

MODELING SPATIAL-TEMPORAL DATA IN VERSION-DIFFERENCE MODEL

Wang Huibing, Tang Xinming, Lei Bing, Yang Ping, Chu Haifeng

Chinese Academy of Surveying and Mapping, Beijing 100039, China – (wanghb,tang)@cas.m.ac.cn

KEY WORDS: spatial-temporal, version-difference, state/event, dynamic correlation

ABSTRACT:

In this paper, a *version-difference* spatial-temporal data model is proposed. The main research interest centers on spatial-temporal data storage and manipulation. This model uses the terms *version* and *difference* to represent spatial-temporal data at an instance and changes over time. In this model, we introduce a default version which focuses on data currency and always represent the current data state, and then data at any instance can be aggregated or reconstructed by version and difference. This model can reduce data reduction and accelerate data access.

1. INTRODUCTION

The spatial-temporal database was developed with the requirement of historical information to monitor and analyze changes over time. Spatial-temporal applications are increasingly the focus of research activities in the geospatial and database communities.

Generally, the capabilities of any information system largely rely on the design of its data models which present the conceptual core of an information system. In its most abstract sense a data model provides the logical framework in which the real world may be describe for computing. Consequently, the spatial-temporal data model is the key to handling spatial and temporal data. Numerous spatial-temporal data models have been constructed, dealing with storage and management of fundamental data. The main models documented are as follows:

- (1) Snapshot model: This model simply gives a new map for each time interval. Each layer is composed of temporally homogeneous unit. When an event occurs, new layer is constructed and occurrence time is stamped to the layer (all of the information, changed or not changed, are stored in the layer)^[9].The snapshot approach usually results in inconsistency and significant data redundancy.
- (2) Space time composite: In STC model, the real world is a collection of spatially homogeneous units in a 2D space that changes over time from one unit to another. Each STC has its unique period of change and can be obtained from temporal overlay of snapshot layers.
- (3) STC can model favorable properties of an object such as situation, but it can not represent changes of attributes such as movement over space. Moreover, the STC approach requires reconstruction of thematic and temporal attribute tables whenever operations involve any changes in spatial objects (shape, size, or configuration)^[8].
- (4) Spatio-temporal object modeling: In this model, the real world is considered as a set of spatial-temporal atoms that constructed from integration of a temporal dimension orthogonal to 2D planimetric space. Each of these spatial-temporal atoms is largest homogeneous unit that can store specific properties in space and time. Thus this model can store changes in both temporal and spatial dimensions^[9].

Whereas these spatial-temporal atoms have a discrete structure, modeling of gradual changes in space or time with this model is impossible. This model as similar to snapshot and STC models can only represent sudden changes.

Other data models, such as event-based model, object-oriented model, etc, were also well documented by many researchers. However, most of existing models cover only partly the requirements (they address either spatial or temporal modeling), and most are at the logical level, hence not well suited for database design^[1].

In this paper, we propose a different data model called version-difference model. In this model, we use the term difference in our approach to reflect the emphasis put on change as the explicit information that is both captured and stored in our spatial-temporal model. Data currency is critical to the system, we proceed by storing the current state of an object (called *default version*) and all historic changes (called *version difference*). Then, an object representation at any instance can be obtained through an aggregation or reconstruction operator of base state and all subsequent changes (difference).

This research was under the support of national fundamental surveying and mapping project—National Fundamental Geospatial Information Dynamic Database (NFGIDD), which aims to construct a general spatial-temporal data model of national fundamental geospatial information to provide a general framework for monitor, analyze and forecast spatial-temporal objects' transition over time.

2. THE VERSION-DIFFERENCE MODEL

Building an appropriate spatial-temporal data model concerning about both spatial and temporal aspects, which can be used to organize and manipulate spatial-temporal feature more effectively and enrich spatial-temporal feature's semantics(attribute, spatial and temporal), is critical to spatial dynamic system. Therefore, we can use this model to retrospect or track historic data, monitor changes and forecast future transition.

For the version-difference model, we need not to store all the information of every state in the area interested, but only the data at base state and changes between the state considered and the base state to minimize storage. Unchanged features are not duplicated. Each change is labeled with a unique state ID (State

ID is incremented). Dataset may look different at different states.

This chapter begins with a brief introduction of some basic conceptions in this model, followed by detailed storage schema and a hierarchical temporal index schema.

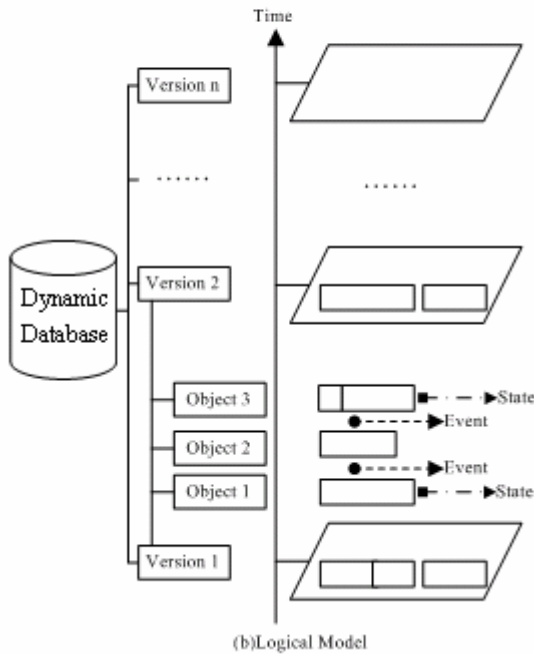
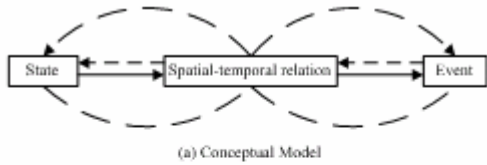


Figure 1. "Version-difference" spatial-temporal data model

2.1 Representation of time attributes

There are mainly two different notions of time which are relevant for temporal databases. One is called *valid time (begin time and end time)*, the other is *transaction time*. The former denotes the time period during which an object is true with respect to the real world. The later is the time period during which an object is stored in the database. Note that these two time periods do not have to be the same for a single object.

In this model, the database is bi-temporal, object's valid time and transaction time are recorded both supporting for temporal queries.

2.2 Event and State

Event and state is one pair of the most important basic conceptions in temporal database. One object may have different states during its life-span; event is the process from one state to another. In general, we record event by time instance and state by time span in database. State is an existent form of geographical entity in a given time range, and it is a comparatively stable process. The state of a spatial-temporal object can be divided into attribute state and spatial state, and then spatial state can be divided into spatial topological state and spatial geometry state (Figure 2).

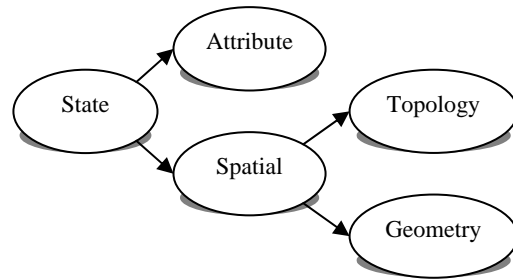


Figure 2. Temporal state

2.3 Dynamic correlation

The disappearance of one object is usually coincident with the involvement of new object's appearance, and vice versa. We record such information in database as dynamic correlation.

Parent object's death and child object's birth was triggered by the same event, in other words, parent(s) object and child object(s) have inheritable relationship in their life span. The relationship of object's variation inheritance is very important in the process of state evolution and temporal data updates, and it is useful to track an object's various states over time, so we need build spatial-temporal dynamic correlation.

Figure 3 shows the state/event concept by an example. For object A, it splits into two objects B and C at time t_1 (an event occurred), and its state began at t_0 and ended at t_1 . As for object B, its parent object is A, children objects are D and E, we recorded this information in database as correlation link. Other object in the figure follows the same rule.

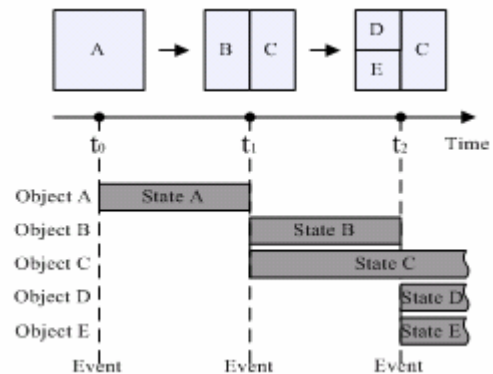


Figure 3. An example of spatial changes over time

2.4 Version and difference

Versioning, as the name implies, explicitly records versions of individual features and objects as they are alternated, added, and deleted through various states. A version explicitly records each state of a feature or object as a row in a table along with important transaction information.

In this model, version is classified into three types:

- ① On-line or On-the-fly Version: It means the version reconstructed by version and difference from database at a given point of time zone. This kind of version has not redundancy basically.
- ② Stand-alone or Snapshot Version: Users can create a version which holds all data belong to a specific state at a given point of time. This version is what so called stand-alone or snapshot version, and it has redundancy.

The snapshot version has some properties, such as: name, owner, description, create time, last modified time, parent version, privilege, etc. The privilege can be divided to three classes as following:

Public: Read/write for all users.

Protected: Read/write for owner, read-only for others.

Private: Read/write for owner.

③ Currency Version or Default Version: It means the version comprised of whole current state dataset. Obviously, it's an example of ②.

Version management should follow rules as follows:

① Only the owner can rename, create, modify or delete a version;

② Create or modify a version can not take effect at once until commit operation done;

③ Every dataset has a default version;

④ The version's owner can restrict others' privilege for his/her version.

⑤ The database should has version verification to identify features' updating.

As for the difference, we must take into consideration several issues as following: (1) Difference comes from comparing information at two temporal states concerned (say A and B), so it must include information of two parts, one belongs to state A but not B , the other is with state B but not state A . In order to use difference to quickly and continuously fetch data at an instance belong to a specific state, we must label the two cases in given difference table identifiably. (2) The object relationship between target state and neighbor states, for example, $0 \rightarrow 1$ (target object emerges.); $1 \rightarrow 0$ (object disappears.); $1 \rightarrow N$ (the number of objects increases from one to more than one.); $N \rightarrow 1$ (the number of objects reduces to one.); spatial geometry remains, but properties change. (3) Similar to most spatial database models, the difference table also consists of spatial information, property information and relation information. So as far as each object is concerned, the relationship among the three kinds of information should be recorded in the difference table.

2.5 Storage schema

Adopt "version-difference" model to support object-oriented design, reduce data redundancy and save space for storage. According to the version-difference model, we store spatial-temporal data into four logical databases to hold different information (Figure 4):

① *Currency database*: It stores the whole spatial-temporal dataset of current state. Because currency dataset will be accessed frequently, we take it as base state in order to improve database performance.

② *Process database*: Transitions are processes representing evolution and therefore subject to constraints, which are preconditions to limit, avoid or force a change. In other words, some transition may be uncorrectable and cannot be happened in reality, we name it "fake transition". When the "fake transition" occurred, the database should discard the alteration and rollback. Obviously, the process database stores the information of dynamic correlation so as to query and trace history.

③ *History database*: The latest state of an object will be stored in the currency database once it was changed at an event,

and also the process of variation will be stored in the database. The event should be arranged by time-order, and the difference between the current state and old state will be stored when all the condition or constraint are all reasonable. Given a specific time, it is easy to obtain a previous state and rollback the current spatial-temporal relations and implement corresponding calculations.

④ *Version database*: Version is a snapshot of dataset at a given point of time. Users can browse the spatial-temporal data of any time instance and make it a version in the version database. After this, the version dataset can be browsed and manipulated quickly, and it can be regard as a base state of other version dataset, whose time stamp is earlier to this one, for hierarchy index schema (Explained in next section).

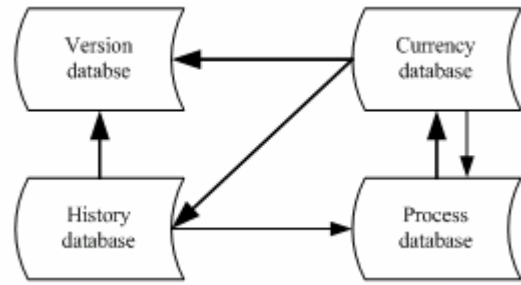


Figure 4. Relationship between the four logical databases

2.6 A hierarchical spatial-temporal index schema

As for the indexing methods of spatial-temporal data models, the large amount of research work that has been done mainly involves two issues: how to build the index and how to design the difference file to precisely measure the changes between two states.

If *direct index* (Figure 5, a) is used to update a state's objects, all the difference files need to be modified; if *indirect index* (Figure 5, b) is implemented, a lot of records in the database have to be searched. However, when the states in the temporal database changed frequently in a quite long period of time, the two methods will require considerable amount of work either on the calculation of data or on the searching and indexing of large databases.

To improve, one way is to establish a hierarchical index schema (Figure 5, c), that is, taking the data at time T_i ($i=0,1,2,\dots$) as base state for the data at time t_j ($j>i$) and then comparing different states indirectly.

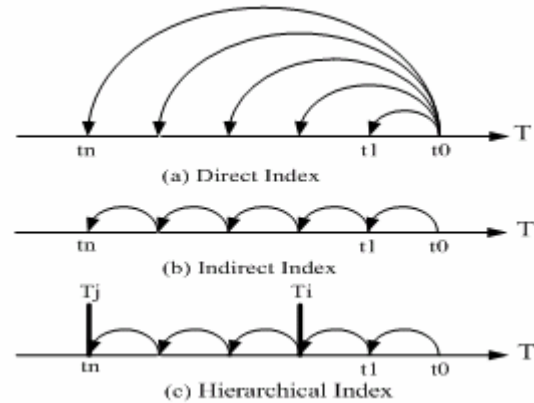


Figure 5. Spatial-temporal index schema

3. IMPLEMENTATION OF NFGIDD PROJECT

We have developed NFGIDD project's application system based on this model. This project aims to construct a general spatial-temporal data model of national fundamental geospatial information to provide a general framework for monitor, analyze and forecast spatial-temporal objects' transition over time.

This project was mainly divided into three sub-systems: Dynamic data process, Spatial-temporal data engine(middle-ware) and Dynamic data management. The screenshots(Figure 6) shows the transition of residential area of Beijing from 1970 to 2000.

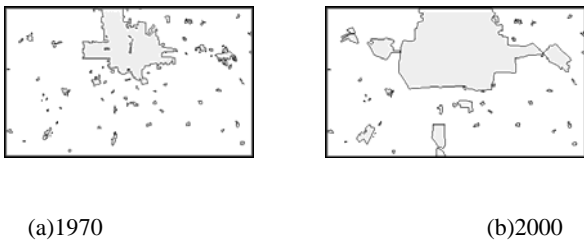


Figure 6. Transition of Residential Area of Beijing

4. CONCLUSION

"Version-difference" spatial-temporal data model presented in this paper organizes and stores the spatial-temporal data through version base state and corresponding difference, and build the dynamic correlation and hierarchical index, which can effectively reduce data redundancy and accelerate data assess and history-trace. An application project's implementation based on this model was also introduced. This model's usability was verified through an application case study.

Further studies will focus on temporal query language and spatial-temporal topology.

ACKNOWLEDGMENT

The work described in this paper was substantially supported by Funded by Chinese National Fundamental Surveying and Mapping Project (No.1460130524207) and Open Research Subject of Key Laboratory of Geo-informatics of State Bureau of Surveying and Mapping.

REFERENCES

Parent, C., S. Spaccapietra., Spatial-Temporal Conceptual Models: Data Structures + Space + Time, Conference on information and knowledge management proceedings of the 7th international symposium on advances in geographic information systems, ACM Press, 1999, pp. 26-33.

Burrough, P. A., McDonnel, R. A., Principles of Geographical Information Systems, Oxford University Press, New york, 1998.

Tang, X.M., 1998, Modeling of Spatiotemporal Data and Identification of Relationships and Change Processes For a

Cadastral Information System in China, ITC MSC thesis, Enschede, The Netherlands, 1998.

Worboys, M.F., 1994c. A Unified Model for Spatial and Temporal Information, The Computer Journal, Vol. 37, No. 1, pp. 26-34.

Allen, J.F., 1983. Maintaining Knowledge about Temporal Intervals, Communications of the ACM 26(11): 832-843.

Langran G., 1989. A Review of Temporal Database Research and it Use in GIS Applications, International Journal of Geographic Information Systems, Vol. 3, No. 3, pp. 215-232.

Langran G. and Chrisman, N.R., 1988, A Framework for Temporal Geographic Information, Cartographica, 25(3), pp. 11-14.

Yuan, M., Temporal GIS and Spatio-Temporal Modeling, http://ncgica.ucsb.edu/conf/SANTA_FE_CD_ROM/sf_papers/yuan_may/may.html.

Nadi, S., Delavar, M.R., Spatio-Temporal Modeling of Dynamic Phenomena in GIS, <http://www.scangis.org/scangis2003/papers/11.pdf>.