Architecture for the Interaction and Access on Multimedia Database Systems in the Context of Mobile Environments

Guntram Flach, Nico Günther

Computer Graphics Center Rostock,
Joachim-Jungius-Str. 11, D-18059 Rostock, Germany
E-mail: {gf, nic}@rostock.zgdv.de

Abstract

In mobile multimedia systems, the mobility of users, devices, and information produces new challenges for processing globally distributed information.

In this paper, we focus our attention on new strategies for the adaptation and integration of database techniques regarding the management of multimedia data types at the particular conditions of mobile infrastructures. We believe that object-relational database systems e.g. Informix are ideal repositories on the stationary server side for next generation multimedia applications, especially in inter-networked LAN or WAN environments.

The basic idea of this paper is the integration of object-relational database technology in a Mobility Information Center (MIC) as a framework which face the common problems of mobile computing like limited bandwidth, end systems with limited resources as well as the problems in the context of multimedia data access and interaction. The major aim is to optimize the mobile data exchange by using of content based information retrieval and reduction methods as well as media specific queries within the database system in order to reduce the amount of data to be transfered. All these methods are influenced by contexts like local resources, available communication channels, and user preferences.

1. Introduction

The presented architecture is one main part of the project "Mobile Visualization" (MOVI)[1] [5,6]. This project is focused on investigations which shall enable the graphic presentation of scientific and of other, even multimedia data in a mobile environment. However, there are the well known problems of mobile computing [10] which should be considered when designing an architecture for this purpose, e.g. resource poor mobile terminals, limited bandwidth of the communication channels, location and time dependencies of data, variations in quality of transfer, and possible disconnections. The conflict of the user's needs and the restricted circumstances can be solved or minimized with several techniques [7, 14]. These techniques minimize the needed bandwidth, save the resources of the mobile system, and enable a comfortable work on mobile devices. Our architecture is designed to combine these methods and to make the best use of them.

Within MOVI, the following research tasks have been covered:

- Basic models,
- visualization and presentation,
- multimedia data management and access,
- data exchange,
- user interfaces,
- intelligent agents.

In the following, we focus our attention on the work packages *multimedia data management and access* as well as *data exchange*. These work packages adress the investigation and development of strategies for the efficient exchange of multimedia objects over wireless and wired networks. All strategies consider the dynamically changing ressources of mobile end systems and network resources.

One typical mobile scenario for us is a person accessing very different public or private data stored on several globally distributed *stationary data servers* (SDS) – the Infoverse – for example from a *mobile end system* (MES) using wireless networks, like GSM (Global System for Mobile Communication). The information contained in the Infoverse could have multimedia character, e.g. WWW data including audios and videos or weather and traffic data with textual descriptions and images that are managed in different database systems.

[1] This research is supported by the DFG (German Research Association) under contract: UR 61/1-3

Proceedings of the 2000 International Database Engineering and Applications Symposium (IDEAS'00)
0-7695-0789-1/00 $10.00 © 2000 IEEE

Therefore we need a suitable architecture which includes the several methods for the access, exchange, and retrieval adapted to the mobile computing environment.

At present in the field of database research, new architectures and models for multimedia database systems [1, 10, 14] are investigated and designed, especially on the base of the object-relational technology [16]. Such a object-relational database management system (ORDBMS) provides a suitable environment for using and managing multimedia information. Therefore, it must support the various multimedia data types (e.g. images, text, audio, video) in addition to providing facilities for traditional DBMS functions like database definition and creation, data retrieval, data access and organization, data independence, integrity control, data replication and concurrency support. The functions of a multimedia DBMS basicly resemble those of a traditional DBMS. However, the nature of multimedia information makes new demands. The requirements are divided into the following broad categories:

- Traditional DBMS capabilities,
- huge capacity storage management,
- information retrieval capabilities,
- media integration, composition, and presentation,
- multimedia query support,
- multimedia interface and interactivity,
- performance.

In the context of mobile environments, the question is how we can integrate known compression/conversion algorithms into the database system or how we can map the already developed request processing and information retrieval methods to server-side built-in operations. The aim is to use the new possibilities of object-relational database systems as stationary data servers by integration of combined features from document/image management, information retrieval and mobile computing. An other issue in our approach is to support several QoS levels. That means that the database system must support the specified QoS levels dependent on user defined parameters and available resources on mobile system.

The remainder of this article is organized as follows: The next section contains a brief survey of a new architecture for selection, access and exchange of information in a mobile computing environment. We present a framework on the base of a Object Bus which enables applications to access globally distributed heterogeneous multimedia information in the Infoverse and to exchange it effectively with respect to mobile resources and other parameters of mobile computing. In Section III, we motivate the need for the integration of object-relational database technology into our framework architecture. Furthermore, we describe and analyze the possibilities and restrictions of these new systems for the management of multimedia data types at the particular conditions of mobile infrastructures. Section IV provides an overview of the information retrieval process and our experimental system based on the Informix DataBlade technology. Finally, we summarize the major results of this investigation and give and outlook on future activities.

2. Object Bus Architecture

2.1 Message Exchange

The Object Bus Architecture of the Mobility Information Center (MIC) has to serve as a flexible platform for the efficient exchange of user data and applications between stationary data servers and mobile end systems. Our main concept is an object oriented approach with the *Object Bus* (OBus) as one central feature. This Object Bus serves as a transparent layer for mobile communication and is responsible for the delivery of messages.

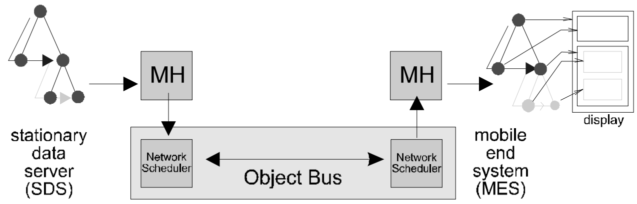

Figure 1. Traversal and exchange of objects

This object bus was extended in order to face and minimize the problems caused by limited bandwidth, end system resource limitations, and frequent disconnections. Basic components of this extended object bus are *Network Schedulers* and context-sensitive *Request/Reply caches*. They use techniques like priority- and QoS-controlled communication, transparent data compression and context-controlled caching to provide basic solutions for these problems. The second main feature is the introduction of *Message Handlers* (MH) [5] that act in place of the communicating processes when exchanging structured objects (see figure 1). They notify each other about transfer procedures and transfer the objects (see figure 2).

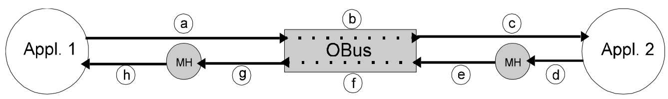

Figure 2. Communication via the Object Bus

That means, that the whole communication between two processes A and B (a) is divided into the following main steps:

(1) Applications generate requests that are handed over to the local Object Bus for transportation (a).
(2) The following transfer inside the Object Bus is carried out by several components (Network Scheduler, Transfer Manager) which reside between application and network layer on every computer (b).
(3) The receiver gets the message (c) and generates reply objects.
(4) The reply objects are handed over to appropriate Message Handler processes (d). The Message Handlers use knowledge about the context to enrich the reply with additional information, that is necessary for an effective scheduling. Additionally they perform operations on the reply objects like data reduction that are influenced by the resources of the mobile host. These operations are necessary in order to save the limited bandwidth of wireless communication channels.
(5) The Message Handlers on both sides transfer the objects with suitable methods via the Object Bus (e/f/g).
(6) The application receives the reply objects (h).

In the following the system components will be explained in more detail.

2.2 Network Scheduler

The Network Scheduler is a unique component on each Computer (mobile and stationary host) and the main component of the Object Bus. It is responsible for the effective transfer of messages. To achieve this aim it manages a list of all requests that it has received and of all replies that have to be send to other NS's together with their context information. Besides the scheduling with priorities and with the influence of Quality-of-service parameters also a compression of messages is initiated by the NS in order to save bandwidth. For incoming new messages the NS starts an appropriate MH that has to handle the message.

2.3 Message Handler

The Message Handler serves to exchange requests and appropriate reply objects between processes and to enable the access to single units of information. That means, that they provide additional functionality to the application to enable a transparent access to remote objects. Due to their *object specific* design and their knowledge about type and structure of the data to be transferred they are able to use type and structure dependent methods for minimizing network traffic and response times like data reduction (e.g. compression, conversion) and detail-on-demand.

2.4 Transfer Manager

The Transfer Manager caries out the lower level network functions like opening and closing of connections as well as the actual send and receive functions. Furthermore, other components like the Network Scheduler can make use of information about the last connections to a special host as a base for planning their work.

2.5 Request/Reply Cache

In the Request/Reply Cache requests and replies are stored at the local host to facilitate the immediate delivery to applications when receiving a following identical request. Due to changing resources it is necessary to hold also additional information about the current contexts in the Bus cache. That means the contexts that were used to determine the reply value will be stored together with request and reply. The Request/Reply Cache Manager is responsible for managing the cached data, that means entering data, deletion and updates by means of given strategies and registration of accesses.

2.6 Prototype: Mobile Multimedia Information Center

A prototype of the Mobile Multimedia Information Center (MIC, see figure 5) was built to demonstrate the use of our object bus architecture, the multimedia database access as well as the deployment of exchange strategies and the information retrieval.

The application consists of several modules: the user interface, a huge database holding actual multimedia data and visualization tools.

Figure 3. MIC application on a mobile client

Proceedings of the 2000 International Database Engineering and Applications Symposium (IDEAS'00)
0-7695-0789-1/00 $10.00 © 2000 IEEE

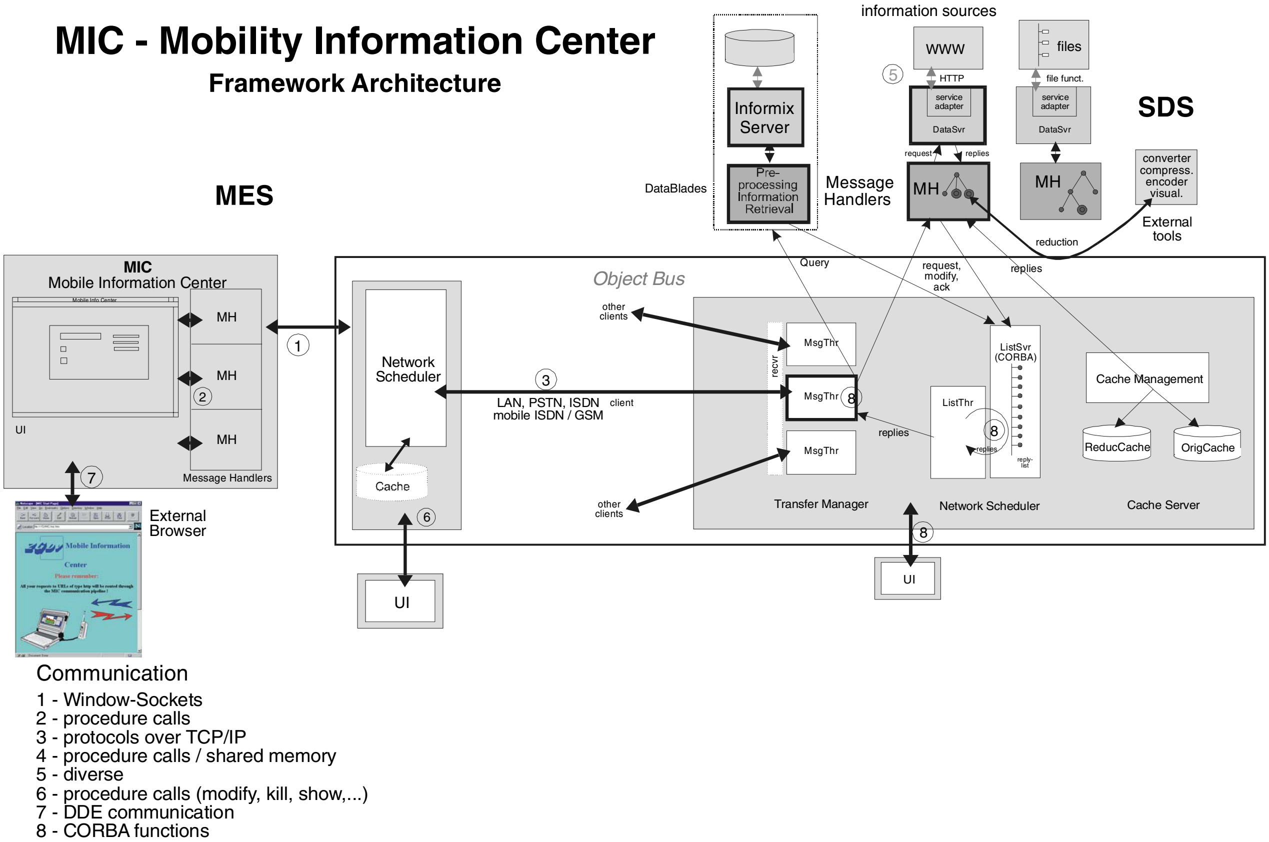

Figure 4. Overview: Components of the System Architecture on MES and SDS

All these modules are implemented as CORBA clients and servers that communicate via the OBus. The complete database is stored at the stationary multimedia support server at the office or can be accessed via this server.

The data to be transferred to the mobile host are reduced according to the resources of the mobile host (see figure 3) in order to save transfer time (and fees) and to enable the use on systems with poor resources. Due to the poor resources of the mobile terminal, we split the application: Only the user interface and the main control reside on the terminal. The computing intensive visualization and the data are located at the workstation.

More details about the components of the Object Bus Architecture and the experimental validation results of our methods and communication protocols you will find in [5, 6]. Based on this architecture, we present in the next section our approach of the integration of an object-relational database system as an extension of the Stationary Data Server (SDS) within the MoVi framework architecture.

3. ORDBMS: Challenges and Issues

At present there is a new family of object-relational DBMS (e.g. Informix , Oracle8, Universal DB2) provides all some new important extensions, especially multimedia extensions [11, 15, 16, 17]. These capabilities allow the behavior of the multimedia objects to be implement directly into the DBMS. Object-relational database systems support a number of multimedia data types - such as images, audio, or video - in their built-in system provide class hierarchies. Forinstance the Informix Dynamic Server provides extensibility in four key areas: datatypes, user defined functions, index structures and the query optimizer. Other areas Informix has addressed are aditional languages for writing extensions and server procedures, ”smarter” BLOBs for enhanced large object support, support for access to external data and the ability to create ”packages” of extensibilty as DataBlade modules.

All of these features are very important to enable the object-relational DBMS to handle a broader set of application requirements.

In the scope of the MoVi project the aim is to integrate known compression or conversion algorithms into the database system or to map the developed request processing and information retrieval methods to server-side built-in operations. So we reach the effect *to relieve the communication components of the MIC object bus architecture (e.g. Message Handlers) from this functionality* and to transfer this to the database backend side (SDS).

As an example Table 1 lists the functions included in the Image DataBlade Module, which we can use within the MIC framework for image preprocessing and data reduction based on the user preferences and the available resources on mobile end systems. As an other fact the querying and searching techniques in DBMS need to extend to information-retrieval capabilities. In information retrieval the emphasis is more on finding the objects that satisfy as closely as possible a user's query and leads so to an *reduction of the amount of information* in the case of limited bandwidth of the communication channels. By using specialised indexes for complex object retrieval we can achieve large performance increases in comparison to blob retrieval for non-indexed.

Function	Description
`ConvertImgFormat`	Converts an image from one format to another
`ConvertImgType`	Converts an image from one image type to another
`ScaleImgTo`	Scales a stored image to a specified pixel width and height
`ScaleImgBy`	Scales a stored image by the specified scaling factor
`ImgCompression`	Returns the compression scheme

Table 1: Image Foundation Functions

To support the information required in the MOVI architecture, the DBMS must also adress the general problems in distributed systems, such as distributed and parallel query processing, distributed mobile transaction management, data location transparency and data replication [4]. In addition, network issues such as limited bandwidth and network delays become important considerations, since they could adverse effects on the QoS supported (data compression and conversion).

The technical realization of the above functionality is done as a *part of the Stationary Data Server (SDS)* in the system architecture of MOVI (see figure 4).

4. Information Retrieval

Beside our data preprocessing approach (e.g. image conversion/compression, see also Table 1) we can use content based retrieval techniques on images (based on the Informix Image DataBlade) in order to reduce the amount of data which have to be transfered (see figure 5).

We believe, that the use of an *object-relational database system (e.g. Informix) for multimedia data management, data preprocessing, QoS and information retrieval will be an excellent, integrated approach* in the MOVI framework (see figure 4).

In this section, the image information retrieval process [8, 9] is briefly presented, restricted to the characteristics relevant to this paper. Then, the main features of our prototype related to the MOVI solution are descripted.

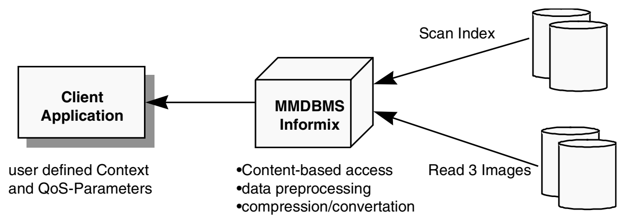

Figure 5. Query Processing

4.1 Searching for Images

When an image is stored in an indexed table, a feature extractor function records the characteristics of the image in an array of bits called a feature vector. The feature vector is indexed, providing content-based access to the images in the database table. In the case of a search for matching similar images in the database, a feature extractor creates a feature vector for a target search image (as an inside database process). The index of the database table is searched for similar vectors, the image associated with each similar vector is retrieved and a ranked list of matching images is returned. Before performing a search, the `frnet` access method calls a feature extractor function to extract features of the searched image and store them in a feature vector. To perform the search, the query optimizer uses the feature extractor function specifications to determine if an `frnet` index is present. After the `frnet` index has been identified, the `frnet` access method invokes the feature extractor function for the `frnet` index to extract the features of the search image into a feature vector. Finally, the `frnet` access method performs a neural network search for similar patterns of feature vectors contained in the `frnet` index and returns the rows that contain similar images. Figure 6 illustrates the search process.

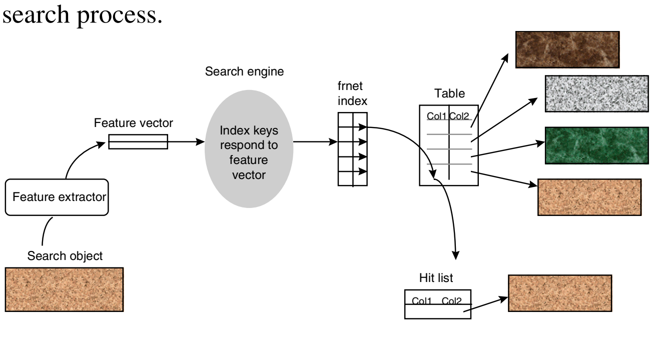

Figure 6. Image Search Process

The `Resembles` function is used to query the database. The Resembles strategy function is a part of the `IfdImgDesc_ops` operator class. The Resembles function is specified in the `WHERE` clause of an SQL `SELECT` statement To access ranking information a statement local variable is used. The following syntax illustrates the `Resembles` function and the rank parameter:

```
   SELECT *, rank FROM table_name,
        WHERE Resembles (
        indexed_column_name,
        row(
   'feature_extractor(feature_extractor_params)')
,
                 search_object,
                 search_options
        ),
        rank #REAL);
```

4.2 Prototype

As a first step we realized an experimental system to demonstrate the presented image retrieval functionality as a part of the Stationary Data Server (SDS) within the MIC framework architecture. This prototype was developed on the base of the Web- and Image-DataBlade of the Informix Dynamic Server/Universal Data Option. On this web-based User-Interface the user can choose an image from a random or default set of images in order to start the content based image retrieval.

Figure 7. User Interface: Image Retrieval

As a first example we choose the sunset-image as the given search object and get the following result list (see figure 7) of similar images, which are ranked by normalized rank factor. Furthermore the user have the possibility to specify the search mode of image retrieval (e.g. colors, textures, shapes). As an other document type we developed a prototype for information retrieval on textual documents (see figure 8).

In addition to the several information retrieval and selection strategies a reduction of delays and of used bandwidth can also be achieved by compression and conversion of media documents before transfering them. As an other aspect, we will integrate Level-of-detail-

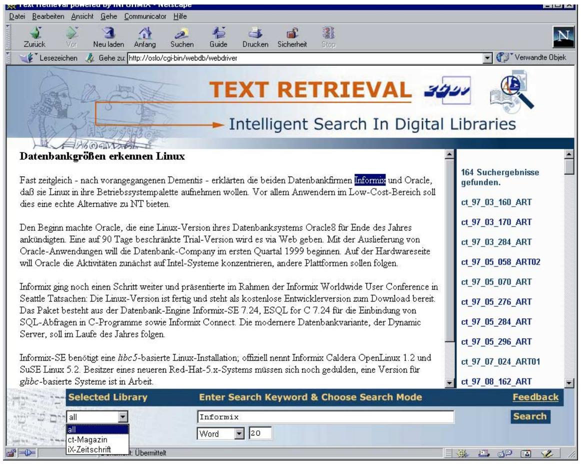

Figure 8. User Interface: Text Retrieval

methods (e.g Detail-on-demand, Progressive Refinement) directly into the database system. These preprocessing methods can be carry out as user defined functions by developed DataBlade modules.

These will be an excellent combination with the already developed client user interfaces for interaction and visualization (see figure 3) on mobile client and framework components (NS, MH) with support for the above methods.

5. Conclusion

By means of the introduced concept, a basic system architecture for mobile multimedia information access, exchange and retrieval is available, allowing applications to access to globally distributed multimedia information and to exchange it effectively with respect to mobile resources and other parameters of the global environment.

Moreover, we considered especially the new possibilities of object-relational database technology by using of content based information retrieval and reduction methods as well as media specific queries within the database system in order to reduce the amount of data to be transferd.

Until now, we could not present a complete experimental validation of this new approach, because the prototype implementation is just in progress.

In future work we will develop new DataBlade modules in order to extent our system with different multimedia datatypes (e.g. video, audio), data preprocessing methods and level-of-detail-concepts as server built-in functionality.

Furthermore, studies will be performed in order to develop a suitable transaction management [12, 13] in the context of a mobile environment. Traditional transaction mechanism and criteria have to be adjusted to accomodate the limitations of a mobile computing environment. The major aim is to develop a new approach of a context based and transaction-driven multimedia data management with support for content based and media dependent request processing.

References

[1] Aberer, K., Thimm, H., Neuhold, E.J. Multimedia Database Systems, In: *Handbook of Multimedia Computing*, Borko Furth (Editor), CRC Press, 1998

[2] Aberer K., Klass W. Multimedia and ist Impact on Database System Architectures. In *Apers, P.M.G, Blanken, H.M., Houtsma, M.A.W (Eds.). Multimedia Databases in Perspective*. Springer , 1997

[3] Apers, P.M.G, Blanken, H.M., Houtsma, M.A.W (Eds.). Multimedia Databases in Perspective. *Springer* , 1997

[4] Boll, S., Klas, W., Löhr, M., Integrated Database Services for Multimedia Presentations In S.M. Chung, editor*, Multimedia Information Storage and Management*. Kluwer Academic Publishers, Dordrecht 1996

[5] Bönigk, J., Lukas,U.v., Interactive Exchange of structured multimedia Data with mobile hosts. In *Global Communication Interactive*, published by Hanson Cooke Ltd, 1997

[6] Bönigk, J., Flach, G., System Architecture and Strategies for the Exchange of structured multimedia Data in the Context of mobile Visualization. In Proc. *5th Intl. Workshop on Mobile Multimedia Communication (MoMuc)*, October 1998

[7] Bellmann, B., Böhmak, W., Kümmel, S., Schill, A., System Support for Mobile Distributed Applications. *IEEE Workshop on Services in Distributed and Networked Environments (SDNE),* Vancouver, June 1995, pp.124-131

[8] Bechhofer, S., Goble C., Classification Based Navigation and Retrieval for Picture Archives. In *Proc. 8th IFIP Conference on Data Semantics (DS-8)*, January 1999, Kluwer Academic Publishers

[9] Baeza-Yates R., Ribiero-Neto B., Modern Information Retrieval. *Addison Wesley*, Harlow, England, 1999

[10] Forman, G.H., Zahorjan, J., The Challenges of Mobile Computing. *IEEE Computer,* 27(4), April 1994

[11] Narasimhalu, A.D.: Multimedia databases, In *Multimedia Systems*, Springer, 1996

[12] Narasayya, V.R.: Distributed transactions in a Mobile Computing System, *First IEEE Workshop on Mobile Computing Systems and Applications*, 1994

[13] Pitoura, E.; Bhargava, B.: Revising Transaction Concepts for Mobile Computing, , *First IEEE Workshop on Mobile Computing Systems and Applications*, 1994

[14] Satyanarayanan, M., Noble, B., Kumar, P., Price, M.. Application-aware adaptation for mobile computing. *Operating Systems* Review, 29(1) January 1995

[15] Subrahmanian, V.S.: Principles of Multimedia Database Systems, *Morgan Kaufmann Publishers Inc*., 1998

[16] Stonebraker, M., Object-Relational DBMSs – The Next Great Wave, Morgan Kaufmann, 1996

[17] Westermann, U., Klas, W., Architecture of a DataBlade Module for the Integrated Management of Multimedia Assets, *First Int. Workshop on Multimedia Intelligent Storage and Retrieval and Retrieval Mangement (MISRM),* October 1999