancy to the query that triggered the creation of the list. However, if document lists are too long, relevant documents may end up at the bottom of such lists, being practically unavailable to users. This suggests that an alternative (or complementary) way should be considered for ranking search results. Thus, the importance of a document is considered as a criterion that, in conjunction with relevancy, provides a more efficient method for ranking search results.

In this paper, we propose a method for measuring the importance of a document that also takes into consideration the number of times it has been

accessed (i.e. selected from a search results list) from the users of a digital library. The temporal aspect of this method is calculated by taking under consideration the lifecycle of each document. An architecture is also presented that enables interactive manipulation of search results as provided from the search engine in terms of filtering and ranking. A distinctive feature of the proposed architecture is that it functions separately from the search engine and that it only uses the resources of the client without putting additional strain on the server. Finally, a prototype implementation of the proposed architecture based on XML/XSL technology is presented.

The rest of this paper is organized as follows: in the next section, related work in the field of information retrieval applications that exploit the importance of search results is presented. In section 3 the current status and problems of the classic search methodology of digital libraries are outlined in favor of a way to interactively re-rank search results using many different and combined criteria. Section 4 clarifies the difference between relevancy and importance and describes how importance could be measured and exploited in existing digital libraries and the Web. Section 5 sketches out the benefits from the interactively reranking Digital Library. Section 6 illustrates an architecture that enables interactive manipulation of search results using XML/XSL technology on the client side and section 7 outlines the experiences gained from a prototype implementation of the above architecture. Finally, the last section summarizes the work presented throughout the paper.

2 Related work

When describing a system that ranks documents according to their importance, it is more appropriate to use the "Recommender System" terminology. Recommender systems, as stated by Oard [3], are systems that exploit ratings provided by an entire user population to reshape an information space. In our case, the user population refers to users of a digital library and the information space consists of documents that appear in a search results list. Recommender systems may employ ratings that are provided *explicitly* by qualified users. However, such strategy has proved to be inadequate and cumbersome for areas with many users and documents, like digital libraries [4,5]. An alternative approach to building such systems is the one that employs *implicit* feedback techniques. This is achieved by inferring something similar to the ratings that a user would assign from observations that are available to the system [4].

The Google search engine [6] relies on PageRank [7] to estimate the importance of a Web site in a search results list. According to PageRank, a page has high rank if the sum of the ranks of its backlinks (i.e. links that point to a given page) is high. This recommender system utilizes the link graph of the Web [8] to estimate the importance of a Web page. Although the employment of PageRank for ranking Web sites (or documents in a digital library environment) in a search results list has proved to be useful in the

sense that it objectively ranks sites according to their importance, still no interaction with users on the search results list is allowed. Instead, if users are not satisfied with the search results, they are forced to address a new query or reformulate their previous one.

ResearchIndex [9] is a digital library of scientific literature that measures the importance of its documents based on the Autonomous Citation Indexing (ACI) system. This implicit recommender system maintains an index that automatically catalogues the citations a document contains by linking documents with the cited works. Thus, documents within the search results list are ranked according to the citations they have. Furthermore, ResearchIndex takes under consideration the elapsed time since a document was first published [10]. Indeed, more recent documents are expected to have fewer citations.

A similar approach (though not as applicable as ResearchIndex since it relies on specific document format) is followed by the Open Journal project. According to this project, a link service is defined that associates every journal with the others through its citation links. Links are provided through a publicly available link database [11].

However, ResearchIndex and the Open Journal project also do not provide dynamic search results lists. Furthermore, the strategies mentioned from the above systems can only be applied to repositories that consist of interconnected documents (i.e. through links and/or citations). Digital libraries of multimedia content for example cannot utilize such strategies for the assessment of the importance of their documents.

The DirectHit [12] search engine implements an implicit recommender system that calculates the importance of a Web site based on the number of times previous users have visited it (i.e. popularity of Web site) [13]. However, the temporal aspect of this strategy is difficult to be accurately estimated due to the chaotic and unstructured nature of the Web. Moreover, similarly to the previous cases, no interaction with the search results is provided to the users of the search engine.

3 Problems in current development in Digital Libraries

One of the features that distinguish digital libraries from the Web is their ability to store data in a consistent way [6]. Although the content of a digital library may be as complex as the content of the Web [14], yet it is always structured according to a set of well-defined rules and is most frequently accompanied by metadata. Yet it seems that many digital libraries do not utilize this information to provide better search results and a friendlier and more effective interface.

In this work, we propose a way of monitoring how often users access the documents of a digital library. Such information is stored in a metadata element (i.e. counter) existing for every document of the digital library. This element is processed from a search results list manipulation application

(acting as a recommender system) that uses the output of a search engine as its input to calculate the importance (in terms of popularity) of the referred documents. Such additional information can be used from users of the digital library to re-rank search results in an interactive way not only according to their relevancy but also by taking under consideration their popularity. More details about this approach are given in the following sections of this paper.

4 Exploiting the importance of documents in Web-based digital libraries

4.1 Defining importance

In 1994, Web search engine designers believed that the best search engine would be the one that would rely on a complete search index [15]. They thought that indexing the whole Web would make it possible to find anything easily. Today, the number of available resources on the Web has increased by many orders of magnitude. Consequently, it is very difficult, if not impossible, to create a complete search index for the entire Web. According to a research performed by Lawrence and Giles [13] in 1998, even if results from multiple search engines are combined through meta-search engines, still it is not possible to cover the whole indexable Web.

Search engines within digital libraries do not face such problems since they always rely on a complete index of the digital library's document set. However, search results from digital libraries are subject to the same problems with Web search results, despite the fact that they contain documents with higher precision and recall. Thus, the importance of a document is identified as a criterion that should be considered when ranking search results lists [7,9,11,12].

According to the special structure of documents within individual digital libraries, the importance of each document can be measured in a variety of ways. Hypertext documents for example, are interconnected through a link graph structure, which provides reliable information about the importance of a hypertext document [7]. In the field of scientific literature, citations contained in each document also provide equivalent information [9].

In this paper, we propose a generic method for measuring the importance of a document in a digital library similar to the one that is followed by the DirectHit Web search engine [12]. Specifically, we exploit the implicit ratings for each document in the digital library, as provided from users that have previously accessed this document from a search results list or by browsing the digital library. In order to equally treat "old" as well as "new" documents, it is necessary to take under consideration the lifecycle of the document (i.e. when it was initially submitted to the digital library). Such information can be monitored directly from the digital library or extracted from the Web server's log files. This method measures the popularity of each

document, which is an indication about its importance among the digital library's users.¹

4.2 Applying importance to existing search results applications

The knowledge of the importance of each document in a digital library gives the ability to use such knowledge for ranking search results. However, importance alone should not be applied as a ranking criterion. If documents within a search results list are not relevant to each other or to the initial query, users are likely to meet irrelevant documents at the top of such lists. Thus, importance in conjunction with relevancy should be considered as ranking criteria for lists of retrieved documents in digital libraries.

Alternatively to search results lists, retrieved documents can be organized in sets that share common features. The field of information retrieval provides several ways of formulating such sets. According to one method, documents may be organized in a number of predefined categories [17]. Examples of applications that implement such concept for the grouping of documents in search results lists are Cat-a-Cone [18] and the Northernlight search engine [19]. A second approach is based on the cluster hypothesis as defined by van Rijsbergen: *Closely associated documents tend to be relevant to the same request* [17]. Applications that are based on the cluster hypothesis for formulating clusters from search results are Scatter/Gather [20], WebLuis [21], Grouper [22], Tilebars [23] and many others. In these cases, importance could be considered as a ranking factor within each individual document set.

5 Towards an interactive search results manipulation architecture

5.1 Current status

Search engines in the Web as well as in the field of digital libraries usually provide some mechanisms for tuning queries or re-querying previously obtained results. Most search engine development though has gone on companies with little publication of technical details. This causes search engine technology to remain largely a black art [6]. Apart from this fact, it is common sense that querying query results allows the search to take place over a restricted set of documents and is potentially much more efficient than a completely new search over the entire space [24].

¹ Of course, popularity is not a synonym to importance. However, in diverse environments like digital libraries, popularity is considered a reliable indication to a document's importance.

However, current search engines are already under severe resource constraints and taking over search results manipulation is not a recommended solution. Indeed, manipulating search results at the server side (i.e. where the search engine resides) is expensive in terms of CPU usage and bandwidth. Furthermore, it is difficult for users to explore search results since they are usually not presented in an efficient way.

In order to avoid abuse of server and network resources and give users the ability to search more efficiently, we propose an interactive search results manipulation architecture that is based on XML/XSL [25] technology and operates separately from the search engine.

5.2 Describing the architecture

According to the proposed approach, the retrieved documents (i.e. their references and brief descriptions) are processed entirely at the client side. Specifically, users are able to interact with search results by filtering and/or ranking the corresponding list. The search engine is thus invoked only if the retrieved document set does not satisfy the user in a sense that more documents are needed. The core functionality of the architecture is achieved by applying XSL transformations to the XML data set.

As soon as a user issues a query, the search engine generates the search results list in XML format and transmits them to the client. At the client side, when an XML-capable Web browser identifies XML data, it renders the data locally according to the directives that are defined in the corresponding XSL file. In our case, the XSL file transforms search results in plain HTML so they can be viewed in a "ranked by <measure>" order where measure is the simple or complex criterion that we have selected as default. Users may interact with search results by applying a number of transformations to the XML data. For example, if the result set contains mostly relevant documents, they can trigger (through a "click button" event) a procedure that re-ranks search results by their popularity and find those documents that are considered as most important by the users of the digital library. In cases where more metadata are provided from a digital library, the result set can be ranked and/or filtered according to the values of such metadata. For example, in a scientific literature digital library where the authors of each document are contained in the resulting XML data, users may trigger a "filter by author" procedure to filter the search results list.

The benefits that derive from the employment of XML/XSL technology in an interactive search results manipulation application are really amazing. However, as it will be demonstrated in the next section, current Web browsers fail to support the standards in a consistent way. Therefore, some functionality that should be supported from a Web browser that claims to be XML capable, remains only theoretically supported. Current trend in the Web though indicates that it is a matter of time before all the major Web browsers will support this new technology.

6 Prototype implementation

In order to evaluate the proposed architecture, we have implemented a search results manipulation application at the University of Piraeus Lecture Notes (UPLN) digital library [26].

6.1 Description of the dataset

The UPLN digital library is a relatively small digital library that contains teaching material from nine departments within the University. Each department has several courses whereas each course may be assigned to two or more departments. Although such courses refer to the same scientific area (e.g. "Statistics"), they are usually assigned to different professors. For example, the department of Economics has an "Algebra" course that is assigned to a certain professor and the department of Informatics also has an "Algebra" course, which is assigned to a different professor. In fact, there are courses within the University that are assigned to three or even four different professors. A bibliographic file containing metadata is also provided to each course. The structure of the document repository is similar to the one described by the "dienst" protocol [27].

6.2 Description of the application

In figure 1 there is an example of a file that the dienst digital library uses to describe an entry. Note that the last field, COUNTER is a new addition (not part of the dienst protocol – that's why it is placed after the END field) that represents the number of times this entry has been accessed since it was first inserted. Special care was taken to protect the digital library from malicious users trying to increase a document's popularity so just by clicking on a link several times quickly wont give more than one "hit" to the document.

In order to apply the proposed concepts to the UPLN digital library, we have modified the existing search engine to generate the search results list in XML format. Specifically, as it is presented in fig. 2, each retrieved document corresponds to a "hit" element containing descriptive information about the course according to its bibliographic file. The "counter" attribute refers to the number of times this course has been accessed and the "popularity" attribute calculates the average number of hits in a month for the specified document.²

² The figures from the UPLN digital library have been manually translated from Greek to English for the needs of this Conference. The reader is prompted to visit the UPLN site (the URL is provided at the "References" section) for real time execution of the application.

```
BIB-VERSION:: CS-TR-v2.1
ID:: unipi.csd//csd-001
TITLE:: "Database Systems"
xed^M
ENTRY:: February 12, 1999
AUTHOR:: George Chondrokoukis
ABSTRACT:: ="Homework for Relational Schemas..."
END:: unipi.csd//csd-001
COUNTER:: 624
```

Fig. 1. A ".bib" file that contains information on a document entry

```
<?xml version="1.0" encoding="ISO-8859-7"?>
<?xml-stylesheet type="text/xsl" href="hits.xsl"
xmlns:xsl="http://www.w3.org/TR/WD-xsl"?>
<hits>
  <hit date="November 13, 1999" counter="450"
title="Database Systems"
url="http://thalis.cs.unipi.gr/Repository/U.I/prod-
001/index.html" popularity="15.3"
abstract="Database Systems..." id="prod-001">
    <author>George Chondrokoukis</author>
  </hit>

  <hit date="November 13, 1999" counter="353"
title="Relational Databases"
url="http://thalis.cs.unipi.gr/Repository/U.I/csd-
003/index.html" popularity="12" abstract="Homework
for Relational Schemas..." id="csd-003">
    <author>George Vasilakopoulos</author>
  </hit>
  .
  .
  .
</hits>
```

Fig. 2. An XML file that contains search results as generated from the search engine

When a student issues a query to the search engine, a search results list in XML format is generated. The XML data migrate to the client side and is transformed in HTML according to the XSL transformations defined in its accompanying hits.xsl file. Hits are ranked by their relevancy as provided from the search engine. Users can click on the “Hits” or “Popularity” buttons at the left frame to re-rank documents according to their overall number of hits or popularity respectively.

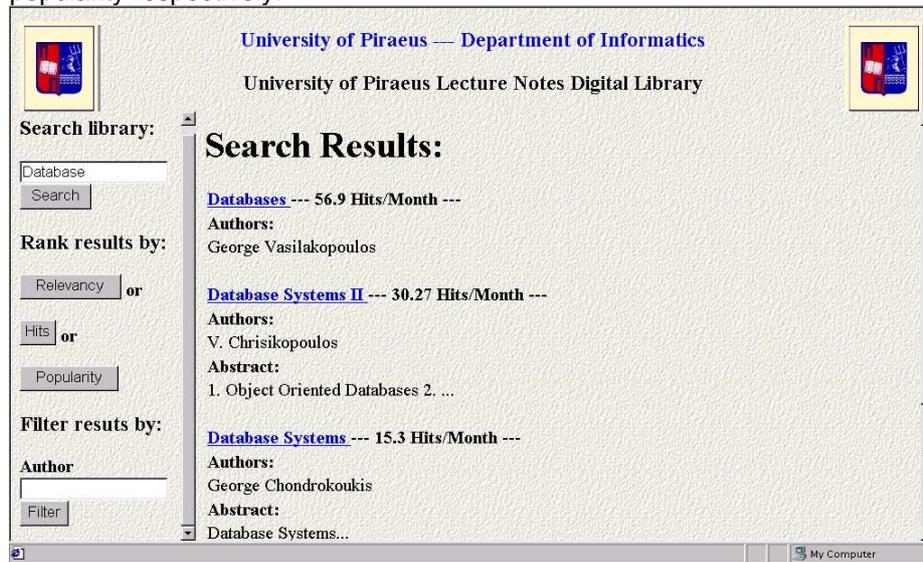


Fig. 3. Ranking search results according to their popularity

Fig. 3 presents a re-ranked search results list according to the popularity of the documents it contains. The query that initially generated this list consisted of the keyword “database”. In case a user wishes to view the lecture notes of a certain professor in the above list, she/he can filter search results by filling the “Author” field and by clicking the “Filter” button. The transformed (i.e. filtered) search result list is presented in fig. 3:

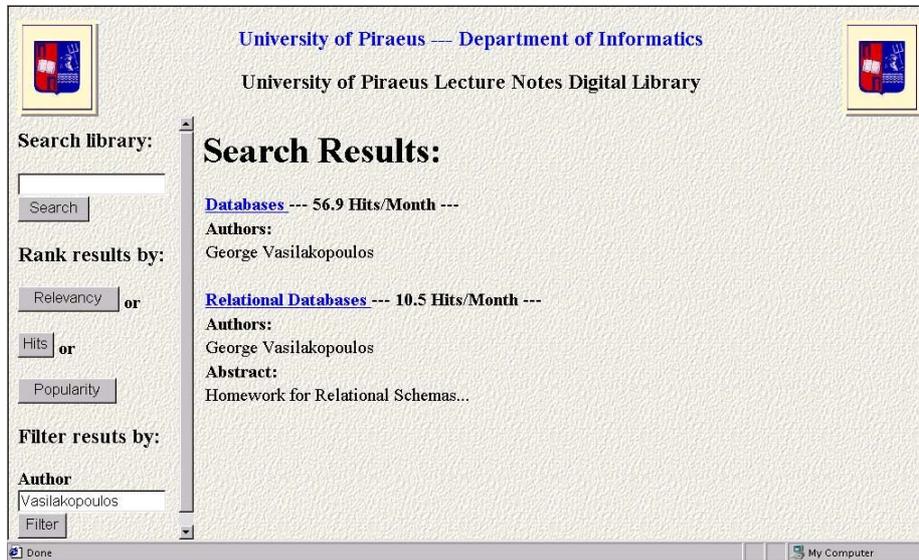


Fig. 3. Filtering search results according to the value of a certain metadata element (*i.e.* “author”)

The described functionality is provided through scripts implemented in JavaScript that apply XSL transformations to the initial XML data. As it has already been mentioned, interaction with users is taking place at the client side using local resources without invoking the search engine. If users need more documents than the ones that constitute the initial search results list, they can address a new query to the search engine or rephrase their previous one.

6.3 Results

While a complete user evaluation of the prototype implementation is beyond the scope of this paper, according to feedback provided by students and according to our own experience, the search results manipulation application provides better understanding of the search results list and significantly improves the performance of the search engine. However, our application depends on the capability of the Web browser to identify XML data and handle it appropriately. So far, only MS Internet Explorer ver. 5.0 and above is compatible with the implementation presented here. Although this is not a serious problem for the specific application since MS Internet Explorer is the dominant Web browser in the University of Piraeus community, we are anticipating XML support from all major Web browsers in the near future in order to be able to apply the proposed architecture to larger digital libraries and ultimately to the Web.

7 Conclusions

In this paper, we addressed the problem of ranking search results according to the importance of the documents they refer to in a digital library. According to related work, importance can be measured in a number of ways in a digital library environment, depending on the overall structure of the underlying document set. The success of Web-based information retrieval applications like Google, ResearchIndex and DirectHit that measure and exploit the importance of documents indicates that relevancy should not be considered as the only criterion for ranking search results.

We have also presented an interactive search results manipulation architecture that, acting as a recommender system, allows users of a digital library to rank and filter search results independently from the search engine. Better control over server resources is provided since search results manipulation is done separately from the search engine. Less network traffic is also achieved due to the fact that search results are managed exclusively at the client side.

Based on the previous remarks, we have implemented an application that manipulates search results as provided from the UPLN digital library 's search engine. The presented application is based on XML/XSL technology and depends entirely on the Web browser to appropriately handle the incoming data. Despite the fact that the current implementation can only be executed in the context of a specific Web browser, it is our strong belief that it is a matter of time before the majority of Web browsers will be able to render XML data efficiently and in a standardized way.

Acknowledgements

The research described here was funded in part by the Greek General Secretariat on Research and Technology under a PENED grant.

References

1. Salton G. Automatic text processing: The transformation, analysis and retrieval of information by computer. Addison Wesley, 1989
2. Pinkerton, B. Finding What People Want: Experiences with the WebCrawler. Proceedings of the Second International WWW Conference (WWW94), pp. 17-20, available at: <http://info.webcrawler.com/bp/WWW94.html>
3. Oard D. W. The State of the Art in Text Filtering. User Modeling and User-Adapted Interaction, Vol. 7, No. 3, pp. 141-178, 1997, available at: <http://www.glue.umd.edu/~oard/research.html>

4. Oard D. W, Kim J. Implicit Feedback for Recommender Systems. Proceedings of the AAAI Workshop on Recommender Systems, 1998
5. Nichols D.M. Implicit Rating and Filtering. Proceedings of the 5th Delos workshop on filtering and Collaborative filtering, pp. 31-36, 1997
6. Page L, Brin S. The Anatomy of a Large-Scale Hypertextual Search Engine. Proceedings of the Seventh International WWW Conference (WWW 98), pp. 14-18, 1998 (System is online at <http://google.com>)
7. Page L, Brin S, Motwani R, Winograd T. The PageRank Citation Ranking: Bringing order to the Web. Stanford Technical Report, available at: <http://www-db.stanford.edu/~backrub/pageranksub.ps>, 1998
8. Kleinberg J. M. Authoritative Sources in a Hyperlinked Environment. Proceedings of ACM-SIAM Symposium on Discrete Algorithms, pp. 668-677, 1998
9. Lawrence S, Giles C. L, Bollacker K. Digital Libraries and Autonomous Citation Indexing. IEEE Computer, Vol. 32, No. 6, pp. 67-71, 1999 (System is online at <http://researchindex.com>)
10. Lawrence S, Bollacker K, Giles C. L. Indexing and retrieval of scientific literature. Proceedings of the Eighth International Conference on Information and Knowledge Management, CIKM 99, pp. 139-146, 1999
11. Hitchcock S, Carr L, Hall W, Harris S. Linking Electronic Journals: Lessons from the Open Journal Project. Dlib e-magazine, ISSN 1082-9873, December 1998, available at: <http://www.dlib.org/dlib/december98/12hitchcock.html>
12. The directHit search engine, available at: www.directhit.com
13. Lawrence S, Giles C. L. Searching the World Wide Web. Science, Vol. 280, No. 4, pp. 98-100, 1998
14. Papadakis J, Despoina Polemi D, Vassileios Chrissikopoulos V. A Secure Web-based Medical Digital Library Architecture based on TTPs. Proceedings of the XVI Medical Infobahn for Europe Conference (XVI MIE2000), pp. 610-617, 2000
15. Plewe B. Popular Navigational Aids, available at: <http://botw.org/1994/awards/navigators.html>
16. Rusch-Feja D. Metadata: Standards for retrieving WWW documents (and other digitized or non-digitized resources). Library and Information Services in Astronomy III, ASP Conference Series, Vol. 153, 1998
17. Sahami M. Using Machine Learning to Improve Information Access: Thesis: Department of Computer Science, Stanford University, 1998
18. Hearst M, Karadi C. Cat-a-Cone: an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 246 – 255, 1997
19. The Northernlight Search Engine, available at: <http://www.northernlight.com>
20. Hearst M, Pedersen J. Reexamining the Cluster Hypothesis: Scatter /Gather on Retrieval Results. Proceedings of the nineteenth annual international ACM SIGIR conference, pp. 76 - 84, 1996

21. Liu Y-H, Dantzig P, Sachs M, Corey J, Hinnebusch M, Sullivan T, Damashek M, Cohen J. Visualizing Document Classification: A Search Aid for the D.L. Proceedings of the second European Conference in Digital Libraries ECDL '98, Springer Verlag, pp. 555-567, 1998
22. Zamir O, Etzioni O. Grouper: A Dynamic Clustering Interface to Web Search Results. Proceedings of the 8th WWW Conference, available at: <http://www8.org/w8-papers/3a-search-query/dynamic/dynamic.html>
23. Hearst M. Tilebars: Visualization of term distribution information in full text information access. Proceedings of CHI '95, pp. 59-66, 1995
24. Wells D, Kurien A. Searching and Indexing, available at: <http://www.objs.com/survey/crawl.htm>
25. Leventhal M, Lewis D, Fuchs M. Designing XML Internet applications. Prentice Hall PTR, ISBN: 0-13-616822-1, 2000
26. University of Piraeus Lecture Notes (UPLN) digital library, available at: <http://thalis.cs.unipi.gr/UPLN/search.html>
27. Lagoze C. Dienst – An Architecture for Distributed Document Libraries. Communications of the ACM, Vol. 38, No 4, 1995.