# On Temporal Modeling in the Context of Object Databases

Niki Pissinou*
Center For Advanced Computer Studies
University of Southwestern Louisiana
Lafayette, Louisiana
pissinou@cacs.usl.edu

Kia Makki
Dept. of Computer Science
University of Nevada
Las Vegas, Nevada
kia@unlv.edu

Yelena Yesha
Computer Science Dept.
University of Maryland
Baltimore, Maryland
yeyesha@cs.umbc.edu

## 1 Introduction

There are currently many data models that have powerful modeling constructs designed to make the tasks of database design, evolution, and manipulation simple and easy for database designers and users [1, 8, 9, 14, 51]. Most of these models do not provide adequate support for representing temporal information. Traditional database management systems that embody these models are therefore limited in their ability to directly record and process time-varying aspects of the "real-world." Such databases represent only "current facts." They do not incorporate the concept of time or provide support for the representation of temporal information. For example, a database records only the most current value of an object's attributes, and when those values change their previous values are erased.

Even though the concept of time is only treated implicitly in existing systems, it is crucial to all databases. There has been growing awareness among researchers of the importance of recording historical information in a database [6, 11, 12, 19, 28]. With these approaches, more complete information about the dynamics of a database's application environment is retained. The vast majority of research on temporal database systems is on relational and pseudo-relational database models, and has focused on the extension of such models to incorporate time. The advances in this area have been dramatic. In just the past decade an expanded body of knowledge of how to model, store and query temporal information in the context of relational databases has been developed, as briefly described in section 2.

However, among the research topics in temporal databases, little work has been reported on time in object [1] databases, compared to other topics such as temporal relational models. We suggest that there are several reasons why this is so. First, temporal/dynamic objects show highly complex structures and a lot of intricate dependencies. Hence, a comprehensive design of a temporal object model can only be attained in degrees and over an extended duration. Second, the potential returns are not clear; there is still no common definition for an object data model and no common consensus over what features are expected in an object database system, let alone a temporal object database system. The rapid and widespread adoption of the "object-oriented" approach has resulted in the underlying principles of the approach being buried under numerous definitions and specific mechanisms which makes the task of incorporating time with object databases even more tedious. Third, the lack of an infrastructure in the temporal relational database area, with no consensus for a common temporal model and modeling concepts, adversely affects progress in temporal object databases.

In this paper, we look at the issues of dealing with temporal modeling in the context of ***object databases***. In particular, we provide a survey of some important research achievements in temporal databases from the past two decades and significant contributions from related areas. We also examine a number of major objectives and areas of challenge which remain for researchers and implementors of temporal object database systems and discuss the time dimension in relation to object data modeling. We investigate the important issues that arise when attempting to integrate time with object databases. Our main objective here, is to provide the necessary background and motivation to design and develop a model that integrates time with objects, thus supporting temporal data and the temporal evolution of data in an object database framework. Such work will present a significant step towards the synthesis of an integrated object data model with a high level of abstraction, that supports the temporal and dynamic aspects of data modeling in addition to the structural and behavioral ones.

The remainder of this paper is organized as follows. Section 2 provides a brief survey of some important research achievements in temporal databases from the past two decades and significant contributions from related areas. To motivate the discussion of areas that need to be investigated in order to increase the acceptance of temporal object databases, in section 3 we give a brief explanation of the important aspects of object databases.

[1] We use the term object to refer to such characteristics as individual object identity, explicitly semantic primitives, active objects and object uniformity, as exhibited by a database model and a database system that embodies that model. A more detailed analysis of these characteristics appears in section 3.

⋆ The National Supercomputing Center For Energy and the Environment, LV, Nevada, pissinoi@nye.nscee.edu

Section 4 examines a number of major objectives and areas of challenge which remain for researchers and implementors of temporal object database systems and briefly describes our approach to temporal object modeling. The last section contains some concluding remarks.

## 2 Research Context

Over the past several years there has been a steady increase of interest in directly modeling time in databases; several research efforts have addressed the problem of supporting such temporal modeling. Some research [5], addresses the issue of time in databases by introducing the concept of an "event" or "process". However, most of these models fail for queries about the state of the world at any given time, rather than just on the occurrence of times of events. The main reason for this failure, is that these models are not concerned with the representation and processing of historical information. We provide here a brief and non exhaustive survey of existing work in this area and present a description of some important storage organizations. Extended surveys of the literature can be found in [46, 30, 3, 39].

One of the earliest efforts in the area was by Bubenko [11]. He developed a notably simple temporal model, by adding an extra field to each tuple of a relation (in a relational database) representing temporal validity. His method is frequently referred to as *tuple versioning.* An obvious drawback of tuple versioning is a high degree of redundancy because of large duplication even when changes are small. An alternative to tuple versioning is attribute versioning [12, 20]. In attribute versioning each dynamic attribute of a tuple is a set of $<$ value, time interval $>$ pairs. Clifford and Tansel [12] formalized the concepts and provided an algebra for databases in attribute versioning.

One of the first attempts to incorporate a semantically substantive concept of time in a database was by Clifford and Warren [13]. Here, intentional logic is used to define the formal semantics of time in database management systems. Their model incorporates time into the relational model at the attribute level. The main idea is to represent time-varying attributes of a database as functions from a set of times into values. Their logical model is somewhat complex but its query language is powerful. Because of its complexity, this model is not widely used. In the Time Relational Model by Ben-Zvi [10], a temporal dimension is added to ordinary relations and a consistent but limited algebra for the model is defined. This approach is similar to the one taken in the historical database model by Clifford [13].

Jones and Mason [22] and Snodgrass [46] considered start and stop times as special attributes and developed a temporal model with query language. The temporal query language (TQUEL) developed by Snodgrass [46] is an extension of QUEL, incorporating notions of time. It supports historical queries by augmenting the retrieve statement with a valid clause, a when predicate, and added constructs such as "start of," "precede" and "overlap." Ariav et al. [7] present an extension to SQL, by incorporating temporal elements into the language. A limitation of both of these efforts is that they do not address the formulation of a historical algebra, thus ignoring many operational issues. Vaishanav [49] and Gadia and Vaishnav [20] developed a query language called HTQUEL with a similar syntax as QUEL. HTQUEL allows two data types: temporal relations and temporal elements. Ben-Zvi [10], Jones and Mason [22], and Jones, Mason and Stamper [23], have also worked on designing temporal query languages.

Jones, Mason and Stamper [23] developed LEGOL 2.0, a formal language for writing rules. The language is based on the relational model. Each relation has the following types of attributes: identifier, characteristic and time. An identifier attribute names an entity. A characteristic attribute specifies a property of an attribute and a time attribute assigns an interval over which an entity holds a property. The interval consists of a start time and a stop time. A relation that has these three attributes is often referred to as a "continuous temporal relation." For a continuous temporal relation, a primitive rule of LEGOL 2.0 is represented as an "append" operation. The set of tuples to be appended is specified in a relational algebra like language. The algebra includes temporal operators such as the "time intersect" operator.

Even though interest on temporal databases is mainly concentrated on developing temporal query languages or on extending relational data models to incorporate time [13, 20], there has also been some effort in either extending other models to incorporate notions of time or in proposing new models for temporal data management. Klopprogge [26] extended the entity relationship model, by developing a set of modeling constructs to handle time and time dependent information. For the definition and manipulation of historical data, he designed a "PASCAL-like" language called TERM. The basic modeling construct in TERM is a structure. TERM has three different types of structures: time structures, value structures and history structures. A *history structure* is defined as a representation structure for histories of an attribute or a role. Its representation set is the powerset of the cross-product of the representation set of a time structure and a value structure. The basic operations in Klopprogge's model are "registration" and "correction." A registration operation could either be an "initiation operation," which is used for the addition of new entities or relationships, or a "completion operation" which is used for the addition of attributes and roles. A correction operation is used for correcting errors, which is a good idea, provided it is used only for editing errors, such as keyboard ones. Used in the right context, any accumulated historical information remains untouched. As such, the user's ability to answer questions about historical facts is not limited.

Shoshani and Kawagoe [45], and Segev and Shoshani [44] proposed a new model for temporal data management. Their main idea is to capture the semantics of ordered sequences of data values in the time domain and the operators over them. They refer to a collection of data values over time as a *"time sequence"* and define a temporal value to be a (surrogate, time, attribute value) triplet where a surrogate is a system defined object-id. Operators over a time sequence can be expressed in terms of the values and of the temporal properties of the sequence. A collection of time sequences will therefore posses the ability to address the temporal attributes of an entire class and relate them to other class attributes. The main contribution of this work is that rather than being an extension of the relational model, it is the step towards the development of a comprehensive model of time.

Physical structures for time sequences were discussed in a paper by Rotem and Segev [41]. Other research (e.g., [28, 21]) has also focused on the implementation of temporal databases. For example, Shoshani and Kawagoe [45] identify properties that are important for operations over time sequences and their physical implementation; these properties are: regularity, type, static/dynamic, and time unit. They also discuss general implementation issues and details for incorporating time in a database. Lum et. al. [28] also discuss some implementation details for including a concept of time in a database.

In fact, during the past decade several attempts have been made to provide efficient storage organizations for temporal data. One of the earliest works in this area is by Ben-Zvi [10], who proposed the temporally partitioned store model. In this model, a database storage is divided into two areas, the *current store* and the *history store*. The current store contains current data and possibly some history data, while the history store holds the remaining data i.e., the history versions of items. This strategy provides efficient access methods for temporal queries without creating an overhead for conventional queries. He also used *reverse chaining* and *secondary indexing* for his proposed storage organization.

The idea of a temporally partitioned store was later used by Katz and Lehman [24] for designing VLSI files. The idea of reverse chaining and indexing was also utilized by Lum et al. [28]. Thereafter, there has been a growing research interest in the area of designing efficient storage structures and access paths for temporal databases [2, 3, 18, 17, 16, 27, 41, 48].

Ahn and Snodgrass [3] also advocated the concept of a temporally partitioned store of Ben-Zvi [10]. In their work they discuss different variations of this concept. They also provide some performance evaluations of these variations, on a set of sample queries. The *append-only tree* was first introduced by Segev and Gunadhi [42, 43]. The *time index* method for temporal databases was introduced in [18, 43]. Elmasri et. al. method uses object intervals for retrieving versions of objects that are valid during that interval. In subsequent works (such as in [17]) different, efficient implementations of the time index strategies were discussed and evaluated. Their simulation results in both works [18, 17], indicate that the time index provides better access structures than other [2] storage organizations which simply link the versions of a given item together. More recently, Elmasri et al. [16], propose an interesting implementation of the time index using a monotonic $B^+$-tree structure, which is suitable for "append-only" temporal databases. Because practical difficulties of implementing an "append-only" temporal database on a large scale remain to be explored, this type of database is not commercially available.

In a recent paper by Makki and Pissinou [29] a storage model for temporal databases is presented, which is a hybrid between the storage structure defined by Ben-Zvi [10] and Ahn and Snodgrass [3] and the storage structure in [18, 17, 16]. Makki and Pissinou [29] utilize Ben-Zvi and Ahn and Snodgrass' idea of maintaining both current data and history data and the idea of time index by Elmasri et al. Their current storage organization follows the storage organization of conventional non temporal databases. However for maintaining the history versions of each item in the database they use a new data structure referred to as a *History Version heap*. The novelty of their approach resides in its ability to insert a new version for an item into the database in $O(1)$ amortized time, and retrieve the oldest version of an item in $O(1)$ amortized time. These bounds are irrespective of the number of existing versions for that item. In addition, they can access or delete any version of a given item in logarithmic amortized time. Since their approach treats the current store and history store differently, it makes use of different access methods for each store possible. As such, their model allows them to handle non-temporal queries in a conventional manner without any extra overhead, while providing an efficient storage organization for temporal queries.

Research also proceeded towards the development of a taxonomy of databases that support the semantics of time. Snodgrass and Ahn [47] proposed four types of databases for supporting time concepts: snapshots, rollback, historical, and temporal databases. Snapshot databases model the "real-world" as it changes dynamically by only keeping the most recent snapshot of a relation. This reflects the most recent state of a database. Rollback databases support the concept of *transaction time*, which can be defined as the time an event was recorded in the database. Historical databases record only one historical state per relation and there is no record of past database states. They support *valid time*, which is the time of occurrence of an event in the real world. A temporal database supports both valid and transaction times.

While the above has highlighted research on time within the context of database systems, the semantics of time has also been addressed in other areas such as artificial intelligence [4], logic, and philosophy. For example, the work of De et. al. [15] involved temporal semantics and natural language processing in a decision support system. They propose a mechanism for representing data and decision models with explicit temporal aspects and address the issue of processing temporal queries.

## 3 Perspective on Object Databases

As impressive as the theoretical accomplishments of basic temporal database research have been, there is a concern among researchers that little attention has been given to the temporal aspects of objects for object database modeling. In what follows, we first examine the principal characteristics of object-based database models. Specifically, we use the term object to refer to the following characteristics, as exhibited by a database model and a database system that embodies that model:

- *Object Identity*: Objects in a database can include not only atomic data values, such as numbers but also abstract objects representing entities, relationships or concepts in the "real world" which can be directly manipulated.
- *Data Abstraction and Encapsulation*: Abstraction consists of emphasizing on the essential and inherent aspects of an object, while encapsulation consists of separating the external aspects of an object which can be accessible to other objects, from the internal implementation details.
- *Explicit Semantic Primitives*: Primitives are provided to support object classification, structuring, semantic integrity constraints, and derived data. These primitive abstraction mechanisms support such features as aggregation, classification, instantiation, and inheritance.
- *Dynamic Objects*: Database objects can be dynamic as well as static, in the sense that they can exhibit behavior and acquire different roles over time. Current approaches to the dynamic modeling of objects include message passing or datatype encapsulation.
- *Inheritance:* Classes share attributes and operations based on a hierarchical relationship. Inheritance is transitive across an arbitrary number of levels.
- *Complex Objects:* The state of an object may refer to another object which in turn may refer to yet another object.
- *Object Uniformity*: Most of the information in a database is described using the same object model, vis. descriptive information (meta-data), and is conceptually represented in the same way as facts.

In addition to these characteristics, there are other features such as polymorphism, selective inheritance, and persistence. Armed with this background, the next section provides an overview of new gaps we are attempting to close, by addressing many of the issues necessary to make the temporal object database technology a practical reality.

## 4 Temporal Object Databases

In this section, we examine a number of major objectives and areas of challenge which remain for researchers and implementors of temporal object database systems. To motivate our discussion of research problems, we raise some intriguing questions that must be solved and where we feel that important research contributions are required in order to make temporal object databases viable.

### 4.1 Research Objectives

We take the view that research work in temporal object databases must satisfy the following broad objectives:

- To study the semantics of time in the context of the object oriented paradigm, identify any potential changes to existing notions of temporal data deemed necessary because of the transition from the relational to the object model, and develop a set of temporal object principles to be used as an underlying basis for the design of a generic temporal object model;
- To study and identify the temporal aspects of objects and develop a model for integrating time with objects, and to provide the necessary theoretical foundation for the proposed model;
- To augment the data modeling power of the object and semantic modeling concepts, particularly versions, with temporal modeling concepts;
- To provide a complete, formal theoretical foundation for a temporal object model;
- To extend OSQL (object SQL) to TOSQL (temporal object SQL);
- To modify and extend current (extensible object database models), to incorporate newly developed notions of time.

### 4.2 Open Questions

With these objectives in mind, we raise some intriguing questions that must be solved and important research contributions that are required in order to make temporal object databases viable. In general, the integration of time into an object database management system raises several fundamental and intriguing open questions [2] such as:

- Do temporal object database systems violate implicit assumptions made in relational environments about how time (and hence the real world) is modeled? What constitutes a temporal object model?
- What are the temporal aspects of objects in the context of object databases and what properties of an object are essential to its existence independent of time?
- How do we describe the structure of an object as it evolves over time and across its multiple representations (i.e versions and history)? How do we characterize time with respect to versions and object migration? How do we handle object migration and multityping?

[2] The questions raised here are not exhaustive but provide an insight into the complexity and the significance of the problem.

- What are the temporal inter-object relationships? How can different objects be related with respect to time? What approaches can be employed to deduce temporal relationships? How can the "temporal behavior of relationships" be represented?
- How do we model the internal actions of an object and its behavioral facets (be it internal or external) as a consequence of an action on transitions and automatic trasitions (actions on transitions are themselves objects)?
- How do we handle objects that are inherently not time-varying (e.g. biological-gender) in a type hierarchy?
- What operations do we need to handle time? Is there a minimum set of such operations? More importantly, is the existing set of temporal operations used for relational databases sufficient, or is it necessary to define an additional set for temporal object databases?
- What constitutes temporal object constraints and does the object nature of our environment and our attempt to model the real world more adequately necessitate additional constraints not usually associate with temporal relational databases?
- What approaches can be employeed to identify generic temporal queries? What types of temporal queries can be expressed and what query processing techniques are appropriate? Should we extend OSQL (object SQL) to TOSQL (Temporal Object SQL)? Can we progress by this extension or should we look into new representation languages for temporal object databases?
- How do we smoothly integrate version databases and temporal databases?
- How do we accommodate the evolvability of the meta-data (conceptual schema) of a database over time? How can a schema be refined to better characterize reality as it is reflected over time (i.e. how do we best describe a schema as it evolves over time?)
- What algorithms, data structures and access methods are best suited for implementing a temporal object model? Are current implementation techniques sufficient?

One of the goals of researchers in this area should be to apply temporal notions to the problem of object and meta-data (conceptual schema) evolution [25]. By including the semantics of temporal aspects of objects into the database schema we intend to provide the added functional capability of supporting the independent existence of a temporal object database, apart from the application programs and systems that manipulate it.

To achieve these objectives and provide some possible solutions to the questions we have raised earlier on, an appropriate data model that is rich in *temporal semantic evolvability* is essential. It is therefore important to define new modeling concepts to capture the temporal aspects of design that are not amenable to description of data models that were developed in the context of non temporal (object) database systems. Any approach in this area should lead to a precise characterization of the properties of temporal objects and operations over them without being unduly influenced by traditional models that were not tailored to model temporal objects. It should also augment the data modeling power of the object model, particularly versions, with temporal modeling concepts. In addition, it should effectively deal with the dynamic aspects of modeling, viz., states and events at the lowest level of granularity, to specify control rather than just constructs. Triggering of events over time should also be modeled. Restrictions in the generalization hierarchy should be enforced.

With respect to adding the temporal dimension to existing object models, any chosen model should include such prominent object oriented features as: (a) the accommodation of objects at various levels of abstraction and granularity, including atomic data values, abstract objects, objects from various media and types (classifications of objects); (b) support for inter-object relationships which represent associations among all varieties of objects, including meta-objects (such as object types); (c) the pre-definition of a set of (extensible) abstractions, including subtyping (specialization) and inheritance of relationships; (d) support for semantic and behavioral integrity rules and derived data (e) definition of operations which support the behavioral and dynamic manipulation of objects (methods).

### 4.3 Our Approach

Given the preceding brief overview of the issues pertaining to temporal object database modeling, it is now possible to briefly describe our approach to the problem based on the recent work by Pissinou [31, 32, 33, 34, 35, 36, 38, 37]. The key observations motivating this approach include:

(a) A realistic world most often supports some notion of time.
(b) Very few events and actions are instantaneous—most such events and actions take time.
(c) More than one event or action may occur at the same time.
(d) Different people may have different views of the same information in the universe at a given time (or sequence of times) with different and possibly conflicting semantics.
(e) An object can have conflicting semantics and behavior.
(f) An object can have different temporal roles with possible conflicting semantics and behavior.
(g) Most actions involve a complex ordering of temporal constraints.

To this end, we have developed a set of principles to which our temporal object model should adhere such as the *temporal representation principle* which states that a "temporal object model should capture and describe events as they occur in the application environment it attempts to represent, and incorporate different views of the same information." Through these principles we have established a set of metrics for determining what constitutes a temporal object model. To satisfy these principles, we defined a new set of abstract concepts to precisely characterize the temporal properties of objects. For example, to describe the structure and dynamic behavior of an object as it evolves over time and across its multiple representations, we introduced the notions of *universal identity* and *possible world identity*. We argue that the universal identity of the object can be derived from all its possible world identities. These two notions lead to several other concepts and definitions as described in [31, 32, 33, 34, 35, 36, 38, 37]. such as the notion of *objects with temporal roles* and the introduction of a new abstraction primitive called *temporal uncertainty*. By adopting modified notions of a time sequence introduced by [44] we introduced the notion of a *time priority sequence* for possible object identity worlds.

To provide a specific context for our approach, we defined a framework for a simple object model. This model addresses temporal problems at the *finest level of data granularity*, viz., the object level. The model provides a user with the basic primitives for temporal object definition, manipulation and retrieval. A set of generic temporal constraints is also defined. The primitive operations of this model may be used as the basis for the specification and stepwise development of object database models and systems of increasing complexity. To demonstrate this, we concentrate on extending a specific rich, extensible object database model to incorporate our new notions.

## 5 Final Remarks

The aim of this paper has been to examine some important research achievements in temporal databases from the past two decades and discuss the time dimension in the context of object databases. Specifically, we concentrated on some crucial issues of integrating time with object databases and identified some important aspects of temporal object databases. We also examined a number of major objectives and areas of challenge which remain for researchers and implementors of temporal object database systems.

Our main objective here was to provide the necessary background and motivation to design and develop a model that integrates time with objects, thus supporting temporal data and the temporal evolution of data in an object database framework. Such work will present a major step towards the synthesis of an integrated object data model that supports the temporal aspects of data modeling in addition to the structural and dynamic ones.

We anticipate that research on time in object databases will have significant impact on several areas. First, this work will be the first step towards the synthesis of an integrated object model that supports the temporal aspects of data modeling in addition to the structural and dynamic ones. As such, work on identifying the temporal aspects of objects and operators over them should provide significant insight into the problem of supporting and manipulating the temporal evolution of data. Any research in the area should establish the first solid framework for integrating time with objects, in the context of object databases.

Second, we expect the work to have a direct impact on how various temporal properties of objects can be incorporated into existing object models. Once research on identifying the temporal aspects of objects and operators over them is complete, these structures and operations can be used in other models by extending or changing these models. Thus the model should be a valuable temporal object model in its own right, and a powerful notation for describing other temporal object models. While the primary context of our study is object databases, many of the results in this area could be applicable to record-oriented (e.g., relational) database management systems as well.

Third, the research will apply temporal notions to the problems of object and meta-data (conceptual schema) evolution. One result of this should be insight into dealing more effectively with the integration of version databases with temporal databases. Subsequently, new evolution principles and techniques useful in object systems, which are of increasing significance and practicality, could be developed. For example, we anticipate results in dealing effectively with schema evolution and object migration over time.

The discussions in this paper seem to lead to a surprising conclusion: at least one of the more promising directions, among current research directions in temporal databases is temporal object databases. In particular, when describing the time dimension in relation to object data modeling it appears that looking beyond the techniques used to model time in the context of relational databases will be the most fruitful and progressive step. A temporal object model with a high level of abstraction will provide a testbed for further scientific research, allow the development of advanced concepts and perhaps more importantly facilitate the efficient and effective implementation of high performance temporal object database management systems.

## 6 Acknowledgment

The authors would like to thank Arie Segev for his comments and suggestions.

## References

[1] S. Abiteboul and R. Hull. Ifo: A formal semantic database model. *ACM Transactions on Database Systems*, 12(4):525–565, 1987.

[2] I. Ahn. Towards an implementation of database management systems with temporal support. In

K. R. Dittrich, editor, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 174–192, 1986.

[3] I. Ahn and R. Snodgrass. Partitioned storage for temporal databases. *Information Systems*, 13(4):369–391, 1988.

[4] J. F. Allen. An interval-based representation of temporal knowledge. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 191–204, Vacouver, B.C., 1981. VLDB Endowment.

[5] T. L. Anderson. Modeling time at the conceptual level. In *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications*, Jerusalem, 1982. 2nd International Conference on Databases.

[6] G. J. Ariav and J. Clifford. A system architecture for temporally oriented data management. In *Proceedings of the International Conference on Information Systems*, pages 177–186, Tucson, Arizona, November 1984.

[7] G. J. Ariav, J. Clifford, and M. Jarke. Panel of time and databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 243–245, May 1983.

[8] F. Bancilhon, G. Barbedette, V. Benzaken, C. Delobel, S. Gamerman, C. Lecluse, P. Pfeffer, P. Richard, and F. Velez. The design and implementation of o2, an object-oriented database system. In K. R. Dittrich, editor, *Advances in Object-Oriented Database Systems*, pages 1–22. Springer-Verlag, 1988.

[9] J. Banerjee, H. Chou, J. Garza, W. Kim, D. Woelk, N. Ballou, and H. Kim. Data model issues for object-oriented applications. *ACM Transactions on Office Information Systems*, 5(1):3–26, January 1987.

[10] J. Ben-Zvi. *The Time Relational Model.* PhD thesis, University of California, Los Angeles, Los Angeles CA, (unpublished), 1982.

[11] J. A. Jr. Bubenko. The temporal dimension in information modeling. Technical Report RC-6187-26479, IBM Thomal J. Watson Research Center, November 1976.

[12] J. Clifford and A. U. Tansel. On an algebra for historical relational database: Two views" clifford james and tansel a. u. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1985.

[13] J. Clifford and D. S. Warren. Formal semantics for time in databases. *ACM Transactions on Database Systems*, 6(2):114–154, 1983.

[14] E. F. Codd. A relational model for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[15] S. De, S. Pan, and A. Whinston. Temporal semantics and natural language processing in a decision support system. *Information Systems*, 12(1):29–47, 1987.

[16] R. Elmasri, M. Jaseemuddin, and V. Kouramajian. Partitioning of time index for optical disks. In *IEEE Database Engineering*, February 1992.

[17] R. Elmasri, Y. Kim, and G. Wuu. Efficient implementation techniques for the time index. In *IEEE Database Engineering*, pages 102–109, April 1991.

[18] R. Elmasri, G. Wuu, and Y.J Kim. The time index: An access structure for temporal data. In *Proceedings of the International Conference on Very Large Databases*, 1990.

[19] S. K. Gadia. Towards a multihomogenous model for a temporal database. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1986.

[20] S. K. Gadia and J. Vaishnav. A query language for a homogenous temporal database. In *Proceedings of the 4th Annual ACM SIGACT-SIGMOD Symposium on Principles of Database Systems*, 1985.

[21] A. Gunadhi and A. Segev. Query processing algorithms for temporal intersection joins. In *Proceedings of the 7th IEEE International Conference on Data Engineering*. Kobe, Japan, April 1991.

[22] S. Jones and P. J. Mason. Handling the time dimension in a data base. In S. M. Deen and P. Hammersley, editors, *Proceedings of the International Conference on Data Bases*, pages 65–83, June 1980.

[23] S. Jones, P. J. Mason, and R. Stamper. Legol 2.0: a relational specification language for complex rules. *Informatin Systems*, 4(4):293–305, November 1979.

[24] R. Katz and T. Lehman. Database support for versions and alternatives of large design files. *Transactions on Software Engineering*, 1984.

[25] W. Kim and H. Chou. Versions of schema for object-oriented databases. In *Proceedings of the International Conference on Very Large Databases*. VLDB Endowment, September 1988.

[26] M. R. Klopprogge. Term: An approach to include the time dimension in the entity-relationship model. In *Proceedings of the Second International Conference on the Entity Relationship Approach*, October 1981.

[27] C. Kolovson and M. Stonebraker. Indexing techniques for historical databases. In *IEEE Database Engineering*, February 1989.

[28] V. Lum, P. Datam, R. Erbe, J. Guenauer, P. Pistor, G. Walch, H. Werner, and J. Woodfill. Designing dbms support for the temporal dimension. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 115–130, June 1984.

[29] K. Makki and N. Pissinou. A new storage organization for temporal databases. *International Journal of Systems and Sciences (to appear)*, 1994.

[30] E. McKenzie. Bibliography: Temporal databases. In *ACM SIGMOD Record*, volume 15, pages 40–52, December 1986.

[31] N. Pissinou. A conceptual framework for time in object databases. Technical Report CRI-90-26, Computer Research Institute, University of Southern California, Los Angeles CA, 1990.

[32] N. Pissinou. Time in object databases. a proposal for the ph.d. thesis (and an nsf proposal) by niki pissinou. Technical report, Computer Research Institute, University of Southern California, Los Angeles CA, May 1990.

[33] N. Pissinou. Extending the 3dis to support time. Technical Report CRI-91-1, Computer Research Institute, University of Southern California, Los Angeles CA, 1991.

[34] N. Pissinou. *Time in Object Databases*. PhD thesis, Department of Computer Science, University of Southern California, Los Angeles, California, December 1991.

[35] N. Pissinou and K. Makki. T-3dis: An approach to temporal object databases. In *Proceedings of the International Conference on Information and Knowledge Management*, 1992.

[36] N. Pissinou and K. Makki. A framework for temporal object database models. In *Advances in Computer Sciences (to appear)*. Springer-Verlag, 1993.

[37] N. Pissinou and K. Makki. Separating semantics from representation in a temporal object databases. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, Washington, DC, November 1993.

[38] N. Pissinou and K. Makki. A unified model and methodoly for temporal object databases. *International Journal on Intelligent and Cooperative Information Systems*, 2(2), 1994.

[39] J. F. Roddick and J. D. Patrick. Temporal semantics in information systems – a survey. *Information Systems*, 17(3), October 1992.

[40] E. Rose and A Segev. A temporal object oriented data model with temporal constraints. In *Proceedings of the 10th International Conference on Entity-Relationship Approach*, pages 205–229. San Mateo,CA, 1991.

[41] D. Rotem and A. Segev. Physical design of temporal databases. In *IEEE Database Engineering*, February 1987.

[42] A. Segev and A Gunadhi. Event-join optimization in temporal databases. In *15th International Conference on Very Large Data Bases*, pages 205–215. 1989.

[43] A. Segev and A. Gunadhi. Efficient indexing methods for temporal relations. *IEEE Transactions on Knowledge & Data Engineering*, June 1993.

[44] A. Segev and A. Shoshani. Logical modeling of temporal data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 454–466, 1987.

[45] A. Shoshani and K. Kawagoe. Temporal data management. In *Proceedings of the International Conference on Very Large Databases*, pages 79–88, August 1986.

[46] R. Snodgrass. The temporal query language tquel. In *Proceedings of the ACM Conference on Principles of Database Systems*, pages 204–213, April 1984.

[47] R. Snodgrass and I. Ahn. Temporal databases. *IEEE Computer*, 19(9):35–42, September 1986.

[48] V. Tsotras and B. Gopinath. Optimal versioning of objects. In *IEEE Database Engineering*, pages 358–365, 1992.

[49] J. H. Vaishanav. A homogeneous temporal extension of quel. Technical report, Texas Tech University, 1986.

[50] G. Wuu and U. Dayal. A uniform model for temporal object-oriented databases. In *IEEE Database Engineering*, pages 584–593. Arizona, February 1992.

[51] S. Zdonik. Object management system concepts. In *Proceedings of the ACM-SIGOA Conference on Office Information Systems*, 1984.