# The Mirror MMDBMS architecture

Arjen P. de Vries  Mark G.L.M. van Doorn  Henk M. Blanken  Peter M.G. Apers

Centre for Telematics and Information Technology
University of Twente
The Netherlands

## 1 Introduction

Handling large collections of digitized multimedia data, usually referred to as multimedia digital libraries, is a major challenge for information technology. The Mirror DBMS is a research database system that is developed to better understand the kind of data management that is required in the context of multimedia digital libraries (see also URL http://www.cs.utwente.nl/~arjen/mmdb.html). Its main features are an integrated approach to both content management and (traditional) structured data management, and the implementation of an extensible object-oriented logical data model on a binary relational physical data model. The focus of this work is aimed at design for scalability.

## 2 Query processing

The query facilities of the Mirror DBMS rely on the **Moa Object Algebra** [BWK98]. Moa constitutes an object data model and query algebra, designed to be used at the logical level of a DBMS. It has been implemented on top of the Monet extensible database management system. The Moa data model is based on the principle of 'structural object-orientation'. Structures, such as tuple and (multi-)set, define complex data types out of the simple base types. The base types, such as integer and string, are inherited from the underlying physical database. This way, Moa introduces the notion of **data independence** into the world of object-oriented databases: the translation from the logical data model into a different physical model (Monet supports a binary relational data model) provides an excellent basis for algebraic query optimization, and allows often for set-at-a-time processing of complex query expressions.

The Moa kernel provides the tuple (or record) structure, and the multi-set structure. The resulting data model is equivalent to what is generally known as the $NF^2$ algebra. However, Moa object algebra is more than 'just' an implementation of $NF^2$ algebra. It is an *open* complex object system, supporting extensibility of structures. Thus, new structures can be added to the system, similar to the well-known principle of base type extensibility in object-relational database systems. Obviously, this enables the definition of new generic structures, such as lists. A more interesting use for structural extensibility is however the definition of *domain specific structures*.

## 3 Content management

The prototype implementation of our database system demonstrates the particular application of Moa's extensibility in the domain of multimedia information retrieval. An information retrieval (IR) model consists of three parts: a document and query representation scheme, a ranking function which determines to which extent a document is relevant to a query, and a query formulation model [WY95]. Documents and queries are usually represented by its terms. The `CONTREP` Moa structure supports the ranking scheme known as the inference network retrieval model. This retrieval model is the basis of the successful IR system InQuery. It allows flexible modeling of the combination of evidence originating from different sources. We have adapted this text retrieval model to handle also multimedia information retrieval [dV98]. New structures in Moa, supported by new probabilistic operators at the physical level, provide an efficient implementation of the inference network retrieval model.

To illustrate the use of these extensions, we model a traditional digital library of (manually) annotated images using these Moa structures. Assume that an image is identified by its URL, and its textual annotation is indexed using the inference network retrieval model:

```
define TraditionalImgLib as
```



**Proceedings of the 25th VLDB Conference, Edinburgh, Scotland, 1999.**

```
SET<
  TUPLE<
    Atomic<URL>:    source,
    CONTREP<Text>: annotation
>>;
```

Ranking the images with respect to a query is then performed with the following query, in which `query` refers to a set of query terms, and `stats` is a structure that represents global statistics of the whole collection:[1]

```
map[sum(THIS)](
  map[getBL(THIS.annotation,
    query, stats)]( TraditionalImgLib ));
```

Because these query expressions can be combined with 'normal' relational operators (such as `select` or `join`), the resulting system is an efficient integration of information and data retrieval. This way, it is possible to refer to both structure and content of multimedia data in a single query (see also [dVW99]).

# 4 Distributed architecture

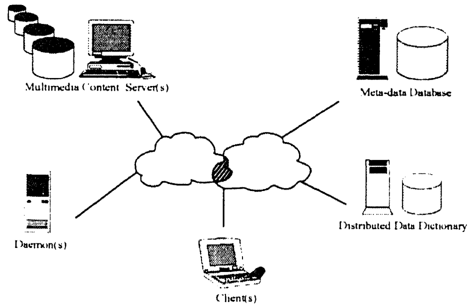


Figure 1: The distributed architecture

Another aspect of our design is related to more practical issues for the creation and maintenance of a multimedia digital library. A digital library involves several more or less independent parties, including human annotators, software to extract meta data automatically, and owners of multimedia footage [dVEK98]. Hence, we believe that a digital library can only be a success if it follows the model of the web. We use an open distributed architecture instead of a monolithic database system, cf. figure 1 [dVB98]. The notion of a 'daemon' abstracts from the various techniques for meta data extraction and query formulation. Using CORBA, we allow distribution of operations, establishing independence between the management of meta data and the parties that create these meta data.

# 5 The demo system

In the Mirror architecture, the retrieval *application* is not integrated in the database system itself, unlike in most 'multimedia databases' found in literature. The Mirror DBMS provides the basic functionality for probabilistic inference, multimedia data types, and feature extraction techniques, just like traditional database systems provide the basic functionality to build administrative applications. In this section, we describe an example image retrieval application, using the functionality provided by the Mirror DBMS. The underlying philosophy of this application has been inspired by theories from cognitive psychology, in particular Paivio's dual coding theory [EK95]. Aspects of its design are similar in spirit to both the Viper [SMMR99] and the FourEyes [MP97] image retrieval systems.

## 5.1 Our prototype environment

The digital library constructed for the demo consists of images collected by a simple web robot. Some of the images in the library are annotated with text. One of the daemons segments the images. Several feature extraction daemons independently create feature representations of the image segments. At the moment of writing, we have implemented two color histogram daemons. In addition, we use the four reference implementations of texture algorithms provided by the MeasTex framework (see URL http://www.cssip.elec.uq.edu.au/~guy/meastex/meastex.html). These feature spaces are then clustered using the public domain clustering package AutoClass [CS95]. Furthermore, we have thesaurus daemons that are interactively used during query formulation. The Mirror DBMS implements the meta data database (see figure 1), which contains the content representations. The media server is a web server.

## 5.2 The example application

The data model of the image library can be specified by the application programmer as:

```
define ImageLibrary as
SET<
  TUPLE<
    Atomic<URL>:    source,
    Atomic<Text>:   annotation,
    Atomic<Image>: image
>>;
```

Next, the daemons in the prototype environment start to work on this schema. Like the example in section 3, the text annotations are indexed and represented as CONTREP structures. The images are segmented, and feature representations are extracted from the segments, creating the following (internal) intermediate schema:

```
SET<
  TUPLE<
    Atomic<URL>: source,
    CONTREP<Text>: annotation,
```

[1]For further details about the query language and the structural extensions, refer to [dV98] and [dVW99]

```
    SET<
      TUPLE<
        Atomic< Image >:  segment,
        Atomic< Vector >: RGB,
        Atomic< Vector >: Gabor,
        ...
      >
    >: image_segments
>>;
```

The feature spaces are clustered with AutoClass, to obtain a representation of the image content that can be queried using the CONTREP structure. We further use the identified clusters as if they are words in text retrieval; they become the basic blocks of 'meaning' for multimedia information retrieval. This results in the following internal schema, which corresponds to the original ImageLibrary schema:

```
define ImageLibraryInternal as
SET<
  TUPLE<
    Atomic<URL>: source,
    CONTREP<Text>: annotation,
    CONTREP<Image>: image
>>;
```

Of course, the clusters in the image content representation (such as 'gabor_21') are not suited for interaction with the users of the digital library. Therefore, we automatically construct a thesaurus, associating words in the textual annotations to the clusters in the image content representation. An interesting aspect of this approach, is that this thesaurus can be considered an implementation of Paivio's dual coding theory. Following the observation used in PhraseFinder [JC94], an association thesaurus can be seen as measuring the belief in a *concept* (instead of in a document) given the query. Thus, the domain specific Moa structures that model IR query evaluation, can also be used for query formulation using the thesaurus.

Querying the digital image library now takes place as follows. First, the user enters an initial (usually textual) query. Next, we use the thesaurus to select clusters from the image content representations that are relevant to this initial query. Assuming that the result is a Moa expression called query, we then retrieve images from the digital library as follows:

```
map[sum(THIS)](
  map[getBL(THIS.image,
    query, stats)]( ImageLibraryInternal ));
```

The results of this query are shown to the user. The user may provide relevance feedback for these images; this relevance feedback is used to improve the current query. A problem for the current retrieval system is that the thesaurus sometimes associates words in the annotations to irrelevant clusters, or AutoClass identifies clusters of little semantic value. To alleviate these problems, we are investigating machine learning techniques to adapt the thesaurus and the content representation, using the relevance feedback across query sessions.

As we demonstrated, the multimedia querying process is expressed in Moa expressions. The Mirror DBMS provides the primitives for managing the images and its meta data, as well as the probabilistic inference required during the interaction with the user. New feature models, different clustering algorithms, or different query modification techniques, can easily be added or modified. Hence, a variety of multimedia retrieval systems can be implemented by simply changing the sequence of Moa expressions issued in the Mirror DBMS.

## 6 Acknowledgments

This demo could never have been created without the Moa kernel, designed and implemented mainly by Annita N. Wilschut and Jan Flokstra. Erik van het Hof has implemented the open distributed architecture of figure 1. Both Henk Ernst Blok, who added the LIST structure to Moa, and Roelof van Zwol, contributed significantly in discussions about the system design. Finally, we want to thank the Monet team, and Peter Boncz in particular, for the support, explanations, and all those nightly hours spent tracing bugs and fixing them.

## References

[BWK98] P. Boncz, A.N. Wilschut, and M.L. Kersten. Flattening an object algebra to provide performance. In *Fourteenth International Conference on Data Engineering*, pages 568-577, Orlando, Florida, February 1998.

[CS95] P. Cheeseman and J. Stutz. Bayesian classification (AutoClass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*. AAAI Press, 1995.

[dV98] A.P. de Vries. Mirror: Multimedia query processing in extensible databases. In *Proceedings of the fourteenth Twente workshop on language technology (TWLT14): Language Technology in Multimedia Information Retrieval*, pages 37-48, Enschede, The Netherlands, December 1998.

[dVB98] A.P. de Vries and H.M. Blanken. Database technology and the management of multimedia data in Mirror. In *Multimedia Storage and Archiving Systems III*, volume 3527 of *Proceedings of SPIE*, pages 443-455, Boston MA, November 1998.

[dVEK98] A.P. de Vries, B. Eberman, and D.E. Kovalcin. The design and implementation of an infrastructure for multimedia digital libraries. In *Proceedings of the 1998 International Database Engineering & Applications Symposium*, pages 103–110, Cardiff, UK, July 1998.

[dVW99] A.P. de Vries and A.N. Wilschut. On the integration of IR and databases. In *Database issues in multimedia; short paper proceedings, international conference on database semantics (DS-8)*, Rotorua, New Zealand, January 1999.

[EK95] M.W. Eysenck and M.T. Keane. *Cognitive Psychology. A student's handbook*, chapter Mental representation. Lawrence Erlbaum Associates, 3 edition, 1995.

[JC94] Y. Jing and W.B. Croft. An association thesaurus for information retrieval. Technical Report 94-17, University of Massachusetts, 1994.

[MP97] T.P. Minka and R.W. Picard. Interactive learning using a "society of models". *Pattern Recognition*, 30(4), 1997.

[SMMR99] D. McG. Squire, W. Müller, H. Müller, and J. Raki. Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback. In *The 11th Scandinavian Conference on Image Analysis*, Kangerlussuaq, Greenland, June 1999. To appear.

[WY95] S.K.M. Wong and Y.Y. Yao. On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems*, 13(1):38–68, January 1995.