A Flexible Framework for Managing Temporal Clinical Trial Data

Michael Souillard

Centre Recherché Informatique, Université Paris I Sorbonne, 90, rue Tolbiac, 75013 Paris – France, souillar@mcs-vdr.fr

Carine Souveyet

Centre Recherché Informatique, Université Paris I Sorbonne, 90, rue Tolbiac, 75013 Paris – France, souveyet@univ-paris1.fr

Costas Vassilakis

Dept. of Computer Science and Technology, University of Peloponnese, Terma Karaiskaki 22100, Tripolis, Greece, costas@uop.gr

Anya Sotiropoulou

Dept. of Computer Science and Technology, University of Peloponnese, Terma Karaiskaki 22100, Tripolis, Greece, anya@uop.gr

Abstract. Clinical trials are processes that produce large volumes of complex data, with inherent temporal requirements, since the state of patients evolves during the trials, and the data acquisition phase itself needs to be monitored. Additionally, since the requirements for all clinical trials have a significant common portion, it is desirable to capture these common requirements in a generalised framework, which will be instantiated for each specific trial by supplementing the trial-specific requirements. In this paper, we present an integral approach to clinical trial management, using a temporal object-oriented methodology to capture and model the requirements, a temporal OODBMS for data storage and a generalised template application, through which trial-specific applications may be generated.

Keywords: e-healthcare, clinical trials, temporal data, medical applications

Biographical notes. Michael Souillard has been a researcher and a programmer for Matra Cap Systems. He has been a co-author of an extensive class system to support multi-granularity calendars and evolving objects. Michael Souillard has also participated in the design of clinical trial frameworks and meta-applications.

Dr. Carine Souveyet is an associate professor at the Universite PARIS-1 Pantheon-Sorbonne in Computer Science. Her research activities are concerned with (1) the verification of the conceptual specification of an information systems (IS), (2) the meta-modelling of IS and (3) the design process model. Dr. Carine Souveyet has extensive research experience and interests in information systems and data base

modelling. She has participated in several European and national projects and has published numerous scientific papers in international journals and conferences.

Dr Costas Vassilakis is an Assistant Professor in the department of Computer Science and Technology of the University of Peloponnese. He holds a B.Sc. in Informatics (1990) from the Department of Informatics, University of Athens and a Ph.D. from the same department in 1995. Dr. Costas Vassilakis has participated in several European and national projects and has published more than 40 scientific papers in international journals and conferences. He has been a consultant for the General Secretariat for Information Systems of the Ministry of Finance, Greece.

Dr. Anya Sotiropoulou has received her BSc in Informatics, MSc in Advanced Information Systems and PhD in Temporal Databases from the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece. She has participated in a number of R&D projects and has given lectures in databases, preservation of digital archives, file management, object-oriented programming - undergraduate and postgraduate level - in the National and Kapodistrian University of Athens, in the Ionio University (Corfu, Greece), and in the University of Peloponnese (Tripolis, Greece). She currently lectures in the University of Peloponnese as a contract Assistant Professor.

1 Introduction

During the last years temporal databases have attracted the interest not only of the database community, but also of certain application fields, such as economics or medicine, where requirements include time reasoning and representation of temporal information. This interest resulted to the organisation of Workshops and seminars (e.g. Texas, 14-16 June 1993, Zürich, 17-18 September 1995, Dagstuhl, 23-27 June 1997). Also books have been written (e.g. [1]; [2]) and a number of proposals ([3]) and implementations ([4]) of temporal database systems have been presented.

On the other hand, medical informatics is a field that exploits computer science results to facilitate the work of researchers in the different medicine areas, spreading across the phases of both diagnosis and prognosis [5]. The requirements of those areas include manipulation of highly complex information that evolves with time. Thus temporal databases have a number of applications in this field, as outlined in ([6]) and in ([7]). Additionally the complexity of the information indicates that a powerful and flexible data model, such as the object-oriented one, should be employed.

In this paper we present a temporal extension to an object-oriented database and a Temporal Object-Oriented Methodology (TOOM), which can be combined to support an application in the clinical research area. The temporal extension can be plugged into any ODMG-compliant OODBMS, formulating thus a TOODBMS, and it has been designed to handle non-temporal and temporal data uniformly. TOOM –which is compatible to the UML OO methodology– is powerful enough to capture and model the temporal requirements of such an application.

The TOODBMS and the TOOM have been used to model and implement a *template application* that focuses on protocol-based research on drugs, but its scope can extend to cover most medical applications with temporal requirements. The template application is used to generate applications that are specific to any trial that will be performed.

The rest of this paper is structured as follows. In section 2, the test case, which has been implemented over the TOODBMS, is described and its temporal requirements are pointed out. In section 3, the TOODBMS components and the TOOM are described, and their suitability in meeting the application requirements is shown. Section 4 describes the template application and the generation of the trial applications. Finally, in section 5 the conclusions are drawn and future work regarding medical applications with temporal characteristics is outlined.

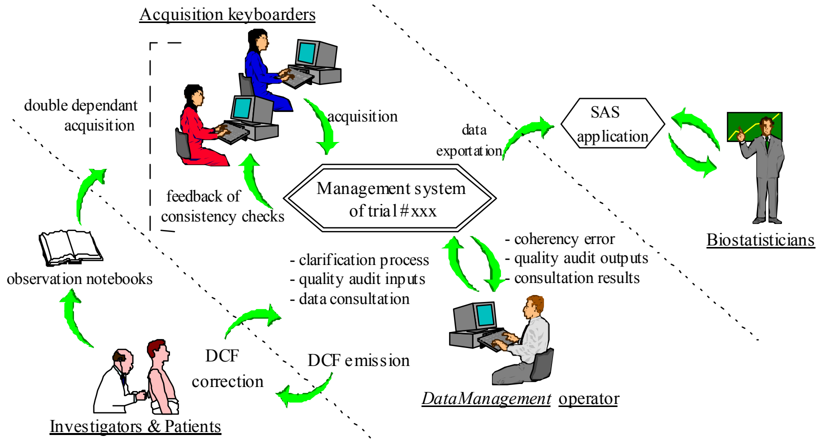
2 Clinical Research context

Clinical Research is one of the medicine research fields where temporal requirements are of major concern ([8], [9]). In the following sections we will describe the context and roles of the application, in the domain of clinical research. This application has been introduced in an earlier phase in [10] and preliminary results have been presented in [11]. We also present the requirements regarding temporal data management and temporal reasoning.

2.1 Data Management of Clinical Trials

The Clinical Research domain covers a large range of areas like cardiology, infectious diseases, paediatrics etc, and thus a wide variety of applications. In this paper, the target area is the Clinical Trials where certain drugs are tested on selected patients, to evaluate the efficiency of certain treatments.

A Clinical Trial Management System includes numerous phases handling different aspects of the clinical trials, with *trial data management* being a horizontal activity supporting data collection, analysis and reporting for all other tasks. The system presented here focuses on the data management phase in which data from the patients' observation notebooks are inserted into the system and its coherency is verified in order to prepare a "clean" database, suitable for biostatistic analysis, trial evaluation and reporting. Figure 1 presents an overview of the data management phase.



DCF: Data Clarification Form

Figure 1 - Overview of the data management phase

Initially, data from the observation notebooks (written by investigators and/or patients in the clinical centres) is inserted into the system. In order to minimise acquisition errors, two different keyboarders type in the same data and the system flags any discrepancies between the two entries, prompting the second keyboarder to choose which of the two values entered is correct. When data have been entered into the system, the *data clarification process* begins, with data management operators launching coherency checks. When incoherence errors are detected, *Data Clarification Forms* –DCFs– are produced and sent to the investigator responsible for the respective patient. Subsequently, the corrected DCFs are returned to the system, and the information in the database is corrected, accordingly. When all the data have been acquired, and no more DCFs are emitted, then the trial database is declared *clean*. Finally, clean data is forwarded to statistical tools for biostatistic analysis. Quality audits regarding the inconsistency and incoherence errors are performed as well, so as to minimise such errors in subsequent trials.

2.2 Temporal Requirements for Clinical Trials

In the context of clinical trial applications temporal requirements may be identified in three time dimensions, namely *user-defined time, valid time* and *transaction time.* In the following paragraphs these requirements are summarised.

2.2.1 User-defined time

The *user-defined time* dimension allows the representation and manipulation of time with semantics known to the application only. In the context of the clinical trials, time quantities like birth dates, treatment periods etc, are examples of user-defined time. Time quantities may be *absolute*, e.g. *08-Sep-2002*, or *relative*, that is expressed as an offset from an absolute time quantities, such as *D0 + 3 days*, indicating an instant three days after the first day of inclusion of the patient within the trial (*D0*).

2.2.2 Valid time requirements

Valid time is employed for maintaining the history of a patient status during a trial. Valid time timestamps may be *instants* if the observations are discrete, or *periods* if the observation is continuous. For example, the measured temperature of a patient pertains to a single point in time, thus it is timestamped using an instant, whereas the investigator to whom a patient is assigned (which may change within a trial) is timestamped using a period. Valid time timestamps stored along with data are used for in trial evaluation, e.g. calculating the percentage of patients presenting an improvement of 10% in the Maximum (Blown) Expiratory Volume between D0 and D14.

The evaluation of the trial is based on main and secondary criteria, in order to assess its efficiency. In the example of the asthma trial, the MBEV values are used in such criteria as for instance: "*How long does it take to obtain a 10% increase of the MBEV, regarding theoretical values, since D0?*" or "*What is the percentage of patients having a 10% increase of the MBEV* etc.

2.2.3 Transaction time requirements

Transaction time is used for maintaining the history of information storage and maintenance within the information system. In the context of clinical trial data management, transaction time is used to satisfy the quality audit requirements and for explanation purposes. When data is deleted from the database or updated, the values prior to the update are maintained, associated with a *non-current* transaction timestamp, whereas newly inserted values are associated with current transaction timestamps. The histories of values stored within the database are used for performing quality audits on the trial data acquisition procedure, by assessing the number and percentage of updates to given information, of DCFs' emission, on the types of detected errors on patients under the responsibility of a given investigator, etc. The second benefit provided by the use of transaction time dimension management is that it allows to explain and justify why certain results obtained or reasoning made at different steps of the trial are different from the final results obtained or reasoning made on the clean system.

2.3 Generic approach in trial systems

Each clinical trial is different from another since it aims to the evaluation of different treatments on different categories of patients. However, the data management phase of clinical trials, as presented above, is generic enough to handle any trial. Naturally, the manipulated data are different, but the tasks performed in the data management phase are always the ones described in section 0. Therefore it is desirable to provide a generic way to manipulate the different trials within the clinical trial data management system. This issue has been tackled by capturing the structure of clinical trials into a *meta-application*, which is then *instantiated* to generate applications specific to any trial. The generated applications implement the data management tasks for the trial they are instantiated for. This approach is illustrated in Figure 2, and described in more details in section 0.

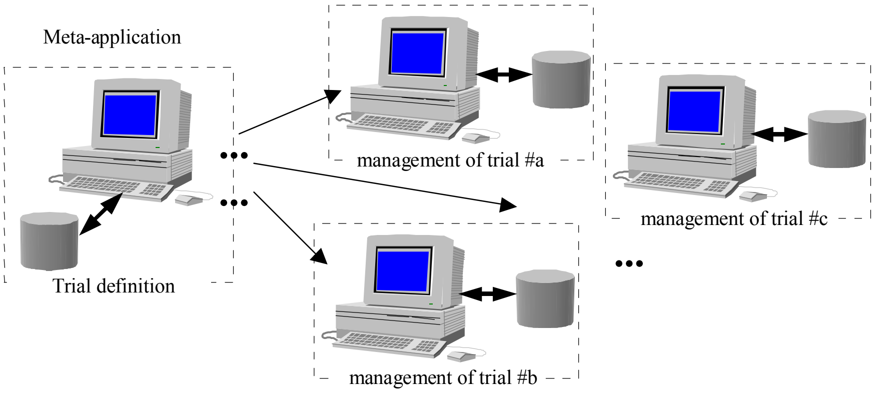


Figure 2 - Managing trials using a meta-application

3 The Clinical Trial Data Management Framework

In order to satisfy the temporal requirements identified for clinical trial data management, a *temporal toolchest* was used, containing the following components:

- A Temporal Object-Oriented Methodology (TOOM) ([12]) for capturing the detailed user requirements regarding the clinical data trials. TOOM has been derived from REMORA ([13]) and O* ([14], [15]) and focuses on the design of temporal applications. TOOM provides advanced design features within a single model to support the design of the application's structural and behavioural aspects ([12]) and offers step-by-step guidance to lead the developers from the conceptual schema to the full implementation of the application, over any temporal OO-DBMS platform ([16]). TOOM also includes advanced features, such as facilities for modelling temporal database applications requiring specific time management, reactive systems, able to trigger automatically operations and workflow systems, which transfer messages among different actors within the system ([17].
- A Temporal Object Data Model (TODM) ([18], [19]), which extends the Object Data Model (ODM) of ODMG ([20]) by accommodating temporal entities that can evolve over valid and/or transaction time dimension. Temporal entities may be temporal objects or temporal instance properties (temporal attributes and temporal relationships). Moreover, TODM provides support for time quantities such as instants, periods, intervals and period sets, which may be expressed in various granularities and calendars. Support for the Gregorian calendar is built into TODM, and users may define additional, application-specific calendars.
- A Temporal Object Definition Language (TODL) ([21], [22]), which extends the Object Definition Language (ODL) of ODMG. Through TODL the user may define classes (*interfaces*, in ODMG terminology). Each interface may have temporal characteristics *as a whole* (resulting to a temporal object of TODM), or may have instance properties with temporal characteristics (resulting to temporal attributes or temporal relationships). Through TODL, the signatures of the

interfaces' methods may also be defined, and operations are extended with the ability to accept as input or produce as output temporal entities. The body of the methods, however, must be implemented in a manipulation language such as C++ or Java, since TODL –in accordance to ODL- is a pure definition language, independent of the manipulation language to be used. The user may also specify that certain attributes are used for user-defined time, by allowing them to have types like instant, period or interval. All time quantities may be expressed in any of the default or user-defined granularities and/or calendars. Finally, used-defined calendars may be defined through TODL.

- A Temporal Object Query Language (TOQL) ([23], [24]), which extends the Object Query Language (OQL) of ODMG. TOQL introduces new operators that provide access to histories of objects, facilitate extraction of values with specific timestamps, allow for restructuring of data over the time axes etc. Methods that have been defined for an interface may be invoked within a TOQL query, providing thus more flexibility to the user. Finally, a C++ Binding for TOQL ([25]) is available, allowing the user to submit TOQL queries from within any C++ program whereas a Java language binding is described in ([26]). The functionality of TOQL has been demonstrated using the TSQL2 ([27]) Benchmarks on Temporal Query Languages ([28]).

4 Clinical Trial Data Management Applications

In this part, we will describe the meta-application that enables the designer to capture the structure of a new trial, and to generate an application specific to the data management of this trial. We will also present an example of such a generated application, in the context of a clinical trial about asthma treatment on children introduced previously. The context and requirements of these applications have been presented previously in this document.

4.1 The Overall Approach

The overall approach undertaken for dealing with the management of data in the context of clinical trials in a generic way is presented in Figure 3.

The first step to perform is the extraction of the generic features that can be found in any trial, both at the data schema level and the dynamics of the data management tasks. To ensure that the characteristics of any trial are covered, a wide set of major clinical trials has been explored, along with inputs from expert people working on clinical trials. This generic analysis has been performed using TOOM, the temporal methodology. Three components have been produced:

- A *data schema* of a generic trial, which may be instantiated to produce the data schema of any new trial. This data schema contains only snapshot information that describes the structures of trials.
- *Mapping rules*, which transform the description of a trial structure to the data schema for this trial, which will accommodate the data resulting from the

observations of patients. By applying these rules, trial-specific schemata are created. These trial data schemata are temporal schemata where the temporal data coming from the trial observations will be managed. The TOOM methodology was used for the definition of these rules, the selection of the temporal structures to be used and the generation of the TODL statements, which are fed to the TOODBMS, in order to create the schema for the new trial.

- The last point is the *dynamic view* of the clinical trial data management applications. A clinical trial data management application is mainly responsible for the acquisition and the coherency of the trial data, which incorporate temporal semantics. The dynamics (or *behaviour*) of such applications have been explored using TOOM, in order to extract the generic dynamics of trial data management applications.

With these three components a data management application for the target trial can be generated.

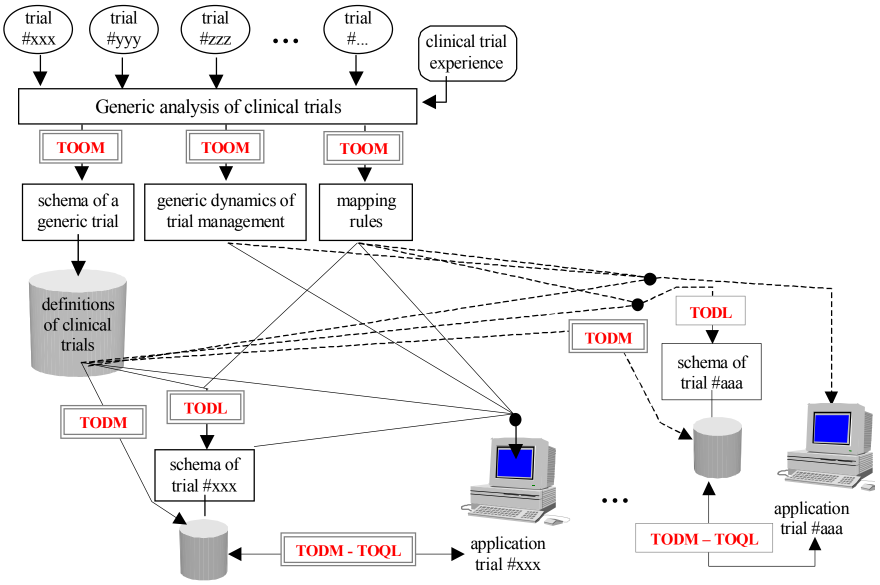


Figure 3 - Generic management of clinical trial data

Figure 3 illustrates the overall approach, beginning from the clinical trial generic expertise and up to the generated applications, which are specific to given trials. The *meta-application*, described in the next part, begins with the definition of a new trial and goes down to the generation of the data management application for this new trial. An example of such a generated application will be described Section 4.3.

4.2 The meta-application

The usage of the meta-application can be split in two steps: first, the description of the target trial has to be acquired and be incorporated into the meta-application. Afterwards, the various components to be generated should be modelled.

The description of a clinical trial contains the planning of regular and supplementary visits of investigators to patients, the observations to perform and the information to collect on each visit, the different observation notebooks, and so on. To define all these structures, the meta-application provides a facility to clone structures from other trials. For instance the physical exam to perform in the asthma trial quite similar to the physical exam of another trial; so we may clone it into the definition of this new trial, and modify just a few things. This feature promotes component reusability, since any components that have been defined for other trials, at any design level may be directly used within the trial under design.

When all the information required for a complete description of the new trial has been acquired into the meta-application, the generation step is launched. Firstly, TODL statements are generated; these statements will be submitted for execution to the TOODBMS in order to create the new temporal data schema specific to the data of this trial. These TODL definitions are generated using the mapping rules defined in the generic analysis phase of the project. When the new schema has been created, the meta-application starts to populate it with information that has been given in the description of the trial, as for instance the planning of the trial, the coherency checks to perform, etc. The API of TODM classes is used here to populate the temporal database associated with the trial schema. The last step is the generation of the application that will deal with the data management of the trial. Using the generic rules about the application dynamics and the trial schema, the trial application is generated. This application will communicate with the temporal database using the TODM API and the TOQL query language.

4.3 A trial-specific application

Recalling the main tasks that the data management module of a trial has to perform, these are:

- the acquisition of the data from the observation notebooks into the system,
- the clarification process to detect and correct incoherence errors, and
- the quality and audit reporting.

All these tasks handle temporal data, either via the TODM class API, or using TOQL queries.

The application is split into modules that take into account these different tasks. The access rights are granted according to the identification of the user, since different persons are responsible for the different actions. Additionally, the system keeps track of the users who make any update to the database, as required for legal purposes.

The same graphical interfaces are used for the two acquisitions, the visualisation and the correction of data. However, the functionality available through the interfaces differs, depending on the current phase of the trial and the role of the current user. For instance, during the second acquisition checks are performed regarding the value entered during the first acquisition, whereas such checks are not performed elsewhere. A piece of data may be corrected only if a DCF about an error on this piece of data has already been issued and not yet archived, as explained in the following paragraphs. These interfaces follow the “look and feel” of the paper version as closely as possible, in order to facilitate the acquisition tasks. The temporal features are encapsulated in series of tabs and sheets.

The coherency check module enables to launch checks on a specific piece of data, such as a given set of patients, the data for given visits or specified observation notebook pages, and so on. Therefore, the clarification process can be launched, even if the acquisition phase has not been completed for all patients or all visits. When errors are detected by these checks, DCFs are emitted by the system and are forwarded to the investigators that are responsible for the patients on whose data incoherence is detected. A DCF is called *current* from the time is has been generated by the system until the time is has been returned by the investigator and the correction to the database has been performed. As soon as the database is updated, the DCF is archived. The system keeps track of the full life cycle of each DCF, starting by registering its creation time when the DCF is emitted. If the DCF has not been returned to the system after a specified time interval, a *temporal event* is raised, causing the system to issue a warning to the data management operator that the return delay for the specified DCF has expired, so that the responsible investigator has to be contacted. This feature allows for the minimisation of the delay and the overall duration of the clarification process. A similar kind of warning is raised by the system when the theoretical end of the trial is arriving, so that personnel responsible for delayed tasks is notified to speed up their activities.

The quality audit and reporting module allows for selection of the target subset of data, similarly to the coherency check module. This way, audit reports may be generated before the end of the trial and feedback to investigators may be produced, so as to improve various processes of the trial. These reports are timestamped using *transaction time* or machine time. This timestamping enables the operator to evaluate the reports and the decisions that may have been taken, in relation to the state of the database that was current when the report was generated or the decision was taken. Since all the database updates are maintained by the system using the transaction time dimension, it is possible to access all the past states of the database, and then to justify and explain past decisions and reports.

5 Conclusions / Future work

In this paper we presented an integral approach to clinical trial management. This approach includes the following modules:

- A temporal object-oriented methodology, which is used to capture the data management requirements of clinical trials both at a generalised level (i.e. requirements that apply to any clinical trial) and at trial-specific level.
- A complete temporal OODBMS which allows the storage and querying of temporal data. Valid time support allows to following up the patient's state during the trial, whereas transaction time facilitates quality audits and evaluation of decisions using the data that were available at decision time. A powerful query language allows for retrieval of the necessary information and provides support for method invocation, leading thus to seamless integration of temporal reasoning and temporal maintenance, as proposed in ([7]).

- A generalised template application, through which trial-specific applications may be generated. This minimises the work that must be performed in order to build an application for a new clinical trial. The application supports the concept of *temporal events*, which allows for automatic scheduling of tasks and employs graphical user interfaces to facilitate the visualisation and manipulation of temporal data.

Future work will include support for uncertain information and provision of graphical query language tools, to support ad-hoc querying. Integrating the system with hospital web-sites to leverage their quality [29] and promote electronic health care [30] will be also investigated.

6 References

1 Tansel, A.U., Clifford, J. et al. (1993). Temporal Databases: Theory, Design and Implementation. Benjamin/Cummings Publications.

2 Date C. J., Darwen H. and Lorentzos N. A. (2002). Temporal Data and the Relational Model, Morgan Kaufmann

3 Tsotras V. J., Kumar A. (1996). Temporal database bibliography update. SIGMOD Record, 25(1), pp.41-51.

4 Boehlen, M.H., (1995). Temporal Database Implementations, SIGMOD Record, Vol. 24, Number 4.

5 Trevor Rohm B.W., Tapie Rohm C.E. Jr. (2004). Evolving medical informatics: from diagnosis to prognosis. International Journal of Electronic Healthcare 2004 - Vol. 1, No.1 pp. 103 – 111.

6 Combi, A. and Shashar, Y. (1997). Temporal reasoning and temporal data maintenance in medicine: Issues and challenges, Computers in Biology and Medicine, Vol. 27-5, pp. 353-368.

7 Shashar, Y. And Combi. C. (1997). Timing is Everything: Time oriented Clinical Information Systems, Western Journal of Medicine, 168, pp. 105-113.

8 Shashar, Y. (1997). A Framework for Knowledge-Based Temporal Abstraction. Artificial Intelligence in Medicine vol. 9(1-2), pp. 79-133.

9 Shashar, Y. And Musen, M.A. (1996). Knowledge-Based Temporal Abstraction in Clinical Domains. Artificial Intelligence in Medicine Vol. 8, pp. 267-298.

10 Souillard, M. and Pollet, Y. (1997). Toobis: une approche pour la gestion de données évolutives et temporelles dans la Recherche Clinique. In Proceedings of 6th International Interfaces'97 Conference, Montpellier, France, pp. 221-225.

11 Souillard, M., Vassilakis, C., and Sotiropoulou, A. (1998). TOOBIS: application de la gestion de données temporelles dans le domaine de la recherche clinique, in Proceedings INFORSID '98 Conference, pp. 145-168.

12 University Of Paris 1 (1997). Temporal Object Oriented Methodology TOOBIS, http://www.di.uoa.gr/~toobis, accessed September 10, 2004.

13 Rolland, C., Richard, C. (1982). The REMORA methodology for Information systems Design and management, in Olle T.W, Sol H.G., Verrijn-Stuart A.A. (eds) Information Systems Design Methodologies: a comparative Review, North-Holland Publications.

14 Brunet, J. (1993). Analyse Conceptuelle orientée-objet, PhD Thesis, University of Paris 6.

15 Lee, S.P. (1994). Formalisation et aide outillée à la modelisation conceptuelle, PhD Thesis, University of Paris 1.

16 University Of Paris 1 (1997). Guidelines for Developing Temporal Applications, http://www.di.uoa.gr/~toobis, accessed September 10, 2004.

17 Deneckere R. (2002). Using Meta-patterns to Construct Patterns, Rébecca Deneckère, in Z. Bellahsène, D. Patel and C. Rolland (Eds): Proceedings of the OOIS 2002 Conference, LNCS 2425, pp. 124-134, 2002, Springer-Verlag Berlin Heidelberg

18 Matra Systèmes & Informations and O2 Technology (1997). TODM Specification and Design, http://www.di.uoa.gr/~toobis/, accessed September 10, 2004.

19 Matra Systèmes & Informations and O2 Technology (1998). TODM Manual, http://www.di.uoa.gr/~toobis/, accessed September 10, 2004.

20 Cattel, R.G.G, Barry, D., et al (Eds.) (1997). The Object Database Standard: ODMG 2.0. The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor.

21 University Of Athens, 01-Pliroforiki and O2 Technology (1997). TODL Specification and Design, http://www.di.uoa.gr/~toobis, accessed September 10, 2004.

22 University Of Athens, 01-Pliroforiki and O2 Technology (1998). TODL and Metadata Manual, http://www.di.uoa.gr/~toobis, accessed September 10, 2004.

23 University Of Athens, 01-Pliroforiki and O2 Technology (1997). TOQL Specification and Design, http://www.di.uoa.gr/~toobis, accessed September 10, 2004.

24 University Of Athens, 01-Pliroforiki and O2 Technology (1998). TOQL Manual, http://www.di.uoa.gr/~toobis, accessed September 10, 2004.

25 University Of Athens, 01-Pliroforiki and O2 Technology (1998). TOQL C++ Binding Manual, http://www.di.uoa.gr/~toobis, accessed September 10, 2004.

26 Sotiropoulou A. (2000). Temporal Databases, PHD Thesis, University of Athens.

27 TSQL2 Language Design Committee (1995). The TSQL2 Temporal Query Language. R.T. Snodgrass (editor). Kluwer Academic Plublishers.

28 Vassilakis, C., and Sotiropoulou, A. (1998). Functionality of TOQL – Application of Temporal Benchmarks. Technical Report. http://www.di.uoa.gr/~toobis, accessed September 10, 2004.

29 Randeree E, Rao H.R. (2004). E-health and assurance: curing hospital websites. International Journal of Electronic Healthcare 2004 - Vol. 1, No.1 pp. 33 - 46

30 Demiris, G. Electronic home healthcare: concepts and challenges (2004). International Journal of Electronic Healthcare 2004 - Vol. 1, No.1 pp. 4 – 16