Dienst Architecture Issues with Respect to Replication

László Kovács, András Micsik

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

1. Introduction

Nowadays digital document libraries are under intensive research and development [1,2]. There are numerous examples of such R&D projects [5]. In these projects the search, retrieval and homogeneous user interface problems are mostly discussed. Little attention is 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 [6] 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.

First the basic structure of cooperative digital document libraries is described, which is a step towards the development of general Reference Model of Digital Document Libraries. After presenting one of the central concepts of this area, the replication of digital data documents is discussed.

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, 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 the distribution of well established software and media libraries is traditionally applied as a special form of replication (mirroring).

2. Replication within Distributed Digital Document Libraries

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.

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 kinds of 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

To date digital document libraries are mostly based on the World Wide Web service. The generalization of Web mirroring therefore is a way towards the development of universal replication services.

3. An Architecture for Distributed Digital Document Library: Dienst

Currently the Dienst protocol is the most promising protocol for communication with distributed digital libraries [5]. 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 [7].

The protocol provides an object-oriented interface to a document model, which allows a user to access complete documents or subparts of multiple document formats. Dienst protocol messages are embedded within HTTP [4], thus using a World Wide Web browser all services of Dienst protocol are accessible.
Distributed digital document library provides four types of independent library services :

user interface services: connect the user to the library and provide data in human readable format
repository services: store and retrieval of digital document
index services: search and maintain index and catalogue data
meta service: provides central directory of location of all other services

Dienst protocol doesn't provide a service for replicating index, data and user interface. This paper suggests a new kind of service called **replication service** which provides mirroring of digital documents, indexes and user interface data.

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. The 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.

3.1 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 have 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 [3]. Imagemap files stay hidden inside the repository data and map coordinates on an image to other images or files, 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 cannot naturally provide full featured HTML documents such as HTML documents containing in-line images and/or split into several files.

Except for these files all other files can be reproduced at another network site, although HTTP-based mirroring software needs 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. From this moment a Dienst server can locally serve requests 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 of advertising a publisher at different sites, but it cannot differentiate between master and mirror sites.

3.2 Dienst Replication Service

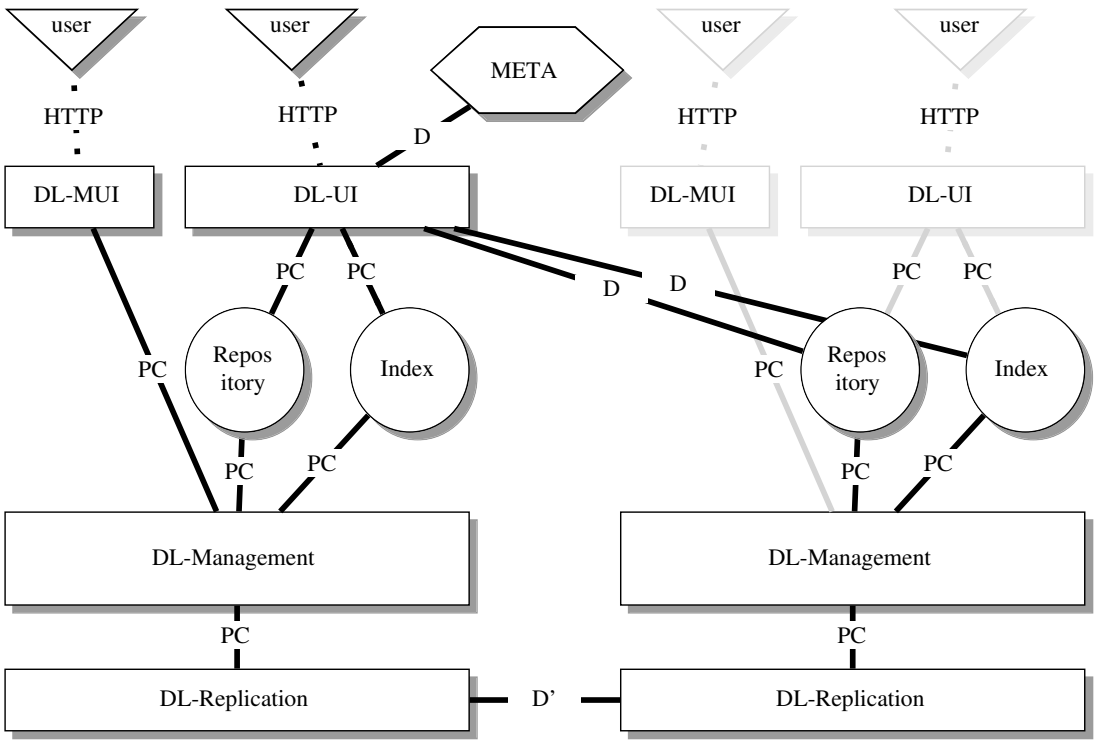
The insertion of new publications will soon be automatized. The Submission Package is already available for Dienst. Updating repositories and indexes will no longer be a manual task. There is also the Dienst Library Management Package which contains the Submission Package and some more utilities (e.g. 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 in 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. 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 = Meta server
DL-UI = Digital Library User Interface
DL-MUI = Digital Library Management User Interface
HTTP = Hypertext Transfer Protocol
D = Dienst Protocol
D' = Dienst augmented by Replication Protocol
PC = 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 the Library Management Unit all the above functionalities must be implemented within the Replication Service.

To demonstrate the replication process the scenario of a document insertion is shown in 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, and 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).

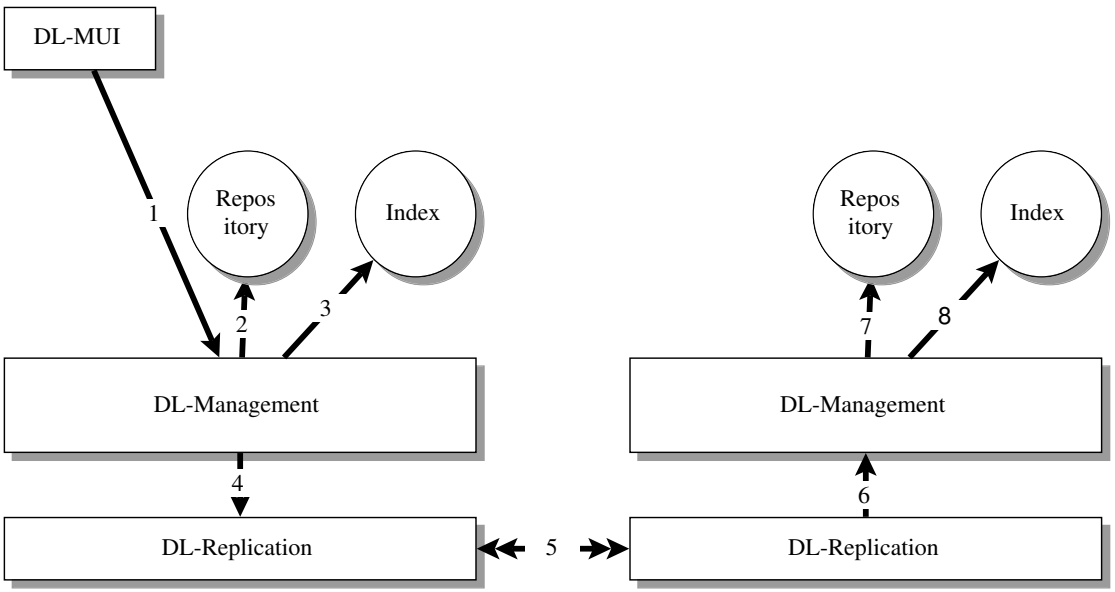


Figure 2 Replication process: the scenario of a document insertion

References

[1] University of Michigan Digital Library Project, URL: http://http2.sils.umich.edu/UMDL/HomePage.html
[2] The Dienst protocol and server, URL: http://cs-tr.cs.cornell.edu/info/server.html
[3] HTML, URL: http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html
[4] HTTP, URL: http://www.w3.org/hypertext/WWW/Protocols/Overview.html
[5] Dienst protocols Release 3.5 Draft, URL http://cs-tr.cs.cornell.edu/info/protocol3.html
[6] Danny Cohen, A Format for E-mailing Bibliographic Records, RFC-1357, URL: ftp://nic.merit.edu/documents/rfc/rfc1357.txt
[7] 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
[8] Andrew Ford: Spinning the Web, How to provide Information on the Internet, International Thomson Publishing, 1995
[9] WWW Names and Addresses, URIs, URLs, URNs, URL: http://www.w3.org/hypertext/WWW/Addressing/Addressing.html