Multimed Tools Appl (2007) 35:91–108
DOI 10.1007/s11042-007-0118-7

Multimedia data warehouses: a multiversion model and a medical application

Anne-Muriel Arigon · Maryvonne Miquel · Anne Tchounikine

Published online: 3 May 2007
© Springer Science + Business Media, LLC 2007

Abstract In field such as Cardiology, data used for clinical studies is not only alphanumeric, but can also be composed of images or signals. Multimedia data warehouse then must be studied in order to provide an efficient environment for the analysis of this data. The analysis environment must include appropriate processing methods in order to compute or extract the knowledge embedded into raw data. Traditional multidimensional models have a static structure which members of dimensions are computed in a unique way. However, multimedia data is often characterized by descriptors that can be obtained by various computation modes. We define these computation modes as "functional versions" of the descriptors. We propose a Functional Multiversion Multidimensional Model by integrating the concept of "version of dimension." This concept defines dimensions with members computed according to various functional versions. This new approach integrates different computation modes of these members into the proposed model, in order to allow the user to select the best representation of data. In this paper, a conceptual model is formally defined and a prototype for this study is presented. A multimedia data warehouse in the medical field has been implemented on a therapeutic study on acute myocardial infarction

Keywords Data warehouse · OLAP · Multimedia data · Functional version · Multimedia data descriptor

A.-M. Arigon (✉)
Laboratoire de Biométrie et Biologie Evolutive, LBBE UMR CNRS 5558, UCBL,
43 boulevard du 11 novembre 1918, 69622 Villeurbanne Cedex, France
e-mail: arigon@biomserv.univ-lyon1.fr

M. Miquel · A. Tchounikine
Laboratoire d'InfoRmatique en Images et Systèmes d'information, LIRIS UMR CNRS 5205, INSA,
7 avenue Capelle, 69621 Villeurbanne Cedex, France

M. Miquel
e-mail: maryvonne.miquel@liris.cnrs.fr

A. Tchounikine
e-mail: anne.tchounikine@insa-lyon.fr

1 Introduction

Modern areas produce increasingly voluminous amounts of electronic complex data. As an example in the medical area conventional administrative data therapeutic and diagnostic data is now completed with complex multimedia data such as X-ray pictures echography electrocardiogram etc... captured by electronic medical devices. This established fact in business or retail areas raised the idea to extract and gather this data coming from a heterogeneous and distributed system such as Hospital Information System (HIS) or Picture Archiving and Communication System) in order to discover useful information. Indeed medical research requires large sets of data coming from various sources collected for the purpose of analysis and extraction of information. These databases are often dedicated to one pathology or a class of pathologies and are used to validate research hypothesis and to build clinical knowledge. Before being introduced into the database data must be validated by experts in order to guarantee the quality and the coherence of the system. In that case the constitution of an expertized database is an expensive work in both time and human investment. Then clinical researchers are naturally interested in data warehousing and in On Line Analysis Processing (OLAP) technologies in order to produce and manage high quality data. With these technologies they also gain the capacity to navigate into large data sets according to their needs (epidemiological studies population follow-up and evaluation of indicators...).

In fields such as Cardiology, the data used for clinical studies is not only alphanumeric, but can also be composed of images or signals. The analysis environment of such data must include processing methods in order to compute or extract the knowledge embedded into raw data. Classical data warehouses are generally used to provide analysis of numeric facts. The use of multimedia data as facts in a warehouse raises numerous pending questions. Among these questions are the storage of voluminous data, the design of suited navigation interface and complex ad-hoc aggregation functions. Another important problem which is barely addressed is related to the semantics of the medical data. In the cardiology area, for instance, the main goal of many researches is to find indicators and descriptors to understand and characterize heart pathologies. For this purpose, the scientists have to developed efficient algorithms (based on signal or image processing, pattern recognition, statistical methodologies...) to transform the initial data (e.g. an electrocardiogram ECG, representing the twelve leads of a standard electrocardiogram) or a set of data (current ECG, past ECG, biological patients' data,..) into relevant information (data descriptors, risk factors, diagnosis class...) and to validate them on a large scale database. Beyond the difficulties encountered in extracting and modeling medical data, this example emphasizes the difficulty in manipulating, interpreting the data and the need in promoting raw data into useful information. These processes are often very hard to elaborate, and one should be able to share this knowledge, and then to allow the user to choose how to calculate his set of indicators. Thus, these processes should be fully, tightly integrated in the warehouse and be part of it.

In this article we propose a model for multimedia data warehouse. This model integrates different functional versions for dimension members and allows navigation and comparison between these different modes of calculus. Section 2 presents definitions and requirements for a clinical and multimedia data warehouse and related works. In Section 3 a case study in the cardiology field is described. The proposed multiversion model is detailed in Section 4. The prototype and data warehouse used for the case study is presented in Section 5 and Section 6 concludes.

2 Multimedia data warehouses

2.1 Data warehouse: definition and architecture

A Data Warehouse is a "subject-oriented, integrated, non-volatile and time-variant collection of data in support of management's decisions" [14]. A data warehouse is a single site repository of information collected from multiple sources. Information in the data warehouse is organized around major subjects and is modeled in order to allow pre-computation and fast access to summarized data. "OLAP" [5, 28] refers to analysis techniques used to explore the data.

Data warehouse has become a leading topic in the commercial world as well as in the research community. For example, until now, data warehouse technology has been mainly used in business world, in retail or finance areas. The leading motivation is to take benefits from the enormous amount of data stored in operational databases. According to Kimball [18], the data-modeling paradigm for a data warehouse must comply with requirements that are totally different from the data models in OLTP environments. The data model of the data warehouse must be easy for the end-user to understand and write queries, and must maximize the efficiency of queries.

Data warehouse models are called multidimensional models or hypercubes and have been formalized by several authors [1, 4, 8, 19, 20, 27]. They are designed to represent measurable facts or indicators and the various dimensions that characterize the facts. As an example, in a retail area, typical facts are the price and the amount of a purchase; dimensions are Product, Location, Time and Customer. A dimension can be organized in hierarchy. For example the Location dimension can be aggregated in City, State, and Country. The "star schema" models the data as a simple cube, in which the hierarchical relationship in a dimension is not explicit but is rather encapsulated in attributes. The "snowflake schema" normalizes dimension tables, and makes it possible to explicitly represent the hierarchies by separately identifying a dimension in its various granularities. At last, when multiple fact tables are required, the "galaxy schema," or "fact constellation" model allows the design of collection of stars schemas.

OLAP architectures are based on a multi-tier architecture (Fig. 1). The first tier is a warehouse server, often implemented using a relational DBMS. Data of interest must be extracted from operational legacy databases, cleaned and transformed by ETL (Extraction, Transformation, and Loading) tools before being loaded in the warehouse. This step aims to consolidate heterogeneous schema (structure heterogeneity, semantic heterogeneity) and to reduce data in order to conform it to the data warehouse model (using aggregation, discretization functions). Then the warehouse contains high quality, historical and homogeneous data.

The second tier is a data mart. A data mart handles data sourced from the data warehouse, reduced for a selected subject. The main advantage of data marts is to isolate data of interest for a smaller scope, thus permitting the focus on optimization needs for this data and increase security control. However this intermediate data mart is optional.

The 3rd level is the OLAP server. It calculates and optimizes the hypercube (Fig. 2), i.e. the set of fact values for all the tuples of instances of dimensions (also called members). In order to optimize accesses to the data, query results are pre-calculated in the form of aggregates. OLAP operators enable the materialization of various views of the hypercube, allowing interactive queries and analysis of the data. Common OLAP operators include roll-up, drill-down, slice and dice, rotate.

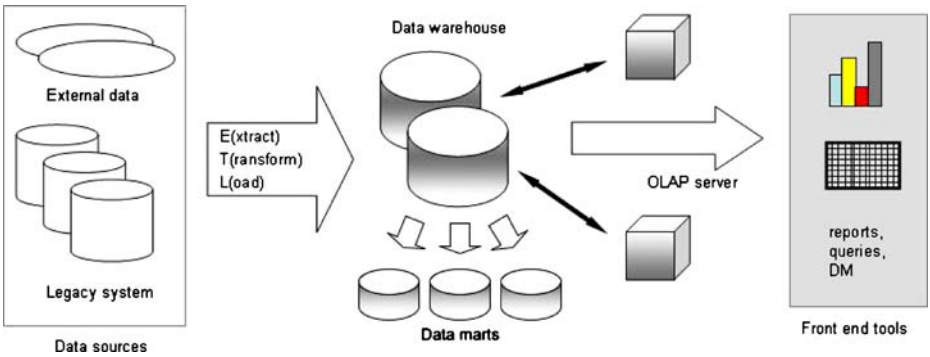


Fig. 1 OLAP multi-tier architecture

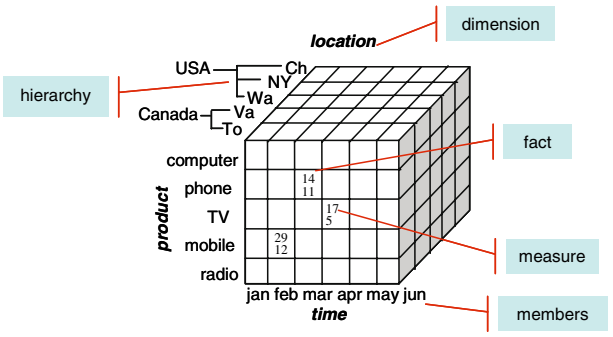
The fourth level is an OLAP client, and provides a user interface with reporting tools, analysis tools and/or data mining tools. Software solutions exist for a traditional use. If the studied data and analysis processes are complex, then a specific interface must be designed.

2.2 Clinical data warehouses

Clinical Decision Support Systems (CDSS) were introduced 25 years ago to help medical decision-making. Compared to CDSS, clinical data warehouses have a much broader scope. Indeed, whereas a CDSS is specialized in one very specific problem, such as drug alerts or comparison of physicians' performance on constructed cases, thus being functionality-centered, data warehouse remains data-centered, providing the possibility of unanticipated queries and encouraging data exploration by free navigation in the hypercube. Data warehousing recently founds its way into healthcare, providing enormous benefit in clinical research, quality improvement and decision support by ensuring efficient access to valuable information.

The most conventional applications for data warehousing, because more easily operational, are aimed at socio-economic evaluation of health network. The purpose is to evaluate the cost of a specific pathology, a drug, a department or a hospital. Another type of use is scheduling means (surgical scheduling, critical care bed allocation) [15]. Finally some studies propose the use of data warehouses in order to improve care or medical

Fig. 2 A hypercube



research: discovering of best practices, protocols, rule-based clinical alert criteria are examples of application areas. For instance, the Turku University Central Hospital in Finland carries out a data warehousing research project for a study on drug-laboratory interference and a study on drug–drug interactions [22]. From a research point of view, three main areas can be distinguished. The first one focuses on extraction of information encapsulated in the Electronic Patient Record to populate the data warehouse. The second challenge is the development of data-mining algorithms for the discovery of rules. [13] emphasizes the needs for life sciences and presents an implementation of a scientific OLAP system and a platform for scientific data mining features. At the conceptual level, researchers underline the interest of combining the performances of OLAP systems with the possibilities of handling complex data offered by statistical and scientific databases [25]. Several experiments show the benefit of the coupling of the two approaches [10, 17]. The Mining Mart project [32] defines M4 (Mining Mart MetaModel), a system for supporting data preprocessing for data mining. Its purpose is a case-based reasoning approach that enables automatization of preprocessing and reusability of defined preprocessing cases. Reference [16] is a proposal to track the various processes that are performed on the data warehouse during its life cycle (cleaning, loading, evolution…).

In all these cases, the data loaded in the data warehouse is conventional alphanumeric data and is modeled using a classical multidimensional model. However, data produced and used by physicians comprises also complex data. Thus, clinical data warehouse should also be able to manage this specific data, and a great challenge of clinical warehousing is the development of powerful data models. Several efforts address multidimensional modeling issues for complex medical data [23, 24]. These studies take place in the 3rd area of research in clinical data warehousing. Advanced requirements for a clinical data warehouse include the support of unconventional hierarchy dimension. Let us suppose a dimension “pathology” and a fact “patient”: the model should allow the representation of multiple hierarchies (a pathology can belong to several class of pathologies), non-onto hierarchies (some pathologies do not belong to a class of pathologies), many-to-many relationships between facts and dimension (a patient can have several pathologies), slowly changing dimension (medical nomenclatures evolve over time, a pathology may be reclassified following new bio-medical investigations) [3]… Medical data is by nature imprecise, uncertain and a medical OLAP server must allow the analysis of this imperfect information, supporting an algebra that provides aggregation of imprecise data for example.

Another aspect related to the complexity of medical data, and more largely of scientific data, is how they are used. An electrocardiogram, an X-ray, a MRI, is information that needs to be interpreted, by the physician and/or a machine to become understandable and to be turned into valuable, relevant information. In the same way, a clinical data warehouse allows to take into account this type of data which is fundamental in the medical decision process. One should be able to store these raw data, mainly captured by electronic medical devices, and then provide advanced analysis features.

2.3 Data warehouses for complex multimedia data

Multidimensional models usually consider facts as the dynamic part of data warehouses, and dimensions as static entities [21]: indeed, members of dimensions are computed once, and in a single way during the ETL step. However, this can restrict the analysis of data, particularly in the case of multimedia data. Indeed, multimedia data is bulky data, with various formats (text, graph, video, sound…). This data is generally described by content-based or description-based descriptors. Description-based descriptors are extrinsic

information (e.g. key words, date of acquisition, author, topic…). Content-based descriptors represent the content of data (e.g. color, texture, form …) and they are generally automatically extracted from the data [9, 30]. Another type of content-based descriptors is data-specific, i.e. descriptors computed by specific processes applied on the multimedia data, such as the various measurements calculated on an ECG signal. In multimedia data warehouses, descriptors frequently become dimensions of the multidimensional model. Warehousing multimedia data require specific extraction processes, adapted tools for visualization and the definition of specific multimedia aggregation functions.

Candidate descriptors for multimedia data are numerous and the same descriptor can be extracted in various ways, giving different, but still correct, values. These different computation modes can be seen as different functional versions of a descriptor. The choice of one of these different functional versions can depend on the user profile, his best-practices or habits, his level of expertise, or the type of the on-going experience, etc. Thus we think that dimensions could gain new semantics in integrating multiple methods to represent the data according to the various functional versions of each descriptor. Therefore, we argue that the integration of functional versions of dimension into the multidimensional model, corresponding to the data descriptors computed by different functions, can improve the characterization and the analysis of the data. Indeed, the user will be able to choose the computation modes for each descriptor in order to define the best interpretation and representation of the multimedia data. He will also be able to compare the different results obtained using different versions of computation modes for the descriptors. The aim of our study is to define a multidimensional conceptual model capable of managing multimedia data characterized by descriptors obtained by various computation modes. This model is illustrated by a case study on a medical multimedia data warehouse.

2.4 Related work on multimedia data warehouses

Various works about the design of multimedia data cubes were undertaken to improve multidimensional analysis of large multimedia databases [29]. The MultiMediaMiner project [30] uses a multimedia data cube in order to store the multidimensional data and to incorporate them with different granularities. In the medical domain, the problems of exploration and analysis of huge multimedia data are omnipresent. For example, [31] is a study undertaken in the domain of breast cancer detection, and proposes a data warehouse integrating numerical mammography. Another study [26] deals with the problem of storage and restitution of medical image data from a warehouse by comparing the data warehouse with a pyramid of means of storage. All these studies use multimedia data modeling based on data descriptors computed at loading time. Then, the design of the multimedia data cube follows the design of traditional data cubes paradigm, i.e. dimension members are computed during the ETL process and remain static.

Dynamic aspects in multidimensional models can be found in studies that deal with time-related evolutions. Two types of approaches are proposed in order to take evolutions of the analysis structures into account: the first one consists in updating models [2, 11, 12] that map data into the most recent version of the structure; the second is based on tracking history models [3, 7, 21, 24] that keep trace of evolutions of the system. This last type of model is particularly interesting because it allows analyzing data in their various versions and their evolutions. However these models focus on temporal evolutions and mapping processes applied on facts. They do not offer navigation into functional version of dimensions which could enable comparison between extraction and/or interpretation processes.

3 A case study in cardiology

In the case of disease studies, researchers try to extract information that best reflect the derangement from a chosen population. It is often difficult to establish, before the study, the set of descriptors that characterizes the multimedia data associated to a patient as well as the methodological approach to process these indicators. However, the design of the methods for information extraction is often a part of the research task. When several methods comete, the aim is to evaluate them. If the purpose is to study the characteristics and the evolution of descriptors taken of a specific population, the descriptor calculation is at least as significant as the selection of the raw data. In all these cases, one notes that an environment of analysis of medical multimedia data must jointly allow the selection of the data to be treated and the methods to be applied on the data for their processes. In order to organize a large set of multimedia data, to allow different ways to calculate their descriptors and to help to constitute study population, we propose a model for a multimedia data warehouse.

We collaborate with an INSERM lab (ERM 107) specialized in methodologies for cardiological information. The EMIAT project (European Myocardial Infarct Amiodarone Trial) objective is to extract knowledge about descriptors that can be used to evaluate a cardiac pathology evolution and to compare a drug called "Amiodarone" to a placebo. The study includes a total of 1,486 survivors of acute myocardial infarction. As a result, the study provides a significant amount of data to be exploited and analyzed. Among these data, are the Electrocardiogram signals (ECGs) (Fig. 3) of the various patients who are involved in the study.

Several descriptors or indicators can be computed from an ECG to characterize the cardiac health state of a patient. The QT interval measure (time after which the ventricles are repolarized) is the most studied [6]. The noise level is an often used criterion for the selection of a study population because it gives a quality indication. Other data are associated with these ECGs, such as the patient's principal pathology. Thus, two types of descriptors characterize ECGs:

- Description-based descriptors: e.g. principal pathology, age, gender of the patient, ECG acquisition slot, technology with which the ECG is obtained
- Content-based descriptors: e.g. the QT duration and the noise level of the signal.

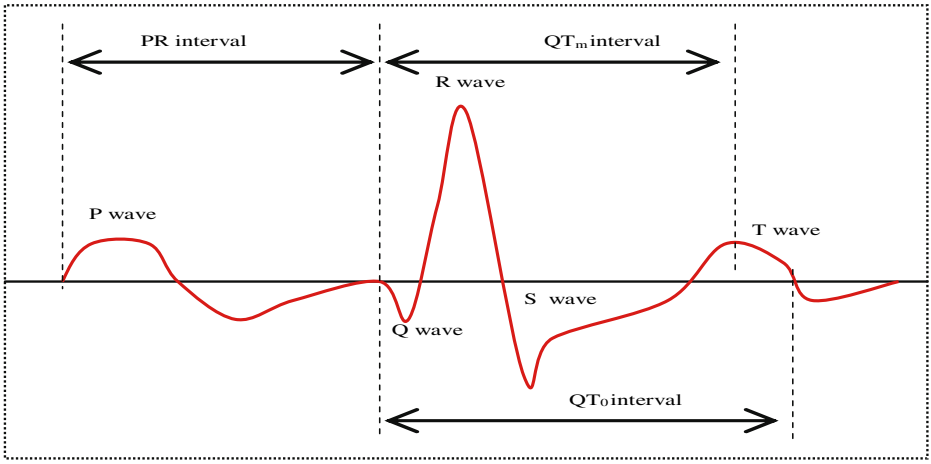


Fig. 3 ECG signal and some descriptors

In the multidimensional model, the studied facts are ECGs, characterized by descriptors corresponding to the dimensions, and organized in complex hierarchies. As an example, the ECG acquisition slots can be classified in hours and then in periods (night, waking, day). Some sets of hours (ex., 6 AM) can belong to several periods (waking and night). Thus, the dimension "Time" corresponding to this descriptor, is organized in a non-strict hierarchy. A simple multidimensional model of the data warehouse shows the fact table (#ECG represents the signal ECG) and the dimensions we use (Time, Duration of QT, gender...) in Fig. 4.

This data warehouse will be used in order to analyze the ECGs of a population with selected age, pathology, etc.... (Fig. 5). Some of the used descriptors can be computed by various computation modes. As an example, the QT duration can be obtained thanks to several algorithms: three threshold methods (T1, T2, T3) and two T wave slope methods (S1, S2). Thus this dimension must be a multiversion dimension in order to allow the user choosing the relevant computation mode (or the functional version) of this descriptor. Specific multiversion dimensions for the case study will be detailed in the prototype section.

On the one hand, our model will incorporate different versions of dimension (i.e. dimensions with members that are calculated according to various functional versions of descriptors), and on the other hand, it will allow explicit and complex hierarchies.

4 The multiversion model

4.1 General principle of our approach

Our approach is based on a fact table that contains the set of measures representing the data to analyze and dimensions representing the descriptors of this multimedia data. Taking into account the problem of functional multiversion, we redefine the multidimensional structure adding the concept of functional version. Therefore, we introduce the concept of version of dimension, multiversion dimension, functional multiversion fact table and function of version of dimension. A multiversion dimension is composed of several versions of a

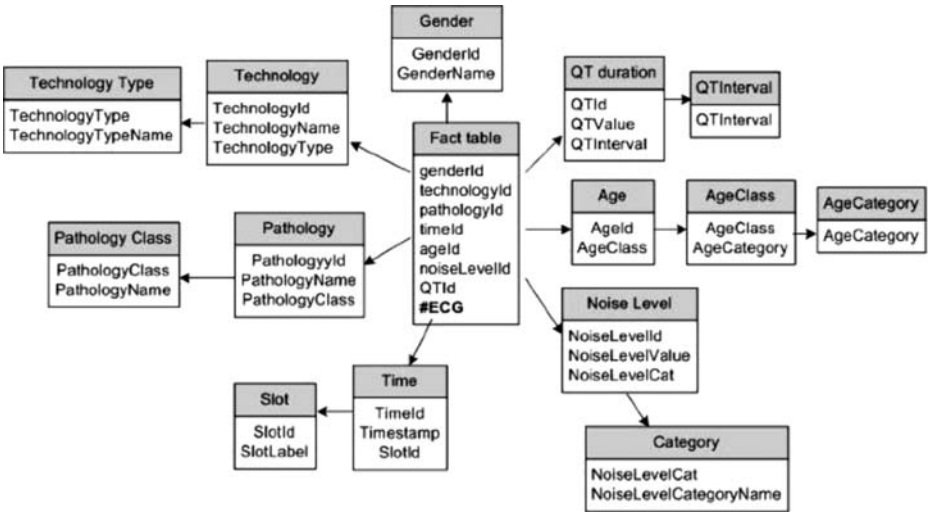


Fig. 4 Schema of the EMIAT data warehouse

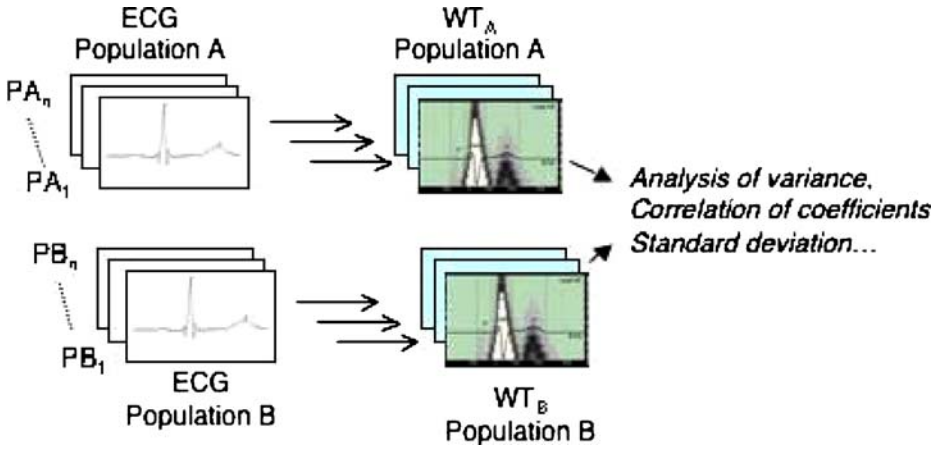


Fig. 5 Analysis process of ECGs

dimension, each one being a dimension for a given version with its own schema and its own instance. The functional multiversion fact table gathers all data, combining the various versions of dimension of a multiversion dimension with the others. Finally, the functions of versions of dimension correspond to the computation modes used to obtain the members of this version. The schemas of various dimensions are described using the hierarchical levels and the links that bind them. We also describe the instances of these dimensions with the set of members belonging to a hierarchical level and their parent–child relationships. Thus, our approach allows the design of explicit dimensions since the schemas of dimension are explicitly defined and our model supports also complex hierarchies (multiple, non-onto, non-strict and non-covering hierarchies) since the instances of dimensions are bottom-up defined from the members and the hierarchical links.

4.2 Concept definitions

Definition 1 (Schema of version of dimension) A schema of version of dimension is a schema of dimension for a given version. A version is a computation mode used to obtain the members of a dimension. The schema S_{VD} of the version of dimension VDid, is a tuple $< \text{VDid}, \mathcal{N}, <_{\text{VDid}} >$ where:

- VDid is the identifier of the version of dimension
- $\mathcal{N} = \{n_j , j=1,...,k\}$ is the set of levels of S_{VD}. A level in S_{VD} represents a set of values with the same granularity associated with the same version of dimension. A level n_j is a tuple $<levelId_j, levelName_j, [A_j], [description_j] >$ where:
  - $levelId_j$ is the identifier for the level of version of dimension
  - $levelName_j$ is the name for the level of version of dimension
  - A_j is an optional property representing descriptive attributes of this level
  - $description_j$ is an optional property representing textual information on the level n_j
- $<_{\text{VDid}}$ is a partial order on the set $\mathcal{N}$ that defines the hierarchical links between the levels of schema S_{VD} when the number of levels is more than 1. The partial order $<_{\text{VDid}}$ is defined such as: $\forall (n_1,n_2) \in \mathcal{N}\times\mathcal{N}$, if $n_1 <_{\text{VDid}} n_2$ then n_1 has a granularity finer than n_2.

Thus, a schema of version of dimension can be seen as a directed graph, where nodes are elements of the set $\mathcal{N}$ and arcs are relations according $<_{\text{VDid}}$. This graph must be acyclic in order to enable aggregations to the least fine hierarchical levels. A level ALL is defined as the root of the hierarchy, i.e. the highest level of granularity.

Example 1 Let suppose we want to analyze the influence of age on ECG. Age is a dimension of the data warehouse but its members can be ordered in various ways, the ages can be classified by intervals of age, i.e. 5 years interval, 10 years, 50 years.

Let $S_{\text{agePerInterval}}$ be a schema of the version of dimension "agePerInterval" of which VDid=1. The schema of this version of dimension is defined by:

$S_{\text{agePerInterval}} = < 1, \{n_1, n_2, n_3\}, <_1 >$ with

$n_1 = < 1, \text{“IntervalOf5”} >, n_2 = < 2, \text{“IntervalOf10”} >, n_3 = < 3, \text{“IntervalOf20”} >$

and the following order : $n_1 < 1 n_2, n_2 <_1 n_3$ *and* $n_3 <_1$ ALL

On the other hand these ages can be collected in age classes (young child, child, teenager, young adult, adult, senior) then in categories (minor, major) and the schema of this version of dimension $S_{\text{agePerClass}}$ can be defined.

Now let us consider the QT duration of these electrocardiograms as another dimension of the data warehouse. It can be computed by several algorithms, i.e. algo1 and algo2. The schema of dimension characterizing the duration of the QT is hierarchically structured by the values of the duration of the QT for the finest level, that are gathered within intervals of 100 ms, then 400 ms. Therefore, the schemas of these two versions of dimension S_{Qtalgo1} and S_{Qtalgo2} can be defined.

Definition 2 (Version of dimension) A version of dimension is a dimension for a given version. The version of dimension VD of schema $S_{\text{VD}} = <$ VDid, $\mathcal{N}$, $<_{\text{VDid}}>$ is a tuple < VDid, VDname, $\mathcal{M}$, $<_{\text{VD}}$, [VDdescription] > where:

- VDid is the unique identifier for the version of dimension
- VDname is the name for the version of dimension
- $\mathcal{M}=\{m_j, j=1...l\}$ is the set of members of this version of dimension. A member of a version of dimension is a member computed by the computation mode related to this version . It belongs to one of the levels of the schema S_{VD}. Thus, level is composed of members gathered with the same granularity. A member m_j is a tuple $< \text{id}_j, \text{val}_j, [a_j], \text{levelId}_j >$ where:
  - id_j is a unique identifier for this member of version of dimension
  - val_j is the value for this member of version of dimension.
  - a_j is an optional property that contains the set of values of the attributes related to this member (corresponding to the level). If this property is defined for the level corresponding to the member, then it must be defined for the member.
  - levelId_j is the identifier for the hierarchical level to which this member of version of dimension belongs.
- $<_{\text{VD}}$ is a partial order on the set $\mathcal{M}$ that defines hierarchical links between the members of the same version of dimension VD. For each pair of levels (n_1,n_2), such as $n_1 <_{\text{VDid}} n_2$, there exist at least a couple $(m_1, m_2) \in \mathcal{M} \times \mathcal{M}$, such as $m_1.\text{levelId}_1 = n_1$ and $m_2.\text{levelId}_2 = n_2$ and $m_1 <_{\text{VD}} m_2$. Thus, m_1 is said to be a member of lower level than m_2, i.e. m_1 has a finer granularity than m_2 .

- VDdescription is an optional property containing possible comments on the version of dimension.

Thus, a version of dimension can be represented by an acyclic directed graph, where nodes are elements of the set $\mathcal{M}$ and arcs are relations according to $<_{VD}$. In the following, we will call "leaf member" of version of dimension, a member of a version of dimension that has no child. Moreover, the member "all" is defined as the unique member contained in level "ALL."

One notes $\mathcal{LM}_{VD}$ the set of leaf members of the version of dimension VD. This set is defined as follows:

$$\mathcal{LM}_{VD} = \{m_j | m_j \in \mathcal{M} \text{ and } \neg\exists m_i \in \mathcal{M} / (i \neq j \text{ and } m_i <_{VD} m_j)\}.$$

Example 2 The version of dimension "agePerInterval" with the schema $S_{agePerInterval}$ is presented in the previous example is defined by the tuple: <1,"agePerInterval", $\{m_1,.....,m_7\}$,$<_a$ > with

$$m_1 = <1, 0-5, 1>, m_2 = <2, 6-10, 1>, m_3 = <3, 11-15, 1>, m_4 = <4, 16-20, 1>,$$

$$m_5 = <5, 0-10, 2>, m_6 = <6, 11-20, 2>, m_7 = <7, 0-50, 3>$$

and the following order :

$$m_1 <_a m_5, m_2 <_a m_5, m_3 <_a m_6, m_4 <_a m_6, m_5 <_a m_7, m_6 <_a m_7 \text{ and } m_7 <_a all.$$

So, the members of the level n_1 ("IntervalOf5") are $\{m_1, m_2, m_3, m_4\}$, those of the level n_2 ("IntervalOf10") are $\{m_5, m_6\}$ and that of the level n_3 ("IntervalOf20") is $\{m_7\}$.

The set of $LM_{agePerInterval}$ is defined by: $LM_{agePerInterval} = \{m_1, m_2, m_3, m_4\}$

The version of dimension "agePerInterval" can be represented by a graph (Fig. 6).

In the same way, the versions of dimension "agePerClass", "QTalgo1" and "QTalgo2" whose the schemas are $S_{agePerClass}$, $S_{QTalgo1}$, $S_{Qtalgo2}$ as well as the sets $\mathcal{LM}_{agePerClass}$, $\mathcal{LM}_{QTalgo1}$ and $\mathcal{LM}_{Qtalgo2}$ can also be defined.

Definition 3 (Multiversion dimension) A multiversion dimension MVD is a dimension that contains *1* to *n* versions of dimension. It is a tuple < MVDId, MVDname, VD, [MVDdescription] > where:

- MVDId is the unique identifier for the multiversion dimension
- MVDname is the name for the multiversion dimension
- VD = *{*VD_i*, i=1,...,n}* is the set of versions of dimension associated with this multiversion dimension

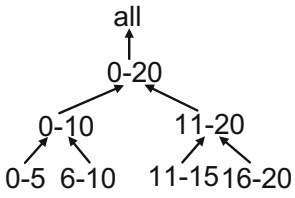


Fig. 6 Version of dimension "agePerInterval"

- MVDdescription is an optional property containing textual information on the multiversion dimension.

One notes $\mathcal{LM}_{\mathrm{MVD}}$ the set of leaf members of the versions of dimension contained in the multiversion dimension *MVD*. This set is defined by:

$\mathcal{LM}_{\mathrm{MVD}} = \bigcup_{i=1}^{n} \mathrm{LM}_{\mathrm{VD}_i}$ with n the number of versions of dimension contained in the multiversion dimension MVD.

Example 3 The versions of dimension "agePerInterval " and "agePerClass" previously defined belong to the multiversion dimension "Age" with the identifier 1. This multiversion dimension is defined by:

$$\mathrm{Age} = < 1, \text{“Age”}, \{\text{“agePerInterval”}, \text{“agePerClass”}\} >$$

In the following the names of the members of versions of dimension will be used in order to identify them. One defines the set $\mathcal{LM}_{\mathrm{Age}}$ by:

$$\mathcal{LM}_{\mathrm{Age}} = \{0-5, 6-10, 11-15, 16-20, \mathrm{youngchild}, \mathrm{child}, \mathrm{teenager}, \mathrm{youngadult}, \mathrm{adult}, \mathrm{senior}\}$$

The versions of dimension "QTalgo1" and "QTalgo2" belong to multiversion dimension "DurationQT" with the identifier 2. This multiversion dimension is defined by:

$$\mathrm{DurationQT} = < 2, \mathrm{QT}, \{\mathrm{QTalgo1}, \mathrm{QTalgo2}\} >$$

The set $\mathcal{LM}_{\mathrm{DurationQT}}$ is defined the same way.

Definition 4 (Functional multiversion fact table) A functional multiversion fact table provides the measures according to various versions of dimension. Let $\{\mu_i, i=1,...,m\}$ be the set of measurements, a functional multiversion fact table *ft* is defined by a function such as:

$$\mathrm{ft} : \mathrm{MVD}_1 \times \mathrm{MVD}_2 \times \ldots \times \mathrm{MVD}_n \rightarrow \mathrm{dom}(\mu_1) \times \ldots \times \mathrm{dom}(\mu_m)$$

$$m_1, m_2, \ldots, m_n \mapsto v_1, \ldots, v_m$$

where:

- n is the number of multiversion dimensions of the data warehouse, $m_i \in \mathcal{LM}\ \mathrm{MD_i}$ with $i=1,...,n$
- $\mathrm{dom}(\mu_k)$ is the range for the measure μ_k .

This function associates the set of the values v_k of measures μ_k with a set of leaf members of the versions of dimension of each multiversion dimension.

Definition 5 (Function of version of dimension) The functions of version of dimension are the computation modes that allow to calculate the members of a version of dimension VD from the data of the production database. A function of version of dimension f_{VD} is a tuple <$\mathrm{functionId_{VD}}$, VDid, $\mathrm{functionName_{VD}}$, $\mathrm{functionDefinition_{VD}}$ > where:

- $\mathrm{functionId_{VD}}$ is the identifier for the function of version of dimension VD
- VDid is the identifier for the version of dimension VD whose members are computed using this function of version of dimension

- functionName$_{\mathrm{VD}}$ is the name for the function of version of dimension
- functionDefinition$_{\mathrm{VD}}$ is the definition for the function of version of dimension

These functions can be formalized as following:

$$f_{\mathrm{VD}} : \mathcal{DB}_f \rightarrow \mathcal{LM}_{\mathrm{VD}}$$

$$d \mapsto m$$

where $\mathcal{DB}_f$ is the set of data of the production database restricted to f_{VD} i.e. used to compute the members of VD. f_{VD} associates a value of the production database and a leaf member of the version of corresponding VD.

Example 4 Let the function $f_{\mathrm{agePerClass}}$ be defined for the version of dimension "agePerClass" and the defined function $f_{\mathrm{agePerInterval}}$ for the version of dimension "agePerInterval." For a 12-year-old patient, we obtain respectively for the versions of dimension, the members:

$$f_{\mathrm{agePerClass}}(12) = \text{“young”} \text{ and } f_{\mathrm{agePerInterval}}(12) = \text{“}11 - 15\text{”}.$$

Let the function f_{Qtalgo1} be defined for the version of dimension "Qtalgo1" and the defined function f_{Qtalgo2} for the version of dimension "Qtalgo2." Then for a given electrocardiogram "ECG5," we obtain, respectively for the versions of dimension, the members:

$$f_{\mathrm{Qtalgo1}}(\mathrm{ECG5}) = 100 \text{ and } f_{\mathrm{Qtalgo2}}(\mathrm{ECG5}) = 110.$$

Definition 6 (Functional multiversion multidimensional structure) A functional multiversion multidimensional structure F2M is a tuple $< \mathcal{MVD}, \mathrm{ft}, \mathcal{F} >$ where:

- $\mathcal{MVD} = \bigcup_{i=1}^{s} \mathrm{MD}_i$ is the set of all the multiversion dimensions
- ft is the functional multiversion fact table
- $F = \bigcup_{j=1}^{r} f_{\mathrm{VD}_j}$ is the set of all the functions of the versions of dimension

Definition 7 (Aggregation of data) Aggregation of data can be computed from the multiversion fact table and the schema of the versions of dimension. Let an aggregation function φ_k be defined for each measure μ_k, m a non-leaf member of the version of dimension VD of the multiversion dimension MVD$_I$ and $m_1^1, m_2^1, \ldots, m_J^1$ its leaf members i.e. such as:

$$\left(m_1^1, m_2^1, \ldots, m_J^1\right) \in \mathcal{LM}_{\mathrm{VD}} \times \ldots \times \mathcal{LM}_{\mathrm{VD}}$$

For each leaf member m_p^1 and the other leaf members $m_2,\ldots, m_n$ of the relying dimensions the ft function gives the value of the measures: $\forall p \in [1, J], ft\left(m_p^1, m_2, \ldots, m_n\right) = v_1^p, \ldots, v_m^p$ with n being the number of multiversion dimensions of the data warehouse.

Thus, one can obtain the aggregated values for m as:

$$\mathrm{ft}(m, m_2, \ldots, m_n) = \underset{p=1}{\overset{J}{\phi}} v_1^p, \ldots, \underset{p=1}{\overset{J}{\phi}} v_m^p$$

i.e. applying the associated aggregation function on each measure. The aggregation function can be a classic function of OLAP technology (sum, min, max, avg) for numeric measures or a more specific function (statistic average...) for signal or image type measures.

5 Prototype

The model has been implemented following three-tier architecture:

- A functional multiversion multimedia data warehouse in which the multiversion dimensions and the functional multiversion fact table are stored,
- An OLAP cube built from the functional multiversion multimedia data warehouse, using aggregations, which enables requests against the functional versions of the dimensions,
- A tool for data navigation and visualization.

We use Microsoft SQL Server and Analysis Services to implement the prototype. The aggregations functions are implemented in Visual Basic; an interface using Proclarity 4.0 is built for navigation. The data is loaded from the production database to the functional multiversion multimedia data warehouse using the functions of versions of dimension.

The data of the EMIAT study have been integrated in the prototype. The multiversion dimension tables and functional multiversion fact tables are populated with an ETL (Extraction, Transformation, and Loading) tool. The data warehouse contains a functional multiversion fact table and seven multiversion dimensions.

The facts are ECGs signals from the EMIAT study. According to the schema of the EMIAT data warehouse (Fig. 4), dimensions represent the description-based descriptors and the content-based descriptors of these ECGs:

- Three dimensions are related to the patient (principal pathology, age, and gender),
- Two dimensions are related to ECG acquisition (time and technology),
- Two dimensions are related to the content of the ECGs (the QT duration, the noise level).

Among these dimensions, we implement three multiversion dimensions: the QT duration dimension and the noise level dimension because members of these dimensions can be computed by various algorithms and the Age dimension because the members of this dimension can be classified following different classifications. Aggregations of data are computed from the functional multiversion fact table and the hierarchical links between the members of the versions of dimension. The aggregation functions allow computing aggregated data according to granularities of the versions of dimension (the levels of the corresponding schemas). In our data warehouse, we define the following aggregation functions:

- “ECG-count”: this function counts the number of ECGs that correspond to the characteristics selected by the user.
- “ECG-list”: this function returns the list of ECGs that correspond to the characteristics selected by the user.
- “Average-ECG”: this function gives the “medium-ECG” calculated on the ECG-list.

The user can visualize in a frame the hierarchies and the members of the versions of dimensions and build his request choosing the appropriate data aggregation operator (“ECG-count,” “ECG-list” or “average-ECG”) (Fig. 7, part C). The result is a

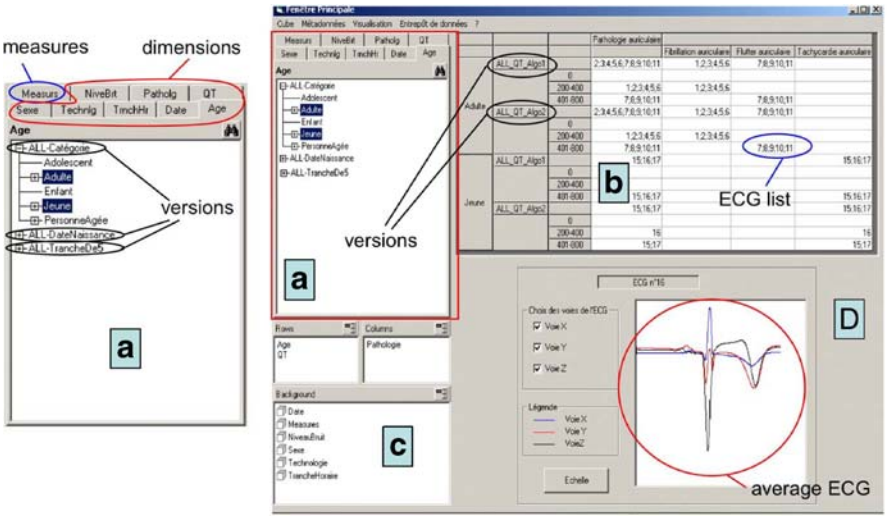


Fig. 7 Navigation interface

multidimensional table (Fig. 7, part B). It is also possible to visualize and to choose several versions of dimension of a multiversion dimension (Fig. 7, part A). In this way, the user can select the computation modes of the members of dimensions and compare resulting measures in the same multidimensional table. The selected multimedia data can be visualized in a frame (Fig. 7, part D). Moreover, metadata can be visualized in order to have a global view of all versions of dimension and to navigate the data cube more easily, e.g. schemas, instances of each version of dimension.

6 Discussion and conclusion

We propose a model that takes functional versions into account. The model helps to manage multimedia data and allows the user to choose different views that represent various functional versions of the descriptors. We define the concept of version and multiversion in dimensions in order to compare results obtained by various versions. This model is particularly well suited to multimedia data because they require various computation modes. We also use specific aggregation functions for multimedia data that are integrated into the data warehouse and in the OLAP engine. This model is used to develop an OLAP application for the navigation into a hypercube integrating signal data. We propose a tool to explore this complex data which improves navigation in the multidimensional data cube. Thus, we enable the visualization of data according to several methods of analysis and we provide the possibility to visualize the representation of this multimedia data.

However, the data storage of our model could be improved. It is possible to have some redundancy in the schemas of versions of dimension, the functional multiversion fact table storage is not optimised, and the treatment of non-strict and multiple hierarchies imply duplications. Our model could be extended by associating the notion of functional version with the facts, in the same way that our model associates functional versions with dimensions. This could be possible by adding a version of fact dimension so as to enable the user to choose the version of fact.

References

1. Agrawal R, Gupta A, Sarawagi S (1995) Modeling multidimensional databases. IBM Research Report, IBM Almaden Research Center, September 25p
2. Blaschka M, Sapia C, Höfling G (1999) On schema evolution in multidimensional databases. In: Data Warehousing and Knowledge Discovery, First International Conference, DaWaK '99, Florence, Italy, August 30–September 1, 1999, Proceedings. Lecture Notes in Computer Science 1676 Springer, ISBN 3-540-66458-0
3. Body M, Miquel M, Bédard Y, Tchounikine A (2003) Handling evolutions in multidimensional structures. In: Proceedings of the 19th International Conference on Data Engineering, March 5–8, 2003, Bangalore, India. IEEE Computer Society, ISBN 0-7803-7665-X
4. Cabibbo L, Torlone R (1998) A logical approach to multidimensional databases. In: Advances in Database Technology—EDBT'98, 6th International Conference on Extending Database Technology, Valencia, Spain, March 23–27, 1998, Proceedings. Lecture Notes in Computer Science 1377 Springer, ISBN 3-540-64264-1.
5. Chaudhuri S, Dayal U (1997) An overview of data warehousing and OLAP technology. SIGMOD Record, 26(1):65–74
6. Chevalier P, Rodriguez C, Bontemps L, Miquel M, Kirkorian G, Rousson R, Potet F, Schott JJ, Baró I, Touboul P (2001) Non-invasive testing of acquired long QT syndrome: evidence for multiple arrhythmogenic substrates. Cardiovasc Res 50(2):386–398 May
7. Eder J, Koncilia C (2001) Evolution of dimension data in temporal data warehouses. In: Proceedings of the Data Warehousing and Knowledge Discovery, Third International Conference, DaWaK 2001, Munich, Germany, September 5–7, 2001, Lecture Notes in Computer Science 2114 Springer, ISBN 3-540-42553
8. Gyssens M, Lakshmanan LVS (1997) A foundation for multi-dimensional databases. In: Proceedings of the 23rd International Conference on Very Large Data 9. Bases, August 25–29, 1997, Athens, Greece. Morgan Kaufmann, ISBN 1-55860-470-7
9. Han J, Kamber M (2001) Data mining, concepts and techniques. Morgan Kaufmann
10. Hristovski D, Rogac M, Markota M (2000) Using data warehousing and OLAP in public health care. In: Proceedings AMIA Symp: 369–373
11. Hurtado C, Mendelzon AO, Vaisman A (1999) Maintaining data cubes under dimension updates. In: Proceedings of the 15th International Conference on Data Engineering, 23–26 March 1999, Sydney, Australia, IEEE Computer Society Press
12. Hurtado C, Mendelzon AO, Vaisman A (1999) Updating OLAP dimensions. In: Proceedings of the ACM Second International Workshop on Data Warehousing and OLAP, November 6, 1999, Kansas City, Missouri, USA, Proceedings. ACM
13. Huyn N (2001) Data analysis and mining in the life sciences. Sigmod Record, 30(3):76–85 September
14. Inmon WH (eds) (1996) Building the data warehouse, 3rd edn, Wiley, New York
15. Isken MW, Littig SJ, West M (2001) A data mart for operations analysis. J Healthc Inf Manag (JHIM) 15 (2):Summer
16. Jarke M, List T, Köller J (2000) The challenge of process data warehousing. In: Proceedings of 26th International Conference on Very Large Data Bases, September 10–14, 2000, Cairo, Egypt. Morgan Kaufmann, ISBN 1-55860-715-3
17. Kamp V, Wietek F (1997) Database system for multidimensional data analysis. In: Proceeding of the International Database Engineering and Application Symposium 1997 (IDEAS) Concordia University, Montreal, Canada, August 25–27
18. Kimball R (1996) The data warehouse toolkit. Wiley, New York
19. Lehner W (1998) Modeling large OLAP scenarios. In: Proceedings of the 1998 International Conference on Extending Database Technology, Valencia, Spain
20. Li C, Sean Wang X (1996) A data model for supporting on-line analytical processing. In: Proceedings of the Fifth International Conference on Information and Knowledge Management, November 12–16, 1996, Rockville, MD, USA. ACM
21. Mendelzon AO, Vaisman A (2000) Temporal queries in OLAP. In: Proceedings of 26th International Conference on Very Large Data Bases, September 10–14, 2000, Cairo, Egypt. Morgan Kaufmann, ISBN 1-55860-715-3.
22. Niinimäki J, Selén G, Kailajärvi M, Grönroos P, Irjala K, Forsström JJ (1996) Medical data warehouse, an investment for better medical care. Medical Informatics in Europe (MIE'96), Copenhagen, Denmark, August
23. Pedersen TB, Jensen C (1998) Research issues in clinical data warehousing. In: 10th International Conference on Scientific and Statistical Database Management, Proceedings, Capri, Italy, July 1–3, 1998. IEEE Computer Society, ISBN 0-8186-8575-1

24. Pedersen TB, Jensen C, Dyreson C (2001) A foundation for capturing and querying complex multidimensional data. Inf Syst 26(5): Special issue: Data Warehousing
25. Tchounikine A, Miquel M, Flory A (2001) Information warehouse for medical research. In: Data Warehousing and Knowledge Discovery, Third International Conference, DaWaK 2001, Munich, Germany, September 5–7, 2001, Proceedings. Lecture Notes in Computer Science 2114 Springer, ISBN 3-540-42553-5
26. Tikekar RV, Fotouhi F, Ragan D (1995) Storage and retrieval of medical images from data warehouses. Digital Image Storage and Archiving Systems
27. Vassiliadis P (1998) Modeling multidimensional databases, cubes and cube operations. In: 10th International Conference on Scientific and Statistical Database Management, Proceedings, Capri, Italy, July 1–3, 1998. IEEE Computer Society, ISBN 0-8186-8575-1
28. Vassiliadis P, Sellis T (1999) A survey of logical models for OLAP databases. SIGMOD Record, 28 (1):64–69, March
29. You J et al. (2001) On hierarchical multimedia information retrieval. In: Proceedings of the 2001 International Conference on Image Processing (ICIP 2001), Thessaloniki, Greece, October 7–10 IEEE.
30. Zaïane OR, Han J, Li ZH, Hou J (1998) Mining multimEdia data. In: Proceedings of CASCON'98: Meeting of Minds, pp 83–96, Toronto, Canada, November
31. Zhang H et al. (2001) Developing a digital mammography data warehouse. Medical Imaging
32. Zücker R, Kietz JU, Vaduva A (2001) Mining mart: metadata-driven preprocessing. In: Principles of Data Mining and Knowledge Discovery, 5th European Conference, PKDD 2001, Freiburg, Germany, September 3–5, 2001, Proceedings. Lecture Notes in Computer Science 2168 Springer, ISBN 3-540-42534-9.

Anne-Muriel Arigon is a Ph.D. candidate in the Laboratory of Biométrie et Biologie Evolutive at the University Claude Bernard of Lyon, France. She received her M.Sc. degree in Computer Science from the French engineering school INSA (Institut National des Sciences Appliquées) of Lyon, France in 2003. Her research interests include Data warehouses and Bioinformatics area.

Maryvonne Miquel is associate professor at the Department of IT and Computer Engineering—INSA (Institut National des Sciences Appliquées de Lyon) since 1989. She received PhD in Computer Sciences in 1987 (INSA Lyon) and habilitation to supervise research in 2005 (University Lyon 1 and INSA Lyon). She is a member of the LIRIS research laboratory (the Lyon Research Center for Images and Intelligent Information Systems, CNRS UMR 5205). Her research interests include data warehouse, advanced multidimensional models and OLAP systems.

Anne Tchounikine received a PhD in computing science in 1993 from the University of Toulouse (France). Since 1996, she is associate professor at LIRIS, the Lyon Research Center for Images and Intelligent Information Systems (France) and at the Computer/IT Department of the National Institute for Applied Sciences (INSA). Her current research is on advanced multidimensional models and OLAP systems.