SMOOTH – A Distributed Multimedia Database System

H.Kosch, L.Böszörményi, A.Bachlechner, C.Hanin, C.Hofbauer, M.Lang, C.Riedler, R. Tusch

Institute of Information Technology, University Klagenfurt
Universitätsstr. 35
Klagenfurt
Austria
harald.kosch@itec.uni-klu.ac.at

Introduction and Main Contributions

Providing mechanisms that allow the user to retrieve desired multimedia information by their semantic content is now an important issue in multimedia databases [1]. However, current prototypes (e.g. Oracle 8i *inter*Media and Informix Datablade Modules) index mostly only low-level features of multimedia objects. Therefore special techniques are needed for semantic indexing and retrieval of multimedia objects.
In this context we present the SMOOTH system, a prototype of a distributed multimedia database system. It implements an integrated querying, annotating, and navigating framework relying on a generic video indexing model. The framework allows the structuring of videos into logical and physical units, and the annotation of these units by typed semantic objects. An index-database stores these structural and semantic information. We provide further a clear concept for capturing and querying the semantic content of multimedia objects, their correlation with low-level objects, as well as their spatio-temporal relationships. The main contributions of SMOOTH are :

1) Integrated Model for Low- and High-Level Video Indexing

SMOOTH relies on an integrated model for low- and high-level video indexing. This model takes advantage of related approaches (see the overview in [2]) and extends them by the introduction of a) means for structuring video steams and b) genericity in the indexing process. The core of the index model defines base classes for an indexing system, while application specific classes are added by declaring subclasses (content classes) of the base classes.

Proceedings of the 27th VLDB Conference, Roma, Italy, 2001

We share with the Multimedia Description Schemes (MDS) of MPEG-7 [3] the manner how to extend base classes to a specific application domain, the general purpose is, however, different from that of MPEG-7. The main goal of MPEG-7 is the standardization of descriptions for communications while we are focusing on modeling of video data supported by a meta-database.

2) Domain Transparency

'Domain Transparency' means that the client software automatically adapts to (correct) extensions made by a new application domain to the base classes. Thus, the GUI of our client implementation is built dynamically from the index database. A special alias table allows the display of the menu-points in different languages.

3) Cache for SQLResults

'Domain Transparency' could result in high traffic network with the database, as every attribute name, label of the GUI ought to be read out of the database, if no supplemental measures were undertaken. In order to manage this problem, we have introduced a cache for attribute names and aliases (i.e. the labels, menu items of the GUI). This reduces the amount of exchanged SQLResults considerably.

4) Integrated Querier, Annotator, Navigator

The client integrates a querier, annotator and navigator interface in a common framework and defines a smooth interaction of the different interfaces in order to allow an effective working with the video material. It realizes an **open architecture** in a way that it enables the integration of existing tools during the video indexing process, e.g. automatic and manual methods for segmenting video streams or methods for extracting and tracking low-level visual objects.

Brief Overview over the Framework

The SMOOTH distributed multimedia database system integrates a query, navigation, and annotation framework for video media material driven by the content of an index database. This index database implements the semantic and structural part of the integrated and general indexing model presented in [4].

The general systems' architecture of SMOOTH consists of a video server which provides selective access to the physical video streams. We currently use the Oracle Video Server with the supported protocol types UDP and RTP. The index database contains the basic set of high-level content classes that are *events*, *objects*, *persons* and *locations*. These classes are subclasses of a general *ContentObject* which may refer to a low-level motion object describing its spatio-temporal characteristics. The model provides means for segmenting video streams in different granularities, such as: *shot*, *scene* and *sequence*.

The functionality of the framework is described throughout the example of an implementation where the base classes of the video indexing model have been extended with a rich set of soccer-specific content classes. The soccer application introduces *goal, penalty, shot, corner, pass, freekick, chance* and *foul* as subclasses of *event*. These events involve one or several *players* (subclass of *person*) and occur in one *match* which is a subclass of *object*. All matches involve two teams. *Team* is also a subclass of *object*. These classes are related to physical or logical video units. Further implemented application examples are medical operations and tele-teaching (see the SMOOTH homepage at *http://www.itec.uni-klu.ac.at/~ smooth*).

Querier, Presentator and Annotator

The JAVA based client provides means to **annotate**, **query** and **navigate** through the video material using the graphical library Swing.
The **annotator** allows the structuring of the video into segments and the annotation of high-level content objects. The **querier** follows a text-based, structured query specification technique. It enables the definition of video queries by specifying conditions on content objects. The query layout follows the pre-categorization of the content into the base classes *events*, *objects*, *persons* and *locations* . Once a base class has been chosen (e.g. *persons*) for the query, the 'Domain Transparency' comes into operation, i.e. the querier[1] builds interfaces dynamically from the specific application part of the index database with the help of the meta-primitives of the JDBC. Thus, the querier presents first all subclasses of *person* by following the inheritance hierarchy in the database. Once a subclass (e.g. *player*) is selected, the user may specify conditions on the attributes of the *player* table from the index database, e.g. name, position, dressnumber.
Example Query: *'Find all video shots, where the player L. Matthäus scores a goal after a pass over 30m'*
Figure 1 shows a snapshot for this query at the moment where the user specifies the two conditions ('AND' combined), the first one on *Person* and the second one on the two events: *goal* and *pass*. The temporal relationship between the two events *goal* and *pass* is entered in the 'Event' dialog (central in figure 1) which opens after the specification of the desired event subclass. Temporal constraints are specified by pairs of events with the declaration of a maximal allowable intermission time between them (set in the 'Interval' menu).

The results of a query are presented in a compact form (server address, length, video description etc.). In the **presentator**, the user can compose presentations (with temporal constraints) from the query results. The **navigator** allows the navigation through the contents of the index database.
The client implementation contains an abstract class which defines all methods a database manager has to provide for the requirements of the client. We provide at time an implementation for the *Oracle8i* DBMS, migrations to other DBMS supporting the JDBC API 2.0 is possible.

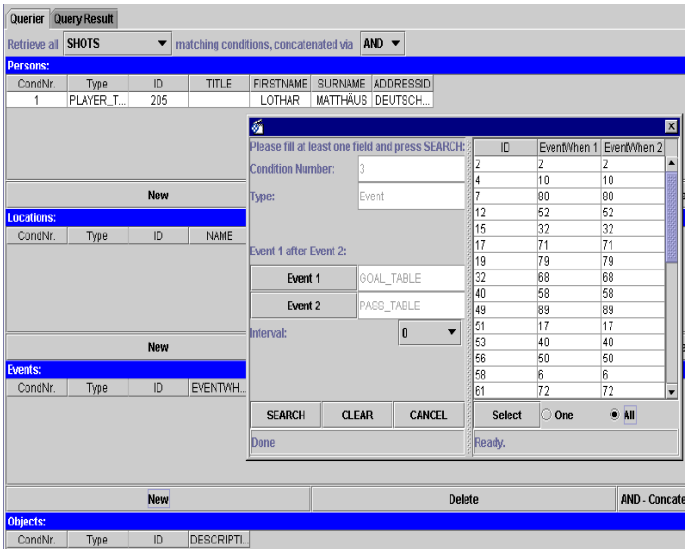

Figure 1: *Extract of the Query Dialog at the moment of the specification of the example query. Central : 'Event' Dialog and specification of temporal constraints*

Further information on SMOOTH can be obtained at *http://www.itec.uni-klu.ac.at/~ smooth*.

References

[1] S.-C. Chen, R.L. Kashyap, and A. Ghafoor. *Semantic Models for Multimedia Database Searching and Browsing*. Kluwer, 2000.

[2] H. Jiang, D. Montesi, and A. K. Elmagarmid. Content-based Access to Video Databases. *Multimedia Tools and Applications*, 9(3):227–249, November 1999.

[3] José M. Martínez. *Overview of the MPEG-7 Standard (v5.0)*. ISO/IEC JTC1/SC29/WG11 N4031, March 2001. www.cselt.it/mpeg/standards/mpeg-7/mpeg-7.html.

[4] R. Tusch, H. Kosch, and L. Böszörményi. VideX: An integrated generic video indexing approach. In *Proceedings of the ACM Multimedia Conference*, pages 448–451, Los Angeles, USA, Oct.-Nov. 2000.

[1] Also the annotator, presentator and navigator rely on this principle.