SPATIO-TEMPORAL DATABASE DESIGN ARCHITECTURE (STDDA): A CONCEPTUAL FRAMEWORK

Nadeem Mahmood[1,3,*], Aqil Burney[3], Asadullah Shah[4], Kashif Rizwan[1,2], Adamu Abu Bakar[4], S A K Bari[1]
[1]Department of Computer Science, University of Karachi;
[2]Department of Computer Science Federal Urdu University Pakistan;
[3]CCSIS, Institute of Business Management IoBM, Karachi, [4]Pakistan; KICT, International Islamic University Malaysia

ABSTRACT— Most of the real world applications involve objects defined over both spatial and temporal attributes, usually termed as spatio-temporal objects. These objects can be found in various applications and in various domains such as traffic monitoring, environmental systems, weather systems, disease spread-out, multimedia content management, bio-diversity, air traffic control systems and many more. Classical database models don't have an inbuilt semantics to represent spatial and temporal attributes in an efficient way. The object-relational database model is more realistic and suitable for representing spatio-temporal objects keeping both the object definition and relational structure together. In this paper we present architecture for spatio-temporal database design (STDDA) and highlight the important concepts for designing a database model for involving both spatial and temporal data semantics.

Keywords—Spatio-temporal objects, relational model, object oriented model, object relational model, spatio-temporal database.

I. INTRODUCTION

A spatio-temporal data model signifies the temporal evolution of spatial objects over time. Spatio-Temporal Database Systems (STDB) have the intrinsic mechanism to offer complete database functionality for managing applications, involves both spatial and temporal characteristics [9,11]. Mostly, the relational database model and object oriented database model are further extended to include spatial and temporal semantics to manage data in various dynamic environments such as air traffic control, traffic monitoring [13] disease spread-out [24], web data semantics [21,29] and environmental changes.

STDB has been an important topic of research for the last decade, as there are several real world applications like, monitoring applications, land control systems, road network management, land management, and geographic information systems, have both spatial and temporal behaviors which should be modeled and stored in the database. Objects having both spatial and temporal behavior are very complex in nature and the traditional database models are deficient in managing multi-dimensional behavior of objects in an effective and efficient manner [9,11,28].

Spatio-temporal object changes its shape and dimension over its itinerary, exhibits both spatial and temporal qualities, it may change its dimension (i.e., shape and location) with respect to time. The question is how to model and store such objects in databases.

The commercial database management systems are constrained in providing full spatial and temporal data support. They are deficient in providing spatio-temporal data types, spatio-temporal operations, indexing mechanisms and efficient spatio-temporal query language support. There are different database layered architectures in use to support spatio-temporal features in existing database applications with limited functionalities. Unlike simple objects, spatio-temporal objects generate continuous data and it is difficult to capture the real semantics in a database. In the case of spatio-temporal objects we have an extra dimension, object value, the time of occurrence and the spatial coordinates. Spatio-temporal nature of objects have become a main aspect of various real world applications and providing DBMS support for these applications is very much needed. For example, in real time monitoring and tracking systems, providing effective efficient and instant handling of spatio-temporal data is of prime importance. Managing spatio-temporal data involves many challenges, in terms of defining appropriate data structures, definition, storage, retrieval and manipulation [9,28].

Although, spatio-temporal semantics of data objects have been studied and various database models and architectures are proposed but there are few complete STDB systems exist to manage ubiquitous nature of complex objects and provide full support for spatio-temporal data types, index mechanisms, operations, spatial and temporal constraints and query processing [6,30]. It is essential to streamline the important aspects of STDBs and to identify key issues involved in STDB design. Therefore, we proposed generic Spatio-Temporal Design Architecture (STDDA) for object relational environment.

The rest of this paper is organized as follows. In section 2 we proposed architecture STDDA. In Section 4, we investigate few spatio-temporal models and architectures proposed in the literature. In section 5, we covered the related work and finally in section 6, we highlight some issues for future work and present our conclusions.

II. METHODS

An Object-Relational Database (ORDB) model is primarily based on relational data model with implicit object oriented features. The key is to maintain the fundamental structure of a relation as per the relational model. However, certain object oriented features can be associated with the definition of a relation which includes definition of abstract data types, user defined types, behavior, class abstraction and object referencing. Modern database systems have already included some of the important object oriented features in the product. ORDB systems have the inbuilt semantics for managing the spatio-temporal data and semantics [9].

The advantages of ORD include impedance mismatch removal, class definition, object referencing, more effective modelling mechanism of complex objects, use of abstract data types, object persistent and object encapsulation. There are some implementations of the spatial and temporal databases based on object relational approaches [18].

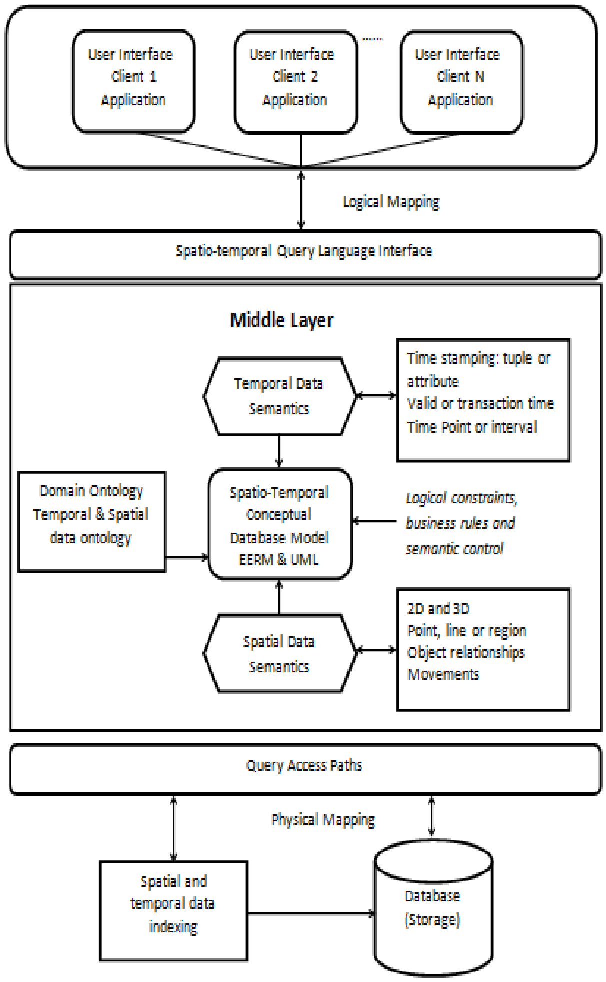
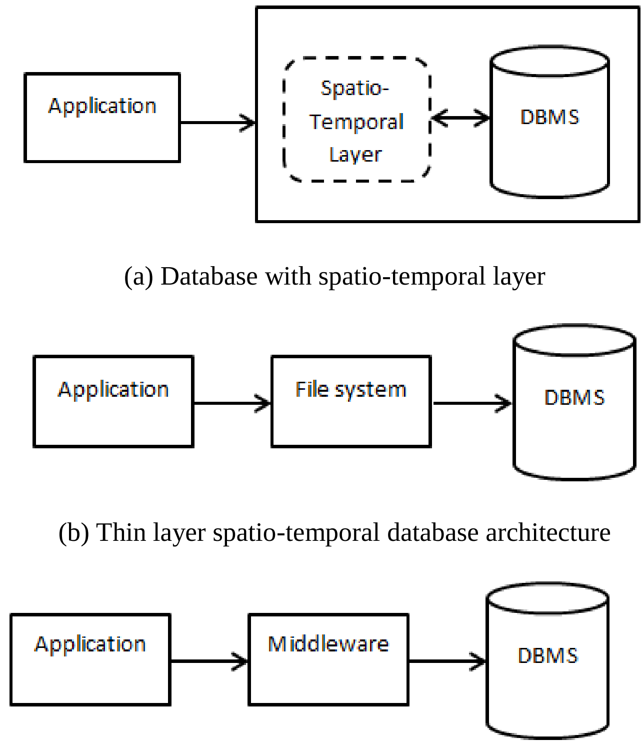


Figure 1: STDB Design Architecture (STDDA)

III. Spatio-Temporal Database Design Architecture (STDDA)

The proposed architecture is based on the existing architectures and it provides a generic approach for the design of STDB. The basic idea is to highlight the temporal and spatial semantics useful in the design of such databases. The proposed architecture STDDA ***"Fig. 1"*** is divided into three layers, presentation (application), middle layer (logical) and physical layer (database). The users at the presentation layer use the database in the same manner and much of the task is done at the middle layer, which has a spatio-temporal query interface with the presentation layer. Users can run queries involving spatial and temporal constructs and access the database. The presentation layer may be designed in accordance with the application and the domain for example, an interface for the physicians to look after the cancer growth in the body or a user interface to monitor weather conditions in a certain region.

Spatio-Temporal Database Systems (STDB) are mainly categorized into three types. First, called standard relational with additional spatio-temporal layer is deployed to work on top of a standard relational DBMS, shown in ***"Fig. 2a"***. Second, thin layer spatio-temporal DBMS architecture with standard DBMS as underlying layer and a file system. Spatial and temporal data and indexes are stored in the file system, represented in ***"Fig. 2b"***. Third, thick layer spatio-temporal DBMS architecture, which is more realistic and deal with the extension in existing DBMS to incorporate complex data types, access methods, storage definitions and query processing methodology. In fact, it redefines the whole database kernel shown in ***"Fig. 2c"***.



(a) Database with spatio-temporal layer

(b) Thin layer spatio-temporal database architecture

(c) Thick layer spatio-temporal database architecture

Figure 2: Database Design Architectures

The middle layer of this architecture is really important and it deals with the temporal and spatial semantics in detail [9][11]. There are many important concepts attached with time such as representation of time, time point to time interval, discrete vs continuous time, valid or transaction time, attribute or tuple time-stamping and time granularity [5][8]. Similarly the spatial objects have their own characteristics, object representation, dimension, direction, movement, and spatial relationships. In many real life applications the spatial object has time dimension called as spatio-temporal object. The object orientation, movement or displacement is with respect to the time value (timestamp) associated with the object. The identification of spatio-temporal objects, their characteristics and behaviors is a difficult task and it must be reflected in the database design. Building domain ontologies for specific applications is another important concern [11][15]. Each application has its own data ontology which is specific for a certain domain. Geographical information systems, land management systems, medical and health applications have their own data ontology and it must be incorporated into final database design.

The conceptual design is based on either the extended version of ERM or UML approach in the case of relational and object oriented databases respectively [15][20]. However, the object-relational database is the best model for the STDB design because it follows the strong mathematical foundations of relational model and also incorporates the some of the important object oriented features like abstract data type, inheritance, and object persistence.

A. Temporal Data Semantics [17,59]

1) Valid Time (VT)

Valid time records the time in the database when a given fact is effective or valid on a certain time point or interval. It captures the time when an event or fact happened in the modelled reality. Valid time is always independent as compare to the recording (transaction) time of a fact. Most of the models use two attributes to represent start and end time of a valid time interval.

2) Transaction time (TT)

The transaction timestamps defines the time when a record is inserted, deleted or updated in temporal database. Transaction time is also called as recording time. It is important to keep versions of the database and rollback operations can be performed on the database. It is represented as a pair of attributes (transaction start time, transaction end time)

3) Bi-Temporal

A bi-temporal database model includes both valid and transaction time dimensions. Complete temporal behaviour (historic and rollback) is only represented in the bi-temporal database model. BCDM model is a bi-temporal model and represents the complete temporal semantics.

4) Time Domain

Time is a very complex phenomenon to model in temporal databases because it is continuous and multi-dimensional in nature. Time domain can be further classified as discrete time domain ((T, ≤), isomorphic to natural numbers) and continuous time domain ((R, ≤), isomorphic to real numbers). Temporal database models are also categorized as linear time and branching time models. Linear time refers to ordered set of time units flow from past to future whereas branching time is divided into many timelines from a certain time point, which are linearly ordered.

5) Time Granularity

Time granularity defines the time units (seconds, minute, hour, day, week) associated with a temporal fact. Usually we define granularity by partitioning the Gregorian calendar into fine granules according to the definition of temporal object. Granularity is defined as a mapping function which maps the time domain into a set of numbers (integers). The smallest possible granule is termed as chronon [2].

Managing mixed granularities in a database is a difficult task [3]. Conversion functions and procedures are required to convert granularities into one another during query processing and data retrieval.

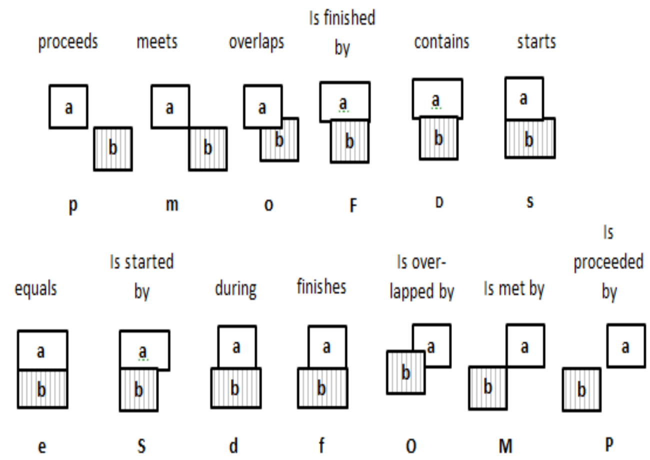


Figure 3: Allen's Basic Relations

6) Time Points and Time Intervals

Time point refers to a single attribute which stores a time value (timestamp) associated with any time varying attribute. In contrast, time interval records a pair of values, start and end time attributes to represent interval. The third approach is to record a set of time intervals associated with a certain fact. Most of the temporal database models use time interval approach for modelling. A very good analysis and study of point and interval types can be found [23]. Allen [1], in his remarkable work, presented a set of thirteen temporal comparisons operators (predicates) for time intervals, given in ***"Fig. 3"***.

7) Tuple Time-Stamping

In this approach timestamp is recorded with a tuple of a temporal relation [8]. Whenever there is a modification (insert, update, delete) in an attribute values a new tuple is added into a relation. The advantage of tuple time stamping approach is its compatibility with the classical relational model as it follows the First Normal Form (FNF) assumption. Most of the proposed temporal database models are the extensions of relational model and used tuple time stamping approach [8,13]. Tuple time stamping can be implemented using both valid time and transaction time attributes. Data redundancy is one of the bigger disadvantages of tuple time stamping technique.

8) Attribute Time-Stamping

In this approach the timestamp is recorded with every time varying attribute in a temporal relation [13]. Different attribute timestamps can be used if there are multiple time varying attributes in a relation. Therefore, there is no data redundancy in attribute time stamping compare to time stamping approach. Historic values are represented in the same tuple for each time-varying attribute of a single relation [8]. Theoretically, the attribute time stamping approach is more convenient and natural but practically it is more complex then tuple time stamping approach.

10) **Temporal Database Types**

The temporal databases are further categorized into four types, on the basis of combination of valid time and transaction time [8]. These four types are snapshot, rollback, historical and bi-temporal databases represented in ***"Fig. 4"***.

	¬ (TT)	(TT)
¬ (VT)	Snapshot Database (Classic approach)	Rollback Database (keep versions)
(VT)	Historical Database (manage histories)	Bi-Temporal Database (Eg. BCDM)

Figure 4: Temporal Database Types

B. ***Spatial Data Semantics***

A spatial data model is a data model which defines the properties of static objects and the operation performed on those objects [11]. These objects may be defined by spatial data types like point, line, and region are represented in ***"Fig. 5a"***. Examples of point, line and region are all the bus stops for a local bus service in a city, connections (pathways) from one railway station to another in a city, and disease spread in certain areas of a city, respectively.

There are certain operations to be performed on spatial objects, which include, the set operations (geometric intersection, union, and difference), other operations such as (overlap, meet, include) and to locate the direction of an object with respect to another object as shown in ***"Fig. 5b"***.

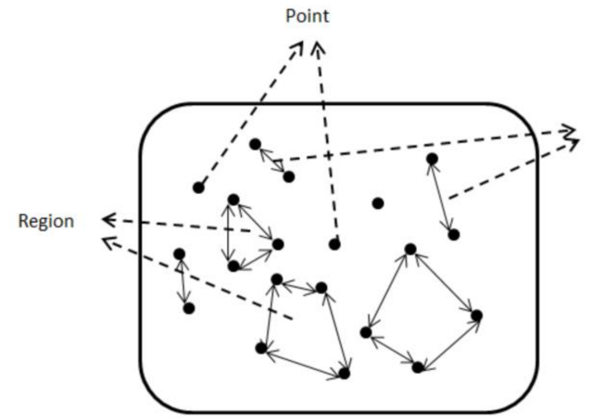
Spatial databases are designed to represent, store and process spatial data efficiently with the help of spatial attributes. Spatial attributes store the position (coordinates) of the object w.r.t. the geographic environment. A spatial attribute on one hand have an attribute value and also have a spatial attribute, for instance "traffic signal" on a road network inside a "city", represents the position of an object and not the attributes of an object itself. Spatial database offers, spatial data types, constraints, data models and query processing.

Table I: Space vs Time

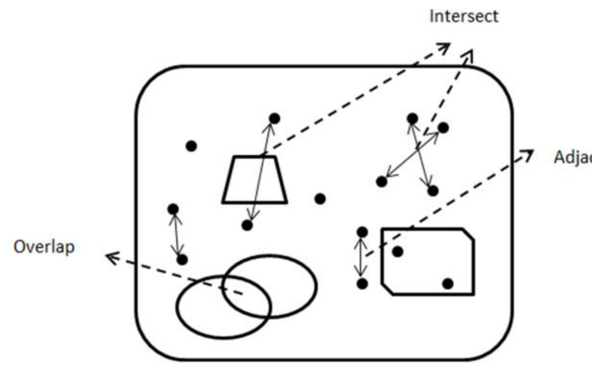
Space	Time
Objects or containers for objects	Objects exist (move) in space with respect to time point or time interval
Spatial domains may be discrete, dense or continuous	Temporal domain may be discrete, dense or continuous
Random movement of objects in space	Time is not random, it is unidirectional
Spatial object in space is dynamic but static in case of container	Time is dynamic
Limited definition	Unlimited definition
Follows non-linear order	Follows linear order (T,<)
Limited dimensions	Unlimited dimensions

C. ***Spatio-Temporal Semantics***

Spatio-temporal semantics (ontology), represents both spatial and temporal concepts as shown in ***"Table I"***. STDB model should incorporate both spatial and temporal database concepts while implementing STDB [4]. Hence, it is vital to



(a) Represents point, line and regions



(b) Represents relationships b/w spatial objects

Figure 5: Spatial object definition

comprehend the important database concepts required to create STDB systems. As a first step, research in spatial and temporal areas was realized independently.

Spatial databases have concentrated on design, modeling and querying dimensions (geometries) associated with objects, but keeping in the database the current status of spatial objects. Thus, spatial databases are static databases. Temporal databases, however, expanded the information in databases so that the previous states of objects were kept along with the current state of the modelled reality.

STDBs have the capacity to manage database applications which possesses both spatial (space coordinates) and temporal (timestamps) attributes. Most of STDB's are the extensions of spatial database models to include temporal features, in order to deal with the complex dynamic environments, like moving objects, traffic flow etc. The need to join spatial and temporal data in a single application arose as a natural fact.

D. ***Spatio-Temporal Modelling Requirements***

There are many considerations in modeling spatio-temporal data in the literature [4][11]. Some of the important spatio-temporal modeling requirements are:

- Object identification and representation with spatial coordinates (attributes) and its existence in time (point or interval).
- Need to capture the change in object movement (displacement) in space with respect to time.

- Definition of spatial attributes and organization into layers.
- Need to capture the change of spatial attributes over time.
- Association of spatial attributes to objects.
- Representation of spatial relationships among different objects with respect to time.
- Representation of relationships among spatial attributes in time.
- Specification of spatiotemporal integrity constraints for data correctness.

E. Spatio-Temporal Query Languages

The standard query language, SQL has no inbuilt semantics for managing spatio-temporal data in a database environment. Extended versions of database query languages are required to handle spatial and temporal data. Various types of query languages that are used for spatio-temporal query processing have been proposed in the literature. Most of them are the extended versions of existing query languages. HQL (Hibernate Query Language) is an extension of DEAL, and it provides almost all programming constructs (nesting, loop, conditional structure, functions etc.). It has operations similar to spatial relationship operators and good choice for spatio-temporal query handling. Spatial query languages in this context include Geo-Quel and GeoQL, STQL (extension of SQL)[10], which may be considered for Spatio-temporal query handling. Aforementioned languages serve for both spatial and temporal characteristics and offers generalized spatio-temporal query language paradigms. Representation of continuously moving objects is a big weakness of STQL. ST4SQL [22]. SQLST [7] is based on Worboy's perception for spatio-temporal model, It is simple and based on time-point representation and provides directed-triangulation-based representation of spatial objects. Viqueira and Lorentzos [25] proposed SQL extension for spatio-temporal data, based on formal extension of the relational model.

IV. RESULTS AND DISCUSSION

It is important to highlight the contribution of Tansel and Guting in the fields of temporal and spatial databases respectively. Their effort is the foundation for STDB models. Guting and Schneider [12] proposed a model which is totally independent of any specific data model. Guting [14] also proposed model for spatio-temporal data management.

The Chorochronos project [4] proposed a framework for spatio-temporal DBMS and extended the entity relationship model to include temporal, spatial and spatio-temporal entities and relationships. Wachowicz and Healy [26] proposed a spatio-temporal data model, based on object oriented approach for handling geometric, topological and thematic properties of an object. Worboys [29], first time proposed the concept of object orientation in the context of modeling spatio-temporal data. He introduced the notion of spatio-temporal object.

Other important models include DOMINO [27] used for moving object databases, build on the concept of application layer functioning on the top of DBMS and focuses on present and future trajectory data. Peiquan [15] proposed the STXER approach, extension of ER model to incorporate spatial and temporal semantics. Matos [18] used abstract data types and modified the internal structure for the representation of spatio-temporal data.

STOC [30] extension in Oracle by using spatio-temporal data types and related operations developed as PL/SQL package. STXER [16] is an extended ER model for STDBs with XML support. Another important approach is called MADS [19], based on object-relational model which incorporates spatial and temporal modeling constructs.

V. CONCLUSION

This research proposes an architecture called STDDA for the design and implementation of STDBs. This framework is flexible because it is independent of any specific technology and product. Secondly, it is not restricted to specific spatial and temporal characteristics of an object but it can be integrated with new data types. Furthermore, this is a generic architecture for STDB design work with both relational and object oriented methodology. The paper presents different modeling approaches specifically in the context of spatial, temporal and STDB management. There are few limitations of the proposed framework. First, STDDA mainly deals with spatial and temporal dimension does not define techniques for multimedia data management. This framework is not applicable to real time database systems. Uncertain data and fuzzy data semantics are not included in the proposed model.In future, we extend this work to implementation phase and to develop spatio-temporal query language constructs and to incorporate more object oriented features in the design. We may further extend this work to introduce fuzzy semantics for spatio-temporal objects.

REFERENCES

[1] Allen J. F., "Maintaining knowledge about temporal intervals," Communications of the. ACM, vol. 26, no. 11, pp. 832–843, Nov. 1983.

[2] Bettini C., Dyreson C., Evans W., Snodgrass, R., and Wang X., "A glossary of time granularity concepts" In Temporal Databases: Research and Practice: State-of-the-Art Survey, O. Etzion, S. Jajodia, and S. Sripada, Eds. Springer-Verlag, New York, 406-413, 1998.

[3] Bettini C., Jajodia S. J., Sean and Wang S. X., Time Granularities in Databases, Data Mining and Temporal Reasoning, Springer-Verlag New York, Inc., Secaucus, NJ, 2000.

[4] Breunig M., Can T., Michael H. B., Dieker S., Jensen C. S., Relly L. and Scholl M. "Architectures and Implementations of STDB Management Systems," In Spatio Temporal Databases, The CHOROCHRONOS Approach. LNCS, 2520Springer, Heidelberg, pp. 263–318, 2003.

[5] Burney A., Mahmood N., and Ahsan K., "TempR-PDM : A Conceptual Temporal Relational Model for Managing Patient Data", Proc. WSEAS Recent Adv. Artificial Intelligence Knowledge Eng. & Databases, pp. 237-243, 2010.

[6] Carvalho A., Ribeiro C., and Sousa A. A., "A STDB System Based on TimeDB and Oracle Spatial," In International Federation for Information Processing,

Volume 205, Research and Practical Issues of Enterprise Information Systems, Springer, vol. 205, pp. 11–20, 2006.

[7] Chen C. X. and Carlo Zaniolo C., "SQLST: a spatio-temporal data model and query language", Proc. of the 19th international conference on Conceptual modeling, Salt Lake City, Utah, USA, Oct. 2000.

[8] Date C. J., Temporal Data and the Relational Model, Morgan Kaufmann Publishers, 2003.

[9] Erwig M., Güting R. H., Schneider M., and Hagen D., "Spatio-Temporal Data Types : An Approach to Modeling and Querying Moving Objects in Databases" Geoinformatica, vol. 3, no. 3, 1999.

[10] Erwig M. and Schneider M., "STQL: A Spatio-Temporal Query Language", In: Mining Spatio-Temporal Information Systems, pp. 105-126, 2002.

[11] Frank A. U., "Ontology for STDBs," in T. Sellis et al. (Eds.): STDBs, LNCS 2520, pp. 9–77, 2003.

[12] Gutting R. H. and Schneider M., "Realm-Based Spatial Data Types: The ROSE Algebra", VLDB Journal, vol. 4, pp. 100-143, 1995.

[13] Jensen C. S., Clifford J., Gadia S., Hayes P., Jajodia S., Wolfgang K., Kline N., Montanari A., Roddick J. F., Sarda N. L., Arie S., Snodgrass R. T. and Tansel, A., "A Consensus Glossary of Temporal Database Concepts," SIGMOD Record, vol. 23, no. 1, pp. 52-64, 1994.

[14] Jensen C. S., Guting R. H., Erwig M., "A foundation for representing and querying moving objects.," ACM Trans. Database Systems, vol. 25, no. 1, pp. 1-42, 2000.

[15] Jin P., Wan S., and Yue L., "Ontology-driven Conceptual Modeling for Spatiotemporal Database Applications," 1st Int. Symp. Data, Privacy, E-Commerce (ISDPE 2007), vol. 4, pp. 110-112, Nov. 2007.

[16] Lin S., Jin P., Zhao L., Wang H., and Zhang L., "Augmenting Traditional ER Method to Support Spatiotemporal Database Applications," Proc. ApWeb 2011, Springer-Verlag, pp. 388-392, 2011.

[17] Mahmood N., Burney A. and Ahsan K., "Generic Temporal and Fuzzy Ontological Framework (GTFOF) for Developing Temporal-Fuzzy Database Model for Managing Patient's Data", Journal Universal Computer Science, vol. 18, no. 2, pp. 177-193, 2012.

[18] Matos L., Moreira J., and Carvalho A., "Representation and management of spatiotemporal data in object-relational databases", Proc. of the 27th Annual ACM Symposium on Applied Computing - SAC '12, p. 13, 2012.

[19] Parent C., Inforge H. E., Spaccapietra S., and Zimányi E., "Spatio-Temporal Conceptual Models : Data Structures + Space + Time," Proc. ACM Int. Symp. Advances in Geographic Information Systems (ACM-GIS), pp. 26-33, 1999.

[20] Pelekis N., Theodoulidis B., Kopanakis I., and Theodoridis Y., "Literature Review of Spatio- Temporal Database Models," The Knowledge Engineering Review, vol. 19, no. 3, pp. 235-274, 2004.

[21] Peuquet D. J. and Duan N., "An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data," International Journal of Geographical Information Systems, vol. 9, no. 1, pp. 7-24, Jan. 1995.

[22] Pozzani G. and Combi C., "ST4SQL: a spatio-temporal query language dealing with granularities", Journal of Multimedia Processing Technology, 2(3):140-157, September 2011.

[23] Toman D., "Point vs. Interval-based Query Languages for Temporal Databases," Proc. of the 15th ACM SIGACTSIGMOD-SIGART Symposium on Principles of Database Systems (PODS), Montreal, Canada, pp. 58-67, 1996.

[24] Torabi M. and Rosychuk R. J., "Spatio-temporal modelling using B-spline for disease mapping: analysis of childhood cancer trends," J. Appl. Stat., vol. 38, no. 9, pp. 1769-1781, Sep. 2011.

[25] Viqueira J. R. and Lorentzos N.A., "SQL extension for spatio-temporal data," VLDB Journal, vol. 16, no. 2, pp. 179-200, Jul. 2006.

[26] Wachowicz M. and Healey R. G., "Towards temporality in GIS," In Worboys, M.F., editor, Innovations in GIS, Taylor & Francis, 1994.

[27] Wolfson O., Xu B., and Chamberlain S., "DOMINO : Databases for Moving Objects tracking," Proc. Of ACM SIGMOD 99 Phildelphia, pp. 547-549, 1999.

[28] Worboys M. F., "A Unified Model for Spatial and Temporal Information," The Computer Journal, vol. 37, no. 1, pp. 26-34, 1994.

[29] Ye X., Peng Z., and Guo H., "Spatio-Temporal Data Model and Spatio-Temporal Databases", Temporal Information Processing Technology and its Applications, Springer, Berlin, pp. 91-112, 2011.

[30] Zhao L., P. Jin, X. Zhang, L. Zhang, and H. Wang, "STOC : Extending Oracle to Support Spatiotemporal," Proc. APWeb 2011, Springer Verlag, no. 1, pp. 393-397, 2011.