Position Statement

Supporting Multimedia Presentations from the Perspective of Multimedia Database Systems

Wolfgang Klas
GMD-IPSI, Darmstadt, Germany
klas @ darmstadt.gmd.de

1. Introduction and Motivation

Applications very often have to handle multimedia data. They have to process, manipulate, present, audio, video, graphics, images, text, and combinations of these media types. One characteristics is that multimedia information often constitutes a huge amount of data. For example, 500 pages of text correspond to about 1 MB data, 500 images of JPEG compressed data amount in about 0.2 GB, 5 minutes of CD-DA quality audio amount in about 53 MB, 5 minutes of PAL-coded video corresponds to about 6.6 GB, and 5 minutes HDTV-coded video is in the order of 30 GB. For that very reason it is often necessary to have the multimedia data managed by a dedicated multimedia storage system.

There are several approaches to the management of multimedia data by means of a database system [1]. One can distinguish between approaches which exploit traditional database technology without any adaptations or extensions of the conventional database system services and approaches which extend conventional database technology in order to provide more suitable support for the handling of multimedia data.

Examples for the first type of approaches are

- managing references to external files containing multimedia data
- storing multimedia data uninterpreted in the database by using, e.g., BLOBS
- making use of external functions not supported by the DBMS
- managing multimedia data by means user (or application) defined types or classes.

All these approaches have serious problems because they do not allow to guarantee consistent management of the data, to support synchronization of time-dependent data, to use efficient media-specific database operations, and to efficiently store and retrieve time-dependent data.

Approaches which extend conventional database system technology try to overcome these limitations. Such extensions, for example, address continuous data management (e.g., [4]), modeling primitives for the handling of continuous data (e.g., [2], that is, built-in data types, constructors of temporal composition of continuous data, description of the content of multimedia data, etc., and synchronization support. These extensions mainly focus on the internal database system support for multimedia data.

Another issue in the context of extending conventional database technology is how to support the presentation of multimedia information by a database system. Presentation in this context means both the visual presentation of multimedia data to the end user

and the delivery of multimedia data to application software according to overall quality of service parameters relevant for the presentation.

Visual presentation usually is required in ad hoc querying situations. That is, the end user can query multimedia data and gets presented the result data of an ad hoc query. For example, a user may get presented the video segments, the audio data stream, or some portion of an animation or simulation he retrieved from the database. Furthermore, the retrieval process itself may be interactive. That is, the user may interact during the query processing in order to determine specific selection criteria to be used for the subsequent query processing.

Delivery of multimedia data at the interface between the database system and some application module usually is needed when multimedia data retrieved from the system is further processed by an application including the presentation of the data by application specific presentation tools.

A database management system should provide basic, adequate support for the two types of presentations mentioned above. In order to provide such presentation support one needs to be able to specify, model, and process preorchestrated presentations. In the following we will discuss some basic features and architectural issues with respect to this problem.

2. Supporting Multimedia Presentations by a Database System

Supporting multimedia presentations by a database requires (1) the description and modelling of preorchestrated presentations and (2) the execution of the presentations.

Specification and Modelling of Presentations

In order to be able to describe a presentation by means of its composition of individual media types and in terms of database modelling techniques, a data model needs to provide appropriate modelling primitives. In [2] we introduced a minimal set of additional modelling primitives which allow to describe presentations. The concepts are *schedules* which describe presentations, *events* which trigger individual presentation activities, and a set of built-in *data types* for multimedia data. The concepts are homogeneously integrated into the object-oriented database modeling language VML [6]. The execution model for schedules differs from the execution model for database methods because it allows to execute schedules in parallel and to synchronize schedule executions with respect to the occurance of events. The following example of a description of a presentation illustrates the concepts. Let us assume that one wants to describe a simple cinema presentation like the following one.

> ***Begin*** *a cinema* ***presentation***
> *with 30 minutes advertisement.*
>
> ***Start*** *with the advertisement.*
>
> ***Pause*** *with advertisement for selling refreshments,*
> ***playing back*** *some music in the* ***meantime****.*

***After** the music*
begin showing** the movie by **playing back** the video **and
***playing back** the selected, e.g., English, sound track.*

After** the movie **terminated close** cinema **presentation

The model of this presentation in terms of extended VML [2] reads as follows.

```
CLASS cinema_presentation
PROPERTIES  advertisement: Slide;
            music: Audio;
            video: Video;
            audio: Audio;
IMPLEMENTATION
      SCHEDULE show(adtime: INT);
            {AT START SELF          advertisement->display();
            AT START SELF+adtime    {advertisement->stop_display();
                                    music->play();}
            AT END  music           video->play();
            AT END  music           audio->play("english");
            AT END video            return; }
END;
```

Although the extensions of VML allow also for the modeling of *temporal compositions* [2] (e.g., sequential composition of two video streams, parallel composition of an audio and a video stream) and of *user interaction* it does not support fine-grained synchronization of continuous data streams. Fine-grained synchronization like lip synchronization needs to be realized on the level of the basic playout operations, e.g., playing back an audio and playing back a video stream.

Execution Support for Presentation Realization

The basic scenario for the execution of a presentation is as follows. The overall control of a presentation is with a playout management component integrated into the database system [3]. The playout management component consists of a controller for controlling and synchronizing individual presentation tasks. Each presentation task is performed by specific Single-Media-Presenters which are capable of presenting a single media type. In addition, there exist specific interaction components which realize user interactions. The playout management component configures an appropriate set of Single-Media-Presenters and interaction components, determines the relevant presentation parameters by means of QoS parameters, initializes the configuration, and finally starts the presentation by generating events for the individual Single-Media-Presenters. The Single-Media-Presenters get the multimedia data to be presented directly from underlying database system modules. In the case of time-dependent data, Single-Media-Presenters request the data from the continuous object manager. In case of conventional (e.g., alphanumeric) data, Single-Media-Presenters request data using the conventional services of the database system.

For example, in the sample presentation mentioned above, the playout management will configure an Image Presenter, an Audio Presenter, and a Video Presenter. In addition, it will configure an interaction component which may be needed to continue the presentation after refreshments have been sold (in the case that one does not want

to wait until the music play back finished). The Video Presenter and the Audio Presenter will request the video and audio data from a continuous object manager. The behavior of the Single-Media-Presenters is under the control of the Playout Manager.

The playout management is based on self-adaptive and reactive strategies in order to allow for adaptations of the presentation with respect to changes of the system behavior and deviations of QoS parameters [5]. That is, the playout manager gets informed about deviations of parameters relevant for the presentation from the continuous object manager, from Single-Media Presenters, and also from the end user. It reacts on this and adapts the entire presentation by adapting the requirements on the behavior of the involved components.

An interesting open issue is whether one can define standardized interfaces for Single-Media Presenters and interaction components which would allow to use arbitrary, customized presenters and interaction components available by, e.g., libraries in this framework. Standardized interfaces for Single-Media Presenters would contribute to the openness of such a system architecture.

Note that, in the case of non-visual presentation, Single-Media-Presenters can be components which just deliver data to other application modules which process multimedia data.

3. References

[1] Wolfgang Klas and Karl Aberer. Multimedia Applications and their Impact on Database Architecture. To appear in: P.Apers, H.Blanken (Eds.): *Multimedia Database Perspectives* Academic Press.

[2] Karl Aberer and Wolfgang Klas. Supporting Temporal Multimedia Operations in Object-Oriented Database Systems. In: *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Boston, May 14-19, 1994, pp. 352-361. Los Alamitos, CA, USA. IEEE 1994

[3] Heiko Thimm and Wolfgang Klas. Playout Management - An Integrated Service of a Multimedia Database Management System. *Procceedings of the First Internatioal Workshop on Multimedia Database Management Systems,* Blue Mountain Lake, NY, August 28-30, 1995. IEEE Computer Society Press, 1995.

[4] Frank Moser, Achim Kraiss, and Wolfgang Klas. L/MRP: A buffer management strategy for interactive continuous data flows in a multimedia DBMS. To appear in: *Procceedings of the 1995 International Conference on Very Large Databases (VLDB)*, USA, Morgan Kaufmann, 1995.

[5] Heiko Thimm and Wolfgang Klas. Reactive Playout Management - Adapting Multimedia Presentations to Contradictory Constraints. Technical Report No. 916, GMD, St. Augustin, Germany, 1995

[6] Wolfgang Klas, Karl Aberer, and Erich Neuhold. Object-Oriented Modelling for Hypermedia Systems Using the VODAK Modelling Language (VML). In: Dogac, A.; Ozsu, T.; Birilis, A.; Sellis, T. (Eds.): *Advances in Object-Oriented Database Systems*: NATO ASI Series F, pp. 389-433. Berlin u.a. Springer 1994