Replication within Distributed Digital Document Libraries

László Kovács, András Micsik

Computer and Automation Institute of the Hungarian Academy of Sciences
Distributed Systems Group
MTA SZTAKI, ASZI, H-1111 Budapest XI. Lágymányosi u. 11. Hungary
laszlo.kovacs@sztaki.hu, micsik@sztaki.hu

Abstract

Cooperative and interoperability aspects of distributed digital document libraries are discussed. The commitment of ERCIM institutions to the Dienst protocol raised the necessity of the development of general detailed architecture of Dienst. A new Dienst architecture augmented by replication service is suggested. Replication is described in details. An abstract Reference Model of distributed digital document libraries can be concluded as well.

1. Introduction

Nowadays digital document libraries are under intensive research and development [Infmed,Stan,Umich]. There are numerous examples of such R&D projects [DigLib]. In these projects the search, retrieval and homogeneous user interface problems are mostly discussed. Little attention was devoted to cooperative and interoperability aspects of distributed digital document libraries.

Digital libraries store documents in different ways, and provide various indexing and searching capabilities. At this moment no uniform and/or standardized data, and catalog record formats are used. Although there are several initiatives for standardization of e.g. catalog record formats [RFC1807] there is no initiative for creating a standardized Reference Model of Distributed Digital Libraries. In this paper a set of digital libraries cooperating to provide uniform user services and digital data is presumed.

A digital library contains digitally encoded information that can be represented by electronic, optical, and magnetic devices and can be transmitted via high-speed network connections. Although there is fast growth in telecommunication services but the available bandwidth is always a limiting factor. Replication was traditionally used within the area of distributed and object-oriented database systems. On the Internet [Comer] the distribution of well established software and media libraries is traditionally applied as a special form of replication (mirroring).

First the basic structure of cooperative digital document libraries is described, then one of the central concepts of this area, the replication of digital data documents is discussed.

2. Architecture of Distributed Digital Document Libraries

General Reference Model of Digital Document Libraries does not exist yet. Instead some of the concepts applied at the construction of experimental Digital Libraries (DLs) are described.

First to mention of these concepts is the Information Bus [Stan] worked out at Stanford University. The Information Bus stands between the Information Sources like WWW servers, OPACs, public databases and the Interface Clients which make the contact with users. This bus contains all the knowledge about formats, protocols and everything else that is needed to connect users and sources. It may contain Protocol Machines and Library Services. Protocol Machines do translations between foreign protocols and the inner protocol of the Information Bus or they can automate more complicated tasks. Library Services can handle specific tasks needed in a DL, like finding relevant items, format conversions, authentication and accounting. As the Information Bus evolves, it can perform commands at higher level, making the DL more and more efficient.

Secondly, Dienst [Dienst, DavLag94] is a widely used Digital Library Server originating from Cornell University. Its protocol has separated basic functionalities of a digital library making it possible to distribute services among different physical locations. This gives us a rather minimal but characteristic set of services in a distributed digital document library:

- **user interface services**: connect the user to the library and provide data in human readable format
- **repository services**: storage and retrieval of digital document
- **index services**: searching and maintainance of index and catalogue data
- **meta service**: provides central directory of all other services

In Dienst each request of a user is translated by the user interface into a set of commands which are executed by a transparent cooperation of the DL services. In case of a search request proper index services are located with the help of the meta service, then searches are performed by the index services and result is presented by the user interface. When the user has selected a document for retrieval the appropriate repository service is asked about the available formats and finally the document is retrieved.

This architecture is suggested to be extended by a new auxiliary service namely replication. It is responsible for managing replicas of repository and/or index data.

3. Replication

Mirroring in the Internet jargon means replicating some data on a physically different server. This technique helps to decrease network traffic if the so-called mirror sites are well-known and everybody retrieves the data from the "closest" server (in the Internet sense). Popular software libraries typically establish mirror sites in Europe, US, Japan, etc. Doing so long-haul network traffic between continents can be avoided. Other benefits of replication: higher availability and faster accessibility.

Mirroring is a straightforward technique with FTP archives, where the protocol itself is used for creating the copy. The copy in this case means a piece of a filesystem. The USENET News service also has a mechanism for replication, a neighbor relationship is established among News servers and differences of the stored data are periodically exchanged between neighbor machines. Mirroring is also applicable for World Wide Web services although it has not developed a well-designed and common style. The aim of mirroring on the Web is to duplicate a hypertext structure. The problem here emerges from the dual nature of WWW data; structure of stored data and structure of the hypertext does not overlap and this makes mirroring a complicated task.

Another tool for replication in use with WWW services is the caching. It does not copy hypertext, only stores pieces of hypertext that a user asked for in the cache. Later requests can be fulfilled from the cache eliminating some of the network traffic. This is very efficiently used nowadays, the only drawback is that only parts of the original data is available in the cache, not a whole copy.

3.1 Replication within Distributed Digital Document Libraries

Replication service is an auxiliary service besides the previously mentioned basic digital library services that can operate independently or in cooperation with other services. There are three possibilities for replication services according to the origin of the replicated data:

- replication of (collections of) digital documents
- replication of index data
- replication of user interface data

Replication of digital documents is the creation of identical copies of digital documents in a separate server site. The identity relation can be defined at structural, presentational, and content level. Stored physical bit images may or may not be preserved. Indexes are specially used digital data structures. Their replication does not necessarily mean the copy of the index data structures but the recreation of their meaning at the separate site. This can be done by (incremental) updating of the index data after the replication of digital documents at the separate site. Replication of user interface data is the synchronized presentation of user interface data and interface events at several user interfaces. This problem is currently out of scope of this paper.

3.2 Techniques of Replication

The first steps in replication include creating the copy of data and set the mirror to work. Replication of large document collections needs large amount of data to be transferred on network, so this action may apply compression and appropriate timing.

Then the fresh replica site has to advertise itself on the network. For example a meta server or a mechanism similar to the Domain Name System can be used. This makes the possibility for a user interface to distribute interactions among master and mirror servers intelligently. The user interface client can measure response times of possible subjects of retrieval and choose the fastest connection available. Later it can abort some very low speed connections and try with other sites or it can start concurrent queries at several replica sites which makes the response quick.

Once a replication is established it has to follow changes in the master database, insertion or withdrawal of publications and modifications on existing publication data.

The simplest case is when there is no protocol support for replication. Servers construct and collect their replications pieces by pieces using the retrieval facilities of the server or the user interface. Every server can make a replication in this way, but these replications must be periodically tested against changes in the original data.

An easy thing for protocol support could be a command which makes the server pack, compress and transfer a copy of its data to the replication site. This mechanism could introduce authorization and checking for permission of replication, so volunteer (or pirate) replication could not be done this way.

There is a good opportunity to develop a protocol for Replication. This would involve the previously mentioned command for initializing a replication database, tools for registering mirror sites, and configurable update mechanisms. Mirror sites could register their address, preferred update mechanism and interested document formats. The two main options for update mechanisms are: master-initiated and mirror-initiated update.

- Master-initiated update means that when anything has changed at the master server, it calls up the registered mirror servers and sends them the modifications coded into an update control file.

- Mirror-initiated update would work in the reverse way; mirror sites would periodically call the master server and download the update control file.

The two kinds of updates could be supported parallel by the master server, its only task is to collect modifications into a timed list. When one of the mirror servers is contacted with either kind of initiation, the update control file sent contains all the modifications having time entries older than the last update sent to the registered mirror site.

This mechanism could be simplified where the automatic recording of modifications is not supported. The master server periodically generates a listing of its publications and the date of the last modification for that publication. This listing is downloaded to the mirror server, the difference from the replica is determined and the modified publications are requested from the master server.

The replication protocol creates the possibility for adoptation of more sophisticated replication techniques in the field of distributed databases and distributed object management systems [BKTJ92, JabWed90]. Statistics and behavior of replication-generated network traffic could be compared to those of distributed database or object management systems and experience could be taken over.

4. Dienst

Currently the Dienst protocol is the most promising protocol for communication with distributed digital libraries [DienstProt]. The Dienst protocol was developed at Cornell University in collaboration with Xerox Corporation and is installed at several major US university sites. The project was partially supported by ARPA and CNRI. The Dienst Digital Library Server has three characteristic features:

1. Unique document identifiers. The identifiers are split into two parts: a publisher and a DOCID. Publisher names are centrally registered, while publishers are responsible for assigning unique DOCIDs to their publications.
2. Message passing scheme, which allows requests to address the whole library, individual services, documents or subparts of multiple document formats.

3. The user interface is written as a World Wide Web service, thus using WWW clients all services of Dienst protocol are accessible.

The current release of the Dienst server contains all four services but not all services have to be operating. The first choice is to operate only a UI service. Second choice is to install the Repository, Index and User Interface services. The Meta service will be run only at sites which hold central authority in some area of publishing (for example there is a Meta service at Cornell for the American publishers of technical reports).

Meta Service creates groups of Dienst servers. The meta-information maps publisher names to addresses of Index and Repository services. All servers replicate the meta-information database of their meta server by downloading it at given intervals.

4.1 Replication within Dienst

Typically a Dienst server functions as a central publishing service for some publishers. Publishers can separate their publication data into different storage places (repositories), because the inner structure of repositories is highly customizable. In fact everybody who installs a Dienst server has to implement his own routines for mapping between document identifiers and place of physical storage. On the other side all publishers on the server has to share one index space which is updated manually. Index mechanisms may also vary on different servers, though changing this needs some programmer skill.

The smallest piece of data which can be replicated is a repository of one publisher. There is no support in the Meta Service for registering smaller entities of publications. Repositories on a Dienst server can be reproduced with the simplest copying technique (mirroring) based on either HTTP or Dienst protocol. A mirroring program could retrieve all document identifiers from a server then explore and download the available formats document by document. Exceptions for this are files that are hidden from Dienst protocol such as imagemap files or in-line images of HTML documents [HTML]. Imagemap files stay hidden inside the repository data and map coordinates on an image to other images or files, and used for example with thumbnail images. Clicking one page on a thumbnail view of document pages, a Dienst server can send you that page in full size. Currently Dienst can not naturally provide full featured HTML documents such as HTML documents containing in-line images and/or split into several files.

Except these files all other files can be reproduced at another network site, although HTTP-based mirroring software need some more built-in intelligence. Since the structure of a Dienst repository is customizable, possibly different directory structures and naming techniques put some extra work on the mirroring software when storing files in the repository. The final steps of mirroring: updating indexes, and setting the replicated publisher name as local in the server configuration. =46rom this moment a Dienst server can locally serverequests concerning the replicated publisher.

To service requests coming from other machines, the User Interface Service and possibly the Meta Service has to be modified. The User Interface has to be prepared to choose intelligently from original and mirror sites and forward requests to the selected one. The Meta Service is capable for advertising a publisher at different sites but it cannot differentiate between master and mirror sites.

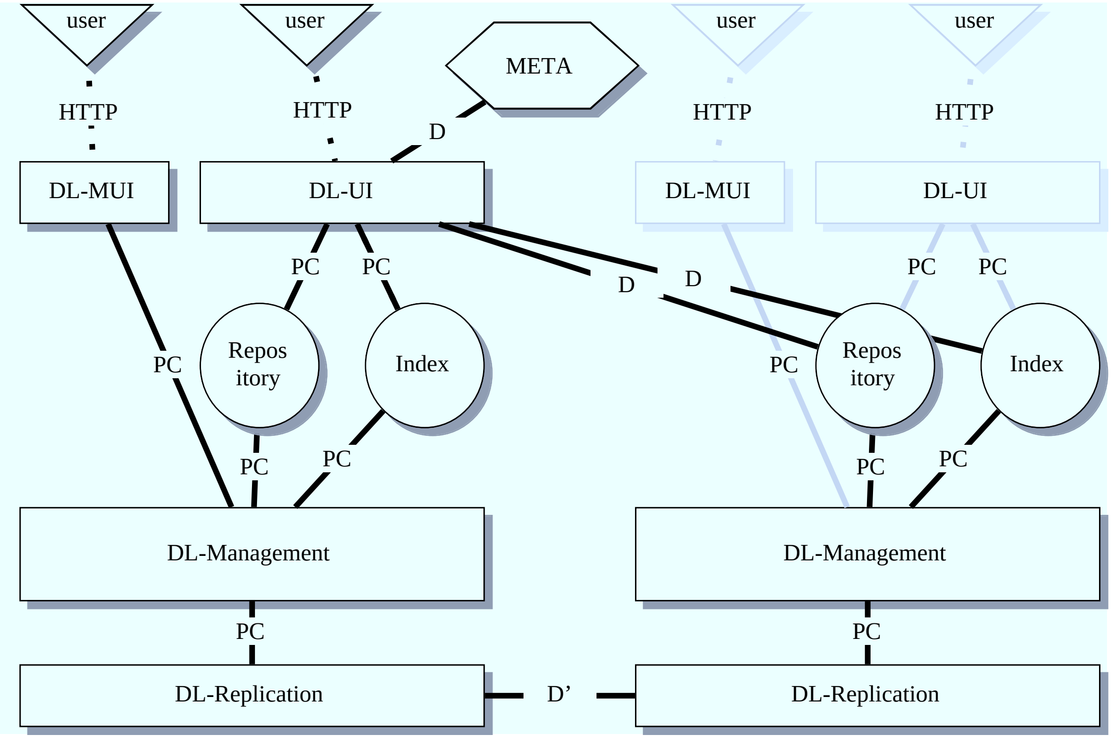
4.2 Dienst Replication Service

The insertion of new publications will soon be automatized. The Submission Package is already available for Dienst. Updating repositories and indexes will no longer be a manual task. There is also the Dienst Library Management Package which contains the Submission Package and some more utilities (e.g. to automatically generate additional document formats, check database integrity).

The above mentioned prospects and the need for a safe replication scheme let us think about a replication mechanism with a two-way cooperation. All sites being a mirror or master site with respect to replication have a list of their cooperating partners. The two basic problems are:

- establish and configure a new mirror
- forward changes at master site to mirror sites

The architecture is shown on Figure 1. The Replication Service functions as a part of the Library Management Unit, and exploits its procedures for index and repository updates. Replication services communicate with a new set of protocol messages which is an extension to the current Dienst protocol. For this communication techniques described in section 3.2 can be utilized. The replication service maintains the list of mirror and master partners, the list of modifications on the local master data, and manages sending and receiving update requests.



META =3D Meta server
DL-UI =3D Digital Library User Interface
DL-MUI =3D Digital Library Management User Interface

HTTP =3D Hypertext Transfer Protocol
D =3D Dienst Protocol
D' =3D Dienst augmented by Replication Protocol
PC =3D Procedure Call

Figure 1. Dienst architecture augmented by the replication service and protocol connection

The librarian interface for the Management Unit also contains tools for replication management, which include configuration of mirror updates and temporary suspension of mirrors. The Library Management Unit provides tools to the Replication Service for tasks such as collecting the repository data, creating a new repository, deleting, modifying and inserting documents and updating the index according to these changes. On servers without Library Management Unit all the above functionality must be implemented within the Replication Service.

To demonstrate the replication process the scenario of a document insertion is shown on Figure 2. The librarian inserts a new document to the DL (1). Through the interface the bibliography file and all document formats are submitted to the Library Management Unit. It inserts all data into the Repository (2), then updates the index (3) with the new bibliography file or a text version of the document, finally notifies the Replication Service about the change (4). The Replication Service collects mirror sites that are to replicate the new document. According to the configuration of the mirror relationship, mirror sites receive the new data (5). They call the Library Management Unit to perform changes (6,7,8).

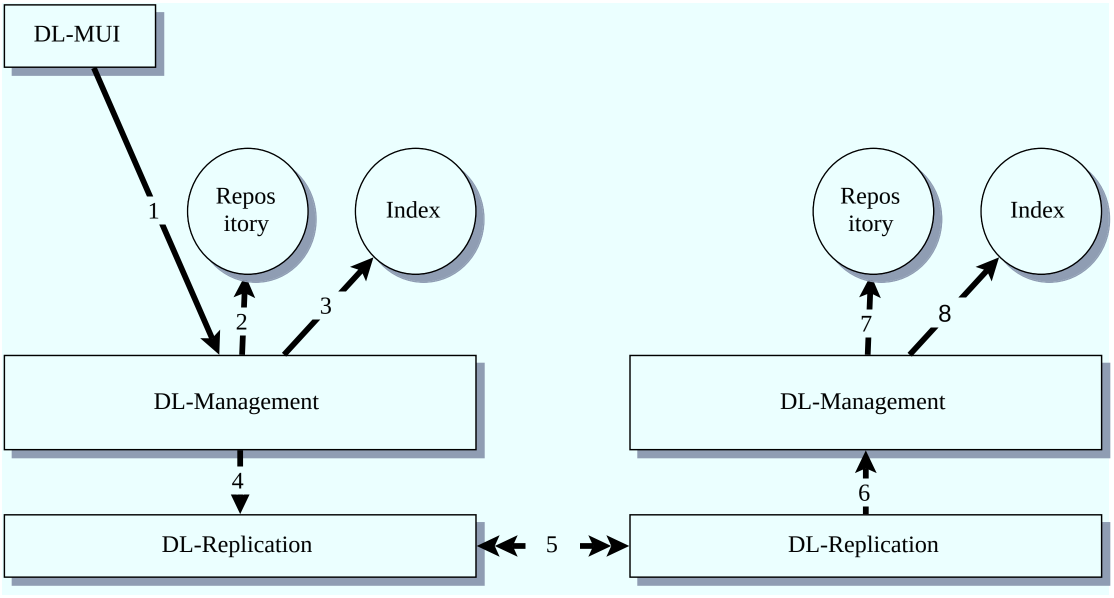


Figure 2. Replication process: the scenario of a document insertion

Summary

During the installation of Dienst server within MTA SZTAKI the problem of replication has been arisen caused by the lack of available network bandwidth to foreign countries. This investigation has led to the general examination of replication within distributed digital document libraries. The commitment of ERCIM institutions to the Dienst protocol raised the necessity of the development of general detailed architecture of Dienst. This may lead to the creation of abstract Reference Model of distributed digital document libraries as well.

Although the problem of replication can be partially solved by similar mechanisms to the available (WWW) mirror software packages, more sophisticated architecture and mirror algorithms are needed. This work aims at the research and development of new Dienst architecture augmented by replication service.

References

[BKTJ92] H.E. Bal, M.F. Kaashoek, A.S. Tanenbaum, J. Jansen: Replication Techniques for Speeding up Parallel Applications on Distributed Systems, Concurrency Practice & Experience,Vol. 4, No. 5, August 1992.

[Comer] Douglas E. Comer: Internetworking with TCP/IP, Volume I., Prentice Hall International Editions, 1991

[DavLag94] Jim Davis, Carl Lagoze: "Drop-in" publishing with the World Wide Web.
URL: http://www.ncsa.uiuc.edu/SDG/IT94/Proceeding/Pub/davis/davis-lagoze.html

[Dienst] The Dienst protocol and server,
URL: http://cs-tr.cs.cornell.edu/info/server.html

[DienstProt] Dienst protocols Release 3.5 Draft,
URL http://cs-tr.cs.cornell.edu/info/protocol3.html

[DigLib] Digital Libraries, Communications of the ACM, April 1995 Vol. 38, No. 4

[Ford] Andrew Ford: Spinning the Web, How to provide Information on the Internet, International Thomson Publishing, 1995

[HTML] Hypertext Markup Language,
URL: http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html

[HTTP] Hypertext Transfer Protocol,
URL: http://www.w3.org/hypertext/WWW/Protocols/Overview.html

[Infmed] Informedia Digital Video Library,
URL: http://fuzine.mt.cs.cmu.edu/im/informedia.html

[JabWed90] S. Jablonski, H. Wedekind: Logical Foundation of Data Replication. Proceedings of Database Systems of the 90s, Müggelsee, Berlin, November 1990, Springer Verlag

[RFC1807] Danny Cohen, Rebecca Lasher: A Format for Bibliographic Records, RFC-1807,
URL: ftp://nic.merit.edu/documents/rfc/rfc1807.txt

[Stan] Stanford Digital Library Project,
URL: http://www-diglib.stanford.edu/diglib/

[Umich] University of Michigan Digital Library Project,
URL: http://http2.sils.umich.edu/UMDL/HomePage.html

[URL] WWW Names and Addresses, URIs, URLs, URNs,
URL: http://www.w3.org/hypertext/WWW/Addressing/Addressing.html