Towards the Unification of Modeling Temporal Aspects of Information Systems

Chafiqa Radjai, Idir Rassoul
Mouloud Mammeri University
LARI laboratory
Tizi-Ouzou, Algeria
Radjai.chafika@gmail.com, Idir_Rassoul@yahoo.fr

ABSTRACT: *The aim of this paper is to model the temporal aspects of information systems by proposing a meta-model based on pre-existing representation formalisms. It will thus be possible to access the successive states (evolution) of IS by the notion of time.*

Keywords: Temporal Aspects, Information System, Object Modeling, Concepts Unification, Instantiation

Received: 28 November 2012, Revised 29 December 2012, Accepted 5 January 2013

1. Introduction

The development of a model that supports all temporal aspects is the main challenge.

The information system is becoming increasingly complex. Changing the state of objects in such a system is characterized by the appearance of a new instance. For this purpose, the information system of the real world is stamped by "*time*". All the changes of these objects in this system over time should be kept to understand the dynamics of information system. This feature requires a mechanism that allows the access to successive states (evolution) of IS by the notion of time. This mechanism involves determining:

- How can we view the status of IS?
- What are the temporal units that punctuate IS?
- What data whose evolution is relevant to keep and manage in the IS?
- What are the constraints to define in order to ensure consistency of past, present and future states of IS?

The other problem is the representation of time. This theme has become even more important in many areas of the society as diagnostic applications, planning, geographic information systems, Computer-Aided Design, Computer-Aided Manufacturing, Computer Aided Construction. The automation of these areas means that we can represent different temporal aspects of the information.

This paper reports about the unification of modeling temporal aspects of information systems. The problem addressed is twofold:

• Proposing a single model of representation of the evolving information in IS.

• Proposing a model of representation of time.

The rest of the paper is organized as follows: in section 2, we present several temporal aspects of information and object evolution. Section 3 we present the main works dealing with the modeling of time and evolution. Section 4 describes the proposed conceptual temporal model. The paper ends with a conclusion.

2. Temporal Modeling

2.1 Modeling time

Several temporal aspects of information have been discussed over the years [1, 2]. First we draw a list of temporal concepts that the proposed formalism should be able to represent them.

The temporal types defined to manipulate time are:

• **Instant:** one point in time as for example, '2006-02-14 3:30:00'.

• **Interval:** time space between two instants, the initial instant and final instant as '[2011-12-14, 2012-10-22)]'.

• **Duration:** the length of an interval.

• **Period:** a period of time is known length time, but is not established on the time axis as 5 days, 4 months.

In timing measurements, we find these definitions:

• **Granularity:** scaling a fixed order or chosen metric e.g. seconds, days, etc.

• **Chronon:** the smallest unit of time in any system.

• **Calendar:** an abstraction of the time axis e.g. the Gregorian calendar consists of units: second, minute, hour, day, month and year.

• Temporal scale defines all (finest) allowed temporal positions within a defined temporal domain.

• Temporal position is the numerical representation of time relative to a temporal scale.

• Uncertain Time is a time which has (uncertain) location and possibly spanning over multiple time points.

There are also three kinds of time:

• **Absolute time:** an entry in the calendar e.g. February 14, 2006, spring 2011.

• **Relative time:** the temporal relationship between one or more absolute time entities e.g. end of 2012, the late spring of 2011, the four days after the election.

• **Periodic time:** an infinite sequence of instants or intervals e.g. the 20th day of each month,

In temporal databases, there are three temporal dimensions:

• **Valid time:** the period of time during which a fact is true in relation to reality.

• **Transaction time:** the period of time for which a fact is stored in the database.

• **Bitemporal time:** combines both.

• **User-defined time:** The value of time associated with a data is assigned by the user e.g. date of birth of an individual.

2.2 Modeling the evolution of objects

The evolution of information systems is an art much more difficult to master. This implies to anticipate change and develop tools for visibility and an appropriate and effective control.

This Section discusses in detail the evolution of object over time that we take into account in our work. We present the basic concepts that have been proposed in [3] to deal with evolution:

• **Life Cycle:** The temporality of objects for their states, rather than their values. Objects can be Scheduled and then Active and can be temporarily suspended, and eventually be disabled. These cases represent the life cycle of an object. The state of an object can be changed continuously. For example, the temperature at a given point changes continuously. We find this case in the fields of medicine, biology, economics, history or environment. Indeed, the visualization of these types of flows would enable us to analyze and anticipate the risks and dangers in order to prevent them. The state of an object can also be modified by an event. We find it especially in land use planning, cadastre, urban or historic planning. An object type can have temporal and non-temporal attributes. Similarly, declaring a type of association to be temporal keeps track of the lifecycle of its instances. These can be also scheduled, active, suspended and disabled.

• **Object Migration:** Object migration expresses that an object of a source class can migrate to a target class. There are two cases of this migration, as whether the object is stored as an instance of the source class. Evolution occurs when the entity ceases to be an instance of the source class. An extension occurs when the entity is an instance of the source class. As the object that undergoes migration retains its identity (since it represents the same real-world entity). Types of source and target object must belong to the same generalization hierarchy.

• **Generation Relationship:** This type of association models the processes that lead to the appearance of new objects. An instance (or set of instances) of a source object type generates an instance (or set of instances) of a target object type. This combination allows us to model causal relationships and temporality associated with the appearance and disappearance of entities in the real world. As in the case of transition, two different types of generation can be distinguished according to whether the instances are preserved sources. A transformation occurs when the instance of the source class is consumed in the production process. Production occurs when the source instance is stored as an instance of the source class.

• **Synchronization relationship:** Combining two types of object with synchronization association is equivalent to specifying an integrity constraint linking temporal life cycles of two types of objects.

3. Related Work

Many researchers have been devoted to model temporal data.

Various extensions of entity-relationship (ER) models have been proposed to model the temporal aspects of information. Following this approach, we include the extension of temporal model ERVT [3]. ERVT model supports timestamp for classes, attributes and relationships. ERVT is able to distinguish between snapshots constructors (snapshot) that are invariant in time; lifespan is associated with each of their instances. Temporary constructors (temporary) changing over time, each of their instances has a limited or mixed lifespan; their instances can have a global or temporary existence. The two time stamps, S (snapshot) and T (temporary) are used to capture such behavior in time. Transitions were introduced to model the phenomenon called object migration. The extension model ERVT introduces new features named class status, relationship generation and inter-temporal relationship to describe the evolution.

However, the model ERVT does not distinguish between different types of relationships used in object-oriented analysis, such as association and aggregation, and it provides no means of notation to describe past states. For this reason, the model ERVT can be increased for use in object-oriented analysis.

The approach relation object has been adopted to represent temporal data. Following this approach, we find MADS [4, 5]. MADS is enriched by the object-oriented structure that provides the usual concepts (temporal attribute, Object Types, temporal generalization link, aggregation link). MADS provides a linear and discrete time, from past to future. A stamp by the time of validity can be associated to attributes values. Associated to objects and associations, it defines their life cycle. Thus, MADS offers management of temporal structures: instant time, interval time and temporal elements. For the time of validity, a particular chronon called "*now*" is proposed. The choice of the calendar is left to the user. The stamping can be defined in absolute or in relative terms with an extension of the calculation of Allen. MADS provides concepts for modeling dynamic relationships between objects, which their semantic have a temporal component like transition relationship, generation relationship, inter-temporal association, snapshots aggregation.

The object-oriented approach has been adopted for the representation of temporal data. Following this approach, we find the temporal extension of the model TOOM [1] includes:

• **Redefining the class calendar:** a calendar is defined by a name, an origin, a measuring system materialized by an ordered list of granules, containing for each of them, upconverting and downconverting conversion. A translation operation of an instant in a period (duration). A conversion operation between granules of the same calendar. Two inter-conversion operations calendars.

• **The definition of temporal domains:** INSTANT, INTERVALLE, and PERIODE. The specialized domains INSTANT-A, INSTANT-R, INTERVALLE-A, INTERVALLE-R allow reasoning in absolute and relative time.

• **The introduction of two types of classes:** instantaneous classes (static objects) and temporal classes (stamping or history management).

• Constraints typology extension.

A model of textual expressions for the temporal component has been proposed in [6]. Following this model, the temporal entities (ET) may be absolute or relative. An absolute temporal entity (ETA) corresponds to dates: a day, a month, a season, a year or a century. They may be complete or incomplete. By incomplete means a day, a month or a season without mentioning the year e.g. January 15. A relative temporal entity (ETR) describes a relationship between one or more temporal entities (ET) (which may be absolute or relative). The temporal entities are represented using a time interval. We distinguish five types of relationships: inclusion, adjacency, orientation, and interval.

However, this model does not retain expressions referring to hours, periods and historical events.

An extension of the UML standard [7] has been proposed to allow the designer to define a set of temporal aspects. In this model, a stereotype called <<Temporal>> is used to define the temporal characteristics of different types of entities and relationships. It contains two tags: *durability* and *frequency* to specify temporal properties of a generalized element (i.e. classifier or association). Durability has four values: *constant, permanent, durable, instantaneous*. The *frequency* value can be *single* or *intermittent*. In addition, a constraint is attached to the stereotype that ensures a designer select valid combinations of temporal properties. The decision to store the history of a certain type is left to the designer who can select, for each type, whether to store the history or not. For non-historical types, the stereotypes <<permanent>> and <<constant>> are proposed to meet the integrity constraints driven by these characteristics.

For historical types, the timestamp attribute is marked with the stereotype <<timestamp>> to distinguish it from other possible temporal attributes defined by the user.

TOS (Temporal Object System) [8, 9] has been proposed to provide better support for complex data processing as clinical data that characterized by schema evolution. TOS has introduced the concept of family to group temporal objects which share a common context i.e. the structure (instance-variables and methods) and the state (data values). A temporal object in TOS changes throughout its life as the change occurs to both the structure and/or the state. The history of changes can be kept by associating time to both the structure and the state. These Changes are maintained throughout its various stages. A new *stage* is appended to a temporal object when a change occurs to the structure and/or state of the temporal object. This stage shares the structure and/or state of previous *stage*, which is not defined in the new stage. An ordered sequence of stages of a temporal object is called *life sequence*. *Stages* are constructed as prototypes. A set of similar structures is called the root-of-family (ROF). TOS also facilitates the construction of a complex family which is an aggregation of temporal objects from different families. Objects in a complex family are called complex temporal objects. An *Offstage object* in a family is created by sharing knowledge of temporal objects of the same family. A temporal object system (TOS) is a collection of families that are defined at different time instances. A change in a particular TO doesn't affect the ROF or any other object of the family. TOS does not allow the root-of-family (ROF) to evolve with the passage of time and imposes a read-only restriction on it. More details can be seen in [8].

Our first observation is that none of the temporal models that we have examined satisfies all aspects of time. Every phenomenon is expressible in at least more one formalism. The empirical and theoretical issues outlined above have led us to draw the lines of research that constitute the plan of our work:

• Is there any possibility to unify the formalisms proposed?

• How to go to a single model?

• Different notions related to the concept of time. Is there a macro concept that encompasses in whole or part these notions?

• Toward what modeling perspective of time?

• The notion of time is connected to the state of the information system (instantiation).

4. Our Proposal

4.1 Objectives

In Section 3, we have listed a series of design objectives. After presenting our proposal, we now examine its design with the mentioned objectives:

• To view and query the status of information system.

• To keep track of the previous state of objects.

• To support a temporal model of the various changes that can undergo an object.

• This model takes into account all the temporal aspects.

4.2 Methodology

To develop our model, we follow the following steps:

• Proposing a model of time.

• Identifying and modeling the Temporal Object (TO) that we use as a basis for capturing the evolution of IS

• Presenting a conceptual modeling of temporal objects (the Conceptual Framework Data) which will be useful for modeling information system versions over time axis.

4.3 UML conceptual data model

Today, object-oriented model has become increasingly popular in the world of IS given the many facilities that provides. The object model provides a better representation of the real world by the definition of complex objects and semantic relations varied as a class of objects, association, aggregation and generalization. This ensures complete management of historicity encapsulated in the object. In addition, the management of temporal objects and non temporal objects will be conducted on the same manner and temporal queries will be easier to implement.

The UML diagram was chosen for its ease of use in the sense that it can model most databases. This model allows a quick understanding of the logical organization of data.

4.4 A Model of Time

After identifying the needs, we define a temporal model for an information system based on the object paradigm.

We prefer a broader meaning of "*Time*" described in the models [1, 4] to include points, intervals and sets of time.

Class Time is a time value which can be an instant (class instant), a time interval (class Interval), a set of instant (Class Sets-of-instant), a set of intervals (Class Sets-of-interval) and a time period (class Period). Class interval has a duration attribute that its value is the interval size. Class INSTANT-A is used to represent instants of absolute time. It has the following attributes: granularity, e.g. in 2013, position, duration, uncertainty, isShifted. These aspects are more detailed in [10].

Class INSTANT-R represents instants of relative time. An interval is called absolute if it is composed of two absolute instants else it is relative, for example, some cases may be expressed in absolute terms (for example, during the interval time [2011/10/02, 2012/01/02]) and other relatively (e.g. between [date-registration, registration date+15 days]). INSTANT and INTERVAL are retained to represent instants or intervals of time whose type is not fixed in advance. Class Period and Class INSTANT-A are expressed relatively to the calendar within the granule used. Calendar class has the attributes: name, origin, granules and a set of methods for converting between different granules. Figure 1 represents the time model proposed:

ISO 8601 is a widely accepted standard regarding the textual representation of dates and time [10]. Concerning the valid time and transaction time, you can use the same model but with constraints such as:

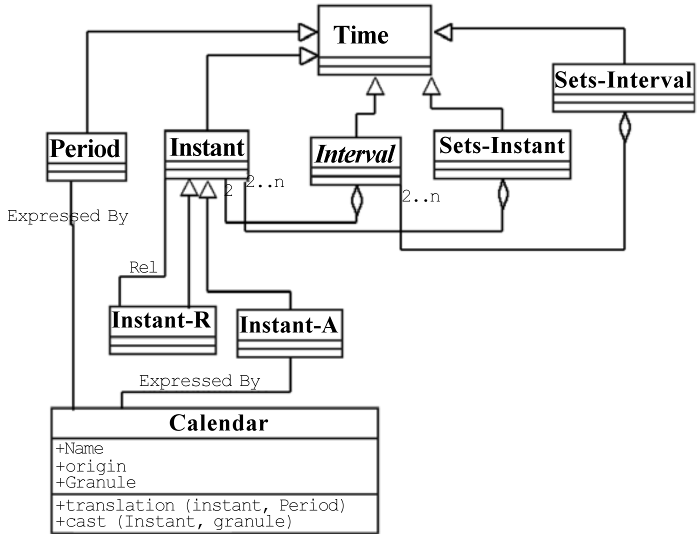


Figure 1. Model of Time

ISO 8601 is a widely accepted standard regarding the textual representation of dates and time [10]. Concerning the valid time and transaction time, you can use the same model but with constraints such as:

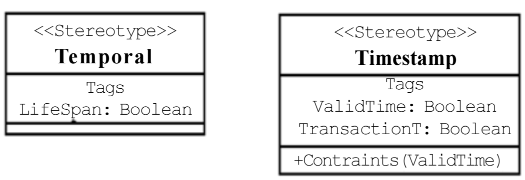
• There is no future time for transaction time. As for Valid time, this constraint depends on the application.

• There is no time now for the valid time.

• A valid time set must not contain intervals that overlap or occur, or moments that are equals.

4.5 Identifying and modeling the temporal object

We have proposed modification to the UML extension proposed in [7] to make it suitable for modeling temporal aspects of temporal attributes. For this reason, the stereotype called <<temporal>> is proposed to define the temporal aspects of object and association. It contains the *LifeSpan* tag that has *boolean* type. If this aspect needs to be captured, we assign to its value True. Temporal attributes are marked with the stereotype <<timestamp>>. It contains two tags *TransactionTime* and *ValidTime* that



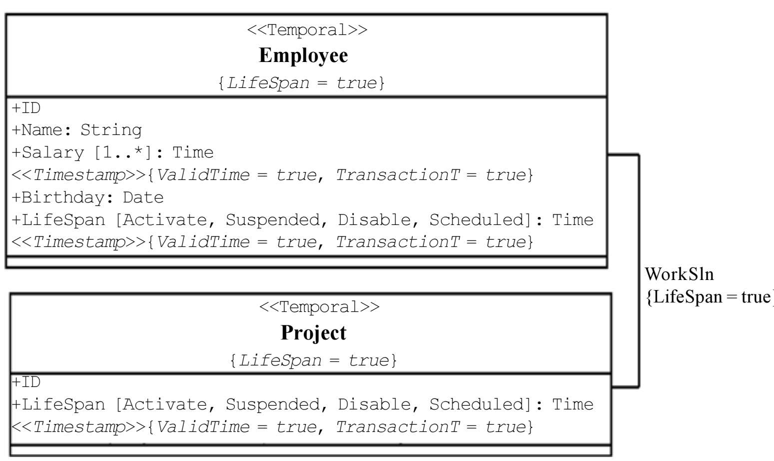


Figure 2. Identification and modeling temporal aspects

have Boolean types. If the database designer decides to capture the valid time of an attribute, the *ValidTime* tag has a value True, if the transaction time is captured, the *TransactionTime* tag has a value True. If both are captured, the two tags have values True. In addition, a constraint is attached to the stereotype that ensures for valid time. If the valid time is a set of intervals, these cannot overlap. If it is a set of instants, these must be distinct. Figure 2 represents the modeling of temporal aspects.

4.6 A conceptual modeling of temporal objects (the Conceptual Framework Data)

This paper proposes the TOS extension model proposed in [9] to model the instantiation of IS or IS States in different time instances. The TOS model has been extended by developing a methodology for the dynamic evolution of the ROF (root of the family) to make TOS an evolutionary model. There are three possible situations for a class to change its structure:

• Adding new instance variables, methods and/or rules

• Deleting an instance variables, methods and/or rules

• Changing in an instance variables, methods and/or rules

When any of above structural change is observed in temporal object of a family, the ROF of this family must be updated. The extension of TOS proposes that the structure of a temporal object should be divided in two logical portions i.e. primary portion will contain information about those members which a temporal object inherits from ROF at his birth time whereas secondary portion of temporal object will retain information of members which specifically belong to this temporal object and are not part of ROF. Any change moves a temporal object to a new stage, first of all will be analyzed to find out the type of change, if structural change is observed then further analysis will identify that whether the change occurred to the primary or secondary portion of a temporal object. ROF of family can be updated only in case of type 1 change. If that change is available in all BTOs (Brother Temporal Objects: the temporal objects belonging to a single family of TOS), the newly added member(s) will be included in ROF as well as moved from secondary portion to primary portion.

Temporal Object Query Language (TOQL) has also been extended to include new operators. Figure 3 shows the structure of a TOS for three Renault's Clio family, where RTOS denotes the root node of the system and there are three families (rectangles): Renault Clio I, Renault Clio II and Renault Clio III:

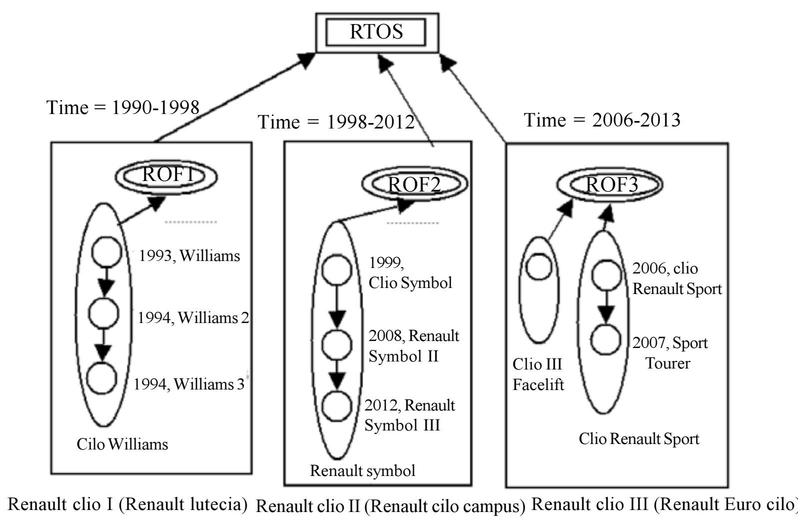


Figure 3. The structure of TOS

5. Conclusion

This paper proposes a unified model for representing temporal aspects of data, which supports several different temporal primitives: time point, interval, duration and temporal sets. Moreover, we can also refer to any past state of the information system.

In our approach we have not worked on enriching temporal conceptual models, nor on extending conceptual models to have the capabilities of modeling temporal aspects, but we have proposed combining several models. The resulting model assures a good representation of time and information system evolution.

We prefer a broader meaning of "*Time*" described in the models TOOM and MADS as the starting point of our model in order to afford a good conceptualization of time. ISO 8601 is a widely accepted standard regarding the textual representation of dates and time. We have proposed modification to the UML extension proposed by Jordi Cabot and all to provide notational procedures to refer to any past state of the information base. Moreover, we are looking in to the possibility of using TOS to model IS versions in different time instances.

References

[1] Souveyet Carine, Rebecca Deneckere. (1998). Conception de bases de données: Aspects temporels. Techniques de l'ingénieur. Informatique H3268: H3268-1.

[2] Matoušek, Kamil, Martin Falc, Zdenìk Kouba. (2012). Extending temporal ontology with uncertain historical time. *Computing and Informatics* 26.3, p. 239-254.

[3] Artale Alessandro, Christine Parent, Stefano Spaccapietra. (2007). Evolving objects in temporal information systems. *Annals of Mathematics and Artificial Intelligence* 50.1, p. 5-38.

[4] Parent Christine, Stefano Spaccapietra, Esteban Zimányi, Pier Donini, Corinne Plazanet, Christelle Vangenot, Nadia Rognon, Pierre-André Crausaz. (1997). MADS, modèle conceptuel spatio-temporel. *Revue internationale de géomatique*, 7 (3/4).

[5] Parent Christine, Stefano Spaccapietra, Esteban Zimányi. (2006). Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach. Berlin, Heidelberg: Springer-Verlag.

[6] Annig Le Parc-Lacayrelle, Mauro Gaio, Christian Sallaberry. (2008). La composante temps dans l'information géographique textuelle, Document numérique 10.2, 129-148.

[7] Cabot Jordi, Antoni Olivé, and Ernest Teniente. (2003). Representing temporal information in UML. «UML» 2003-The Unified Modeling Language. *Modeling Languages and Applications*, p. 44-59.

[8] Shah, A., Fotouhi, F., Grosky, W., Al-Muhtadi, J. (2004). Operators of the temporal object system and their implementation. *Information Sciences*, 158, 37-68.

[9] Abad Shah, Syed Ahsan, Ali Jaffer. (2009). Temporal object-oriented system (TOS) for modeling biological data. *Journal of American Science* 5.3, 63-73.

[10] Benzler Justus, Samuel J. Clark. (2005). Toward a Unified Timestamp with explicit precision. *Demographic research* 12.6, 107.