5 Spatio-Temporal Data Model and Spatio-Temporal Databases

Xiaoping Ye[1,2], Zewu Peng[2], and Huan Guo[2]

[1] Computer School, South China Normal University, Guangzhou 510631, P.R. China
[2] Department of Computer Science, Sun Yat-sen University, Guangzhou 510275, P.R. China

Abstract Spatio-temporal database can deal with the objects' temporal and spatial properties uniformly, and it can manage data that is evolving with time, i.e., the information of spatial objects whose shape and position evolve with time. In spatio-temporal database, spatial data has one more time dimension, which increases the complexity of data management. In this chapter, we first introduce the concepts of spatio-temporal database, and then introduce the spatio-temporal data model, query types of spatio-temporal data and the architecture of spatio-temporal database system.

Keywords *spatio-temporal database, spatio-temporal data model, query types of spatio-temporal data, architecture of spatio-temporal database system*

5.1 Introduction

Spatio-temporal database is the combination of temporal database and spatial database. Its basic idea is to create a three-dimensional or four-dimensional database by adding time constraints for spatial database, which is mainly used to store and manage various types of spatial objects with temporal information. The main purpose of spatio-temporal database is to deal with spatio-temporal information, which usually involves the expression of spatio-temporal objects, its modeling, indexing and query of spatio-temporal data, the system structure, prototype systems and application of spatio-temporal database, and so on.

Spatio-temporal database can support complex applications with **spatio-temporal objects** and their relationships by the database core, so it has a wide range of applications. According to the types of spatio-temporal objects, the application of spatio-temporal database can be divided into the following categories:

- The application of spatio-temporal objects with continuous movement: In such

mcsyxp@mail.sysu.edu.cn

Y. Tang et al. (eds.), *Temporal Information Processing Technology and Its Application*

applications, the location of spatio-temporal objects continuously changes with time, but their shapes remain unchanged. The traffic-related spatio-temporal database applications, such as vehicle traffic management, vessel traffic management and aircraft flight management, can be classified as such applications.

- The application of spatio-temporal objects with discrete change: The spatio-temporal objects involved in this application have a spatial location, and their spatial attributes, such as shape and location, may change discretely over time. Such applications include cadastral management, city zoning management and surface vegetation changes, as well as virus, disease region detection and so on.
- The application of spatio-temporal objects with continuous movement and shape changing: Such applications usually refer to environment-related applications, such as storm monitoring and prediction, forest-fire monitoring, offshore oil pollution monitoring, as well as migration of species and so on. In addition, the biological information processing also belongs to such applications, for example, a cell's shape may change during its moving process.

5.2 Spatio-Temporal Data Model

In order to devise a spatio-temporal data model, we first discuss the basics of spatio-temporal objects.

5.2.1 Spatio-Temporal Object

The spatial objects whose spatial locations or extents change with time are called spatio-temporal objects, for example, flying aircraft, highway vehicles, the forest whose boundaries change (broadened or narrowed) over time and so on. Many practical applications need effective storing and management of spatio-temporal objects to query the objects' locations and their information in the past, at present, and to speculate their behavior in the future.

Spatio-temporal databases store time-varying spatial objects, and there are three kinds of basic spatial objects: points, lines and regions. Points represent the spatial objects whose spatial extension are zero or the ones for which we only concern on their spatial locations rather than their sizes, for example, a city in a large-scale map. Lines, including the space curves, are used to describe the facilities' traces in space or the space connection, such as roads, rivers, telephone lines, wires. Objects with certain space range are called regions, for example, a country's administration regional divisions, a mountain or lake and so on. Currently, the study of spatio-temporal data objects is mainly focused on two basic cases: moving points and moving regions and the study of line (curve) is usually converted to the study of moving point's trajectory.

A spatio-temporal object O_i can be expressed by tuple $\langle oid, p_i(t), e_i(t), t \rangle$, where *oid* is object O_i's unique identifier, $p_i(t)$ and $e_i(t)$ correspond to the location and extent of object O_i in time t, respectively. The application of spatio-temporal database is usually to insert and delete objects at any time. The time period from an object's insert time to its delete time is called the object's survival time. During the survival time, the object is alive. Spatio-temporal database's state $S(t)$ in time stamp t includes all activity objects in time t, and $S(t)$ can be seen as the snapshot of spatio-temporal object's spatial location and extent in time t. Spatio-temporal databases store time-varying spatial objects, so it needs to preserve all of the spatio-temporal statuses $S(t)$, and can query on any time's state $S(t)$. This means that in spatio-temporal database, the deletion of spatio-temporal data object is only a logical deletion. The records of deleted objects are still preserved in the database.

Spatio-temporal database can be seen as spatial database's extension in the time dimension, but the time dimension itself has the following characteristics:

- Monotony increment: Time information's change is always monotone increasing.
- Two time dimensions: Temporal information of data object usually has two time dimensions—valid time and transaction time. Spatio-temporal database needs to support valid time or transaction time or both.
- Discrete and continuous: Spatial objects always change with time evolution. Its representation method in database is related to its changing frequency and can be generally divided into two cases: discrete and continuous. In the discrete case, the changing frequency of spatial objects is slow. We can use the snapshot of spatio-temporal state to express spatio-temporal data and update data when the spatio-temporal object is changed. In the continuous case, the changing frequency of spatial objects is rapid. The corresponding spatial information change can be expressed as a location function $p_i(t)$ and scope function $e_i(t)$, respectively. t is variable time and an update only happens when corresponding time function changes.

5.2.2 Basic Considerations of Spatio-Temporal Modeling

At present, there are no uniform modeling standards for spatio-temporal data. Most of them are driven by specific applications and are devised to solve a specific problem. Judging from the current work, there are three major phases for spatio-temporal data modeling. First, determine the smallest-scale GIS unit in certain scope. Second, devise a reasonable, accurate and tight spatio-temporal relationship expression. Third, determine the model's layered architecture.

1. Spatio-temporal data information unit

Spatio-temporal information belongs to multi-dimensional data, which generally can be divided into space dimension, time dimension and attribute dimension.

- The basic element of space dimension is the spatial coordinates and it determines the spatial relationship between spatio-temporal entities (adjacent, intersection, separation and contains, etc.). For example, arbitrary point on the earth is uniquely determined by the spatial coordinates (X, Y, Z), where X, Y and Z are the longitude, latitude and altitude coordinates, respectively. In order to facilitate the query, the entire earth's surface can be cut into a number of relatively independent parts by the projection. The corresponding relationship can also be transformed from three-dimensional space to two-dimensional plane.
- Time dimension, as important data information, describes the creating, developing and deleting of spatio-temporal entities, and it can be seen as the life cycle of spatio-temporal entity or as a valid time or version information.
- Properties in attribute dimension can be divided into two types: core property and non-core property. The core property is the only identifier of the spatio-temporal entities or spatio-temporal process. The changes of the core property imply the deleting of the spatio-temporal objects (entity or process).

Only when studied with a certain scope, spatio-temporal information can be meaningful. In spatio-temporal modeling process, it is necessary to determine the basic unit in the spatio-temporal system research. There is a "constant" relationship between space and process, which is the smallest unit of spatio-temporal information: spatial and temporal information unit. As the unit of the smallest "space" with the shortest "time", spatio-temporal information unit is the appropriate "granularity" to study this problem, and it can further guarantee the "heterogeneity" of space and the "indivisibility" of time. The spatio-temporal information unit can be seen as same unit in the spatio-temporal system analysis.

The key to characterize the spatio-temporal information unit is through a suitable spatio-temporal data model. With rich, accurate expressive power, it can describe the complex, dynamic and interrelated spatio-temporal information unit. Theoretically, spatial and temporal information unit can be abstracted as the following function: $\mathrm{STIC} = F(x, y, z, A, T)$. STIC (spatial temporal Information cell) is the temporal and spatial information unit, where, x, y, z are the coordinates of spatial location, A is the set of properties and T is time.

2. Description of spatio-temporal relations

Study on spatio-temporal data entities is in fact the study on spatio-temporal relations between the information units, whose spatial relationship is the topological relationship, and time relationship is the temporal relationship. With reference to spatial databases, Egenhofer describes topological relations of two-dimensional space as {adjacency, contained, including, covering, covered, overlap, separation} (Egenhofer 1991). With reference to temporal databases, Allen divided the time relationship into {equal, before, meet, overlap, start, period, end} (Allen 2005). In spatio-temporal database, it is necessary to consider spatio-temporal relationship unitedly. Therefore, the spatio-temporal relationships that may exist between spatio-temporal data elements are shown in Fig. 5.1.

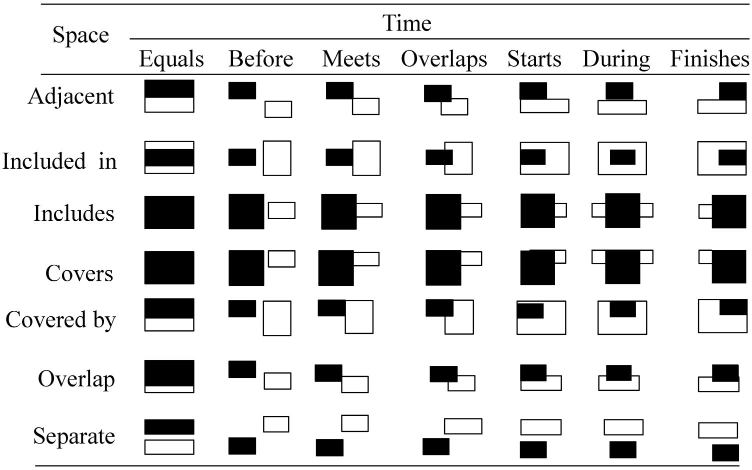


Figure 5.1 Spatio-temporal relationship

3. Hierarchical structure of spatio-temporal data model

After abstraction, classification, calculation and association, spatio-temporal information units can be implemented in the computer system. Similar to general data modeling, spatio-temporal data modeling also has a stepwise layer process, which is shown in Fig. 5.2.

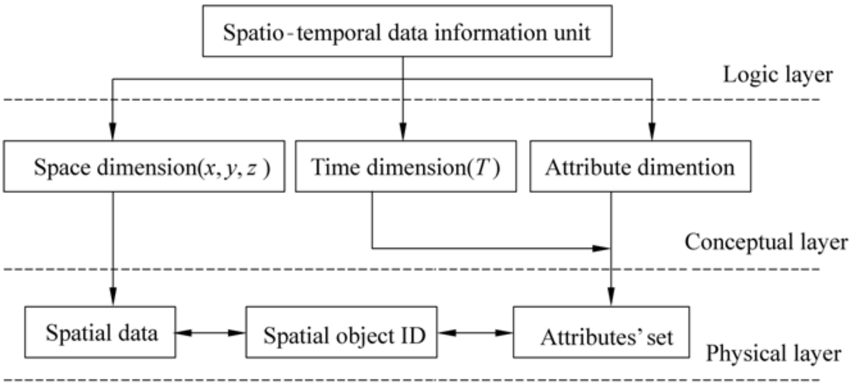


Figure 5.2 Modeling layers of spatio-temporal data

- Logic layer: The main purpose of this layer is to abstract the spatio-temporal information units from different dimensions, i.e., give their representations in space, time or property dimensions.
- Conceptual layer: The main purpose of this layer is to express the concept of spatio-temporal information units and to establish the relationships between them.
- Physical layer: The main purpose of this layer is to devise a physical model that is applicable in computer hardware. Most of the existing spatio-temporal

systems manage spatial data and common attributes separately, while the temporal information is seen as part of common attributes. Spatial data and common attributes are connected by ID. However, the connection between spatial data and common attributes is usually relatively weak and it is difficult to reflect the complex spatio-temporal relations.

5.2.3 Version Based Data Model

The basic idea of the version based spatio-temporal data model is to record the states of spatial objects in different events in order to track the spatial information changing over time. Based on different versioning technologies, researchers have proposed a number of spatio-temporal data models: sequential snapshots model, base state with amendments model, space-time cubic model, space-time composite model, object-oriented spatio-temporal model.

1. Sequential snapshots model

Armstrong (1988) discussed **sequential snapshots model.** The sequential snapshots model adopts the database versioning technology, using a series of database snapshots to represent the evolving process of spatial object with time changes. Each snapshot records the database state at that time. The sequential snapshots are a series of snapshots. Each of these snapshots corresponds to a layer in state diagram at some time (as shown in Fig. 5.3). Spatio-temporal data is organized according to sequential snapshot to store all data at some time or after an interval as a new layer in the database.

In a snapshots model, each layer stores all the information at a specific time point. It shows a changing spatio-temporal distribution over time. Therefore, there is no definite temporal relation between layers. The time interval between any two layers may be different. Moreover, it is uncertain whether there are any changes in objects in any two layers. Generally, sequential snapshot has two categories: vector snapshots model and grid snapshots model. Temporal Map Sets (TMS) may be regarded as an expansion of the snapshots model. The advantage of sequential snapshots model is that it can be achieved in current spatio-temporal information system directly. The disadvantage is that massive changeless spatio-temporal data in different layers cause data redundancies and this may lead to data inconsistency. In addition, the effect is not direct for snapshot when representing changes. In order to obtain different states at two different times, complete comparison between corresponding snapshots is indispensable.

2. Base state with amendments model

To avoid storing changeless data in consecutive snapshots, Langran and Chrisman (1988) proposed the **base state with amendments model**. Base state with

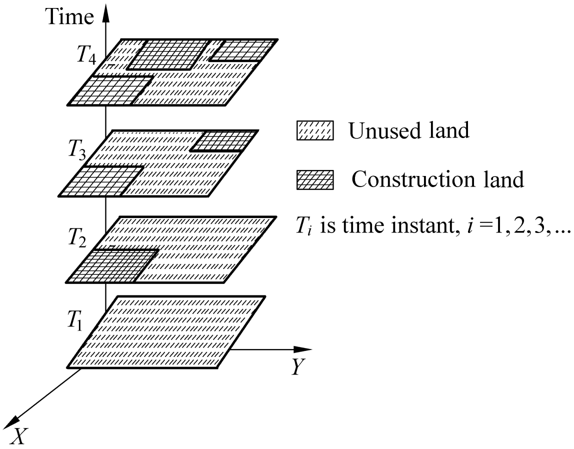
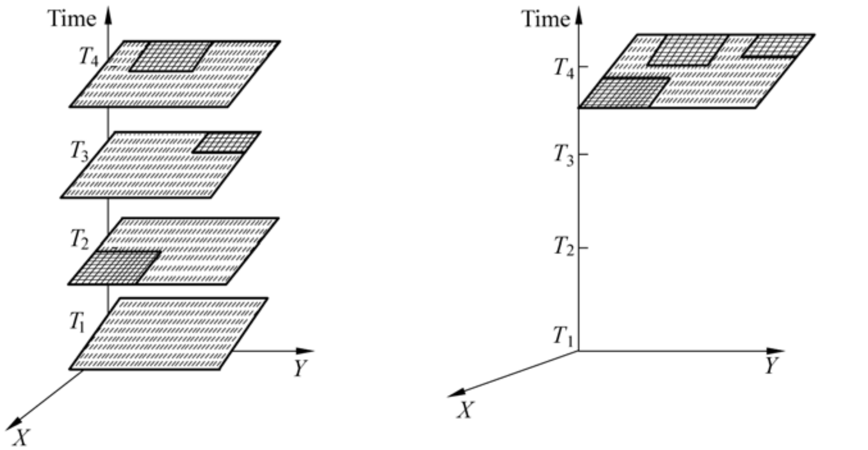


Figure 5.3 Sequential snapshots model

amendments model is based on original database state (snapshots). It just stores a basic data state (base state) at a certain time, and then records all the changes relative to previous state after every predefined time interval. This model updates data only when the data is changed, and it only records the differences between current state and previous state. Compared to the spatio-temporal snapshot model, the base state with amendments model decreases the data redundancy. However, every time, in order to calculate the current state of database, it needs to retrieve the records of base state and all the changes, so the query efficiency is still low. The base state model is shown in Fig. 5.4.



(a) Records of changed area (b) Records superposition of the in time T_1, T_2, T_3, T_4

Figure 5.4 Base state with amendments model

3. Space-time cube model

Hagerstrand first proposed the **space-time cube model** (Hägerstrand 1970). Usually the space-time cube model is composed of two-dimensional space data and one-dimensional time data. It describes the evolving procedure of the two-dimensional objects along the time dimension. The evolutional process of any spatial object is an entity in the space-time cube. Spatio-temporal query is to retrieve the corresponding sub-cube. The characteristic of space-time cube model is that it uses the geometric features of the time-dimension and represents the changes of the spatio-temporal state in a simple and clear way. However, it is difficult to implement the three-dimensional cube. Moreover, with the increase of data, the operations in the cube will become more and more complicated. Space-time cube model is shown in Fig. 5.5.

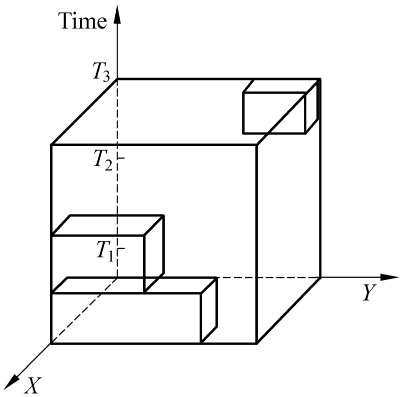


Figure 5.5 Space-time cube model

4. Space-time composite model

Langran proposed the **space-time composite model** (Langran and Chrisman 1988), which combines the characteristics of database versioning and object versioning. It marks up the versioning information on a combinational complex of time and space. The complex of time and space is the combination of a number of objects that are isomorphic in space and consistent in time. Spatio-temporal database, in this case, is a collection of these space-time complexes. Each space-time complex can individually store its states. This model can effectively answer the queries of the spatial and temporal changes over time. However, each space-time complex independently represents its own information. Hence, it is difficult to answer queries for relations between different space-time complexes. In addition, if there is an update operation, it could lead to many space-time complex reconstructions. Space-time composite model is shown in Fig. 5.6.

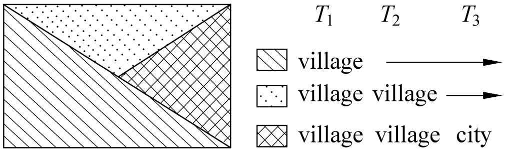


Figure 5.6 Space-time composite model

5. Object-oriented spatio-temporal model

Object-oriented spatio-temporal model (object oriented spatial temporal data model) organizes spatio-temporal data based on the object-oriented idea. Each spatio-temporal data object is packaged independently and each entity has a unique mark. It packages the object's temporal and spatial characteristics, the field properties, related operations and relations with other objects. Object-oriented spatio-temporal model presents the spatio-temporal database model as a collection of spatio-temporal objects and each object contains a number of space-time spatio-temporal atoms. A spatio-temporal atom is a part of some special properties of the spatio-temporal object. These properties remain unchanged for a long time. Although the spatio-temporal atoms themselves do not express any temporal and spatial variation, but projecting the spatio-temporal atom of a spatio-temporal object onto the time dimension or the space dimension, we can get the states of the spatio-temporal object for different times. Therefore, it can express the states and changes of spatio-temporal objects.

Worboys in 1992 presented object-oriented spatio-temporal model based on three-dimensional spatio-temporal characteristics (Worboys et al. 2006). Its basic idea was to use generalization, inheritance, aggregation, composition and an orderly combination of the object-oriented technology to expand data modeling methods based on the entity. The idea was also to use (two-dimensional) spatial objects together with the information of its time axis to compose a complete three-dimensional spatio-temporal object. Since then, people present or improve the object-oriented spatio-temporal data model. These spatio-temporal data models use or partly use object-oriented methods of collection, organization and ordered organization, describe the level hierarchy relationship (such as a large object by a number of small objects) and the mutual reference relationship (such as a piece of land was owned by an individual, a person living in a city). This is done by defining the appropriate object's characteristic structures. The model, however, ignores the data expression patterns of the interaction course between the objects. Therefore, it can be seen that this model was founded partly based on the object-oriented idea.

Object Data Management Group has put forward a series of standards about storing objects, including specification of the object-definition language and query-objects language (Group ODM 2005). In the standards, we use series of changes in the expression of spatio-temporal data to describe the object state. Object

Behavior is a set of predefined operations. It uses the program logic to achieve some necessary and special features to help complete the standardization of data management, as well as to obtain a certain degree of intelligent data management. In fact, with the use of the object-oriented methods and techniques, the standard becomes an expanded program for the features of object-oriented database. It builds an object-oriented data management model. At present, some commercial DBMS offers some basic functions of object-oriented data management. Spatio-temporal object model is shown in Fig. 5.7.

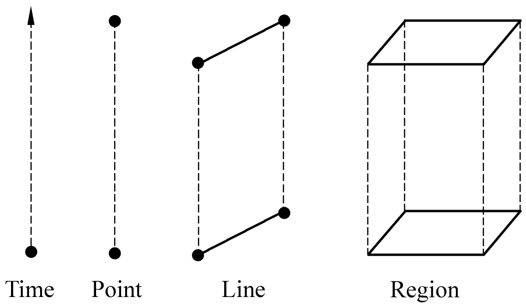


Figure 5.7 Object-oriented spatio-temporal model

Object-oriented spatio-temporal model data structure is simple. If we take full advantage of object-oriented software technology, it is helpful in expansion and the temporal operation of the spatio-temporal data model. However, the spatial and temporal object model is similar to the space-time snapshot model and the space-time composite model. It can only express some discrete changes, but cannot express the continuous changes. At present, the object-oriented space system is rare, because many questions about the model are unresolved.

5.2.4 Event-Based Data Model

Because spatio-temporal data model based on versioning has many disadvantages, researchers began to model the spatio-temporal objects through event or process. The **ESTDM** and **Three Domains Model** are models of this type. Event is a fact in a specific time. The basic idea of models based on event is to track the spatio-temporal changes in different events and every event records a change of a spatio-temporal object. In the event-based model, every event records a change and its state information before and after the change. The characteristic of event-based spatio-temporal data model is to explicitly keep all the spatio-temporal information changes, but it also imposes great overhead on the system. An example of spatio-temporal object is depicted as in Fig. 5.8.

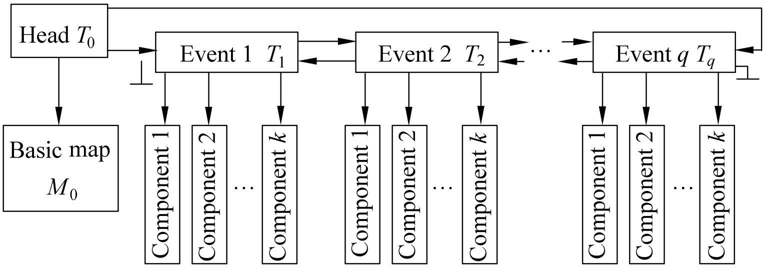


Figure 5.8 Event-based spatio-temporal data model

1. ESTDM and TRIAD

Peuquet and Duan (1995) proposed an event-based spatio-temporal data model. They believe that the spatio-temporal model based on location or trait is not fit for analyzing the spatio-temporal relation. Therefore, they proposed a new data model based on raster named ESTDM (event-based spatio-temporal data) to organize the spatio-temporal information about location changes. ESTDM depicts the spatio-temporal observation of a single event in the spatio-temporal list by organizing the levels based on time stamp. Compared to time list snapshot, the advantages of ESTDM are efficiency in dealing with data and analyzing the spatio-temporal model and relation. The snapshot only stores one state and the related changes from the former state. One head file in ESTDM stores the information of state and pointers in ESTDM point to basic map and start-end event list. The basic map appears on the initial snapshot of the content needed in a spatio-temporal area. An event-based list includes the spatio-temporal information in this spatio-temporal area. Every event has a time stamp and is associated with the time component list. The event component list stores the change of a unit in a predefined location and a given time.

ESTDM effectively supports the temporal query as follows:

- At ESTDM's specified time, change location to the specified value.
- After a specified time interval, change location to a specific value.
- After a specified time interval, change region to the union of some regions with specified values.

If we are using ESTDM in a system based on vector, then it needs to redesign event component. When spatial object or topological relation is changed, the historical information of entity may be sliced as transition information. This mechanism needs event component to track the spatial information and change in entity properties for a predefined entity.

As the extension of ESTDM, the model named TRIAD uses an integrated view (including trait, location and time) to represent spatio-temporal data. It can be implemented through object-oriented method. TRIAD implements a universal and general representation by integrating attribute dimension, space dimension and time dimension. The characteristics of TRIAD are as follows:

- It can support query according to any combination of time, location and properties.
- The specific accessing sequence of view can be determined easily according to query.

The pointer structure of ESTDM is shown in Fig. 5.9.

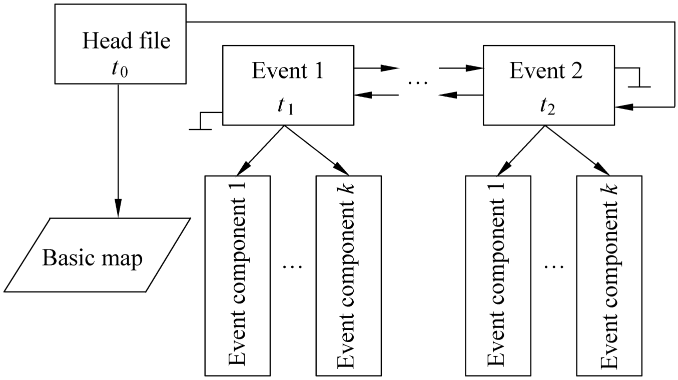


Figure 5.9 ESTDM's pointer structure

2. Three Domains Model

Three Domains Model is proposed to meet the needs of analyzing spatio-temporal information during the process of research on wildfire (Yuan 1994). The information ring of wildfire includes many kinds of data representation, such as snapshot state, fire entity process, snapshot change of entity, representation history of fire block. They can be utilized for research on fire-alarm prediction, fire behavior, and fire impact and fire history of wildfire. Because the application data are separated on concrete semantic, relevant spatio-temporal information needs to support this representation dynamically. Three Domains Model defines semantic and spatio-temporal object in each independent area. In the snapshot model of time list, time is an attribute of location. In space-time composite model and space-time object model, time is a part of space entity. In Three Domains Model, time models as an independent concept. It represents the real world from three angles: making location, entity or time as center. Hence, it can depict various basic changes of spatio-temporal information, such as attribute change, static space distribution, static space change, dynamic space change, transit of process and movement of entity. Three Domains Model is shown in Fig. 5.10.

Three Domains Model uses its dynamic linking relative object to represent spatio-temporal entity. These links between objects can be numerical type or fuzzy member function. Member function is especially very effective in representing dynamic boundary, such as the change of oil distribution, seasonal lake boundary and coastline.

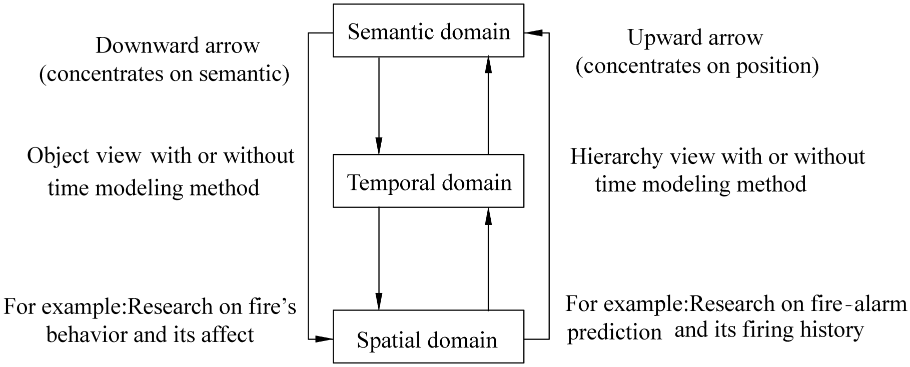


Figure 5.10 Three domains model

5.2.5 Constraint-Based Data Model

As a new data model, **constraint-based data model** extends traditional relational data model through "generalized tuple". A constraint tuple of constraint relation R is defined as a constraint conjunct on a variable set. For example, "$((1<x<3)\wedge(2<y<5))$" can be a constraint tuple defined on variable set $\{x, y\}$ and can correspond to rectangular region of two-dimensional plane. Every constraint tuple can depict a point set, which may be infinite. Therefore, constraint data model can represent multi-dimensional information, such as spatio-temporal data in the form of constraint. Constraint-based spatio-temporal model can adopt relational algebra or relational calculus as its data operation. Compared to other models, constraint-based spatio-temporal data model has better presentation ability. It can not only represent continuous changes, but also depict spatio-temporal semantics based on version.

The performance of spatio-temporal data model is shown in Fig. 5.11.

5.2.6 Moving Objects Data Model

At present, the technique of MOD (Moving Objects Databases) has already become hot spots of spatio-temporal database. Erwig and others believe the essence of spatio-temporal database is to store the moving objects (Erwig et al. 1998). MOD is a database that manages the moving objects and location. MOD can be used in many domains such as civil air regulation, traffic management, military command and location-based information service. *Now*, many researchers have studied the key points of this domain, such as representation, modeling, index and query of moving objects.

1. Continuous and discrete model

Moving object is the geometry object whose figure, location and domain attribute

	Snapshot (Langran and Chrisman 1988)	Base state with amendments (Langran and Chrisman 1992)	Space-time composite (Langran and Chrisman 1988)	Space-time object (Worboys et al. 1992)	Three domains (Yuan 1994)	Event-based (ESTDM)
Data redundancy	big	small	small	small	small	small
Data query	weak	weak	weak	relative strong	strong	weak
Data obtaining	easy	difficult	easy	easy	easy	difficult
Change obtaining	difficult	easy	difficult	easy	easy	easy
Space dependent	yes	yes	yes	yes	no	yes
Attributes change	partly	partly	partly	no	yes	partly
Discrete change	yes	yes	yes	yes	yes	yes
Continuous change	no	no	no	yes	yes	no
Motion support	no	no	no	no	yes	no
Event consideration	no	no	no	no	no	yes
Time participation	edition map layer	edition map layer	edition space object	edition space object	independent time object	edision event
Time dimensionality	VT	VT	VT	BT	BT	VT
Space expression	vector/grid	vector/grid	vector	vector	vector	grid

Figure 5.11 Summarization of spatio-temporal database

is changing along with the time. These geometry objects can be divided into moving points and moving regions. The change of moving objects is either continuous or discrete. Hence, the continuous model and discrete model can be established considering only the change of moving objects.

- Continuous model depicts the moving object as a set of infinite moving points and takes moving points as a continuous curve in a three dimensional space. This model can depict the moving information of moving object precisely. Actually, computer cannot store and operate on infinite points, so this model will have problems in implementation.
- Discrete model depicts the moving object as a set of finite moving points and takes moving points as a continuous broken line in a three dimensional space. Discrete model can depict the moving information of moving object by approximate value. Although the change of actual moving objects' location is a continuous concept, modeling using discrete concept is still needed and is feasible for the limitation of computer system resources.

2. Moving data model

For time and space, discrete model is a high-level criterion of DBMS of data structure. In the process of implementing a spatio-temporal subsystem, discrete model needs to be mapped to actual data structure as the attribute data type of object. Spatio-temporal data are of four types: basic data, spatial data, basic spatio-temporal data and moving spatial object data.

- **Basic Data Type**: Basic data type is the standard data type in regular DBMS, such as *int*, *real*, *boolean* and *string*. It can be supported, making full use of current database platform.

- **Spatial Data Type**: Spatial data type is the data type in spatial database, such as *points*, *lines* and *regions*. Every spatial object has a corresponding *SID*. Spatial object deals in the Realms Criterion. A complex moving spatial object, such as *area*, is constituted of hundreds or even thousands of points. Its data structure should support the basic operations on *point*, *thread* and *area* of moving spatial objects, such as increase and delete. At the same time, it should also support most of the operations in spatial analyze. In spatial database, balanced binary tree can be chosen as the basic data structure of moving spatial object. The time interval of every unit is regularly defined as right open and the time factor is the start time of this interval. Following the regulation of discrete model, unit function x (for continuous change) or unit object value (for discrete change) should be defined in the time interval of this unit to decide the object's value in this unit.
- **Basic spatial-temporal data type**: This type is represented as an ordered tuple of time factor and basic data type, such as *mint* (instant, int), *mreal* (instant, real) and *mstring* (instant, string). In practice, unit function of linear simulation is usually used for *mreal* class.
- **Moving spatial object data type**: This type is represented as an ordered tuple of time factor and the spatial data type. For example, many moving objects can be expressed as *mpoint* (instant, point), *mline* (instant, lines) and *mregions* (instant). In implementation of the algorithm, unit function of linear simulation can be used for *mreal* type. However, for the moving spatial object, such as *mpoint* and *mregion*, the selection of time factor must follow the discretization criterion for the constraint of implementation condition of computer.

3. Framework of abstract data type

Currently, people have done a lot of work on modeling of moving objects. Erwig and Güting (1998) define a frame of abstract data type for moving objects. This frame includes standard data type (*int* and *Boolean*), spatial data type (*point* and *area*) and spatio-temporal type (*time* and *interval*). Through type construction, a new type is created from basic types.

In fact, the modeling method by Erwig defines two models on different levels, abstract and discrete models. Abstract model considers the independence of a system, generality and consistency of operation, closure and consistency of the relation between non-temporal and temporal data structure and operation. Simple query operation can be designed using abstract model. Discrete model is a higher-level standard of data structure. It defines the possibility of implementation of the corresponding data operation in the computer system when the representation is limited. This data type of moving object can be used as the extended type in traditional relation model and could also be integrated into object-oriented or object-relational model. In the meantime, it can support the analyzing operation of spatio-temporal data, based on DBMS, and it can apply the defined spatio-temporal analyzing function in the query.

5.3 Query on Spatio-Temporal Data

Since a long time, the expression and modeling of spatio-temporal data are the most popular problems in the area of spatio-temporal database. In recent years, the research of queries on spatio-temporal data has become a hot spot. The types of current spatio-temporal queries are as follows:

- **Window query**: Given query area *QR* and time interval *QT*, to find all the objects that have intersection with *QT* and *QE*.
- **Recent adjacent query**: Given query point *qP* and time interval *qT*, to find all the adjacent objects with the smallest time distance to *QT*.
- **Spatio-temporal join query**: Given two sets S_1, S_2 of moving objects, a future time stamp t_q and a distance threshold *d*, a spatio-temporal join retrieves all pairs of objects that are within distance *d* at t_q.
- ***k*-CPQ (*k* Closest Pair Query)**: It discovers the *k* pairs of spatial objects formed from two data sets that have the *k* smallest distances between them, where $k \geqslant 1$.
- **Navigational WQ**: Based on historical databases, Pfoser et al. (2000) proposed navigational WQ, that is, given two query regions QR_1, QR_2, and two time stamps QT_1 and QT_2, retrieve all the objects that intersect with region QR_1 at time QT_1 and intersect with region QR_2 at time QT_2.
- **TP (time-parameterized) query**: For predictive spatio-temporal database, Tao and Papadias (2002) point out that the results of traditional query (such as WQ, *k*NN, WDJ, *k*CP) may change because of the movement of projects. Therefore, results of traditional query are not enough for spatio-temporal database. According to this situation, TP query is proposed. This kind of query can be applied to any traditional query method, and the query result could return the result *R* by normal traditional query, the expiry time *TC* of the result *R* and the change result *C* after *T*. Then Tao and Papadias extended the TP concept to continuous query, intending to track the change of query result until some conditions are satisfied.
- **LB (location-based) query**: Zhang et al. (2004) puts LB query forward. This query could be applied to WQ and *k*NN queries to get query results and valid range of query.

In addition to the above normal spatio-temporal queries, some researchers of spatio-temporal database have done some researches on the model of spatio-temporal data query cost. Choi proposes a model of spatio-temporal cost on TPR-tree. Tao and Papadias (2001) put forward a cost model according to overlapping B-tree and multi-version B-tree. This model can evaluate the query tree, accessing point and selection size of query. Lately, Zhang et al. (2005) proposed a cost model of NN query.

5.3.1 Spatio-Temporal Data Query

There are several kinds of spatio-temporal data queries, such as window query, nearest neighbor query and TP and LB queries.

1. Window query

Window query can be represented as $\{O \in DB \mid O(t) \in queryR \wedge t \in [t_1, t_2]\}$. $O(t)$ represents the spatial location of object O at time t. *queryR* represents a rectangular window. The semantic of window query is: find all the moving objects whose locations are in *queryR* during t_1 to t_2. If the window query does not indicate the query time directly, it means that the query time is current time.

2. Nearest neighbor query

The mathematical definition of nearest neighbor query is: for any given object q and object set $P = \{p_1, p_2, \cdots, p_m\} (m \geqslant 1)$, find the p_i, which makes $|q, p_i|$ the smallest. $|q, p_i|$ represents the spatial distance between object q and object p_i. In traditional NN query, all the objects are stationary. Spatio-temporal database uses the concept of nearest neighbor query in spatial database and extends the concept that all objects can be either stationary or moving. Nearest neighbor query on moving objects is a critical technique in spatio-temporal database. It is widely used in many domains, such as intelligent navigation, modern communication, traffic control and weather forecast. It is the key point of spatio-temporal database. The nearest neighbor query of moving objects can be abstracted as: Given query object q and its movement status (velocity and direction), find a series of nearest neighbors of q from start time s to end time e. If the shape of the object is not considered, the object can be treated as a point. Then the nearest neighbor query is the nearest neighbor query of moving points.

3. Approximate query

Approximate query can be represented as $\{O \in DB \mid dist(Q(t), O(t)) \leqslant \varepsilon\}$, where $Q(t)$ is the object to be queried, *dist* the "distance" between object $Q(t)$ and $O(t)$, ε the given range of distance, and $t \in [t_1, t_2]$. The semantic of approximate query is: retrieve all the objects $O(t)$ whose distance from $Q(t)$ is in the given range ε during t_1 to t_2.

5.3.2 Moving Data Query

Moving data query can be divided into several basic types, and those are queries on historical location, current location and future location. The future location query is to predict the future location of moving objects based on the current location, its velocity and direction. Historical and current location query can be further divided into two subcategories: coordinate-based query and trajectory-based query. Coordinate-based query includes time slice query and time interval query. Trajectory-based query includes topological query, navigation query, and combination query.

1. Coordinate-based query

- **Time slice query**: Time slice query can be represented as $Q = (R, t)$, where R is a hypercube at time t. For example, finding the traffic cars in area R at time t is a simple and regular spatio-temporal query of traffic monitoring. Time slice query could be seen as the product of the select operation in spatial database and select operation in temporal database.
- **Time interval query**: Time interval query can be represented as $Q = (R, t_1, t_2)$, where R is a hypercube of k-dimensional space, and Q a $(k+1)$-dimensional hypercube constructed by R and time interval (t_1, t_2). In the application of traffic monitoring, this kind of query returns all the vehicles passing through the area R during t_1 to t_2.

2. Trajectory-based query

- **Topology query:** Topology query retrieves part of or the whole trajectory of moving objects. The basic predicates of topology query are “center”, “cross”, “pass by”, and “leave”. In order to judge whether a moving object satisfies those predicates, multiple line segments need to be examined.
- **Navigation query:** The information of moving objects that is not directly kept in database but can be inferred from its trajectory is called navigation information. For example, the average and maximum speed of the object can be obtained by the distance it made in some time interval. The moving vector can be calculated by two vectors at different locations. The queries on speed and moving vector are very popular in applications. For example, “What is the current speed of this car? What is its maximum speed?” The first question is related to current time and the second question is related to the data in some time interval.
- **Combination query:** The trajectory-based query turns out to be more complicated in practical applications. It could be a combination of several query types. For example, if every car has an exclusive *ID* number, there may be queries like: “Where will the vehicles on the 5th Zhongshan Road from 9 to 11 o’clock this morning be in the following one hour?” In this case, it needs to find all the vehicles that are on the 5th Zhongshan Road from 9 to 11 o’clock and then it needs to predict where those vehicles would be in the following one hour.

5.3.3 Spatio-Temporal Database Language

Spatio-temporal database language and spatio-temporal data model are closely related. Every spatio-temporal data model needs its query language. There are mainly two directions in current spatio-temporal database language: spatio-temporal database based on SQL and spatio-temporal database based on OQL. As the popular language in relational database, SQL has been supported by most of commercial databases such as Oracle and SQLServer. OQL is an object-oriented database query language proposed by ODMG and has become the standard language of object-oriented database. It has been supported by OODBMS such as O2 and Versant. STSQL is a query language of spatio-temporal database based on SQL. Because SQL3 has scalability in its abstract data types, the spatio-temporal data model based on data type can be combined with SQL3.

The characteristic of STSQL is that it is compatible with SQL. It does not extend clauses of SQL language, but just extends data types and operations. Therefore, STSQL is applicable in many applications.

STQL is another spatio-temporal database query language based on SQL. The data model of STQL is based on relational model and it represents the spatio-temporal change in the form of tuple versioning. Since STQL extends the clauses of SQL, it is incompatible with SQL3. The query clause of STQL extends the traditional SQL in two ways. The first one is using WHEN clause to represent the temporal query condition and the second one is to express the spatial query condition by adopting spatial operator and spatial predicate. Therefore, STQL uses the combination of two kinds of conditions (space and time) to represent a spatio-temporal query. Compared to STSQL, STQL introduces transaction time-dimension. It can deal with bitemporal data. However, it splits the spatial and temporal query criteria artificially and its spatio-temporal data model is based on versioning. It cannot deal with continuous spatio-temporal query.

For the spatio-temporal data model and spatio-temporal database language based on ODMG, object-oriented data model is more expressive in semantics than relation model (such as polymorphism, inherit and aggregation). Therefore, it has some advantages in the representation of spatio-temporal data.

5.4 Structure of Spatio-Temporal Database System

Because of the special requirements in spatio-temporal applications, it is critical to design an effective system structure that supports STDBMS (spatio-temporal database management systems). The system structure of spatio-temporal database system utilizes the achievements of the former spatial database and temporal database. There are three basic types of system structures: complete type, layered type, and extended type.

5.4.1 Structure of Complete Type

The complete type implements an STDBMS from bottom to top. This type needs to implement all the modules of DBMS including query compilation and execution, transaction management, storage management, and even the drivers for the spatio-temporal database. Its workload is too heavy for many spatio-temporal applications.

5.4.2 Structure of Layered Type

Implementation of layered structure (such as in Fig. 5.12) adds a spatio-temporal layer on the traditional RDBMS. Dealing with spatio-temporal data in spatio-temporal layer, it does not need to change the core of DBMS. Spatio-temporal layer does all the work related with spatial temporal information, such as translation of spatio-temporal database language and SQL, spatio-temporal query optimization. However, the SQL translated from spatio-temporal query is very complicated. It is very difficult for the RDBMS to do the query optimization. Because all the queries need to be transformed to standard SQL by the spatio-temporal layer first, this layer probably will become the bottleneck of applications.

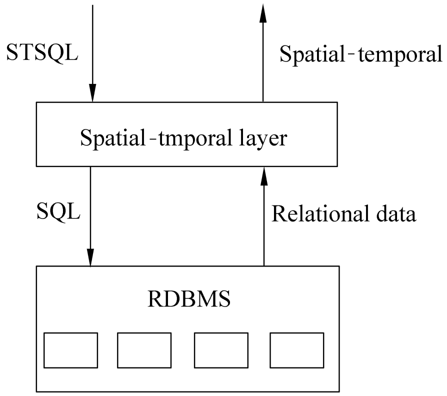


Figure 5.12 Layered architecture of STDBMS

5.4.3 Structure of Extended Type

The extended structure is to execute a spatio-temporal extension on the ORDBMS. Figure 5.13 shows such a case. Since ORDBMS provides the function of UDT (user-defined data type) and UDR (user-defined routine), it can be used to extend new spatio-temporal types and operations. This structure is the most popular one right now. The extended system structure makes the implementation and application of spatio-temporal DBMS possible. Its main problem is: though original DBMS can accelerate the query by extended spatio-temporal index, the strategies of query optimization is still the old ones of relation database, and it is not appropriate for spatio-temporal query.

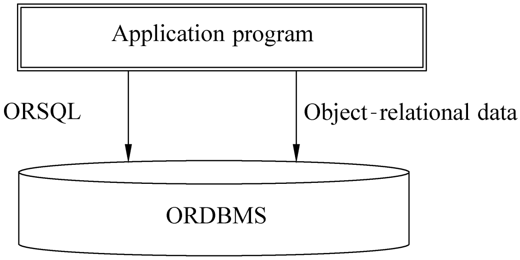


Figure 5.13 Extending architecture of STDBMS

Reference

[1] Allen J (**2005**) Towards a general theory of action and time. The Language of Time: A Reader: 251

[2] Armstrong M (**1988**) Temporality in spatial databases. In: Proceedings of GIS/LIS'88, pp 880 – 889

[3] Cho H J, Chung C W (**2005**). An efficient and scalable approach to CNN queries in a road network. In: Proceedings of the 31st International Conference on Very Large Data Bases (VLDB), Trondheim, Norway, 865 – 876

[4] Egenhofer M (**1991**) Reasoning about binary topological relations. Springer-Verlag London, UK

[5] Erwig M, Güting RH, et al. (**1998**) Abstract and discrete modeling of spatio-temporal data types. In: Proceedings of the 6th ACM International Symposium on Advances in Geographic Information Systems. Washington, D.C., USA

[6] Group ODM (**2005**) The standard for storing objects. Retrieved 1-18, 2005. http://www.odmg.org

[7] Hägerstrand T (**1970**) What about people in regional science? Regional Science **24**(1): 6 – 21

[8] Langran G, Chrisman N (**1988**) A framework for temporal geographic information. Cartographica: The International Journal for Geographic Information and Geovisualization **25**(3): 1 – 14

[9] Peuquet D, Duan N (**1995**) An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. International Journal of Geographical Information Science **9**(1): 7 – 24

[10] Pfoser D, Jensen CS, Theodoridis Y (**2000**) Novel approaches in query processing for moving object trajectories. In: Proceeding of the International Conference on Very Large Databases (VLDB), 395 – 406

[11] Tao Y, PaPadias D (**2001**) MV3R-tree: a spatio-temporal access method of timestamp and interval queries. In: Proceedings of the 27th International Conference on Very Large Databases (VLDB), 431 – 440

[12] Tao Y, Papadias D (**2002**) Time parameterized queries in spatio-temporal databases. In: Proceedings of the ACM International Conference on Management of Data (SIGMOD), 334 – 345

[13] Worboys M, Hearnshaw H, et al. (**2006**) Object-oriented data modeling for spatial databases. Classics from IJGIS: Twenty Years of the International Journal of Geographical Information Science and Systems **4**(4): 119

[14] Yuan M (**1994**) Representation of wildfire in geographic information systems. State University of New York at Buffalo

[15] Zhang J, Mamoulis N, PapadiasD, Tao Y (**2004**) All-nearest-neighbors queries in spatial databases. In: Proceedings of the 15th IEEE International Conference on Scientific and Statistical Database Management (SSDBM), 297 – 306

[16] Zhang J, Papadias D, Mouratidis K, Zhu M (**2005**) Query proceeding in spatial databases containing obstacles. International Journal of Geogrophic Information Science(IJGIS), **19**(10): 1091 – 1111