

FLYINGDOC: An Architecture for Distributed, User-friendly, and Personalized Information Systems

Ilvio Bruder, Andre Zeitz, Holger Meyer, Birger Hänsel, and Andreas Heuer
University of Rostock, Germany
{ilr, zeitz, hme, bigmac, heuer}@informatik.uni-rostock.de

The need for personal information management using distributed, user-friendly, and personalized document management systems is obvious. State of the art document management systems such as digital libraries provide support for the whole document lifecycle. To enhance such document management systems to get a personalized, distributed and user-friendly information system we present techniques for a simple import of collections, documents, and data, for generic and concrete data modeling, replication, and, personalization. These techniques were employed for the implementation of a personal conference assistant which was used for the first time at the VLDB conference 2003 in Berlin, Germany.

Architecture Our Client-Server architecture provides an information server with different services and different kinds of clients. The most interesting type of clients are those using their own information system. Each contains a personal document management application for managing local, user-supplied, or downloaded documents. There are also tools and interfaces for connecting and using the services of the server. These services comprise a distribution and replication service, a collection integration service, a data management unit, and, a query processing service.

Modeling and Import Aspects Our data model is a trade-off between a pure generic approach and a specialized approach. A pure generic model would consist of very few data types or maybe of only one data type that is able to represent every kind of data. This kind of modelling leads to large and degenerated data because the structures would be quite large and hard to query. The other way is to build a specialized model that consists of a quite large number of data types. This would be easy to use but you need to adapt it every time you would like to use it for another domain.

The import service is used for loading a collection. One main feature of the system is user-friendliness and a simple but effective management of collections. Therefore we provide import declarations with very simple rules. Additionally, we try to recognize metadata using lexica. Management-relevant data are also structure data and layout data. Such data are used for managing collections and

documents, mainly for modeling internal data and for providing standard query forms and representation patterns.

Replication and Personalization Replication is a technique to overcome dysfunction in mobile environments. It may reduce communication and increases availability and locality during the retrieval process. In a typical digital library system three types of data are replicated: metadata, index information (access paths), and the documents themselves. Besides that, system functionality and query processing, e.g., representation patterns and query forms are replicated and/or distributed. Our implementation replicates all metadata and index information needed on the local client. Which information is needed is derived from the user context, i.e., the tasks the user has to fulfill, the locations the user will be at, and the features the device in use has. The documents from retrieved result sets are cached as documents that will be used in the near future. This prediction is based on the same context information. To access and retrieve documents from a digital library system users usually have to search in document collections. In order to simplify searching and accessing documents we provide the personalization concept of *relevance* with respect to *time* and *space*. To support mobile devices retrieving documents implies replicating them to local memory and building a personal document collection.

Application To apply some of the presented concepts we implemented a conference assistant, FlyingProc, that was used at the VLDB conference 2003. Using this planner users can retrieve documents that are *relevant* with respect to the current talk, e.g., the paper and the slides of the speaker. To permit this *access by relevance* users have to select talks of the conference that they will attend. A full-text search is also provided.

To support small mobile devices, documents are replicated from the server to the mobile device. Users can access these replicates in off-line mode. This replication of documents is triggered. Thus, only requested documents are stored locally to save local memory. To provide users with up-to-date documents our implementation can detect new versions on the server.