시공간 데이터를 위한 공간 및 시간 관계 연산자의 통합

이 종 연† · 류 근 호††

요 약

이 논문은 시공간 데이터의 통합된 연산을 위하여 공간 연산자와 시간 연산자의 결합 접속을 연구한다. 시공간 연산을 위한 통합 연구는 시공간 참조 매크로에 의한 시공간 위상관계 연산자의 공통 사용을 의미한다. 또한 이 논문은 지형 객체의 이력 정보를 검색하는 이력 연산자와 시간관계 비교 연산자의 통합 알고리즘을 제안한다. 제안된 확장 알고리즘은 기존의 GIS(Geographic Information System) 공간 데이터베이스를 기반으로 구현되며, 아울러 시공간 질의 표현의 예를 통해 평가된다. 여기서 연구된 시공간 연산자의 통합은 통일된 시공간 질의를 지원하는데 유용한 기반구조를 제공할 것이다.

An Integration of Spatial and Temporal Relationship Operators for Spatiotemporal Data

Jong-Yun Lee† · Keun-Ho Ryu††

ABSTRACT

In this paper we study on an interface connection between spatial operators and temporal operators to support the unified operations for spatiotemporal data. The integration development of the spatiotemporal operations will be defined as a common use of spatiotemporal relationship operators by spatiotemporal referencing macros. Additionally, we propose an integration algorithm of a history operator and temporal relationship operators. Then the proposed system will be implemented on the commercial GIS software and evaluated by examples of spatiotemporal query expressions. Here, our integration of spatial and temporal operators will provide spatiotemporal query expressions with a useful framework.

1. Introduction

Actually the conventional general geographic information systems and data warehousing systems as well as real-time data acquisition systems cannot handle their historical data since the legacy database management systems(DBMSs) manage the snapshot image of current objects in database. Additionally, even though many studies have been conducted toward the topological comparison operations in space and time earlier, they did not support the uniformed interface in spatiotemporal query expressions. For example, Clarament[4] addressed the need to merge spatial and temporal relationship operations and Worboy[18, 19] tried to the handling of the spatial and temporal references in a uniform fashion.

※ 이 논문은 한국통신의 '98년도 정보통신 기초연구과제와 과기처 NGIS과제 연구비지원에 의해 수행되었음.
† 준 회 원 : 충북대학교 대학원 컴퓨터과학과
†† 종신회원 : 충북대학교 컴퓨터과학과 교수
논문접수 : 1998년 8월 3일, 심사완료 : 1998년 11월 4일

In this paper we therefore propose the following three spatiotemporal referencing macros to integrate with the spatial and temporal relationship operations, each of which references an additional information of the spatiotemporal object: Spatial, Valid, Transaction. First, the 'Spatial' macro is an extracting operation of spatial elements for designated object in the spatiotemporal database. Second, the 'Valid' and 'Transaction' macros reference an additional information of either valid time or transaction time for a designated object, respectively. The additional development of spatial and temporal reference macros will provide our spatiotemporal query expressions with a unified interface.

Additionally, we propose not only a classification of spatiotemporal relationship operators such as spatial relationship operators, temporal relationship operators, spatiotemporal common relationship operators, but also an integration algorithm of the history operator. The history operator is an operation which extracts all historical information of given objects in the spatiotemporal database. Especially, we also propose several of temporal comparison operators to implement them on a legacy GIS system and describe their source code in Appendix.

This paper is organized as follows : In Section 2, we review the previous temporal geographical information systems and spatiotemporal databases. Section 3 defines three spatiotemporal referencing macros in detail. Section 4 and 5 present some spatiotemporal relationship operators and the history operator. Additionally, a retrieval statement describes for spatiotemporal database. In section 6, the proposed spatiotemporal operations will be implemented and evaluated with some examples of query expressions. Finally, Section 7 summarize our work and future directions.

2. Related works

One of solutions which solve the problems not only to support time dimensions in GIS, but also to manage time-varying information of spatial objects in the real world is either spatiotemporal database or temporal geographical information system(TGIS). The temporal geographic information system is an extension of temporal elements for history management of objects in GIS, which manages additional time dimensions such as valid time, transaction time, and other time information. A number of frameworks in GIS have been studied together with the development of temporal GISs such as Claramunt[4], Langran[6, 7], Renolen's[15] and Montgomery's temporal reviews[11]. There are also Beller's development of a temporal GIS prototype in 1991[2] and Wuu's uniform model for object-oriented database[20].

The conventional GIS data models emphasize static representations of reality[21]. In the spatiotemporal database, however, the geographic objects in the world are represented by incorporating temporal elements with the two dimensional space and attributes. The geographic objects in spatiotemporal database are able to possess all changes in both space and time. Therefore, the spatiotemporal model can record all changes of attributes of a spatiotemporal object in both spatial and temporal dimensions together or separately.

In the last decade many studies have been conducted toward the development of spatiotemporal data models and query languages. One is a design of time-based spatiotemporal data model and the other is spatiotemporal extensions of previous query language. ESTDM[13], Langran[6], Mont[11], Price[14], Worboys[18, 19], and Yuan[21] are examples of previous spatiotemporal data models. Query extensions such as IXSQL[12], ORParaSQL[3], SQL/ST[8], Claramunt[4] have been also studied on the development of spatiotemporal DBMS and TGIS. Yuan[21] proposed a tightly-coupled event-based spatiotemporal data model and showed efficient, conceptually straightforward implementation of temporally-based queries relating to locations, In addition, Yuan investigated the development of temporal GIS and its

〈Table 1〉 Summary of major previous spatiotemporal databases

Authors or models	Major characteristics		
	Data model	Query language	Other properties
ESTDM	· Event-based spatiotemporal data model(ESTDM)	· Description of three temporal based queries	· Facilitate analysis of temporal relationships over time
IXSQL	· Directly supporting representation and manipulation of interval data · Interval data types & literals	· Interval extension to SQL · Interval-extended relational algebra /select/insert/update/create statement · Interval set/value functions	· An extension to SQL89 for the management of interval data · prototype implementation
Clarament	· A framework for spatiotemporal modeling · Extended-versioning model	· Typology of spatiotemporal processes · Timestamp-based operators · Events-based operators	· Addressing the need to merge space and time and to relate them
Worboy	· Unified spatio-bitemporal model · Supporting valid/transaction time concurrently	· Classification of spatiobitemporal relationship operations(relational algebra)	· Conceptually handling of the spatial and temporal references in a uniform fashion

applicability to support spatiotemporal modeling. Worboy[18] also proposed a unified model for spatiobitemporal information which has spatial and temporal components. The model was fully described in terms of its object set and the classification of spatiotemporal relationship operations upon the object set.

In summary, the characteristics of major previous works in spatiotemporal database are very theoretical and impractical, as summarized in Table 1. Clarament addressed the need to merge space and time and to relate them[4] earlier and Worboy's unified spatio-bitemporal model conceptually supported the handling of spatial and temproal references in a uniform manner[18]. Although many spatial, historical databases and temporal GISs have been also studied recently, the previous commercial spatiotemporal toolkits cannot be used directly to handle all time-varying information of spatial objects[8, 9, 10] and to retrieve all their historical information. In the following sections, we therefore suggest an extension of the spatiotemporal search statement which are very helpful to retrieve any historical changes of the geographic objects and to manage all history management of objects in the world as well.

3. Spatiotemporal reference macros

In this section we design the spatiotemporal referencing macros to integrate with spatial and temporal relationship operators in a uniform fashion. In the spatiotemporal database it is necessary to define some macros for referencing a dimensional information of spatiotemporal objects because the operands of spatiotemporal operators could be either spatial or temporal object, depending upon user's query statements. In this section we describe three referencing macros of spatiotemporal data such as Spatial(), Valid(), Transaction(), as described in Definition 3.1 through 3.3, where each i, m, n is a natural number.

First, spatial referencing macro, Spatial() is used for indicating spatial components of tuples from which spatiotemporal objects are current or historical independently. It is defined as follows:

Definition 3.1 Spatial(object) = {SC_i | spatial components SC_i from spatial attributes of either current spatiotemporal object <fid', F_1', F_2',..., F_n', SC'_i> or historical spatiotemporal object <fid'', F_1'', F_2'',..., F_n'', SC''_i>}.

As an example, there is an explicit spatial reference such as Spatial(schools), where schools in the spatiotemporal database may be a layer relation. It is very similar to the valid timestamp referencing macro in temporal database(ex. Validtime)[17].

Second, Valid() is a reference macro for the valid time of spatiotemporal objects, which indicates a pair of valid timestamps of tuples from which spatiotemporal objects are valid currently or historically in the spatiotemporal database. It is defined as follows:

Definition 3.2 Valid(object) = {VT_i | valid timestamps <VTs, VTe> from attribute data of either current spatiotemporal object <fid′, A_1′, A_2′,...., A_m′, VT′, TT′, prev′> or its spatiotemporal history object <fid″, A_1″, A_2″,...., A_m″, VT″, TT″, prev″>.

Next, Transaction() is used for referencing a pair of transaction timestamps of spatiotemporal objects, as defined as follows:

Definition 3.3 Transaction(object) = {TT_i | transaction timestamps <TTs, TTe> from attribute data of either spatiotemporal object <A_1′, A_2′,...., A_m′, VT′, TT′, prev′> or historical spatiotemporal object <A_1″, A_2″,...., A_m″, VT″, TT″, prev″>.

As defined above, we can summarize spatial and temporal referencing macros, as shown in Table 2. In this paper we adopted the prefix notation to avoid confusion and to convey clearness. There are only three possibile combinations of spatiotemporal referencing macros for the spatiotemporal query expressions such as Spatial()/Spatial(), Valid()/Valid(), Transaction()/Transaction() since the referencing macros between Valid() and Transaction() cannot be exchanged with each other, where a spatiotemporal relationship operators among them are used together.

〈Table 2〉 Spatiotemporal Referencing Macros

Macros	Examples of use
Spatial()	• Spatial(a) disjointss Spatial(b) • Spatial(b) contains Spatial(a)
Valid()	• Valid(a) precedes Timestamp 'Nov-31-1998' • Valid(a) meets Valid(b)
Transaction()	• Transaction(a) starts PERIOD 'Jan-01-1998, Nov-31-1998'

4. Integration of spatial and temporal relationship operators

In this section we are going to try to integrate with space and time reference methods in a uniform manner. The spatiotemporal relationship operators are integrated by using the spatiotemporal referencing macros defined earlier in spatiotemporal query expressions.

4.1 Spatial relationship operators

Spatiotemporal database manages five dimensional(5D) information of geographic objects, which include attributes, spatial data, valid time, transaction time, and their historical information. It basically requires extensions of its data type and query language such as data definition language(DDL) and data manipulation language(DML) for spatiotemporal data. In this section, we therefore focus on the topological comparison operators of spatiotemporal data in which include meets, overlaps, contains, disjoints, equals, and so on.

In this paper the spatiotemporal relationship operators are classified into the following three groups: spatial relationship operators for an exclusive spatial use, temporal relationship operators for an exclusive temporal use, and spatiotemporal common operators for either spatial or temporal use. The spatiotemporal database contains lots of spatial relationship operators to compare with spatial topological relationship among geographic objects. The examples of spatial relationship operators are common point, line cross, common line, area intersect, disjoints, point in polygon, contains, centroid inside polygon, inside envelope.

In this paper, futhermore, general spatial relationship operators as meets, equals, contains, overlaps are defined as spatiotemporal common operators. Other spatial relationship operators are also defined as an operation group for exclusive spatial use, named spatial relationship operators. In other words, the conventional general spatial relationship operators are classified into spatiotemporal common relationship operators and spatial relationship operators. The spatial relationship operators only include disjoints, point in polygon, inside envelope, centroid inside polygon, shortest path, and closest path for an exclusive spatial use.

4.2 Temporal relationship operators

Allen[1] proposed a complete set of period relations which includes before, after, starts, started_by, finishes, finished_by, during, contains, meets, met-by, equals, overlaps, overlapped_by. In this paper, however, the temporal relationship operators are defined as specific time-comparison operators such as before, after, starts, finishes because other temporal operators are classified into spatiotemporal common relationship operators for common use. The semantics of proposed temporal relationship operators are expressed in Table 3, where each Begin() and End() macro means an extraction of the beginning time and the ending time of a given period, respectively. They are then implemented by converting their semantics into their relational expressions, as listed in Appendix A.

<Table 3>Time relational expressions of exclusive temporal relationship operators

Temporal operators	Temporal semantics
a before b	End(a) < Begin(b)
a after b	End(b) < Begin(a)
a starts b	Begin(a) = Begin(b) and End(a) < End(b)
a finishes b	Begin(b) < Begin(a) and End(a) = End(b)

4.3 Spatiotemporal Common Relationship Operators

As Table 4 shows, the third group of proposed spatiotemporal operations is the spatiotemporal common relationship operators which their operands could be either spatial objects or temporal objects, depending on user's query statements. At that case the comparison operators in Table 4 may be used in conjunction with dimensional referencing macros as Spatial(), Valid(), and Transaction(). The spatiotemporal common relationship operators are therefore operated differently as either a spatial relationship operator or a temporal relationship operator, depending upon its operands as Spatial(), Valid(), and Transaction().

In this paper, the proposed spatiotemporal common relationship operators for temporal use are also implemented by converting operator's temporal semantics into relational expressions, as listed in Appendix A. The spatiotemporal relationship operators for spatial use are implemented by calling the spatial operations of conventional spatial database engine in which are passed by given spatial relationship macros of the spatial database engine.

5. History Operator

5.1 What is the history operator?

Geographic objects in the spatiotemporal database are represented by their states and the timestamp information, when an event happens. In the spatiotemporal database all spatial objects which happen to change their states in the real world are also stored as their historical data which are consisted of a pair of time and non spatial data with spatial information. Through these historical data, we would like to both retrieve and manage spatial, non-spatial histories of geographic objects. Hence, we propose the following history operator to support all historical interrogations for spatial objects which are keep changing their states over time.

As described the spatiotemporal reference macros earlier, we also adopt the prefix notation as an interface of the history operator. The efficiency and convenience of use for the history operator are explained by the following queries:

[Q1] List all the history of the owner whose the lot number of a building is 100.

〈Table 4〉 Spatial and temporal semantics of spatiotemporal common relationship operators

Spatial semantics	Spatial and temporal uses for spatiotemporal common operators	Temporal semantics
boundary(a) ∩ boundary(b) ≠ ∅ interior(a) ∩ interior(b) = ∅	Spatial(a) meets Spatial(b) Valid(a) meets Valid(b) Transaction(a) meets Transaction(b)	End(a) + '1 granule' = Begin(b)
boundary(a) ∩ boundary(b) ≠ ∅ interior(a) ∩ interior(b) ≠ ∅	Spatial(a) overlaps Spatial(b) Valid(a) overlaps Valid(b) Transaction(a) overlaps Transaction(b)	Begin(a) < Begin(b) and End(a) < End(Y) and End(a) > Begin(b)
boundary(a) ∩ boundary(b) = ∅ interior(a) ∩ interior(b) ≠ ∅	Spatial(b) contains Spatial(a) Valid(b) contains Valid(a) Transaction(b) contains Transaction(a)	(Begin(a) < Begin(b) or Begin(a) = Begin(b)) and (End(a) > End(b) or End(a) = End(b))
boundary(a) = boundary(b) interior(a) = interior(b)	Spatial(a) equals Spatial(b) Valid(a) equals Valid(b) Transaction(a) equals Transaction(b)	Begin(a) = Begin(b) and End(a) = End(b)

```
SQL/ST> SELECT history(name, address)
      > FROM buildings
      > WHERE gid = 100;
```

[Q2] List all the spatial histories of a building whose number is 100, where the Spatial() macro means an extraction of spatial components for the object in the real world.

```
SQL/ST> SELECT history(Spatial(*))
      > FROM buildings
      > WHERE gid = 100;
```

5.2 Processing algorithm for history operator

It is often required that we interrogate all histories of designated spatial objects. All historical data of objects in proposed spatiotemporal database can be retrieved by historical pointers of spatiotemporal objects since all histories of either their spatial or attribute data shall be managed by either historical pointer or extra attribute tables. The history operator is a selection and projection operation which chooses all histories of attributes and/or non-spatial data of objects in the spatiotemporal database. The processing results may be many historical set or empty, depending on the estate state. The proposed algorithm of the history operator is described clearly, as shown in Figure 1.

The spatiotemporal queries can be expressed very shortly by using the proposed referencing macros and the history operator, as shown in the following example. In the spatial case the results will be visualized into maps on GIS maps. Otherwise, attribute data will be viewed as a table structure having its attributes, valid time, and transaction time.

[Q3] Find all the movement paths of vehicle 'Seoul 4-6179' on May 1, 1998.

```
SQL/ST> SELECT History(location)
      > FROM Vehicle v
      > WHERE v.cno = 'Seoul 4-6179' and
        Valid(v) overlaps PERIOD '01-May-1998, 01-May-98';
```

```
PROCEDURE History(feature)
BEGIN
    a. Check the current read mask from its feature layer;
    b. If (the current layer is not readable)
         set the mask of the current layer into readable;
    c. Set the valid period by the given period, gvp.
    d. If (the given feature exists in current feature table)
         Read its previous pointer from the current feature table;
       else
         Retrieve its previous pointer from the feature history table;
    e. For (; its previous pointer != NULL; ) {
         If (a feature overlaps the valid time of the given feature) {
           e1. Display the spatial component of history
               of its feature;
           e2. Display the attribute information of history
               of its feature;
           e3. Set the previous pointer of designated object
               into its historical pointer;
         } /* the overlapped valid period */
       }  /* for statement */
END;
```

(Fig. 1) Proposed algorithm of the history operator

5.3 Grammar extension for spatiotemporal search statement

In this section, we describe the data retrieval statement for spatiotemporal data, called SQL/ST. The following search statement in SQL2 is extended by proposed spatiotemporal referencing macros, spatial and temporal relationship operators, and a history operator to support historical operations and handle spatiotemporal query expressions in a uniform manner.

```
SELECT <attribute list>
FROM   <relation list>
WHERE  <attribute qualifications>
```

The Appendix B is a brief BNF notation of search statement of the spatiotemporal database in SQL/ST, where the attribute qualifications on the where clause may be combined with other attribute predicates and it is similar to the previous query expressions in SQL2. In lines (2)-(3) of Appendix B are executed as in SQL2, but lines (1), (4)-(6) in the where clause should be extended for a retrieval of spatiotemporal data. Hence, spatial, temporal, and spatiotemporal patterns in spatiotemporal queries should be integrated with a unified fashion.

6. Implementation and review

In this section the proposed spatiotemporal database schema and its operations are implemented and their results are evaluated with examples of spatiotemporal queries.

6.1 Implementation

The integrated spatiotemporal operations were implemented by using Oracle DBMS 7.2.3 and SDE (Spatial Database Engine) 2.1[5] on Solaris 2.5.

As described earlier, the spatial relationship operators were also implemented by calling the spatial searching functions on which are passed by the spatial comparison macros in SDE. Additionally, both of the proposed temporal relationship operators and the spatiotemporal common relationship operators for temporal use were implemented by converting their temporal semantics into relational conditional expressions, as listed in Appendix A. Then they consist of a spatiotemporal search condition which is combined with the conventional attribute predicate and the additional spatial, temporal elements in search functions, named by application program interface(API) functions in SDE(Spatial Database Engine).

In summary, both of the temporal relationship operators and the spatiotemporal common relationship operators for temporal use are implemented by converting the meanings into relational expressions using C language, as listed in Appendix A. The converted relational expressions are used as the spatiotemporal retrieval predicate for searching functions using SDE 2.1, which consists of spatial, temporal predicate as well as the conventional attribute predicate.

6.2 Examples of integrated queries

In this section the convenience and efficiency of proposed spatiotemporal operations are explained by examples of spatiotemporal queries including spatial relationship operators, temporal relationship operators, spatiotemporal relationship common operators, history operator, and spatiotemporal referencing macros. First, we show query expressions frequently being closed when we use the automatic vehicle location system, as listed follows:

[Q1] Find the ambulance car be the closest distance from which an accident breaks out in Ahnguk-dong of Jongro gu ?

```
SQL/ST> SELECT v.cno, v.location
      > FROM Vehicles v, Accidents a
      > WHERE a.county = 'Jongro-gu' and
          a.address = 'Ahnguk-dong' and
          Spatial(v) is_closest Spatial(a) ;
```

[Q2] Find all of fire engines contained in the same county which a fire breaks out at a 'Jong' builing of Jongro gu on May 1,1998 ?

```
SQL/ST> SELECT v.cno, v.location
      > FROM Vehicles v, Buildings b
      > WHERE b.county = 'Jongro gu' and
        b.name = 'Jong' and
        Spatial(b.county) contains Spatial(v)
        and Valid (b) during Period
        '01-May-98, 01-May-98' ;
```

[Q3] Find all movement paths of vehicle A on 28 April, 1998 today ?

```
SQL/ST> SELECT History(location, VTs, VTe)
      > FROM Vehicles v
      > WHERE v.cno = 'A' and
        Valid(v) overlaps CURRENT_DATE;
```

In the above examples, [Q1] is a simple spatial query, but [Q2] and [Q3] queries are spatiotemporal queries which are combined with spatial and temporal predicates.

Next, we represent the spatiotemporal queries of spatial objects consisted of the temporal GIS. For example, we can express the historical interrogation of either spatial or non-spatial data in GIS as follows:

[Q5] List all histories of the owner and his or her address histories of the 'Jongro-gu A' building since 1990.

```
SQL/ST> SELECT History(owner, address)
      > FROM Buildings b
      > WHERE b.name = 'A' and
        b.county = 'Jongro-gu' and
      Valid(b) overlaps PERIOD '01-Jan-90,
          CURRENT_DATE';
```

[Q6] List all of the spatial histories of the 'Jongro-gu A' building since 1990.

```
SQL/ST> SELECT History(Spatial(*))
      > FROM Buildings b
      > WHERE b.name = 'A' and
        b.county = 'Jongro-gu' and
        Valid(b) overlaps PERIOD
        '01-Jan-90, CURRENT_DATE';
```

6.3 Evaluation

In this paper, we designed the spatiotemporal operations including spatiotemporal topological relationship operators, a history operator, spatiotemporal referencing macros, and implemented them. Hence, we report briefly on the prototype implementation results and other considerations as follows:

Firstly, many studies conducted toward the development of spatiotemporal DBMS and TGIS with special purposes. For examples, spatiotemporal DBMS, IXSQL were proposed for the management of interval data[15] and ORParaSQL[5] for pattern matching. In this paper, our system SQL/ST, however, handles spatial, temporal, spatiotemporal queries in a uniform manner since we completely unified the spatial and temporal relationship operation for spatiotemporal data.

Secondly, we classified the spatiotemporal relationship operators into three groups like spatial relationship operators, temporal relationship operators, spatiotemporal common relationship operators. Additionally, we proposed the spatiotemporal referencing macros for extracting either spatial or temporal values of spatiotemporal data and the history operator. The spatiotemporal queries should be therefore

〈Table 5〉 Major results of our proposed SQL/ST

Proposed operations	Results of implementation
· Unification of spatiotemporal operations · Definition of spatial/temporal/spatiotemporal common relationship operators · Spatiotemporal referencing macros · A 'history' operator · A brief BNF expression of select statement	· Integrating with spatial and temporal relationship operators by using referencing macros. · Extension of spatiotemporal queries · Based on Oralce DBMS & SDE 2.1

integrated with spatiotemporal queries by using the spatiotemporal referencing macros.

Thirdly, we proposed a brief BNF statement of select statement in SQL3 which was extended by reference macros, spatiotemporal relationship operators, history operator, and both spatial and temporal clause.

In the majority of temporal extensions to the relational model, valid and transaction times are represented as single chronon, sets of consecutive chronons, and arbitrary sets of chronons. However, there are extensions in which time is represented either as sets of consecutive chronons or as a union of disjoint intervals[21]. In our approach we represented the structure of valid and transaction time as a pair of time points (start, end) because we can use the same query processing algorithm in conventional relational model.

Finally, our contribution and results in space and time database can be described as summarized in Table 5.

7. Conclusion

The characteristics of spatiotemporal data in the real world are varying over time, which are managed in GIS software. However, the conventional GIS software cannot handle time-varying data because it neither control the historical information of spatial objects nor support their spatiotemporal operations. Additionally, the previous spatiotemporal query languages cannot be unified the spatial and temporal relationship operators.

In this paper, we therefore suggested an integration of the spatial and temporal relationship operators by the spatiotemporal referencing macros for extracting the designated dimensional information of spatiotemporal data. We also proposed a history operator as well as a brief BNF statement of search statement in SQL3 which are extended by data types, time clause, and spatiotemporal operations to be able to support both spatial and temporal domains. Then, they were implemented and reviewed by examples of spatiotemporal queries.

Finally, it is obvious that the proposed spatiotemporal system should integrate with the spatial, temporal, spatiotemporal relationship operators in a uniform manner and support all historical interrogations of spatiotemporal data from users as well. In the future, we have plan to study an additional spatiotemporal indices and spatiotemporal query language to support the efficient and convenient query processing completely.

References

[1] Allen, J. F., "Maintaining Knowledge about Temporal Intervals," Communications of the Association of Computing Machinery, Vol.26, No. 11, pp.823-843, Nov. 1983.

[2] Beller, A., T. Giblin, K. V. Le, S. Litz, T. Kittel, and D. Schimel, "Temporal GIS prototype for global change research," GIS/LIS 1991 proceedings, Vol.2, pp.752-765, 1991.

[3] Cheng, Tsz S. and Shashi K. Gadia, "A pattern matching language for spatio-temporal databases," CIKM'94, pp.287-295, 1994.

[4] Claramunt, C., "Managing Time in GIS : An Event-Oriented Approach," Recent Advances in Temporal Databases, Workshops in Computing Series, Edited by J. Clifford, and A.Tuzhilin(eds) (Berlin:Springer-Verlag), pp.23-42, 1995.

[5] ESRI, Introduction to SDE^{TM}, Environ-mental Systems Research Institute, Inc., 1996.

[6] Langran, G., "A framewrok for temporal geographic information systems," Cartographica, Vol. 25, No.3, 1988.

[7] Langran, G., Time in Geographic Information Systems, London : Tayor & Francis, 1993.

[8] Lee, Jong, Seong Lee, and Keun Ryu, "SQL Unification of Spatiotemporal Operations for AVLS : SQL/ST," Proceeding of the 16Th Annual Conference of the International Association of Management, Chicago, USA, Aug.5-8, 1998.

[9] Lee, Jong, Byoung. Ahn, and Keun Ryu, "A Historical Extension for SDE Data Model," The Transaction of the Korea Information Processing Society, Vol.5, No.9, 1998.

[10] Lee, Jong and Keun Ryu, "Extension to Temporal Relationship Macros for History Mangement in SDE™," Journal of Korea Information Science Society(C), 1998.

[11] Montgomery, L. D., Temporal geographical infor mation systems technology and requirements : Where we are today, Master's thesis, The Ohio State University, 1995.

[12] Lorentzos, Nikos A. and Yannis G. Mitsopoulos, "SQL Extension for Interval Data," IEEE Transactions on Knowedge and Data Engineering, Vol.9, No.3, pp.480-499, May/June 1997.

[13] Peuquet, D. and N. Duan, "An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data," Information Systems, Vol.9, No.1, pp.7-24, 1995.

[14] Price, Stephen, "Modeling the temporal element in land information systems," Int. Journal of Geographic Information systems, Vol.3, No.3, 1989.

[15] Renolen, A., "Temporal Maps and Temporal Geographical Information Systems," Dept. of Surveying and Mapping(IKO), The Norwegian Institute of Technology, Feb. 14, 1996.

[16] Snodgress, R. T., "The Temporal Query Language TQuel," ACM Transactions on Database Systems, Vol.12, No.2, pp.247-298, June 1987.

[17] Snodgrass, R. T., The TSQL2 Temporal Query Language, Tucson AZ : Kluwer Academic Publishers, 1995.

[18] Worboys, Michael F., "A data model for information with spatial and bitemporal components," TR93-06, Feb. 1993.

[19] Worboys, Michael F., "A Unified Model for Spatial and Temporal Information," the Computer Journal, Vol.37, No.1, 1994.

[20] Wuu, G. and U. Dayal, "A Uniform model for temporal object-oriented databases," In Proceedings of the International Conference on Data Engineering, pp.584-593, 1992.

[21] Yuan, May, "Temporal GIS and Spatio-temporal Modeling," the 3rd Internationl Conference/Workshop Integrating GIS and Environmental Modeling, Jun. 21, 1996.

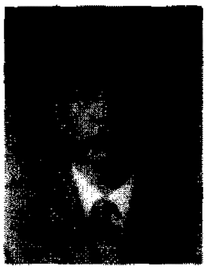
Appendix A. Source code for temporal predicate operations

```
/*---------------------------------------*/
/* Compute the temporal predicate        */
/*    t_predicate=temporal    predicate,op=temporal
operator */
/* begin=the start valid time,end=the end of valid
time*/
/*---------------------------------------*/
int compute_temporal_predicate(
char *t_predicate, int op, char *begin, char *end)
{
  switch (op) {
   case BEFORE : /* BEFORE = PRECEDES */
   case PRECEDES :
     strcat(t_predicate, "VTe < '");
     strcat(t_predicate, begin);
     strcat(t_predicate, "'"); break;
   case AFTER :
     strcat(t_predicate, "VTs > '");
     strcat(t_predicate, end);
     strcat(t_predicate, "'"); break;
   case OVERLAPS :
     strcat(t_predicate, "(VTs < '");
     strcat(t_predicate, begin);
     strcat(t_predicate, "' and ");
     strcat(t_predicate, "VTe < '");
     strcat(t_predicate, end);
     strcat(t_predicate, "' and ");
     strcat(t_predicate, "VTe > '");
     strcat(t_predicate, begin);
     strcat(t_predicate, "' or ");
     strcat(t_predicate, "VTs > '");
     strcat(t_predicate, begin);
     strcat(t_predicate, "' and ");
     strcat(t_predicate, "VTe > '");
     strcat(t_predicate, end);
     strcat(t_predicate, "' and ");
     strcat(t_predicate, "VTs < '");
     strcat(t_predicate, end);
     strcat(t_predicate, "')"); break;
   case CONTAINS :
```

```
      strcat(t_predicate, "(VTs > '");
      strcat(t_predicate, begin);
      strcat(t_predicate, "' or VTs = '");
      strcat(t_predicate, begin);
      strcat(t_predicate, "') and (VTe < '");
      strcat(t_predicate, end);
      strcat(t_predicate, "' or VTe = '");
      strcat(t_predicate, end);
      strcat(t_predicate, "')"); break;
    case MEETS :
      strcat(t_predicate, "(VTe + 1) = '");
      strcat(t_predicate, begin);
      strcat(t_predicate, "'"); break;
    case STARTS :
      strcat(t_predicate, "VTs = '");
      strcat(t_predicate, begin);
      strcat(t_predicate, "' and VTe < '");
      strcat(t_predicate, end);
      strcat(t_predicate, "'"); break;
    case FINISHES :
      strcat(t_predicate, "VTs > '");
      strcat(t_predicate, begin);
      strcat(t_predicate, "' and VTe = '");
      strcat(t_predicate, end);
      strcat(t_predicate, "'"); break;
    default: return TCL_ERROR;
  }
  return TCL_OK;
}
```

Appendix B. A brief BNF description of search statement in SQL3

(1) SELECT **<extended attribute list>**
(2) FROM <relation list>
(3) WHERE <attribute qualifications>
(4) [AND Valid(object) <temporal relationship operators> <Timestamp reference expressions>
(5) [AND Transaction(object) <temporal relationship operators>
<Timestamp reference expressions>
(6) [AND Spatial(object) <spatial relationship operators> Spatial(object)]
(7) <timestamp reference expressions>
::= [Valid(object) | Transaction(object)] |
Timestamp '<datetime>' |
PERIOD '<datetime> - <datetime>' |
(8) <spatial relationship operators>
::= DISJOINTS | IS_CLOSEST | IS_SHORTEST |
PIP | INS_EVELOPE | CENTROID_INS_POLYGON |
<spatiotemporal common relationship operators>
(9 <temporal relationship operators>
::= BEFORE | AFTER | PRECEDES | STARTS |
FINISHES | **<spatiotemporal common relationship operators>**
(9) <spatiotemporal common relationship operators> ::= EQUALS | MEETS |
OVERLAPS |CONTAINS
(10) **<extended attribute list>**
::= history(<attribute list>) | <attribute list>



이 종 연

e-mail : jylee@dblab.chungbuk.ac.kr
1985년 2월 충북대학교 전자계산학과(공학사)
1987년 2월 충북대학교 대학원 전자계산기공학과(공학석사)
1999년 2월 충북대학교 대학원 전자계산학과(이학박사)
1990년~1994년 현대전자(주) 소프트웨어연구소(주임)
1994년~1996년 현대정보기술(주) CIM사업부 근무(책임)
관심분야 : 시공간 데이터베이스, 객체 모델링, GIS, CIM

류 근 호

e-mail : khryu@dblab.chungbuk.ac.kr
1976년 숭실대학교 전산학과 졸업(이학사)
1980년 연세대학교 산업대학원 전산전공(공학석사)
1988년 연세대학교 대학원 전산전공(공학박사)
1976년~1986년 육군 군수 지원사 전산실(ROTC 장교), 한국전자통신연구소(연구원), 한국방송통신대학교 전산학과(조교수) 근무
1989년~1991년 Univ. of Arizona Research Staff (TempIS 연구원, Temporal DB)
1986년~현재 충북대학교 컴퓨터과학과 교수 겸 컴퓨터정보통신연구소장
관심분야 : 시간지원 데이터베이스, 시공간 데이터베이스, 정보검색, 객체 및 지식베이스 시스템