A Multimedia Meta-Database Model for Distributed MultiMedia DBMS

Georges CHALHOUB, Samir SAAD, Richard CHBEIR, Kokou YETONGNON
Computer Science Department
LE2I – Bourgogne University
BP 47870 21078 Dijon CEDEX - FRANCE
Phone: (+33) 3 80 39 36 55, Fax: +333 80 39 68 69

Abstract: - In this paper, we address the issue of Distributed MultiMedia DBMS (DM²DBMS) where traditional meta-database used to describe the database schema is no longer appropriate. The meta-database is the kernel of the DBMS and we do believe that new generation of meta-database is required for DM²DBMS. For this, we provide a multimedia meta-database model M² able to improve multimedia management in DM²DBMS in terms of distributed data storage and retrieval. The proposed multimedia meta-database model is independent (but compatible) of all current data format models (MEPG-4, MPEG-7, etc.). We show how M² can allow to DBMS to easily respond to new requirements imposed by distributed multimedia data.

Key-Words: Distributed Multimedia data, Database Management System, meta-database.

1 Introduction

In the past few years, multimedia data have become available at an increasing rate, especially in digital format. There has been a tremendous need for the ability to store, query and process non-traditional data in a wide variety of applications. For example, medical applications create and use large amount of X-ray and ultrasound pictures; geographical information systems and location-based applications often manipulate digital maps; satellite based applications routinely generate and use large amount of images; video surveillance cameras such as those used in criminal investigations generates large number of video frames; and web-based applications have access to heterogeneous multimedia data composed of different data types and formats. Early on, the particular needs and requirements of multimedia database management systems (MMDBMS) have been recognized and their differences from traditional database management systems have been pointed out. These differences stem from the diversity of the data formats and media sources that must be handled by multimedia systems including image, video, audio, text document and other pictorial data. Therefore, it has become naturally important to focus several research efforts on extending and using traditional DBMS technologies to develop multimedia management systems that are able to not only store but also filter, retrieve, and organize the mass of available multimedia data.

A lot of work has been done in the past to increase the efficiency of multimedia management in DBMS and to integrate data in the standard data processing environments [3, 4, 6]. Early research in multimedia data processing has been carried out separately in the database and computer vision communities. The database approach focuses on metadata management and content-based semantic annotations for storage and retrieval of multimedia data. This approach has several inadequacies as it is time-consuming, subjective, and cannot adequately describe the content of multimedia data [4, 7]. The computer vision approach has addressed content-based issues such as information coding, lossless data compression, image segmentation. This approach is based on low level features such as color, texture, shape, layout etc. [1, 2, 8]. To integrate the two approaches, several research activities have focused on defining new representation formats and standards allowing the description of multimedia data through several dimensions. For example, the MPEG family of standards [5] aims to define a framework for the efficient representation of multimedia data (MPEG-4, 7 and 21). Their goal is to provide core technologies for efficient storage, transmission and manipulation of multimedia data.

The need for a full fledge multimedia DBMS becomes even more apparent when one considers distributed processing environments (such as P2P architectures) in which complex multimedia objects can be shared on demand and replicated over several sites. To provide database functionalities and meet the growing demands for efficient processing of the vast quantities of data, multimedia data management systems must incorporate several of the following capabilities:

- Multimedia-oriented operations: For instance, users need queries that involve "*similarity-based*" selection and join operations that use both content-based and metadata representation of multimedia. Such a "similarity-based join" operation on multimedia tables is not considered by existing systems. For example, in a firm time management application, we stock in a EMPL table the employees names, addresses, and images and in ENTRANCE table, the video captured by a monitoring camera at different times. A multimedia-join operation between the two tables, can be used to determine the name of employees entering (or leaving) the firm at a given time.

- Appropriate security: Current security policies are no longer appropriate for multimedia objects where several layers (sub-objects) and parameters need to be considered such as user profiles, network, and media type.
- Appropriate data storage: classical methods used to achieve textual data storage are no longer applicable (index, cluster, fragment, etc.) where criteria are built on identical attributes. When managing multimedia data and an alternative solution should be found.
- Appropriate multimedia query model and optimization techniques: the system uniform query capabilities over the diverse multimedia data types. The query interface and the traditional SQL query standard must be extended to deal with not only traditional relational data but also the image, audio and video data types. Moreover, the various similarity based, cluster based and different range queries must be taken into account.
- Appropriate relation abstraction capabilities: it is widely accepted by multimedia and database communities that multimedia data model must include different layers of abstractions to better capture the relationships that may be multimedia objects at different levels. For example, two or more multimedia objects can be related because of similarities in their low level features values. It is important therefore to allow classification schemes to define classes or clusters of similar object on the basis of traditional textual, physical and/or semantic features of different media types. Moreover, multimedia object can be linked by higher level relationships types. Spatial and temporal relationships can be defined on the objects. Composition and other semantic relationships (generalization, specialization and instance of) can be established among classes of multimedia objects.
- Appropriate meta-database management: in reality, for processing almost all functions, the DBMS accesses the meta-database to find details about the tables or objects. As an example let us consider the SQL query in Relational DBMS: "*SELECT ename, age FROM Emp WHERE sal= 2500*". Before executing this query on Emp table, the DBMS checks in the meta-database the existence of Emp and its attributes ename, age and sal.

In this paper, we present a multimedia meta-data model M^2 to address some of the above issues and to support the design of efficient multimedia meta-database model able to improve multimedia management. The goal is to provide a modeling framework to express the properties of data items and the meta-data that are necessary for organizing multimedia management systems at different levels. Built on relational-object paradigm (to support both of them), our multimedia meta-database model is independent (but compatible) of all current data format models (MEPG-4, MPEG-7, etc.) and able to organize distributed multimedia data in an efficient manner in order to optimize queries response. The key feature of the model is that it captures in a single modeling meta-concept the low-level features, the structural and semantic properties, and the relationship descriptions of both multimedia object and meta-object. The meta-model is the core component of an ongoing research on distributed multimedia management environment which aims to address design issues involving security and fragmentation.

The rest of the paper is organized as follows. Section 2 presents the M^2 model. Section 3 presents some examples of descriptions of multimedia data based on the M^2 model and how multimedia distribution is done. And finally section 5 concludes the paper.

2 A multimedia meta-database model M^2

Operations on the meta-database include creation, modification, and access organizing operations. Below, we will explain our proposal for structuring meta-database for DM^2DBMS. The proposed meta-database is built on relational-object paradigm is order to be able to consider both relational and object-oriented DBMS. It can also be used on XML-Based DBMS. Our proposal is built upon a main component M^2 detailed here below.

2.1 Definition

In essence, the multimedia model M^2 extends a previous repository model for the management of image databases [9] which describes the image data through several abstraction levels. This model has been used to establish an algebra for image databases where SQL and image-oriented operations can be written. Two basic concepts are provided by the multimedia repository model M^2: a meta-object and a meta-class.

A **meta-object** has a set of properties used to capture the descriptions of an object at different levels of description and can be related to other meta-objects via one or more relationships. The representation M² (id, O, F, A, R) of a meta-object consists of:

- **Id** is a unique identifier associated to a meta-object. It is used to differentiate an object from any other object. It represents an instance of a multimedia object or a record. The id includes the location of the instance (or record) which allows considering data distribution and global unique identification.
- **O** is a reference to the raw data of the object (or the file). For complex multimedia data, O is the actual (image, video, or audio) object file which can be stored as BLOB. For set oriented data, O is an index for the data structure used to store the elements of the set. O can be null for some meta-object.
- **F (Descriptor, Model, Value)** is a feature vector representation of the object O.
  - Descriptor: is the type of representation (such as Color Histogram, Color distribution, Texture Histogram, Start Time, End Time, Duration, Motion, Camera Motion, Audio freq., Amplitude, Band no., Power (dB), etc.)
  - Model: is the description format (such as RGB, RHV, etc.)
  - Value: is the content descriptor. This component contains the physical, visual, spatial and temporal feature data value.
- **A (ES, Sem_F)** contains meta-data where:
  - ES: is the External Space descriptions consisting of:
    - Context-Oriented (CO) data that are completely independent of the multimedia object content. For example, in a route monitoring application, it contains the name of the monitoring center.
    - Domain-Oriented (DO) data are directly or indirectly related to the object. For example, it contains the traffic state of the route or the street.
    - Multimedia-oriented (MO) data are directly associated to the multimedia object creation such as compression type (MPEG, MP3, etc.) and type (movie, home media, video, video shot, region, filming date, etc.).
  - Sem_F(Type, Description):
    - Type: defines the type and the semantic feature (keyword, scene, etc.)
    - Description: is a textual representation
- **R ({{S1={$id_{i\ i=1..n}$}, S2={$id_{j\ j=1..m}$}, Re={$Rel_{k\ k=1..p}$}}}):** This component represents zero or more relationships between objects. The description of each relationships consists of:
  - The set of the identities of objects participating in the relation. These may be from different tables. The sets S1 and S2 can be empty when they represent the meta-object itself.
  - Re represents a set of relationships between two sets of objects. Each triplet (S1, S2, Re) means that for any couple of S1 and of S2, each element in Re is valid, e.g. ({id_1}, {id_2}, {R_1, R_2}) $\Rightarrow$ id_1 R_1 id_2 and id_1 R_2 id_2. Each relation can be a spatial (directional, metrical, topological), semantic, temporal, and similarity relation. Using the relations, we can easily identify the spatial and semantic hierarchies between multimedia objects represented in our model. This component also implements the traditional composition and membership relations.

In our approach, a **meta-class** is used to construct sets of objects which verify a membership relation. Contrary to traditional database model, the meta-class is schemaless. The meta-model M² defines self-describing objects which encapsulate their description schema with their values. A meta-class therefore does not define a structure or a set of properties that is shared by all its members. A meta-class is represented as a meta-object M² (id, O, F, A, R) where:

- id is the unique identifier of the meta-class
- O is a null reference.
- F: contains a median feature vector of the meta-class. This is very useful for data organizing and accessing.
- A: contains a representative meta-data set. This would be very important for indexing purposes.
- The R component includes a mandatory Instance_of relationship between a meta-class and its member. It is defined by:

  R= {(S1= {id_i}, S2= {}, Re= {Instance_of})}

  The **Instance_of** relationship specifies the members of a meta-class based on a meta-class membership condition or predicate which is verified by the instances of the meta-class. The set S1 contains the identifiers id_i of the meta-objects. The definition of the meta-class membership predicate can be based on:
  - The meta-properties defined by parameter A of the member objects
  - Similarity expression defined on the low level features (component F of the representation) of the objects. For example, one can consider the class of images that are predominantly blue. In this case, the meta-class regroups objects that verify this

condition regardless of the values of the other components of the object.
- o A semantic expression based on the context-based annotations of the objects.

R can be extended with additional meta-class relationships. For example, generalization and specialization relationships between a meta-class and one or more meta-classes or meta-objects can be defined by:

R= {(S1= {idi}, S2= {}, Re= {Instance_of}), (S1= {idi}, S2= {}, Re= {Generalization}), (S1= {idi}, S2= {}, Re= {Specialization})}

- The **Generalization** relationship defines a link between a meta-class and its sub-meta-classes. It can be used to express traditional super-class relationships between classes. A super-meta-class in this case defines a membership relationship that can be subsumed by the sub-classes.
- The **Specialization** relationship is the inverse of the generalization relationship. A sub-meta-class inherits and subsumes the membership condition or predicate of its super-meta-class.

2.2 Example

Using the proposed multimedia meta-database model, either static object (e.g. image), dynamic object (e.g. movie), or a set (or a table) of media objects can be represented in the DBMS. Here below, we give an example concerning the representation of a movie object (or a record). As we will see, the R component of M^2 plays a major role here.

Let us study the movie components appearing in Fig. 1. The hierarchical relations between objects in M^2 are represented by a N-ary tree where the root M represents the entire multimedia object and where each node is a static or dynamic object having one or several outgoing edges. A tree leaf includes either still or moving regions, audio file, or annotation data.

In M^2, this movie M is represented as follows. The components O and F will not be presented in this example. Here below, we only represent relations between object and direct sub-objects. We think that it is appropriate for multimedia objects level. However, any link of relations could be use at meta-class level:

- o M.A component:
  - ES.MO contains the object type: movie
  - Sem_f.type decribes the movie type: comedy
- o M.R component:
  - ({M_1.id}, {M_2.id}, {*before*}): allows to both implicitly identify the composition of the object M, and the sequencing between the movie scene M_1 and M_2 present in object M
- o M_1A component:
  - ES.MO contains the object type: movie scene
  - Sem_f.type: contains the location
  - Sem_f.description: *road and trees in Spain*
- o M_1R component:
  - ({}, {M_{11}.id, M_{12}.id, M_{13}.id}, {*Equal*}): It expresses the composition and the parallel executing of video, audio and annotation in M_1.
    - …
- o M_{211}.A component:
  - ES.MO: moving region
  - Sem_f.type: color and name of the moving region
  - Sem_f.description: red car
- o M_{211}.R component:
  - ({}, {M_{212}.id}, {*left, Disjoint*}): identifies the spatial (directional, topological and metric relations between M_{211} and M_{212}

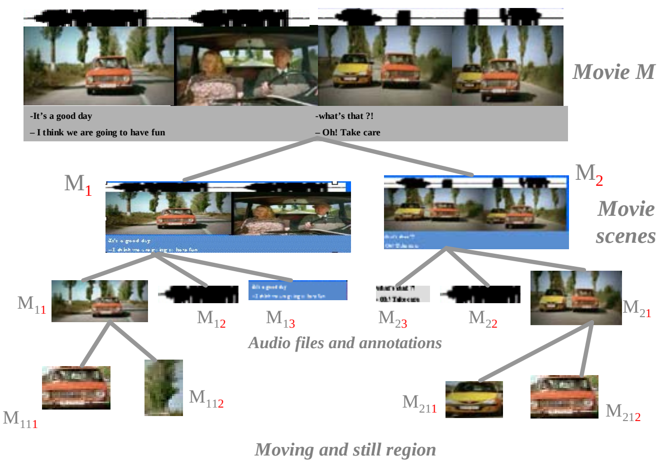


Fig. 1: Movie representation

3 Applications

The applications of our multimedia meta-model M^2 are various. In this section, we show how queries can be designed, and how data distribution and clustering can be performed.

3.1 Multimedia Query Model M^2Q

We define here the query model of M^2. We consider several types of query: metadata-based, content-based and multicriteria-based query. A meta-data query is based on meta-data while content-based query addresses feature vectors. The initial content-based query input could be an image, a video sequence, a movie scene, or a set (or table) of objects. A multicriteria-based query includes meta-data and feature vectors parameters.

The proposed query model can support these query types. It is expressed as follows $M^2Q(id_q, O_q, F_q, A_q, R_q) \rightarrow M^2QR$ where:

- id_q is a unique identifier of an instance of Q. ID is useful when a table of query objects or records is submitted.
- O_q is a reference of the query object itself that can be stored as BLOB. It has a null value in case of metadata-based query
- F_q, A_q, R_q have the same roles of F, A, R in the multimedia meta-model M^2.
- M^2QR is the Query Result. It allows to identify the desired attributes or properties.

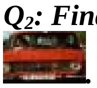
Here below, we show how to use the M^2Q to answer metadata-based, content-based, and multicriteria-based queries. We give query examples with their corresponding SQL statements.

Q_1: Find all comedy movies where Eddy Murphy plays? *is expressed as:*

$M^2Q_1(i_{q1}$, null, null, A.={movie, comedy, Eddy Murphy}, null) $\rightarrow M^2QR_1$(,*(movie), , ,)*

***M_Q_1**: SELECT M^2.O FROM M^2 WHERE (A.ES.MO="Movie" AND A.Sem_f.Type="movie type") AND (A.* contains "comedy") AND (A.* contains "Eddy Murphy").*

Note that the operator *contains* in the predicate A.* is used to verify whether the corresponding meta-data ("comedy", "Eddy Murphy") belong to any A components

Q_2: Find all movie scenes containing the car driving  *is expressed as:*

$M^2Q_2(i_{q2}$,  *, F_2[1], A.*={movie, scene, car, driving}, null) $\rightarrow M^2QR_2$ (,*(movie scene), , ,)*

*M^2QR_2_1: SELECT * FROM M^2 WHERE (A.ES.MO="Moving region") AND (A.* contains "car" AND A.* contains "driving")*
*M^2QR_2_2: SELECT * FROM M^2QR_2_1 WHERE (M^2QR_2_1.F SIMILAR $Q_2.F_q$) AND (A.ES.MO="frame")*

*M^2QR_2_3: SELECT * FROM M^2 WHERE (A.ES.MO="Video") AND (null, {M^2QR_2_2.id}, {Contain}) **IN** R*

***M^2QR_2**: SELECT O FROM M^2QR_2_2 WHERE (A.ES.MO="Movie scene") AND (null, {M^2QR_2_3.id}, {Contain}) IN R*

Note that **IN** is an operator to express the membership inside the R component:
$(\{id_i\},\{id_j\},\{R_k\})$ **IN** R $\Leftrightarrow$ $\{id_i\} \subseteq R.S_1$, $\{id_j\} \subseteq R.S_2$ and $\{R_k\} \subseteq$ R.Re

[1] The feature vector computing on the basis of the media

The same reasoning can be used for Audio or other media type query using the correspondent feature vectors (frequency, amplitude, etc.).

Q3: Find all movies containing the following movie scene  ***with total similarity[2] (Video Audio, text), is expressed as:***

$M^2Q_3(i_{q3}$,  *, F_3, A.*={movie, scene, car, driving}, R SIMILAR (Video, Audio, text)) $\rightarrow M^2QR_3$ (,*(movie),,,)*

In this case, we represent the query M^2Q_3 according to M^2 as follows:

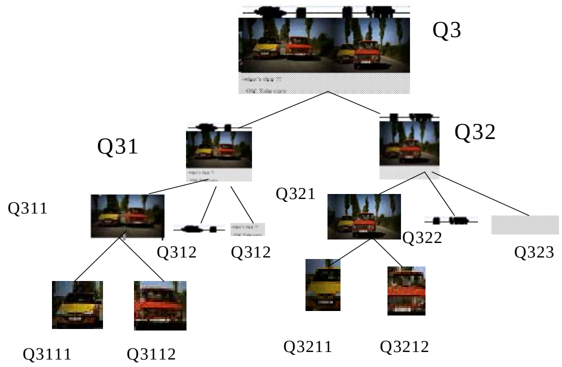


Fig. 2: M^2 Query representation

*M^2QR_3_1: SELECT * FROM M^2 WHERE (A.ES.MO="Moving region") AND (A.* contains "car" AND A.* contains "driving")*
*M^2QR_3_2: SELECT * FROM M^2QR_3_1 WHERE (F SIMILAR Q3111.Fq)*
*M^2QR_3_3: SELECT * FROM M^2QR_3_1 WHERE (F SIMILAR Q3112.Fq)*
*M^2QR_3_4: SELECT * FROM M^2QR_3_1 WHERE (F SIMILAR Q3211.Fq)*
*M^2QR_3_5: SELECT * FROM M^2QR_3_1 WHERE (F SIMILAR Q3212.Fq)*

*M^2QR_3_6: SELECT * FROM M^2 WHERE (A.ES.MO="Video") AND (A.* contains "location" AND A.* contains "Road") AND (null, {M^2QR_3_2.id, M^2QR_3_3.id}, {Contain}) IN R AND ({M^2QR_3_2.id}, {M^2QR_3_3.id}, {Parallel}) IN R*

*M^2QR_3_7: SELECT * FROM M^2 WHERE (A.ES.MO="Audio") AND (F SIMILAR Q312.F)*

*M^2QR_3_8: SELECT * FROM M^2 WHERE (A.ES.MO="text") AND (F SIMILAR Q313.F)*

*M^2QR_3_9: SELECT * FROM M^2 WHERE (A.ES.MO="Movie Scene") AND (null, {M^2QR_3_6.id, M^2QR_3_7.id, M^2QR_3_8.id}, {Contain, Equal}) IN R*

[2] The different types of similarity are out of scope of this paper

*M^2QR_3_10: SELECT * FROM M^2 WHERE (A.ES.MO="Video") AND (A.* contains "location" AND A.* contains "Road") AND ({M^2QR_3_5.id}, {M^2QR_3_4.id}, {Parallel}) IN R*

*M^2QR_3_11: SELECT * FROM M^2 WHERE (A.ES.MO="Audio") AND (F SIMILAR Q322.F)*

*M^2QR_3_12: SELECT * FROM M^2 WHERE (A.ES.MO="text") AND (F SIMILAR Q323.F)*

*M^2QR_3_13: SELECT * From M^2 WHERE (A.ES.MO="Movie Scene") AND (null, {M²QR3_10.id, M^2QR_3_11.id, M^2QR_3_12.id}, {Contain, Equal}) IN R*

***M^2QR_3:** SELECT O FROM M^2 WHERE (A.ES.MO="Movie") AND (A.Sem_f.Type="Scene" AND A.S* contains "two cars driving on a road between trees") AND ({M^2QR_3_13.id}, {M^2QR_3_9.id}, {immediately sequential}) IN R*

In partial similarity (Audio only, Video only, or Audio and text only, etc.), we can omit the correspondent sub-query.

3.2 Data Distribution

Data distribution consists of creating and allocating data units, which are elementary fragments or clusters, among a set of distributed (logical or real network) sites [10, 11]. In our approach, the identification of multimedia objects and distribution units (fragment, object, table, etc.) is carried out using the extended URL based object identifier. The object id and various fragmentation meta-data are integrated in the meta-database. The distribution of multimedia data can depend on various media. We distinguish 3 types of data distribution: the Logical-based media: where several logical criteria decide about data distribution, the Physical-based media: where media characteristics (type, size, quality, etc.) determine data distribution, And Mixed media: where criteria are physical and logical. We do believe that this kind of distribution is very important for multimedia applications.

4 Conclusion and future work

This paper describes an original manner to address distributed multimedia DBMS which consists of providing an appropriate multimedia meta-database model. The meta-database is the kernel of the DBMS which allows managing all internal functions. The proposed model, called M², is based on relational-object paradigm. It is able to consider both object (and record) representation and meta-class of media objects. Several examples and a query model were presented. We also discussed how to apply M² to achieve data distribution.

The future directions will be focused on: 1- security policies on multimedia objects where several layers (sub-objects) and parameters need to be considered such as user profiles, network, and media type. 2- multimedia data fragmentation which consists of dividing data into several fragments in order to reduce processing cost and to minimize execution time. We are investigating how to extend traditional fragmentation techniques to take into account the main characteristics of multimedia data. 3- clustering algorithms where several criteria are involved.

References:

[1] J.K.Wu and A.D. Narasimhalu and B.M. Mehtre and C.P. Lam and Y.J. Gao, CORE: A Content-Based Retrieval Engine for Multimedia Information Systems, Multimedia Systems, 1995, Vol. 3, pp. 25-41.

[2] S. Berchtold and C. Boehm and B. Braunmueller and D. A. Keim and H. P. Kriegel, Fast Parallel Similarity Search in Multimedia Databases, SIGMOD Conference, AZ, USA, 1997, pp. 1-12.

[3] A. Yoshitaka and T. Ichikawa, A Survey on Content-Based Retrieval for Multimedia Databases, IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 1, 1999, pp.81-93.

[4] Y. Rui and T.S. Huang and S.F. Chang, Image Retrieval: Past, Present, and Future, Journal of Visual Communication and Image Representation, 1999, Vol. 10, pp.1-23.

[5]http://www.chiariglione.org/mpeg/working_documents.htm (visited at April 4, 2004)

[6] W. I. Grosky, Managing Multimedia Information in Database Systems, Communications of the ACM, 1997, Vol. 40, No. 12, pp. 72-80.

[7] J. P. Eakins and M. E. Graham, Content-Based Image Retrieval: A Report to the JISC Technology Applications Program, January, 1999, Inst. for Image Data Research, Univ. of Northumbria at Newcastle.

[8] William I. Grosky; Managing Multimedia Information in Database Systems, Communications of the ACM, Vol. 40, No. 12, 1997, pp. 72-80.

[9]Atnafu S., Chbeir R., Brunie L., Integrating Similarity-Based Queries in Image DBMSs, ACM SAC'04, Cyprus, march 04, To appear.

[10]Ezeife, C.I. and Ken Barker, A Comprehensive Approach to Horizontal Class Fragmentation in a Distributed Object Based System, International Journal of Distributed and Parallel Databases , Kluwer Academic Publishers, Volume 3, No. 3, pp. 247-273, July 1995

[11]Ezeife, C.I. and Ken Barker, "Distributed Object Based Design: Vertical Fragmentation of Classes", International Journal of Distributed and Parallel Databases, Vol. 6, No. 4, pp. 327-360, Kluwer Academic Publishers, October 1998.

[12] *A Tutorial on Clustering Algorithms* http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/index.html (visited at April 4, 2004)