RESEARCH ON UNIFIED SPATIOTEMPORAL DATA MODEL

Peiquan Jin [a, *], Lihua Yue [a], Yuchang Gong [a]

[a] Dept. of Computer Science and Technology, University of Science and Technology of China,
Jinzhai Road 96#, Hefei, 230027, P.R. China - (jpq, llyue, ycgong)@ustc.edu.cn

KEY WORDS: Spatiotemporal Data model, Spatiotemporal Changes, Object-Relational Database

ABSTRACT:

Spatiotemporal data model, which is the basis of spatiotemporal data management, deals with the representation and manipulation of spatiotemporal data. So far many spatiotemporal data models have been proposed, but most of them are towards specific applications and few are for general purposes. It is not feasible to design respective data management systems (DBMS) for different applications because it is expensive and exhausting. Aiming to provide general support for spatiotemporal applications in DBMS, our idea is to design a unified spatiotemporal data model and then to implement a general spatiotemporal DBMS. This idea is much like the development of relational database technology. Traditional relational DBMS can support general applications, and is not for specific applications, and most importantly it is successful in real applications. In this paper, we study several issues on the unified spatiotemporal data model, including the data structure, the querying operations and the implementation issues. The unified spatiotemporal data model is based on the object-relational data model. Spatiotemporal objects are represented as spatiotemporal relations and spatiotemporal tuples. And queries on spatiotemporal relations are implemented through an extended relational algebra. The spatiotemporal changes of spatiotemporal objects are represented by spatiotemporal data types, and queries on spatiotemporal data and changes are supported by the operations defined on those data types. The data structure of the unified spatiotemporal data model, the querying operations along with the algorithm to transform the data model to object-relational database management systems are discussed in detail. The unified spatiotemporal data model has been partly implemented on Informix, and a real example about China Historical Geographic Information System (CHGIS) is discussed. The experimental results show that the unified spatiotemporal data model is able to represent, store and query spatiotemporal changes successfully.

1. INTRODUCTION

Spatiotemporal data model deals with the representation and operation of spatiotemporal data, and has taken an important role in the research on spatiotemporal database and temporal GIS. The key and difficult issue of spatiotemporal data model is the representation and querying of spatiotemporal changes, since spatiotemporal change is the main property that spatiotemporal database differs from spatial database.

To represent spatiotemporal changes of spatial objects, many researchers added timestamps into the spatial attributes and thematic attributes of spatial objects, and developed a lot of spatiotemporal data models, e.g. the spatiotemporal snapshot model [Langran, G., 1992], the base state and amendment model [Langran, G., 1992], the spatiotemporal cube model [Langran, G., 1992], the spatiotemporal composite model [Langran, G., 1992] and the spatiotemporal object model [Worboys, M., 1994]. In these models, timestamp and version were used to represent the states of spatiotemporal objects, and spatiotemporal changes could be queried through the comparison on versions. However, these models provided weak supports for spatiotemporal changes, e.g. they could not represent those changes that involve more than one objects (split, mergence, et al.). Meanwhile, the querying efficiency was very low, since in each spatiotemporal query a lot of versions must be referred in order to know the spatiotemporal changes.

Forlizzi et al. presented a spatiotemporal data model based on data types [Forlizzi, L., Güting, R. H., et al. 2000]. In this model, type constructors are used to represent spatiotemporal changes. A type constructor is a function that defines new types depending upon time. Spatiotemporal changes are queried by the operations defined on type constructors. This model is more practical that other models, because we can implement it in commercial object-relational database management systems using their extended abilities. But it still can not model changes related to more than one objects.

The event-based spatiotemporal data model [Peuquet, D. J., et al. 199] looked each spatiotemporal change as an event, and represented spatiotemporal changes through an event list. However, it still does not support changes related to many objects. Chen Jun et al. improved this model by introducing changing reasons [Chen, J., et al. 2000], but their model can only support land subdivision applications and not for general purposes.

Other researchers studied spatiotemporal data modelling technologies from an object-oriented view. Hornsby qt al. presented an identity-based framework for the representation of spatiotemporal changes [Hornsby, K., et al. 2000]. But this approach can not support changes of thematic attributes and spatial attributes which do not change any object identities. The history-graph model [Renolen, A., 1997] utilized the OMT (Object Modelling Technology) model and presented a semantic data model for spatiotemporal information representation. But it can only represent discrete spatiotemporal changes. The same problems also exist in the models developed in [Yi, S., Zhang, Y., Zhou, L., 2002, Zheng, K., Tan, S., Pan, Y., 2001].

* Corresponding author.

The same limitation of the above spatiotemporal data models is that they do not support continuous spatiotemporal changes. In [Sistla, A., Wolfson, O., et al. 1997] a moving-object data model was proposed to modelling the changes of moving objects such as cars and ships. But the moving-object data model can only suit those applications that only need to trace the continuous changes of locations. Cai Mengchu[Cai, M., Keshwani, D., et al. 2000] and Yujun Wang [Wang, Y., Wang, W., Shi, B., 1999] researched spatiotemporal data model based on constraint database. The parametric rectangle model represented a spatial object as a set of parametric rectangles. And the continuous changes of a spatial object were represented by the changes of all parametric rectangles that the object contained. The problem of this model is that the set of parametric rectangles is difficult to obtain in real applications. This problem exists in all data models based on constraint database.

The problems of previous spatiotemporal data models can be summarized as follows:

(1) They did not present a unified way to represent and query spatiotemporal changes.

(2) They did not support both discrete spatiotemporal changes and continuous spatiotemporal changes.

(3) They were short of effective technologies to implement their data model [Paton, N., Fernandes, A., Griffiths, T., 1998].

To solve these problems, a new unified spatiotemporal data model is proposed in this paper. The new model is based on the object-relational data model. It uses abstract data types (ADT) to represent spatiotemporal changes and operations on ADTs to query spatiotemporal changes. Different ADTs are designed for different types of spatiotemporal changes, and they can be transformed into objtec-relational database management systems (ORDBMS) for implementation.

The remainder of this paper is structured as follows: Section 2 discusses the data structure of the unified spatiotemporal data model. Section 3 presents the operations for manipulating spatiotemporal data. Section 4 introduces the algorithm transforming the unified spatiotemporal data model into ORDBMS for implementation. Section 5 discusses the experiment of implementing the unified spatiotemporal data model. And conclusions are presented in Section 6.

2. DATA STRUCTURES

In the unified spatiotemporal data model, a spatiotemporal object is represented as a tuple, and we define it as a *spatiotemporal tuple*. A spatiotemporal relation is a set of spatiotemporal tuples. The key idea of our data model is to design new abstract data types for spatiotemporal changes and new operations to query spatiotemporal changes.

DEFINITION 1. Given n domains D_1, D_2,…, D_n, and at least one domain is a spatiotemporal domain, every element in the product $D_1 \times D_2 \times \ldots \times D_n$ is called a spatiotemporal tuple.

DEFINITION 2. A spatiotemporal relation is a set of spatiotemporal tuples.

The basic data structure of spatiotemporal objects is shown in Fig.1, in which the spatiotemporal attributes are extended data types representing spatiotemporal changes.

Primary Key	Thematic Attributes	Spatiotemporal Attributes

Figure 1. The data structure of the data model

The type system of the unified spatiotemporal data model is shown in Fig.2, which is defined by the *Second-Order Signature* [Güting, R., 1993]. Second-Order Signature is a tool for the formal definition of data mode. It can define not only the relational data model but also other advanced data models, e.g. image data model and spatial data model.

According to the *Second-Order Signature*, a data model is defined by two signatures. The first signature defines the type system of the data model, and the second signature defines the operations on the type system. In this section, we discuss the type system, which is first signature. The second signature defining the operations on the type system will be discussed in the Section 3.

In the type system shown in Fig.2, the *kind* is a set of *types*. The *type constructor* is a function which produces new types from existing types. For example, the type constructor *std* and *stc* produces data types such as *std*(*point*) and *stc*(*region*) to represent discrete spatial changes and continuous spatial changes.

Actually, in our model, five types of spatiotemporal changes are concerned, which are listed as follows:

(1) **TYPE 1** (*continuous spatial processes*): the spatial attributes of an object change continuously with time, such as spread of fire, flowing of flood and moving of a car. These changes are always related to a period of time.

(2) **TYPE 2** (*discrete spatial processes*): the spatial attributes stay static during a period and suddenly change to another value. A typical example is the change of the boundary of a land. Discrete spatial processes always happen in a specific instant.

(3) **TYPE 3** (*continuous thematic processes*): the thematic attributes of an object change continuously with time, such as changes of soil type.

(4) **TYPE 4** (*discrete thematic processes*): the thematic attributes are basically static, but they may be changed into another values suddenly, e.g. the change of the ownership of a land.

(5) **TYPE 5** (*discrete life*): sudden changes that result in creation of new objects or deletion of existing objects, such as split of a land or mergence of several lands.

Most of spatiotemporal changes fall in these five spatiotemporal changes. Since the unified spatiotemporal data model is designed to represent these five types of spatiotemporal changes, the model can suit most spatiotemporal applications. Table 1 shows the corresponding types in the unififed spatiotemporal data model for these five changes.

Spatiotemporal	Types in the unified

changes	spatiotemporal data model
TYPE 1 and TYPE 3	Types produced by the type constructor *std*
TYPE 2 and TYPE 4	Types produced by the type constructor *stc*
TYPE 5	*mht*

Table 1. Corresponding data types in the unified spatiotemporal data model for the five types of spatiotemporal changes

The kind *HT* contains only one const type, which is *mht*. This type is also used to represent spatiotemporal changes, but it differs from the type constructor *std* and *stc*. The types produced by the type constructor *std* and *stc* represent spatial processes and thematic processes, which only result in changes of attributes of an object, e.g. moving of an object (in this case only the position of the object changes with time). The type *mht* represents the life cycle of a spatial object. A spatial object may be created or deleted and we use *mht* to represent such changes. The *mht* is a list of *htstate*, each of which represents one life change of the object.

The kind *IDENT* contains only one const type *ident* which represents a domain of identifiers (these identifiers can used as attribute names). For any spatial data type *s* in the kind *SDATA*, the type constructor *std(s)* or *stc(s)* constructs the corresponding spatiotemporal data type which is in the kind *STDATA*, e.g. *stc(point)* and *stc(region)* are spatiotemporal types representing continuously moving points and discrete moving regions respectively. For a type *t* in the kind *STTUPLE*, *strel(t)* is a data type in the kind *STREL*. The type *strel(t)* represents a spatiotemporal relational schema, and *t* represents a spatiotemporal tuple. The symbol S^+ means a list of *S*, thus $(ident, DATA)^+$ represents a definition list of attributes. The symbol [*S*] means *S* is optional. The following is an example about the definition of type *sttuple* and type *strel*:

sttuple(<(No,*int*),(owner,*string*)>,<(boundary,*std(region)*)>,(history, *mht*))
strel(*sttuple*(<(No,*int*),(owner,*string*)>,<(boundary,*std(region)*)>,(history, *mht*)))

kinds	IDENT, DATA, SDATA, STDATA, HTSTATE, HT, STREL, STTUPLE
type constructors	
→ IDENT	*ident*
→ DATA	*int, char, bool, real, point, line, region, composition, instant, period*
→ SDATA	*point, line, region, composition*
→ HTSTATE	*htstate*
→ HT	*mht*
DATA → STDATA	*std, stc*
$(ident \times \mathrm{DATA})^+ \times (ident \times \mathrm{STDATA})^+ \times [ident \times \mathrm{HT}]$ → STTUPLE	*sttuple*
STTUPLE → STREL	*strel*

Figure 2. The type system of the data model

3. DATA OPERATIONS

The operations on spatiotemporal relation are similar to the relational algebra. There are five basic operations in relational algebra, and other operations can be expressed by these five operations. The following definitions are about the five basic extended relational algebraic operations in the unified spatiotemporal data model. Since the five operations form a complete set of relational algebra, other operations on spatiotemporal relation can also be implemented using the five basic operations.

(1) $R \cup^{st} S$ (*Spatiotemporal Union*): suppose *R* and *S* are two spatiotemporal relations, the spatiotemporal union of *R* and *S* is the set of *R*'s tuples and *S*'s tuples. Spatiotemporal union $R \cup^{st} S$ is defined as follows:

$$R \cup^{st} S \triangleq \{ t \mid t \in R \vee t \in S \}$$, where *t is a variable representing any spatiotemporal tuple.*

(2) $R -^{st} S$ (*Spatiotemporal Difference*): The spatiotemporal difference $R -^{st} S$ is a set containing those tupes in *R* but not in *S*.

$$R -^{st} S \triangleq \{ t \mid t \in R \wedge t \notin S \}$$

(3) $R \times^{st} S$ (*Spatiotemporal Product*): suppose a spatiotemporal relation *R* has *r* attributes, and *S* has *s* attributes. The spatiotemporal product of *R* and *S* is a set of spatiotemporal tuples. And each spatiotemporal tuple has (*r* + *s*) attributes. The former *r* attributes of each spatiotemporal tuple come from *R*, and the latter *s* attributes come from *S*.

$$R \times^{st} S \triangleq \{ t \mid t = < t^r, t^s > \wedge t^r \in R \wedge t^s \notin S \}$$

(4) $\sigma^{st}{}_F(R)$ (*Spatiotemporal Selection*): The spatiotemporal selection makes a horizontal partition on a spatiotemporal relation based on some conditions. That means to select the spatiotemporal tuples that satisfy the given conditions. The conditions are represented by a predication formula *F*.

$$\sigma^{st}{}_F(R) \triangleq \{ t \mid t \in R \wedge F(t) = true \}$$

(5) $\pi^{st}{}_{A_{i1}, A_{i2}, \ldots, A_{im}}(R)$ (*Spatiotemporal Projection*): The spatiotemporal projection makes a vertical partition on a spatiotemporal relation. Suppose the attributes of *R* are denoted as ($A_1, A_2, \ldots, A_r$), the value of attribute A_i is denoted as t_{Ai}, the spatiotemporal projection $\pi^{st}{}_{A_{i1}, A_{i2}, \ldots, A_{im}}(R)$ is a set of tuple, each of which has *m* attributes.

$$\pi^{st}{}_{A_{i1}, A_{i2}, \ldots, A_{im}}(R) \triangleq \{ t \mid t = < t_{A_{i1}}, t_{A_{i2}}, \ldots, t_{A_{im}} > \wedge < t_{A_1}, t_{A_2}, \ldots, t_{A_r} > \in R \}$$

Fig.3 shows the signature of the above operations.

types	*strel, sttuple, bool*
operators	
$\cup^{st}$	$strel \times strel \rightarrow strel$
$-^{st}$	$strel \times strel \rightarrow strel$
$\times^{st}$	$strel \times strel \rightarrow strel$
σ^{st}	$strel \times (sttuple \rightarrow bool) \rightarrow strel$
π^{st}	$strel \rightarrow strel$

Figure 3. The operations on spatiotemporal relation

In order to query spatiotemporal changes, we also need to define some other operations on spatiotemporal data types. These operations include temporal operations, spatial operations, life cycle operations and spatiotemporal operations. Fig.4 to Fig.7 shows the signatures of these operations.

Some symbols are used in the Fig.4 to Fig.7, as described in the follows:

α: zero dimension spatial data types.

β: one dimension spatial data types.

γ: two dimensions spatial data types.

ε: zero, one or two dimensions spatial data types

$\otimes$: $a \otimes b$ means $a \times b$ or $b \times a$。

types	*period, bool*
operators	
coalesce	$period \times period \rightarrow period$
duplicate	$period \times period \rightarrow period$
before	$period \times period \rightarrow bool$
equals	$period \times period \rightarrow bool$
meets	$period \times period \rightarrow bool$
overlaps	$period \times period \rightarrow bool$
during	$period \times period \rightarrow bool$
starts	$period \times period \rightarrow bool$
finishes	$period \times period \rightarrow bool$

Figure 4. The temporal operations

types	*real, bool,* $\alpha, \beta, \gamma, \varepsilon$
operators	
intersect	$\gamma \times \gamma \rightarrow \gamma$
cross	$\beta \times \beta \rightarrow \beta$
	$\beta \times \gamma \rightarrow \gamma$
union	$\gamma \times \gamma \rightarrow \gamma$
center	$\beta \rightarrow \alpha$
	$\gamma \rightarrow \alpha$
disjoint	$\varepsilon \times \varepsilon \rightarrow bool$
meet	$\varepsilon \times \varepsilon \rightarrow bool$
overlap	$\gamma \times \gamma \rightarrow bool$
	$\beta \times \beta \rightarrow bool$
intersects	$\beta \otimes \gamma \rightarrow bool$
contain	$\gamma \times \alpha \rightarrow bool$
	$\gamma \times \beta \rightarrow bool$
	$\gamma \times \gamma \rightarrow bool$
	$\beta \times \alpha \rightarrow bool$
	$\beta \times \beta \rightarrow bool$
equal	$\varepsilon \times \varepsilon \rightarrow bool$
distance	$\alpha \times \alpha \rightarrow real$
area	$\gamma \rightarrow real$
perimeter	$\gamma \rightarrow real$
length	$\beta \rightarrow real$

Figure 5. The spatial operations

types	*instant, period,* $\alpha^{st}, \beta^{st}, \gamma^{st}, \varepsilon^{st}$, *bool*
operators	
when	$\varepsilon^{st} \times instant \rightarrow \varepsilon$
history	$\varepsilon^{st} \times period \rightarrow \varepsilon$
stDisjoint	$\varepsilon^{st} \times \varepsilon^{st} \times period \rightarrow bool$
stMeet	$\varepsilon^{st} \times \varepsilon^{st} \times period \rightarrow bool$
stOverlap	$\beta^{st} \times \beta^{st} \times period \rightarrow bool$
	$\gamma^{st} \times \gamma^{st} \times period \rightarrow bool$
stIntersects	$(\beta^{st} \otimes \gamma^{st}) \times period \rightarrow bool$
stContain	$\gamma^{st} \times \alpha^{st} \times period \rightarrow bool$
	$\gamma^{st} \times \alpha^{st} \times period \rightarrow bool$
	$\gamma^{st} \times \alpha^{st} \times period \rightarrow bool$
	$\beta^{st} \times \alpha^{st} \times period \rightarrow bool$
	$\beta^{st} \times \beta^{st} \times period \rightarrow bool$
stEqual	$\varepsilon^{st} \times \varepsilon^{st} \times period \rightarrow bool$

Figure 6. The spatiotemporal operations

types	*htstate, mht, instant, period*
operators	
htWhen	$mht \times instant \rightarrow htstate$
htHistory	$mht \times period \rightarrow mht$

Figure 7. The life cycle operations

4. TRANSFORMING THE UNFIED SPATIOTEMPORAL DATA MODEL INTO ORDBMS

The unified spatiotemporal data model can be transformed into object-relational database management systems (ORDBMS) for implementation. Actually, in recent years more and more people advocate that the ORDBMS is the appropriate fundamental DBMS to implement a spatiotemporal DBMS [Forlizzi, L., Güting, R. H., et al. 2000, Yang, J., Cheng, H., Ying, C., Widom, J., 2000, Cindy, C., Zaniolo, C., 2000], because the ORDBMS offers extending facilities for users to add supports for complex data management. An ORDBMS provides extensibility of User Defined Types (UDT) and User Defined Routines (UDR) [Stonebraker, M., Moor, D., 1996,]. The following algorithm transforms the unified spatiotemporal data model into a set of UDTs and UDRs in an ORDBMS.

Algorithm 1: *Transforming the unified spatiotemporal data model into ORDBMS*

Input: *The type system (as shown in Fig.2) and the operations system (as shown in Fig.3 to Fig.7) of the unified spatiotemporal data model.*

Output: *Set U and R, where U is the set of UDTs and R is the set of UDRs.*

Preliminary: *The kind set in the type system is denoted as K, while the operation system is denoted as F. Both U and R are initially empty.*

```
BEGIN
  // The types in kinds IDENT, STTUPLE ands STREL already
  // exist in ORDBMS. We needn't transform them.
X:= K－ {IDENT, STTUPLE, STREL, STDATA};
For Each S In K Do
   For Each p In S Do
         U:= U ∪ {p};
   End For
End For
  // Transforming operations
For Each q In F Do
   R:= R ∪ {q};
End For
END
```

5. EXPERIMENT

In order to demonstrate the implemental feasibility of the unified spatiotemporal data model, we make some experiments on Informix Dynamic Server 9.21. Informix Dynamic Server 9.21 is an ORDBMS. We implement a spatiotemporal data management system on Informix using the extended facilities it provides. The implemented spatiotemporal data management system can represent and store spatiotemporal data. It can also query spatiotemporal changes effectively. We use the data set from China Historical Geographic Information System (CHGIS) [The CHGIS Project, 2003]. This data set contains the boundary changes and location changes of each county in Shanghai District since 751 A.D.

First, according to the experimental data set, we create two spatiotemporal relations to store the boundary changes and location changes, as shown in the follows.

His_Boundary(no *int*, name *char(50)*, boundary *region*, life *period*)

His_Location(no *int*, name *char(50)*, location *point*, life *period*)

The original data set is stored in Mapinfo format, so we first need to transform the data into the spatiotemporal database in Informix. This is done by the COM interfaces provided by Mapinfo. Fig.8 shows the final data transformed into the spatiotemporal database.

Since we build our spatiotemporal database on the ORDBMS Informix, we can directly use standard SQL to access the spatiotemporal database. The underlying interface we use in the experiment is ODBC. By using spatiotemporal operations defined in Fig.4 to Fig.7 we can query spatiotemporal changes. For instance, if we want to find the boundary changes of Shanghai from 1937 to 1978, we can issue the following SQL statements to get the result:

SELECT no, name, boundary, duplicate(life,'"1937-01-01 00:00:00,1978-12-31 23:59:59"'::period) as time
FROM his_boundary
WHERE overlaps(life,'"1937-01-01 00:00:00,1978-12-31 23:59:59"'::period) and name='Shanghai Shi'

Fig.9 shows the executing result of the SQL query.

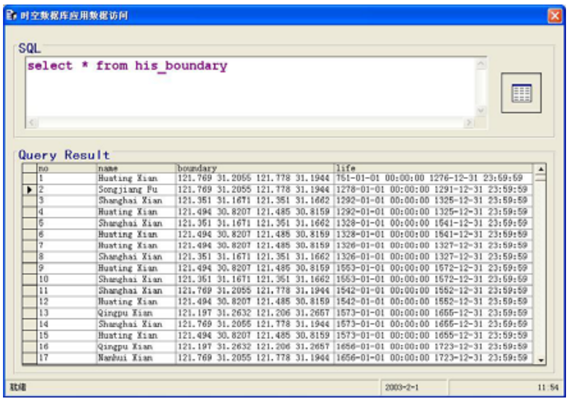

Fig.8 Spatiotemporal data transformed from CHGIS

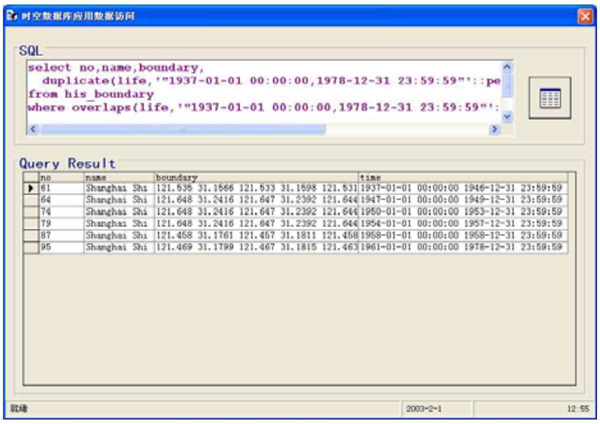

Fig.9 Find the boundary changes of Shanghai from 1937 to 1978

6. CONCLUSIONS

A unified spatiotemporal data model that can support different types of spatiotemporal changes is the foundation of spatiotemporal data management. In this paper, we present a unified spatiotemporal data model based on the object-relational data model. According to the new model, spatiotemporal objects are represented as spatiotemporal relations and spatiotemporal tuples, and querying operations are realized by extended relational algebra. Spatiotemporal changes are represented by extended spatiotemporal data types and queried through the operations defined on spatiotemporal data types. By defining new spatiotemporal data types, we can enhance the data model with better support to spatiotemporal changes. Meanwhile, this data model can be easily transformed into a typical ORDBMS for implementation. We have made some experiments on the ORDBMS-based implementation of the unified spatiotemporal data model, and the results show it is a practical way to implement a spatiotemporal DBMS on an ORDBMS.

ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation of China under the grant number 60403020 and by the High Technology Development Project of China grant 2002AA783055 from the Ministry of Technology of China.

The authors would like to thank the CHGIS project hosted by Harvard University and Fudan University for providing downloadable spatiotemporal dataset.

REFERENCES

Langran, G., 1992, *Time in Geographic Information Systems*. Taylor & Francis Ltd.

Worboys, M., 1994, A unified model for spatial and temporal information. *The Computer Journal*, 37(1), pp.26-34

Forlizzi, L., Güting, R. H., et al. 2000, A data model and data structures for moving objects databases. In: *Proceedings of SIGMOD Conference 2000*, pp.319-330

Peuquet, D. J., et al. 1995, An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *International Journal of Geographical Information Systems*, 1, pp. 7-24

Chen, J., et al. 2000, An event-based approach to spatiotemporal data modelling in land subdivision systems. *GeoInformatica*, 4, pp. 387-402

Hornsby, K., et al. 2000, Identity-based change: a foundation for spatio-temporal knowledge representation. *International Journal of Geographical Information Science,* 3, pp. 207-224

Renolen, A., 1997, History graphs: Conceptual modeling of spatiotemporal data. In: *Proceedings of GIS Frontiers in Business and Science*, Brno, Czech Republic: International Cartographic Association

Yi, S., Zhang, Y., Zhou, L., 2002, A spatiotemporal data model for plane moving objects, *Journal of Software*, 13(8), pp.1658-1665

Zheng, K., Tan, S., Pan, Y., 2001, A unified spatiotemporal data model based on status and change, *Journal of Software*, 12(9), pp. 1360-1365

Sistla, A., Wolfson, O., et al. 1997, Modeling and querying moving objects. In: *Proceedings of ICDE 1997*, pp.422-432

Cai, M., Keshwani, D., et al. 2000, Parametric rectangles: A model for querying and animation of spatiotemporal databases, In: *Proceedingd of the 7th International Conference on Extending Database Technology*, pp.430-444

Wang, Y., Wang, W., Shi, B., 1999, Interval constraint and its algebraic query language, *Journal of Computer*, 22(5), pp.550-554

Paton, N., Fernandes, A., Griffiths, T., 1998, Spatiotemporal databases: contentions, components and consolidation. In: *Proceedings of 11th DEXA Workshop on Advanced Spatial Databases (ASDM)*, A.M. Tjoa et al. (eds), IEEE CS Press, pp.851-855

Güting, R., 1993, Second-order signature: a tool for specifying data models, query processing, and optimization. In: *Proceedings of SIGMOD Conference 1993*, pp.277-286

Stonebraker, M., Moor, D., 1996, Object-Relational DBMS: The Next Great Wave. Morgan Kaufmann

Yang, J., Cheng, H., Ying, C., Widom, J., 2000, TIP: a temporal extension to Informix, In: *Proceedings of the ACM SIGMOD*, Dallas, Texas, pp.213-223

Cindy, C., Zaniolo, C., 2000, SQLST: A spatiotemporal data model and query language, In: *Proceedings of the 19th International Conference on Conceptual Modeling(ER'00)*, Salt Lake City, USA, pp.96-111

The CHGIS Project, 2003, http://www.people.fas.harvard.edu/~chgis/