Spatio-temporal Data Model for Road Traffic Information System

Maw Maw Oo[1] and Toncan Duong[2]
1. CSIRO Land and Water; 2. School of Computer Science and Engineering, UNSW

Abstract: Road Traffic Information may change in both temporal and spatial dimensions. Its textural (non-spatial), morphological (geometry) and topological (connectivity and contiguity) attributes need to be considered together with the time attribute. The integration of time and space leads to many interesting research problems in particular the updating of spatio-temporal object which is complicated due to difficulties in tracking, error correction and retrieval by version. In this paper, we will discuss our experience in the building of a prototype system for Road Traffic Information System with respect to its conceptual model and query handling. We found that there are distinctive differences in spatio-temporal behavior for different kinds of objects based on their settings of topology and geometry. Object oriented approach is used to model this hierarchical arrangement. The notion of version has been used extensively in the other database applications like CAD and CAM; here we extend the concept to include the interaction between time and space. The choice of version management type and data storage structure should reflect the demand on data retrieval (i.e. - whether old version retrieval or current version retrieval)and the update operation of the system. The prototyping confirms our belief that object oriented approach is appropriate for spatio-temporal information system.

1. INTRODUCTION

Spatio-temporal system is required for many applications. The system such as Road Traffic Information System is an example of it. This system requires the management of spatial information together with time.

There are unique issues related to the spatio-temporal systems including data model and query processing. Various researchers have reported on their works relating to - spatial database issues (Brown, 1996; Gunther, 1989), temporal database issues (Snodgrass, 1990) and spatio-temporal issues (Armstrong, 1988; Langran and Chrisman, 1988; Worboys, 1992; Kemp and Kowalczyk, 1993).

In this paper, we discuss the prototyping of Spatio-Temporal Data Model for Road Traffic Information System which emphasize Version management(tracking) and Query Handling.

2. SPATIO-TEMPORAL DATA MODEL

Spatio-temporal data model can be generally defined as abstraction of a real life system with respect to space and time. In the following, we present a Spatio-Temporal model for the Road Traffic Information System. There, we will define its conceptual definitions, version management and updating, then the requirements for spatial and temporal resolutions as well as data storage versus processing time are discussed. This follows by a general discussion on Road Traffic Information System schema, and to access the model, query handling is discussed.

2.1 Conceptual Definition

2.1.1 Object - The Basic Element

In our proposal, The basic element is object.

- Object is an entity which possesses spatial and/or non spatial attributes.
- **Spatial object** is an object which has both spatial and non spatial attributes.
- **Non spatial object** is an object which has non spatial attributes only.
- Object has its own identification to distinguish from the others.
- Every object has to have a parent object except new independent objects.
- **Independent object** is an object which comes into the system as an orphan.

- **Dependent object** is an object derived from the other object(which will become parent object).

Spatial attributes can be grouped into geometrical attribute (location) and topological attribute (connectivity and continuity). It is necessary to understand the nature of spatial attributes changes along with time as some changes may lead to the creation of new version while the other changes may lead to the creation of new object. For instance, changes in topological attribute (connectivity and contiguity) of a spatial object lead to the creation of new object. Changing the direction attribute of road segment object will create a new version of the existing segment object while connecting new segment object to an existing segment object will create new objects. (see Appendix A)

2.1.2 Object, Version and Updating

Along with time, object may have more than one version in its life span. Thus, management of object and version is crucial for the model design in order to keep track the information required together with time.
Two approaches can be employed in object and version management. One approach uses separate identification for each version of the same object. The other approach uses single object identifier for all versions. One version can be distinguished from the other by their difference in time stamp. The comparison on efficiency of these two approaches is currently not yet resolved (Wachowicz and Healey, 1993). We adopt the latter approach and define the notion of version as follows:

- Every Object must have an initial version.
- Different versions of an object have separate time stamps but the same object identifier.
- Changing the identifier will lead to the creation of a new object rather than a new version.

Thus each version can be considered as a unique image of an object at a particular time. Version inherits basic attributes from the object and these basic attributes can not be changed. The object identifier concept is essential for the tracking of versioning as well as object which will be used in update operations.

In **Update** operation, object and version management is important. We categorize the update operation as follows:

- Inserting new object (i.e. - an independent object creation and it may trigger the birth of other new dependent object from the existing object)
- Inserting new version of an existing object (i.e. - geometrical and non spatial attributes changes)
- Error correction (i.e. - correction may be on the hierarchical structure of versions or single version)

Regarding '**inserting new object**', it is required to check the type of object before adding it into the system (i.e. - new region or zone, new road segment, accident and etc.) This information is used to find out the associated spatial data type (i.e. - polygon, arc(segment), point). If the spatial data type is "point", there will be no creation of new dependent object. If not "point" data type (i.e. - arc(segment), polygon), check whether it would create new object or not. If it would create the new object, all the affected objects have to be updated accordingly. If it would not create the new object, commit the inserting new object.
In **inserting new version,** changes are based on the object's geometrical and other non spatial attributes. It would involve the identifier checking and based on this checking the system would give error response to the user or proceed the insertion. As it is a creation of new version based on the current version, the object identifiers and others attributes can be inherited from the current version.
Error correction can be considered as a part of update operations. In error correction, the operation can be on current version of an object or on versions other than the current one.
If error is from the current version, we need to update the "valid end time" and "transaction end time" to current time and then add a new version containing corrected data into the system with new transaction time. But it needs to make sure that the transaction time of the error version is earlier than that with the corrected data and these two transaction times can not be same. If error is not from the current version, the following algorithm is used:

- Finding out the affected versions of the same object or dependent objects including their versions.
- Finding out whether the error correction will cause the creation of new object or creation of

new version(i.e. - whether the correction is based on spatial attributes).

- IF error correction is based on spatial attributes, then
    - check whether it would change topological attributes or not
    - IF it would change the topological attributes THEN
        - find out the affect versions and objects and perform correction for them
    - ELSE enter the corrected information with new "start and end transaction time" for all affected versions.
    - END IF
- ELSE enter the corrected information with new "start and end transaction time" for all affected versions
- END IF

The old transaction time would be earlier than that of the updated one. In this case, it is easy to check which one is corrected information and which one is not by checking valid time and transaction time.

2.1.3 Semantics of Time

Researchers have represented time as a parameter, property or dimension (Effenberg, 1992). Among them, time as a property has been widely adopted. Here, we use the same approach. Time can be captured at when an event occurs or at when an event is recorded. Snodgrass and Ahn (1985) call them valid time and transactional time. Both the valid time and transaction time have a duration denoted by the Start time and End time.

In this work, we defined another type of time called event time. In fact it is a special case of valid time in which valid start time and valid end time are same. This type of time is used in Road Traffic Information System car accident information. The accident starts and end less than the temporal resolution used in the Road Traffic Information System (i.e. - in our case - day).

The other relevant issue is - in which level time-stamping should be attached. Time stamping can be attached at relation level, tuple level or attribute level (Langran, 1992).

Here we use tuple level versioning of Snodgrass and Ahn in Road Traffic Information System as the frequency of change is small but the frequency of query may be daily or more.

2.2 Spatial and Temporal Resolutions

When modeling the spatio-temporal dimension, the spatial and temporal resolutions requirement for the system should be taken into account. If high spatial resolution is required vector data structure should be employed otherwise raster or quatree data structures should be used. These data structures have their own pros and cons. In this work, vector data structure is used because of its ability to store a complex data in a minimum storage space and its ability to represent spatial feature more accurately than the raster's. The vertices are stored in encoded form in order to save the storage space. However, when spatial query is performed, decoding is required.

regarding the temporal resolution, some systems may need smaller time period between significant changes while the others may need larger time period between significant changes. This time period between significant changes is called time granularity. In our work, "day" is chosen as the time granularity.

2.3 Data Storage Design and Processing Time

In general, spatio-temporal data need more storage space than non spatio-temporal data. However, the data volume can be greatly reduce if only changes to existing data are stored. The unchanged data can be obtained from the previous version. The query response time is longer due to the linked structure.

The trade-off between data storage and processing time should be considered according to the requirements of specific system. The system like Road Traffic System may have changes yearly basis. For example, construction of new link road for highway happens once in a decade. On the other hand, car accident may happen in daily basis. Based on these factors, we choose to store all data at one place rather than separately.

2.4 Road Traffic Information System Data Schema

In this model, information such as car accident, service station data, road data, divisional data and etc. was stored in tables - Division, Road, RoadSegment, Accident and Feature. It is required that the model should be able to manage the spatial information with time while providing the data accessibility and maintaining data integrity. And to represent the spatial information of the object, all relevant spatial information was stored in tables - Point, Arc, ArcNode, L_R list and Polygon. The entity relationship diagram for Road Traffic Information System is shown in Appendix B.

2.5 Query Handling

The other important aspect is query handling. To handle these operations, spatial as well as temporal operators are required. The spatial operators like Intersect, WithinPolygon, Overlap and the temporal operators such as During, Start and Before are commonly used. For spatio-temporal query, integration of spatial operators and temporal operators is required.

In spatio-temporal system, query types may vary. Langran, 1992, also discussed the query types in spatio-temporal application. Some of query types in Road Traffic Information System are:

- Spatial query (e.g. - Is there any petrol station in this zone?)
- Temporal query (e.g. - what is the road segment type for road segment "S007" at July 10, 1995?)
- Temporal range query (e.g. - On which road segment, 'shell' petrol station opened in 1991?)
- Spatio-temporal query (e.g. - How many petrol stations being operated on July 5, 1993 in this division?)
- Spatio-temporal range query (e.g. - How many 'Drive Slowly' sign-post were installed in division "D007" in 1991?)

To perform these different types of query, spatial as well as temporal operators have to be defined and implemented according to the business requirement and database schema used.

Spatial query needs spatial operators like "Intersect", "Overlap" and "WithinPolygon" which are commonly used. For instance, "WithinPolygon" operator tests whether the point type object is in the polygon type object or not. Though these spatial operators are built to perform the different tasks, some of their requirements are similar. In the following example, the spatial operator "WinthinPolygon" illustrates the tasks required which are similar to those of the other spatial operators except the portion that test for spatial criteria (step 4).

1. Acquiring objects(e.g. - petrol station and zone from tables "Feature" and Division") and their information.
2. Selecting the valid version (by checking transaction time).
3. Decoding these spatial attributes (geometrical attributes) for processing.
4. Testing spatial criteria.

```
IF summation of all angle which are formed by
   connecting the vertex of point type object
   (e.g. -petrol station) and the vertices of the
   polygon type object (e.g.- zone) are
   360 degree THEN this petrol station is
   within this zone
ELSE this petrol station is not in this zone
END IF
```

The implementation of step 4 depends on the type of objects and type of functionality required. The step 1 to 3 are common and can be implemented as library functions.

Temporal query and **temporal range query** need temporal operators like "During", "Start", "Before" and "TemporalWithin". For instance, "Before" operator tests whether the object's version is earlier than the specific time or not. Temporal query tests only at the specific time while temporal range query tests for the period of time.

The query like "What are the pavement type and road segment type of this road segment while the "drive slowly" signpost are installed on that segment? And what type of pavement before this?", would need temporal join and temporal operator "Before" to retrieve the required information. The following steps would be performed for this operation.

1. Acquiring the lifespan and road segment identifier from the table "Feature" based on the condition "signpost is equivalent to drive slowly"
2. Then join the table "RoadSegment" whose object versions' lifespan intersect with the lifespan and feature identifier which have been retrieved previously.
3. Retrieving the information which meets the join condition.
4. Retrieving the different version of pavement type of the same object identifier whose time stamp is just earlier than that of that version but later than those of earlier version.

Spatio-temporal query and **spatio-temporal range query** need spatial operators as well as temporal operators to retrieve the desired information. For example, if the divisional manager want to see whether road warning sign-post "Drive-slowly" were installed at the road segment (after or before the fatal accident occurred in 1990 at that segment) and if it is installed, find when was it and how far from the accident point?, the following procedures (which includes spatial and temporal operations) is required for retrieving this information.

1. Acquiring error free objects information (i.e. - fatal accident at the specific road segment from "Accident" table) within 1990 in the division. (It needs - error checking operation and TemporalWithin operation)

2. Testing whether "Drive slowly" sign-post exists in that road segment or not (i.e. - "Drive slowly" sign-post information from the "Feature" table).

IF there is no "Drive slowly" sign-post THEN
    give "No sign-post in this road segment" message to user and terminate the operation.
ELSE
    Acquiring the error free object information at the same road segment. (It needs - error checking operation) and move to step 3.
END IF

3. DO Loop based on total number of fatal accident happened at that road segment in 1990 and sign-post "Drive slowly" at the same road segment (regardless of after or before 1990) to retrieve the spatial attribute and decoding for spatial operation.

4. DO Outer and Inner Loop based on total number of fatal accident and total number of sign-post respectively to test:

(a). the spatial criteria (i.e.- how far a particular accident point and a particular sign-post)

(b). the temporal criteria (i.e. - when did a particular sign-post installed before or after a particular fatal accident happened).

It is found that depending on the type of query and database schema used the algorithm should be implemented to achieve the efficient processing time especially in the operations which need join operation especially spatial join and temporal join.

3. CONCLUSION

Spatio-temporal data model for Road Traffic Information system has been prototyped. With the use of new data types such as point, segment and polygon types, we can achieve better performance in data access, data manipulation and data management. By having these built-in data types, we can reduce the processing time required for encoding and decoding of spatial information in additions to reduction in data volume which plays as an important part in database management. The choice of version management type and data storage structure should reflect the demand on data retrieval (i.e. - whether old version retrieval or current version retrieval)and the update operation of the system. The prototyping confirms our belief that object oriented approach is appropriate for spatio-temporal information system.

Using object oriented database system would be more efficient for management of spatial and temporal data. However, a system which can provide the advantage of relational database system as well as object oriented database system is the most desirable as the former is widely used. As such the object relational approach is the most likely environment for the spatio-temporal system development. The combination of new built-in data types and operators, ability to implement user define operator(function) as well as user defined data type, ability to use conventional SQL, and ability to implement indexing should be considered for spatiotemporal application.

References

Armstrong, M. P., Temporality in spatial database. Proceedings of the GIS/LIS '88 Conference, San Antonio, Texas, pp. 880-889, 1988.

Brown, P., Spatial indexing in modern database systems, Informix White Paper, 1996.

Effenberg, W. W., Time in spatial information systems. First Regional Conference on GIS Research in Victoria and Tasmania, Ballarat Victoria, 1992.

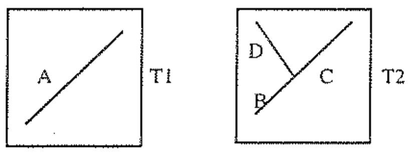
Gunther, O., The design of the cell tree: an object-oriented index structure for geo-metric databases, Proc.IEEE 5th Int. Conf. On Data Engineering (Los Angeles), 598-605, 1989.

Kemp, Z., and A. Kowalczyk., Incorporating the temporal dimension in a GIS. Innovations in GIS 1, pp. 89-102, 1993 .

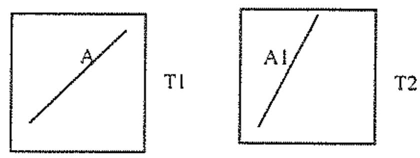
Lai, S.V., J.P. Kuilboer, and J.L. Guynes, Temporal database: Model design and commercialization prospects. Database - Vol.25, No.3, pp. 6-18, 1994.

Langran, G., and N. R. Chrisman, A frame work for temporal geographic information. *Cartographica*, 25, 1, pp. 1-14, 1988.
Langran, G., Time in geographic information systems, Taylor & Francis 1992.
Lin, H., and H. W. Calkins, A Rationale for spatiotemporal intersection, in Proceedings of the ACSM-ASPRS Annual Convention. 1991, pp. 204-213, 1991.
Saab, A., and T. W Haythornthwaite, Application of spatio-temporal database models to street network files. GIS for the 1990s Conference Proceedings, pp. 1072-1085, 1990.
Snodgrass, R and I. Ahn, A taxonomy of time in databases. In Proceedings of the SIGMOD ' 85 Conference, pp 236-245, 1985.
Snodgrass, R., Temporal database status and research directions, SIGMOD RECORD, Vol. 19 No. 4, December 1990.
Stonebraker, M., Limitations of spatial simulators for relational DBMSs, Informix White Paper, 1997.
Wachowicz, M. and R. G. Healey, Towards temporality in GIS. Innovations in GIS 1, pp. 105-115, 1993.
Worboys, M.F., Object-oriented models of spatiotemporal information. Proceedings GIS/LIS'92, San Jose, California, USA, November, 2, 825-834, 1992.

Appendix A



(a)



(b)

Figure 1(a): Creation of new objects due to topological change.

Figure 1(b): Creation of new version due to geometrical change.

Appendix B
Entity Relationship Diagram