Document Management as a Database Problem

Rudolf Bayer
Institut für Informatik, Technische Universität München
e-mail: bayer@informatik.tu-muenchen.de

Extended abstract of invited paper

Document management has many aspects, among them acquisition, storage, retrieval, presentation and processing of documents (work flow). These aspects will be covered and illustrated by examples from the library system OMNIS/Myriad.

1. Document Acquisition

There is a broad spectrum of techniques how to acquire documents in such a way, that they are in computer-readable form and can be stored in a document base. This spectrum ranges from fully automatic at low cost via semiautomatic using tools like scanners and optical character recognition (OCR) to manual acquisition according to elaborate rules and regulations. The purpose of high quality document acquisition is to capture the structure and the semantic content of a document not as far as possible but as far as affordable. Presently the state of the art of semiautomatic acquisition of paper documents is scanning followed by OCR. This yields a facsimile image and the text content, but no structure and no real semantics. In many cases the text produced by OCR is low quality and must be corrected to be useful for effective retrieval. Various projects are under way to capture structure and semantics automatically [1,12], but they have not reached sufficient maturity to be used in production environments like libraries or businesses.

The cost of document acquisition is probably the most important, but also the least adressed issue in building and maintaining large document bases. Cost is not critical for fully automatic systems, but very critical for semiautomatic and manual acquisition processes. Acquisition cost varies easily by a factor of 50. The acquisition of paper documents with scanning and OCR in OMNIS/Myriad [4] costs about 2 DM per page presently. Our cost goal, which we hope to reach with the next version, is 1 DM. The cost of "acquiring" the catalogue entry of a book using the cataloguing rules [5] of the German library community is as high as 50 DM.

Considering large document bases with millions of documents it becomes obvious, that the cost of acquisition and maintenance is a critical issue and often dwarfs the system cost (hardware, software, storage system, network) by a factor of 100 and even more. Therefore it must be the overriding issue in designing a document management application and in selecting a document management system.

2. Architecture of Document Storage System

Storage Requirements vary widely, from a few kilobytes per page for coded text information to about 25 MB for a high quality colour image. For non-coded information or pixel images therefore compression is essential. This brings storage requirements down to about 35 KB per black and white image of a text page and to about 165 KB for colour images of pages of rare books [7]. I estimate that the 20 million volumes of the Library of Congress would require about 180 Terabytes of storage, which is well within the reach of todays archive stores based on high density video tapes and jukeboxes.

Storage Hierarchies: Large document stores require hierarchies consisting of main storage caches, hard disk, hard disk caches for slower media like CD-ROM and tapes, and even offline stores like cabinets full of CD-ROM and tape cartridges which are then mounted on demand. When data are migrated, e.g. from hard disk to CD-ROM, they must be reorganized in order to match the access characteristics and speeds of the new media. The slow access speed of CD-ROM can e.g. largely be compensated by caching the relevant parts of indexes - which is typically less than 1% of the CD-ROM content - on hard disk. Also the text part of a document base which is needed for search and retrieval can be separated from the image data, which are only needed for viewing a document. The text requires high frequency, high speed access. The text is only a few percent of the total document base and can be therefore be

Proceedings of the 21st VLDB Conference
Zurich, Switzerland, 1995

kept on faster storage devices. This requires, however, that the document base must be split technically into several databases which are distributed over a variety of storage devices. Furthermore, such storage hierarchies must be managed completely transparently for the user. Therefore complex software and a variety of tools to support the document base administrator and the archiving personel are needed.

Networking: Document bases live in networked environments, usually mixes of LANs and WANs. Applications designed for LAN environments do not easily port to WANs. The reason is that in LAN applications the slowest device is the hard disk with about 10 ms latency time. Therefore applications are architected to cope with disks and to hide their latency time as much as possible. In WAN environments the slowest device is the network itself with latency times of about 5 sec. This is 500 times slower than disk latency. As a consequence, applications often have to be rearchitected when moved from LAN to WAN.

Client Heterogenity: This is another serious problem, especially in WAN applications. Document base applications must perform a lot of work on the client, e.g. image decompression, supporting and checking user activity, preprocessing and optimizing queries for document retrieval, supporting work flow, etc. This yields highly complex client software. But, whereas the document base is usually kept on one or several homogeneous servers, the client machines are rather heterogeneous. The consequences are that many versions of the complex client software for a variety of platforms must be produced, distributed and - most of all - installed and maintained on a large number of individual client machines with widely different configurations. Solutions in the spirit of WWW and web viewers like NetScape and XMOSAIC seem promising, but so far one must pay a very high price in functionality and performance for such general as compared to customized solutions.

Intelligent Document Store: Today's database systems treat documents as binary large objects (BLOBs), i.e. they know nothing about the internal structure and the semantics of a document. All this must be taken care of by the application.

Obviously, it is very desirable to integrate the semantics of a document as much as possible into the document management system. This would make application development much easier and document processing probably much more efficient.

3. Retrieval

Searching and retrieving of documents is intimately tied to the acquisition techniques. The more care and cost is invested in the acquisition process, the more convenient and effective retrieval techniques can be expected. Today retrieval via hierarchical classification techniques and via so called information retrieval based on text search is used. These basic search techniques are often enhanced by fuzzy search, soundex methods and relevance ranking.

The underlying basic search engines use classical database access techniques like B-tree indexes, hashing and signature files. A very careful and extensive study [10] recently found, that for text retrieval purposes inverted files based on B-tree indexes are far superior to signature files both in space requirements and in runtime efficiency.

Hypertext techniques for document linking and document retrieval have become very popular and widespread through the world wide web, but they suffer from the "lost in hyperspace" syndrome, which seems partially due to the lack of semantic meaning of the links.

Recently investigations have started [6,8] to integrate classical information retrieval, hypertext linking and metainformation in the form of semantic networks. The hope is that the document retrieval can be made more convenient and more effective by providing this high level semantic information. It is interesting that logic programming techniques are used to implement the basic search engine. A key question in this research will be, to what extent the semantic network construction can be automated, which determines largely the cost of document acquisition.

4. Presentation

A document management system stores documents in several forms for different purposes. Typically a document is represented as a triple (structure, text, image). The structured information and the text are needed for searching and retrieval. The structure is also needed for additional management and workflow purposes like reserving and ordering a book from a library. The image is needed for presentation and viewing, e.g. for looking at the electronic image of the table of contents of a text book or of the abstract of a journal article before ordering the document from the library, from the publisher or from some other document delivery service.

Presentation of images requires transmission over the network in compressed form and decompression in the client. For this, asymmetric compression/decompression methods are needed which are optimized for fast decompression in the client. Such documents are typically compressed once in a powerful server which could even employ special purpose hardware, as is done today for video compression. But the document is decompressed many times in computionally weaker clients, which most of all must rely on general viewers and cannot employ special purpose decompression hardware.

Document presentation also raises the issue of confidentiality and security. It is clear that not every user can see or even manipulate every document. Therefore the usual techniques of managing users with various privilege classes and documents with various access restrictions must be part of a document management system. In WAN applications there is the additional problem of documents being passed through intermediate network nodes of unknown security and trustability. Therefore there is considerable need for anonymity or pseudonymity of both senders and receivers of documents, and for encrypting the transmitted documents. For encryption and decryption analogous arguments can be made as for compression and decompression w. r. to splitting the work between client and servers. Again, as many of these issues as possible should be solved once and for all by the document management system rather than many times by all the applications.

5. Document Authoring:

So far we considered the management of essentially finished and therefore static documents. Another dimension of complexity and challenge opens up if we take into account the process of authoring and constructing the documents. Often the internal structure of documents closely mirrors the structure of the documented system, take e.g. the UNIX manual or the operating manual of a car. In OSIDOK [9] complex technical systems like automobiles are modelled using an object oriented database system. Then the operating and maintenance manuals are generated semiautomatically from text and image modules corresponding to the technical subsystems out of which the physical automobile is constructed.

In addition to the tasks of document management systems considered so far workflow management techniques must be added. In such environments the design steps are long and complex and require much more general transaction concepts than the classical ACID transcaction.

A number of generalized transaction concepts like nested transactions, sagas, contracts have been developed. For MoodBase [2,9] the concept of Δ-transaction was developed [11], which uses a check-out and check-in mechanism in combination with local, private extensions (deltas) of the

database to manage all the changes to the design. The Δ-transactions weaken the ACID properties by giving up symmetric isolation in the sense that consistent updates - due to check-ins of other users - become visible and lead to non-repeatable reading. This weakening seems to be quite natural in the sense that it mirrors reality closely.

In this way the fairly long working steps of a community of users can be coordinated in a natural way. Therefore a generalized transaction concept should be part of a document management system.

On the other hand, the working steps themselves seem to be highly dependent on a particular application domain. So far at least generic models are missing. For that reason it does not seem to be appropriate to incorporate the coding of the working steps themselves into the document management system proper. However, rather general solutions for specific domains like office automation [3] or CASE or automobile documentation seem to be feasible and lead to an interesting layer of middleware between the document management system and the genuine application.

Bibliograhy:

[1] **Neuhold, E., Turan, V.:** *Database Research at IPSI,* SIGMOD Record, Vol. 21, No. 1, March 92, p. 133

[2] **Bayer, R.:** *Mood: A Knowledgebase System with Objektoriented Deduction,* DASFAA '91 Proc., pp. 320-329, Tokyo 1991

[3] **Karbe, B., Ramsperger, N.:** *Office Work Coordination Using a Distributed Database System,* DASFAA '91 Proc., pp. 439-443, Tokyo 1991

[4] **Bayer, R.:** *OMNIS/Myriad Elektronische Verwaltung und Publikation von Multimedialen Dokumenten.* In: Informatik, Wirtschaft und Gesellschaft. 23. GI-Jahrestagung, Dresden 1993, pp. 482-487, Berlin: Springer 1993

[5] **Regeln für die alphabetische Katalogisierung RAK** Band 1 bis 6, Dr. Ludwig Reichert Verlag, Wiesbaden 1983-1989

[6] **Wiesener, S., Kowarschick, W., Vogel, P., Bayer, R.:** *Semantic Hypermedia Retrieval in Digital Libraries.* To appear in: ADL '95, Advances in Digital Libraries, 1995

[7] **Böhm, C., Opitz, A., Vogel, P., Wiesener, S.:** *Prints of the 17th Century in a Distributed Library System.* To appear in: Database and Expert Systems Applications, DEXA '95

[8] **Kowarschick, W., Roth, C., Vogel, P., Wiesener, S., Bayer, R.:** *OMNIS/Myriad on its Way to a Full Hypermedia Information System,* 1st European Workshop on Human Comfort and Security, EITC, Brussels 1994

[9] **Höfling, G., Kempe, J., Bayer, R.:** *The Application of an Object-Oriented Database System, Exemplified*

by an Information and Document Mangemant System. Proc. SI-DBTA-Workshop: Object-Oriented Database Systems at Work., 1992

[10] **Zobel, J., Moffat, A., Ramamohanarao:** *Inverted Files versus Signature Files for Text Indexing,* CITRI/TR95-5, Melbourne, Feb. 1995

[11] **Kempe, J., Höfling, G.,:** *Differential Transactions - A Model for long-lived Transaction,* Bayerisches Forschungszentrum für Wissensbasierte Systeme, München, Mai 1993

[12] **Werner, J., Güntzer, U.,:** *XTUMIS: A step towards a true electronic library.* Proc. 2nd International Conference on Information Technology for Training and Education, ITTE 92, Brisbane, Sept. 1992 The University of Queensland, 1992, pp. 614-631