## Health Care Management Information Systems with Object Oriented Methodology

Kanta Jangra<sup>1</sup> and Kavita Dua<sup>2</sup>

<sup>[1]</sup>Research Scholar, NIMS University, Jaipur <sup>[2]</sup>Associate Professor, OITM, Hisar <u>kanta.anuj@gmail.com</u>

ABSTRACT: - In this paper, we present the methodology we adopted in designing and developing an objectoriented database system for the management of medical records. The designed system provides technical solutions to important requirements of most clinical information systems, such as 1) the support of tools to create and manage views on data and view schemas, offering to different users specific perspective on data tailored to their needs; 2) the capability to handle in a suitable way the temporal aspects related to clinical information; and 3) the effective integration of multimedia data. Remote data access for authorized users is also considered. As clinical application, we describe here the prototype of a user-oriented clinical information system for the archiving and the management of multimedia and temporally oriented clinical data related to percutaneous transluminal coronary angioplasty (PTCA) patients. Suitable view schemas for various user roles (cath-lab physician, ward nurse, general practitioner) have been modeled and implemented on the basis of a detailed analysis of the considered clinical environment, carried out by an object-oriented approach.

*Index Terms*—Cardiology data, clinical information systems, object-oriented databases, object-oriented views, temporal databases, WWW-based information access.

### 1. INTRODUCTION

Usually, in an information system, data archiving and management functions are performed using database management systems (DBMS). Depending on the complexity level of the considered organization, a DBMS may have various groups of users, wishing to access stored information with a different perspective as function of their role and of the performed activities. Some users may be interested in data not representative for others, or they may be interested in the same information but presented in a different way (as for instance a table containing data or a graphic representing the trend of the same data). The head of a cardiology department could be interested in overviews of multipatient data, significant to evaluate the quality of care and the work charges of the division, to schedule forthcoming interventions, to make technology and costs assessments, and so on. On the other hand, a physician working in the ward would be more interested in data related to a single patient, for clinical purposes. His/her goals can be the serial comparison of results of exams performed in different follow-up visits, the management of the clinical history of a patient, or the diagnostic process in uncertainty conditions.

Although data organization in patient folders has been the object of research and discussion for decades, no unique solution able to satisfy requirements of all the users has been found [1], [2]. Several different medical record structures have been proposed in the literature (source-oriented, problem-oriented, timeoriented, management-oriented, specialty-based, and so on), but no model has emerged that can be sharable by everyone [2].

In our approach we consider the following as strategic items for a successful system design and development.

- 1. A comprehensive analysis of the studied scenario, in particular concerning the user roles and activities, to be carried out according to an object-oriented methodology coming from the field of business process modeling [3].
- 2. The use of object-oriented data models and languages for the development of the clinical information system. This choice allows a suitable modeling of temporal and multimedia aspects related to patient data, and the development of tools to support the creation and the management of views and view schemas. In fact, objectoriented views offer additional functions to those supported

by relational ones, induced by the higher expressive power of the object-oriented approach. Among these, the possibility to update and create new database objects through views, the integration of heterogeneous databases, the potentiality to provide multiple interfaces to the same conceptual entity, and the simulation of database schema evolution [14]–[19].

3. The use of a system architecture allowing remote data access to authorized users through the Internet and the WWW.

### 2. BACKGROUND

## A. Business Modeling Techniques Applied to the Health Car Environment

An information system can be defined as the totality of the activities and structures dealing with the management of information flows within an organization, independently from the level of automation involved [21], [22]. The development process of an information system and of the related hardware and software infrastructure can be decomposed in three main steps [22]:

- analysis of the considered application context and of system requirements (particular attention should be devoted to the comprehension of users needs);
- design of a suitable system architecture;
- implementation of the information system.

## B. Views

Views are constructs allowing the creation of "dynamic windows," defined according to the needs of different users or group of users, on data stored in a database [3]. The use of views permits to profile user needs and to create a suitable organization of pertinent data and knowledge. A view may be a subset of the database or it may contain *virtual* data that is derived from the database but is not explicitly stored. Views can be applied to integrate data from different databases, to define content-based authorizations, and to simulate schema evolution.

Most of the medical database systems supporting views are based on the use of relational databases both for data archiving and management and for view definition. Additional functions must be supported by object-oriented views compared to those supported by views in relational databases: the possibility to create and update database objects through views, thanks to object identifiers; overriding, projection and addition of new attributes and methods to database types; the management of the view inheritance relationships, and so on.

### **3. DESIGN CONSIDERATION**

### A. Analysis of Application Contexts

To design and implement a clinical database system, the first task is the identification of the application domain (the universe of discourse), of the environment, its services and its objectives. Then, an important step concerns the identification of all possible users of the system and of their needs when performing the activities they carry out in normal clinical practice. We adopted the semantic object model (SOM), an object- oriented approach previously developed in the field of economics for the analysis and the design of business processes and business application systems [11] SOM supports a stepwise and hierarchical modeling procedure allowing multiple levels of abstractions.



Fig. 1. Interaction schema related to the management of a PTCA patient in a clinical division according to the SOM

Using SOM, hospital processes can be represented by two graphs: the *interaction schema* and the *task-event schema*. The first one is devoted to model the structural view of processes: all the objects belonging to the considered context are represented, together with their interactions, as client-server connections. Task-event schemas model the corresponding behavioral view with a time-dependent representation of sequences of tasks that are to be carried out when performing an activity.

Fig. 1 shows the interaction schema related to the management of a PTCA patient within a clinical division. All the objects belonging to the considered scenario, both internal (represented by squared boxes) and external to the hospital (represented by circles), are modeled. Arrows

represent interactions occurring between objects, called transactions, that can be both information flows and services provided by an object to another. To limit the complexity of the schema, the transactions indicated in Fig. 1 are not labeled by their specific names. Fig. 2 depicts the task-event schema corresponding to the interaction schema in Fig. 1, concerning the case of a patient undergoing an emergency PTCA treatment. The modeled situation regards the case of a successful PTCA treatment: the patient after the treatment is not transferred to the intensive care unit (ICU).

#### B. Supporting Views and View Schemas

1) The Data Model: The designed system is based on an object- oriented data model, which supports all the main characteristics of object-oriented systems (object identity, encapsulation, persistence, single inheritance, polymorphism, management of complex objects, etc.) [1]. Each entity of the modeled world is represented by an object. Objects are created as instances of a *class* (the classes belonging to the model have here the notation LMI\_*name of the class*, while the classes belonging to a given application have the notation LMI\_A\_*name of the class*). In the following, we will use the term *class* and the term *type* as synonyms, to describe the proposed data model [5].

Besides some basic types (*char, string, int, real, array*, ) the proposed data model, based on granular clinical history—object oriented data model (GCH-OODM) [6] supports other predefined data types to represent the temporal aspects related to clinical information.



Fig. 2. Task-event schema corresponding to the interaction schema represented in Fig. 1, related to the management of a patient undergoing an emergency successful PTCA treatment (the patient is not transferred to the ICU).

In fact, time is important in clinical medicine for almost every decision making action (diagnosis, therapy, prognosis): each action occurs at some time point and is based on temporal information. In order to make a diagnosis, for instance, the physician needs to know the clinical history of the patient, composed by previous pathologies, therapies and symptoms. All these facts have a temporal dimension that can be expressed either by using the concept of interval, for facts having a time span, or the concept of instant, for events occurring at time points [7]. In particular, the model allows the specification of the valid time of information, i.e., the time during which the information is true in the modeled world. The model does not handle the transaction time related to data entry in the database.

2) Extending GCH-OODM With Views: The GCH-OODM data model has been extended with new tools and constructs to support the definition and the management of views. Several approaches have been proposed to model views in object-oriented database systems [8]. Views can be either not materialized or materialized. In the first case the view extent, i.e., the instances of the view, are computed on demand when the query related to the view is processed. In the case of materialized views, instead, the view population is stored in the database. In general, besides being based upon different data models and exploiting different query languages to express view populations, the proposals described in the literature differ by: 1) the functions supported by the view mechanism (shorthand for queries, external schema definition, schema evolution, data protection, etc.); 2) the positioning of views in the database schema; 3) the properties assigned to view objects (depending on whether or not view objects have persistent object identifiers); and 4) the possibility to perform updates of base objects through views [3].

Finally, the view AggregationView allows the definition of views based on more than one base class; this kind of view has been designed mainly to aggregate data otherwise spread in several classes. For example, we can define a view LMI A ClinicalHistory AggregationView, defined on the base classes LMI\_A\_Patient, LMI A Angiovisit, LMI A Diagnosis, and LMI A Therapy aggregating all the events belonging to the clinical history of a patient. Aggregation views can be either materialized or not materialized. The views View and MaterializedView can be used for data entry, deletion and update. Results of these operations are consistently propagated to the instances of the base classes of the database, as well as to other possible views defined on the base of the same classes. To support the creation of views, we developed a suitable viewdefinition language. This language offers to the database administrator three different statements for view definition:

CreateView, CreateMaterializedView, and CreateAggregationView.

By these statements it is possible:

- to define views derived from base classes of the database;
- to specify their structure (attributes and methods);
- to define the query that has to be executed on the instances of the pertinent database class or classes to obtain the view extent, i.e., the object instances populating the view;
- to indicate which are the user roles for whom the view has been defined (the considered view will belong to all the view schemas defined for the user roles indicated in the statement).

The view definition statements CreateMaterialized- View and CreateAggregationView allow also the specification of the temporal validity of the created view. This is not related to the valid time of the instances of the related base class, but to the scope for which the view has been defined: for instance, a view can be defined for a particular study that has a limited time duration

3) View Schemas: Although no universally accepted definition of "view schema" has been found in the literature, a common definition considers a view schema as a set of base classes and views linked by some kind of relationship. In the proposed system, since classes and views belong to two different inheritance hierarchies, a view schema can contain just HAS\_A and view-derivation relationships.

The system has been designed to support multiple view schemas in order to satisfy the needs of all the considered user roles. It has to be noted that, while the database schema has to be defined during the design phase, view schemas can be defined and modified in every moment of the life of the database. View schemas provide thus a powerful way to adapt the database system to the evolving needs of its users. In the considered application domain, the object database schema allows the user to integrate time-oriented clinical data and images acquired during hospitalization, follow-up visits, and diagnostic examinations of patients subjected to PTCA [4], [9]

The view schema defined for the cath-lab physicians, , contains views ruling the access modality (read, insert, modify) to patient data: information modeled by the classes *LMI\_A\_person*, *LMI\_A\_demographic\_data*, *LMI\_A\_risk\_factor* and *LMI\_A\_angiovisit* can only be read (for this reason corresponding views have been defined),

while all access modalities are allowed for the classes *LMI\_A\_diagnosis*, *LMI\_A\_therapy* and *LMI\_A\_angio\_exam*, thus belonging to the view schema.

### C. Enabling Remote Access to Clinical Data

As mentioned above, the main goal of the proposed system is to provide, to each group of users working in the considered context, a suitable perspective of the managed clinical information. Some of the system users perform extramural activities, this is the case of general practitioners referring patients to the clinical division. To offer also these users the possibility to access data related to their patients without leaving their usual working place, we adopted an architecture allowing a remote consultation of patient medical folders. The communication infrastructure is based on the use of Internet and the WWW [2].

## 4. SYSTEM DESCRIPTION

The clinical database system has been implemented using the object oriented database management system Ode for Unix workstations [5]. To support remote data access we used the Apache Web server, a secure HTTP server available on most of the existing hardware platforms. The user interface has been developed using Java language; client application requirements are limited to a Web browser supporting Java applets.

In this prototyping phase, data security and privacy issues have not been faced in deep in implementing the system. Nevertheless, we considered and analyzed the problem of data security for clinical information. The protocol SHTTP, based on SSL (secure sockets layer), seems to have some suitable features for the exchange of clinical data. On the other hand, Java classes are commercially available, allowing the use of cryptographic techniques.

The field of security and privacy on the WWW is rapidly changing and growing. In this context, it is worth to underline that the object-oriented architecture of the proposed system allows the modification of the adopted (secure) communication protocols, without affecting the software related to the clinical application. The database system has a modular architecture. Moving from the bottom to the top, at the lower level are indicated the tools provided by the OODBMS Ode and some further base classes implemented to support features not yet provided by the current version of Ode database (*LMI\_Object, LMI\_Set, LMI\_iterator*)



Fig. 8 illustrates the overall architecture of the proposed

system.

Then there are the classes we defined to handle the related clinical information temporal aspects to (LMI\_el\_time, LMI\_instant, LMI\_duration, LMI\_interval, LMI\_t\_object, LMI\_granularity, LMI\_bool3) and to support management view creation and (views View. MaterializedView, AggregationView). On the top of these class libraries and tools, we find the classes belonging to the clinical database and, above these, the view schemas defined for different user roles. This is the layer immediately under the client application, developed

as Java applet and interacting with the database server through the communication infrastructure previously illustrated.

## 5. DISCUSSION

### A. View Mechanisms for OODBMS

Table I summarizes a feature by feature comparison of view mechanisms described in the literature neluding the characteristics supported by the system we proposed here.

### B. OODBMS Supporting Views for Clinical Applications

In the previous sections of this paper we have already mentioned two other systems based on object-oriented DBMS supporting views and used for the development of clinical applications: they are described in and respectively. In views are essentially used as shorthand for queries and for providing different objects compositions for different users. The concepts of materialized views, capacity augmenting views and view schemas are not addressed. Views are built using a

template, called *presentation description object*, which contains no actual data but provides the framework for both retrieving and storing different parts of an object. The system designer can define different presentation descriptors for different roles and contexts within the application. In this way, the same underlying data objects are viewed differently depending on the role of the user and the context he or she has entered. Presentation description objects are objects in the database.

## 6. CONCLUSION

A first result of the presented work consists of the output of the analysis phase of the clinical scenario. Thanks to the use of the adopted object-oriented business modeling technique, the considered context has been studied in detail using a formal approach. Different user roles have been identified and investigated, together with the activities they carry out. The outcomes of this analysis phase have been very important to design and

implement a system suitably modeling the considered part of real world.

A second important result is represented by the design

and implementation of an object-oriented database system supporting simultaneously three significant features for clinical information systems:

- 1. availability of tools to define and handle views and view schemas tailored to user needs;
- 2. data modeling approach allowing the adequate modeling
- 3. of temporal aspects related to clinical information; effective integrated management of multimedia data.

The overall achievement, building on the previous results, is the design and the development of a prototype of information system for the management of PTCA patient medical folders. The system provides each user group with a tailored view of the relevant data. User needs and hospital processes have been analyzed in detail and modeled

adopting an object oriented approach, more suited than other approaches in modeling such a complex application context. The adopted approach allows a very general view of the considered scenario covering both automated and non automated tasks of both the service system and the information system.

View and view schema support is based on an advanced view model offering features not provided by any other clinical applications described in the literature. A discretionary data access control policy has been implemented through views and view schemas defined for specific user-roles.

A communication architecture based on the use of Internet and the WWW has been adopted to enable a remote access to patient data to authorized users. In fact, some of the potential system users characterizing the studied context do not perform their activities locally (for instance, this is the case of the general practitioner). A prototype of a client application designed for the role of the general practitioner has been designed and implemented.

### 7. REFERENCES

- R. S. Dick and E. B. Steen, Eds., *The Computer-Based* Patient Record—An Essential Technology for Health Care. Committee on Improving the Patient Record, Division of Health Care Services, Institute of Medicine, Washington, DC: National Academy Press, 1991.
- F. L. Ricci, D. M. Pisanelli, F. Ferri, and F. Consorti, "Modeling the structural and behavioral features of medical concepts: a tool for generating multimedia patient folders," in *Proc. 12th Int. Congr. Eur. Federation for Medical Informatics—MIE 94*, 1994, pp. 470–474.
- T. Barsalou, "View objects for relational databases," Ph.D. dissertation, Dept. Comput. Sci. Med., Stanford University, Stanford, CA, Technical Report no. STAN-CS-90-1310, 1990.
- 4. R. Elmasri and S. Navathe, *Fundamentals of Database Systems*. Reading, MA: Addison-Wesley, 2000.
- Q. Zeng, J. J. Cimino, and K. H. Zou, "Providing concept-oriented views for clinical data using a knowledge-based system: An evaluation," *J. Amer. Med. Informatics Assoc.*, vol. 9, no. 3, pp. 294–305, 2002.
- J. S. Ash, P. N. Gorman, M. Lavelle, and J. Lyman, "Multiple erspectives on physician order entry," in *Proc. AMIA Symp.*, 2000, pp. 27–31.

- F. Ferri, D. M. Pisanelli, F. L. Ricci, and F. Consorti, "An object-oriented information model for a patient definition and management system," in *Proc. Applied Computing*, 1996, pp. 50–57.
- L. Padgham and J. Löwgren, "A user interface management approach for object-oriented data base applications," *J. Syst. Software*, vol. 27, pp. 183–193, 1994.
- W. Sujansky, D. Zingmond, M. Toshiyuki, and T. Barsalou, PENGUIN: An intelligent system for modeling and sharing declarative knowledge stored in a relational database," in *Proc. Medinfo*. Amsterdam, The Netherlands: North Holland, 1992, pp. 466–471.
- H. Tange, "How to approach the structuring of the medical record? Toward a model for flexible access to free text medical data," *Int. J. Biomed. Comput.*, vol. 42, pp. 27–34, 1996.
- Q. Zeng and J. J. Cimino, "A knowledge-based, concept-oriented view generation system for clinical data," *J. Biomed. Informatics*, vol. 34, no. 2, pp. 112– 128, 2001.