

관계형 시공간 데이터베이스 질의언어와 연산

김 동 호[†] · 류 근 호^{††}

요 약

시공간 데이터베이스는 현실세계에 존재하는 다양한 유형의 객체에 대한 공간 관리와 이력 정보를 동시에 제공함으로써 지리정보시스템, 도시계획시스템, 자동차주행시스템 등과 같이 다양한 응용분야에서 사용된다. 지금까지 시공간 데이터베이스 모델에 관한 기본적인 연구가 진행되었으나, 시공간 데이터베이스 질의언어에 대한 연구는 매우 미비한 상태이다. 따라서 이 논문에서는 먼저 시공간 객체를 데이터베이스화하기 위한 모델을 살펴보고, 시공간 질의언어가 제공하는 기능들을 정리한다. 또한 관계형 질의 언어인 SQL을 기반으로 효율적인 이력 정보와 공간 관리기능을 제공하는 시공간 데이터베이스 질의언어인 STQL을 제안하며, 제안된 질의언어의 시공간 질의연산 수행을 보인다.

A Relational Spatiotemporal Database Query Language and Their Operation

Dong-Ho Kim[†] · Keun-Ho Ryu^{††}

ABSTRACT

The spatiotemporal databases support historical informations as well as spatial managements for the objects in the real world, and can be efficiently used to various applications such as geographic information system, urban plan system, car navigation system. Although there are so far several pioneering works in spatiotemporal data modeling, it is little to study on spatiotemporal query language. So in this paper, we at first survey a modeling technique that makes spatiotemporal objects into databases, and then show some significant functions supported by query language. Also we suggest a new spatiotemporal database query language, entitled as STQL, which make users to enable to manage efficient historical information as well as spatial management function on the basis of relational query language, SQL. Also the proposed STQL shows their operations in spatiotemporal databases.

1. 서 론

시공간 데이터베이스(spatiotemporal databases)는 현실세계에 존재하는 다양하고 복잡한 객체에 대하여 공간 관리와 이력 정보를 동시에 제공함으로써, 지리

정보시스템, 도서관리시스템, 환경관리시스템, 자동차주행시스템 등과 같이 광범위한 응용분야에서 적용된다[8,11].

지난 30여년간 시간지원 데이터베이스(temporal databases)[19] 및 공간 데이터베이스(spatial databases)[6]와 관련된 왕성한 연구가 수행되었지만, 이들은 서로를 고려하지 않고 독립적인 영역으로서만 진행되어왔기 때문에 이들을 직접적으로 시공간 데이터베이스에 적용하기 어려운 문제점을 갖는다[8]. 최근들어

* 이 연구는 정보통신부 대학기초연구지원사업 및 한국과학재단 핵심전문연구(과제번호: 971-0902 015 2)에 의하여 수행되었음.
[†] 준 회 원 : 충북대학교 대학원 전자계산학과
^{††} 종신회원 : 충북대학교 컴퓨터과학과 교수
 논문접수 : 1998년 6월 22일, 심사완료 : 1998년 8월 12일

시공간 데이터베이스에 관한 관심이 증가하면서 주로 선진국을 중심으로 시공간 데이터베이스 연산 및 모델에 대한 연구[3,10,12,20]들이 진행되고 있지만, 이력정보와 공간정보를 동시에 효율적으로 관리하기 위한 시공간 질의 언어의 연구는 미비한 실정이다.

따라서 이 논문에서는 지금까지 수행된 시간지원 데이터베이스, 공간 데이터베이스, 그리고 시공간 데이터베이스 모델 및 질의언어에 관련된 연구들을 살펴보고, 이를 바탕으로 현실세계의 객체에 대하여 공간 관리와 이력 관리를 효율적으로 동시에 지원하는 선형 삼차원 시공간 데이터베이스 모델과 이를 기반으로 하는 관계형 시공간 데이터베이스 질의언어인 STQL을 제안한다. 효율적인 논문 전개를 위해 다음과 같이 구성하였다. 먼저 2장에서는 대표적인 시간지원 데이터베이스 질의 언어와 공간 데이터베이스 질의 언어들의 특징을 간략히 살펴보고 아울러 시공간 데이터베이스의 관련된 이전 연구들을 정리한다. 3장에서는 선형 삼차원 시공간 데이터 모델의 구조와 관계형 시공간 데이터베이스에서의 테이블 구조를 소개하며, 4장에서는 시공간 질의 언어를 설계하며, 관계형 시공간 데이터베이스 질의 언어인 STQL의 특징과 세부적인 기능들을 5장에서 설명한다. 6장에서는 제안된 STQL과 기존의 SQL과의 동질성을 위한 고려사항과 특수한 시공간 상수값을 살펴보고, 마지막으로 7장에서는 이 연구의 요약 및 앞으로의 연구를 제시한다.

2. 관련 연구

시간지원 데이터베이스 질의언어에는 대표적으로 TQuel[15], TSQL2[16], HQuel[18] 등이 있다. 특히 TQuel은 관계 해석(relational calculus)을 기반으로 하는 Quel을 시간차원으로 확장한 언어로서 구문상의 명료함과 시간 연산에 대한 수학적 증명의 용이함의 장점을 갖는 언어이며, TSQL2는 시간지원 데이터베이스 표준 질의어로 목표를 두어 연구된 언어로서 이전의 질의언어들이 제공하는 대부분의 기능을 호환하여 제공하는 장점을 갖는다. 하지만 이들 질의 언어는 다양한 유형의 이력연산을 제공할 수 있는 장점을 갖지만, 기본적인 타입만을 대상으로 이력을 지원하며 공간 속성을 처리하기 위한 확장 방안이 매우 미비한 문제점을 갖는다. 그리고 공간 데이터베이스 질의언어에는 Spatial SQL[4], PSQL[14], Geo-SAL[17] 등이 있다.

그러나 이들 질의 언어는 오직 현재 시점에서 유효한 공간 정보만을 대상으로 연산이 이루어지기 때문에 이력을 제공하기 위한 질의 작성이 매우 어려운 문제점을 갖는다. 반면 시공간 데이터베이스는 기본적인 속성뿐만 아니라 공간 속성에 대한 용이한 이력 연산을 지원할 뿐만 아니라 효율적인 저장관리를 제공하는 장점을 갖는다. 하지만 초창기에 있는 시공간 데이터베이스의 이전의 연구들은 데이터베이스 모델에 치중하여 진행되었다. 세부적인 연구들로는 시간지원 데이터베이스와 공간 데이터베이스에 관련된 연구조사[1], 시공간 데이터베이스 모델링[3,8,10,12,20] 및 시공간 연산자[11] 등이 수행되었으며, 개념적인 시공간 데이터베이스 모델링에는 시간분할 스냅모델(time-slice snapshot model)[10], 기본상태와 개정모델(base state with amendments model)[12], 공간-시간 복합 모델(space-time composite model)[10], Tempest 삼원 모델(tempest triad model)[16] 등이 있다. 아울러 일부 연구에서는 기존의 SDE[5] 또는 고딕(Gothic)과 같은 공간 데이터베이스를 기반으로 그래픽 사용자 접속기상에서 시간 연산을 위한 간단한 메뉴를 두어 내부 함수를 호출하는 형태를 가지므로 초보 사용자에게는 비교적 용이한 장점을 갖지만 전문 사용자의 다양한 유형을 표현하기가 매우 어려울 뿐만 아니라 처리 비용의 관점에서 하위의 거대한 기존 시스템과 상위의 부가적인 시스템간의 연동을 위한 과도한 비용과 불필요한 과정이 야기되는 문제점을 갖는다.

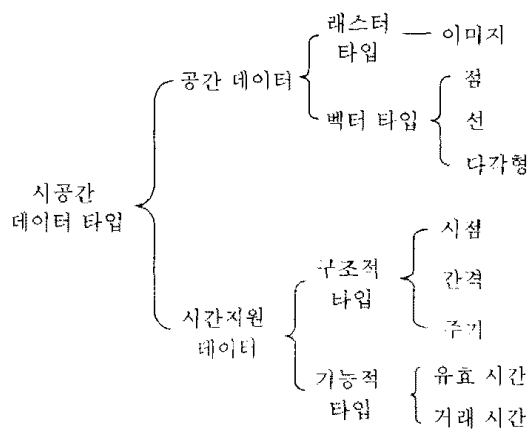
따라서 이 논문에서는 이전 연구를 바탕으로 현실세계에 존재하는 객체에 대한 시공간 정보를 데이터베이스에서 관리하기 위한 모델로서 선형 삼차원 시공간 데이터베이스 모델을 제안하고, 기존의 SQL를 기반으로 시간지원 연산과 공간 연산을 표현할 수 있도록 확장하며 세부적인 질의 표현 예와 연산의 수행을 보인다.

3. 데이터베이스 모델

3.1 시공간 자료형

기존의 데이터베이스에서 정의된 정수형, 실수형, 문자형, 부울형과 같은 일반적인 자료형과 함께 시공간 자료형은 시간지원 자료형과 공간 자료형의 두 가지 주된 부류로 세분화된다. 시간지원 자료형은 구조적 관점과 사용 관점에 따라서 구조적 자료형과 기능적 자료형으로 분류된다. 그림 1과 같이 구조적 관점

의 시간지원 자료형에는 객체에 대한 시점(event), 주기(interval) 및 간격(period) 자료형이 있다. 시점 자료형은 오직 하나의 시간값으로 구성되며 해당 객체가 현실세계에서 유효했던 시간을 의미한다. 그리고 주기 자료형은 해당 객체가 현실세계에서 유효했던 시간들의 길이를 의미하며, 간격 자료형은 객체가 유효했던 시간의 범위로서 시작시간과 종료시간의 쌍으로 구성된다. 반면 사용 관점에서 유효 시간(valid time)은 해당 객체가 현실세계에서 실제 유효했던 시간값을 의미하며, 거래 시간(transaction time)은 해당 객체가 데이터베이스에서 처리되어진 시간값을 의미한다. 또한 공간 자료형은 크게 벡터형(vector type) 자료와 래스터형(raster type) 자료로 구분된다. 래스터 자료형은 해당 객체에 대한 이미지 또는 그림을 의미하며 인간에게 공간 정보의 이해를 증진하기 위해 사용된다. 그리고 벡터형 자료는 공간 정보를 계산하기 위해 사용되며, 점(point), 선(line), 다각형(polygon)과 같은 세부적인 자료형으로 구분되어 일련의 공간 좌표의 쌍으로 표현된다. 점 자료형은 가장 단순한 형태로서 하나의 X좌표와 Y좌표로 구성된 이차원 원소로 구성되며, 선 자료형은 시작점과 끝점을 의미하는 하나 이상의 점 자료형의 순서쌍으로 표현된다. 마지막으로 다각형 자료형은 시작점과 끝점이 일치하는 폐곡선의 형태로써 일련의 점 자료형의 집합으로 구성된다.



(그림 1) 시공간 자료형
(Fig. 1) Spatiotemporal data type

3.2 시공간 연산

시공간 연산에는 그림 2와 같이 기본적으로 일반 속성값을 대상으로 하는 기존의 관계형 연산(relational

operation)과 시간 속성을 대상으로 하는 시간지원 연산(temporal operation) 및 공간 속성을 대상으로 하는 공간 연산(spatial operation)으로 구분된다. 관계형 연산에는 튜플의 선정(select), 속성의 추출(project) 및 조인(join)등이 있으며, 시간 연산에는 객체에 유효한 시간 정보를 부여하는 시간지원 구성자 연산(temporal construct operation)과 시간상의 인접(adjacent), 포함(contain), 중첩(overlap)등과 같이 저장된 객체로부터 지정된 시간조건을 만족하는 객체를 검색하는 시간지원 조건자 연산(temporal predicate operation)등이 있다. 아울러 공간 연산에는 해당 객체에 공간 정보를 부여하는 연산과 함께 공간 조건을 기술하는 두 가지 유형의 공간 조건 연산(spatial predicate operation)이 제공된다. 공간 조건 연산은 교차(cross), 인접(contact), 포함(enclose), 분리(disjoint), 방향(direction)등과 같이 두 객체의 공간상의 위치 관계를 판별하는 위상 관계 연산(topological operation)과 거리(distance), 길이(length), 반경(perimeter) 등과 같이 객체 또는 객체 간 유클리드 공간상의 값을 계산하는 기하 연산(geometric operation)이 있다. 특히 위상 관계 연산과 시간지원 관계 연산은 모두가 Allen[2]의 연산에 기반을 두고 있으며 매우 유사한 특징을 갖는다.

O 시간지원 연산

시간지원 관계 연산[2]

adjacent, contain, equal, precede, overlap

- 시간지원 구성 연산[15,16]

valid, transaction, scale, cast, bind, nobind

O 공간 연산

- 기하 연산[4]

distance, length, perimeter, area, position

- 위상 관계 연산[14]

cross, disjoint, contact, enclose,

east, west, south, north

(그림 2) 시공간 연산자

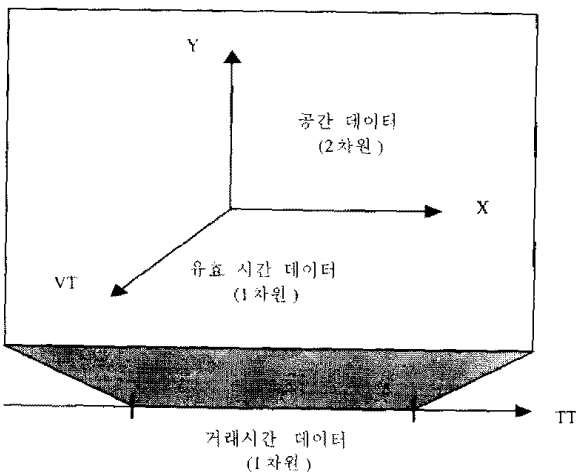
(Fig. 2) Spatiotemporal operators

3.3 데이터베이스 구조

현실 세계에 존재하는 객체는 기존의 데이터베이스에서 정의된 일반적인 정보와 함께 이차원의 공간 정

보와 일차원의 시간정보로 구성된 삼차원 형태를 갖는다. 시간 영역의 관점에서 객체에 대한 시간 정보는 해당 객체가 현실 세계에서 유효한 시간 값을 시점 또는 간격의 형태로 표현된다. 결국 객체에 대한 시공간 정보는 최소 삼차원의 구조를 형성하며, 이를 통해 주어진 시간조건을 만족하는 공간 정보를 추출하거나 해당 객체의 공간 정보가 변화된 이력을 추적할 수 있다. 하지만 시공간 객체를 데이터베이스에서 관리하기 위해서는 거래 시간이라는 부가적인 시간 영역이 추가되어야만 정확한 이력관리가 가능하다. 이것은 시간지원 데이터베이스에서 설명되는 시간 영역의 고유한 특성에서 기인한다. 따라서 이 논문에서는 (식 1)과 그림 3에서 표현한바와 같이 선형 일차원 시간영역에서 정의되는 삼차원 시공간 데이터 모델을 제안한다.

$$\lambda = (TT, \{VT, x, y\}) \quad (식 1)$$

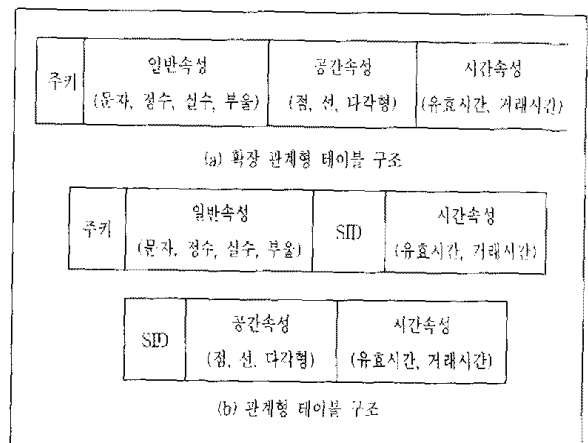


(그림 3) 이원시간을 지원하는 삼차원 구조
(Fig. 3) 3-D Architecture with Bitemporal time

즉, 선형 삼차원 시공간 데이터 영역은 그림 3과 같이 일차원 선형 거래 시간 영역에서 정의되는 3차원 시공간 요소를 계층적으로 표현하였는데, 시공간 영역을 구성하는 요소들은 유효 시간과 거래 시간으로 구성되는 이차원의 시간 영역과 이차원의 공간 영역으로 구성되기 때문에 이를 통합하는 구조를 표현하기 위해서는 최소 사차원의 구조를 가져야하지만 이론적으로 삼차원 이상의 구조를 형성할 수 없는 이유로 선형 삼차원 구조를 사용한 것이다.

3.4 테이블 구조

선형 삼차원 시공간 데이터베이스 모델에서 하나의 테이블에는 그림 4와 같이 사용자가 정의하는 일반 속성과 공간 속성, 그리고 시스템에 의해 관리되는 유효 시간과 거래 시간 속성으로 구성된다. 여기서 공간 속성은 하나 이상의 원자값(atomic value)으로 구성됨을 가정한다. 즉, POINT 속성 타입은 하나의 공간 좌표쌍을 가지며, LINE 속성 타입은 두 개의 공간 좌표쌍으로 구성된다. 그리고 POLYGON 속성 타입은 두 개 이상의 공간 좌표쌍으로 구성된다. 아울러 테이블에 대하여 속성 단위의 유효시간과 튜플 단위의 거래시간을 지원함으로써 유효시간 테이블, 거래시간 테이블 및 이원시간 테이블 구조를 지원한다. 이때 거래시간은 시간 특성상 항상 간격의 형태를 유지하며, 유효시간의 경우는 시점과 간격의 두 가지 유효시간 구조를 선택할 수 있다. 그림 4(a)의 경우는 정보의 변화에 대하여 중복 저장 비율이 높으며 처리 계산이 비교적 복잡한 문제점을 가진다. 그림 4(b)의 경우는 구조가 비교적 간단하며, 정보의 변화가 발생한 경우 부분적인 처리가 가능하다는 장점을 갖지만 빈번한 조인 연산이 수행되기 때문에 처리비용이 큰 문제점을 갖는다. 더욱이 외래 키(foreign key)에 의한 탐색 알고리즘이 사용되는 경우에는 검색 속도가 매우 느린 문제점을 갖는다. 따라서 그림 4(a)의 구조를 기반으로 튜플간 차분에 의한 저장 방식(difference store technique)을 적용한 경우가 가장 효율적인 시공간 테이블 구조가 되며, 다음 장에서는 이 개념을 기반으로 하는 시공간 질의 언어를 설계한다.



(그림 4) 시공간 테이블 구조
(Fig. 4) Spatiotemporal Table Structure

4. 질의 언어 설계

이 장에서는 3장에서 제안한 시공간 데이터 모델용 데이터베이스에서 처리하기 위하여 SQL을 기반으로 기존의 공간 데이터베이스에서 정의된 공간 속성 및 연산과 시간지원 데이터베이스 질의언어인 TSQL2에서 정의된 이력 표현 및 연산을 추가한 관계형 시공간 데이터베이스 질의언어의 설계에 대하여 설명한다.

기존의 질의언어와 마찬가지로 시공간 질의 언어는 크게 시공간 자료 정의 언어와 시공간 조작 언어로 구분할 수 있다. 시공간 자료 정의 언어(spatiotemporal data definition language : STDDL)에는 시공간 테이블 생성문, 색인 및 뷰 정의문과 변경문 등이 있으며, 이들 문장은 기존의 일반 속성 뿐만 아니라 시간 속성 및 공간 속성을 연산의 구성요소로 사용한다. 그림 5(a)에서 표현된 시공간 테이블 생성문의 구문에서 사용되는 속성 타입에는 CHAR, INTEGER, FLOAT등의 기본적인 속성 타입뿐만 아니라 POINT, LINE, LINES, TRIANGLE, RECTANGLE, CIRCLE, 그리고 POLYGON등의 공간 속성 타입이 있다. 선택적으로 사용되는 AS절에서 VALID 구문은 해당 테이블이 유효시간을 지원함을 가리키며, STATE 구문은 시점이 아닌 시작시간과 종료시간으로 구성된 간격 형태를 의미한다. 아울러 TRANSACTION 구문은 해당 테이블이 거래시간을 지원함을 의미하는데, 거래 시간의 경우 항상 간격의 형태를 갖는다. 만일 AS절이 생략되면 기존의 데이터베이스와 동일한 개념인 스냅 테이블(snapshot table)로 정의된다.

시공간 자료 조작 언어(spatiotemporal data manipulation language)에는 객체에 대한 삽입, 삭제, 변경 및 검색문 등이 있다. 이들 문장들은 기존의 일반 조건뿐만 아니라 시간 연산자와 공간 연산자를 조건절에서 사용할 수 있으며, 저장될 자료에 대한 유효 시간 정보를 표현하는 구문을 추가로 갖는다. 즉, 일반 속성을 대상으로 하는 조건절인 WHAT절과 공간 속성을 대상으로 하는 조건절인 WHERE절, 그리고 시간 속성을 대상으로 하는 조건절인 WHEN절은 시공간 자료 조작 언어에서 선택적으로 사용된다. 일반적으로 데이터베이스에서 검색문은 다른 문장의 구조의 기반이 되는 가장 기본적인 문장이다. 그림 5(b)에서 표현된 시공간 검색문(spatiotemporal select statement)은 데이터베이스에 저장된 객체에 대하여 공간 조건과 시

간 조건을 동시에 지정함으로써 과거로부터 현재에 이르는 객체에 대한 공간 정보를 사용자에게 제공한다. 시공간 검색문의 목적질에는 일반 속성과 공간 속성이 사용되며, FROM절에서는 지정된 테이블에 대한 별칭을 사용함으로써 2개 이상의 테이블로 구성된 복잡한 질의 구조를 보다 간단하게 표현해준다. 그림 5(c)에서 표현된 시공간 삽입문(spatiotemporal insert statement)의 구문은 기존의 데이터베이스 질의어와 달리 공간 속성에 대한 이력 정보를 삽입하기 때문에 다소 복잡한 구조를 갖는다. 아울러 VALUE절은 삽입될 자료의 일반 속성 및 공간 속성 값을 지정하는 기능을 제공하며, VALID절은 해당 객체의 자료에 대한 유효 시간을 표현하기 위하여 사용된다. 그리고 그림 5(d)에서 표현된 시공간 삭제문(spatiotemporal delete statement)에서 VALID절은 삭제될 시공간 객체의 유효시간을 표현하며, 시간지원 연산의 특성상 거래 시간은 연산 대상에서 제외되는 중요한 특징이 있다. 마지막으로 그림 5(e)에서 표현된 시공간 변경 구문에서 SET 절에는 일반 속성 값과 공간 속성 값을 포함하는 변경될 새로운 속성 값들이 기술되며, VALID절에서 기술되는 유효시간 값은 삭제될 자료의 유효시간 값과 새로이 추가될 자료의 유효시간 값을 의미한다.

```
CREATE TABLE table_name
    ( attribute_name attribute_type
      [, attribute_name attribute_type] )
[AS [VALID {STATE | EVENT}]
 [AND] [TRANSACTION]]
```

(a) 시공간 테이블 생성문의 문법

```
SELECT attribute_name [, attribute_name]*
FROM table_name [AS alias_name]
    [, table_name [AS alias_name]]
[VALID temporal_expression]
[WHAT general_predicate]
[WHERE spatial_expression]
[WHEN temporal_expression]
```

(b) 시공간 검색문의 문법

```
INSERT INTO table_name
    ( attribute_name [, attribute_name] )
VALUES ( attribute_value [, attribute_value] )
[VALID temporal_expression]
[WHAT general_predicate]
[WHERE spatial_expression]
[WHEN temporal_expression]
```

(c) 시공간 삽입문의 문법

```
DELETE FROM table_name
[VALID temporal_expression]
[WHAT general_predicate]
[WHERE spatial_expression]
[WHEN temporal_expression]
    (d) 시공간 삭제문의 문법
```

```
UPDATE table_name
SET ( attribute_name TO attribute_value
    [, attribute_name TO attribute_value]* )
[VALID temporal_expression]
[WHAT general_predicate]
[WHERE spatial_expression]
[WHEN temporal_expression]
    (e) 시공간 변경문의 문법
```

(그림 5) 시공간 질의 구문
(Fig. 5) Spatiotemporal query syntax

5. 언어 연산의 수행

이 장에서는 4장에서 설계된 시공간 테이블 생성, 시공간 검색, 시공간 삽입, 시공간 삭제 및 시공간 변경과 같은 시공간 연산의 수행을 설명한다.

5.1 시공간 테이블 생성문

[질의 1]에서는 공간 속성을 포함하는 이원시간 테이블을 생성하는 질의 예이며, 그림 6과 같은 이원시간 테이블이 생성된다.

[질의 1] "건물(이름, 소유자, 가격, 주소, 위치, 모양)의 스키마를 갖는 이원시간 테이블 building을 생성하라."

```
CREATE TABLE building
    (name CHAR, owner CHAR, price INT,
    address CHAR, location POINT, shape POLYGON)
AS VALID STATE AND TRANSACTION
```

[질의 1]에서 location 속성은 POINT 타입의 공간 속성이며, shape 속성은 POLYGON 타입의 공간 속성이다. 아울러 그림 6에서와 같이 AS절에서 유효 시간과 거래 시간을 동시에 지정하였으므로 암시적으로 유효 시간과 거래 시간을 저장할 속성이 생성되며 튜플 타임스탬핑(tuple timestamping) 구조를 갖는다.

일반속성

Name (CHAR)	Owner (CHAR)	Price (INT)	Address (CHAR)		
Location (POINT)	Shape (POLYGON)	VF	VT	TS	TE

공간속성

시간속성

(그림 6) 건물 테이블 스키마
(Fig. 6) Building Table Schema

5.2 시공간 검색문

[질의 2]에서는 시간 조건과 공간 조건으로 구성된 시공간 검색문을 통하여 과거로부터 현재에 이르는 공간 객체에 대한 이력을 제공한다.

[질의 2] 건물 테이블로부터 "1986년 3월 1일 이전에 준공된 50 m²이상의 규모를 갖는 건물의 이름을 출력하라."

```
SELECT name
FROM building AS B
WHEN BEGIN(VALID(B)) PRECEDE
    TIMESTAMP 1986-03-01
WHERE AREA(B.shape) > 50 squaremeter
```

즉, 유효시간이 1995년 3월 1일 이전에 시작된 공간 객체의 면적이 50 m²인 건물을 검색하여 건물의 이름과 준공된 날짜를 출력한 예이며, 그림 7(a)의 테이블이 주어졌을 때 수행한 결과는 그림 7(b)와 같다. 그림 7(b)에서 일반 속성은 지정된 속성만이 출력되며, 공간 속성과 유효시간 속성은 자동적으로 추출되고 계산된 후 출력된다. 그림 7(a)의 테이블에서 속성 shape의 값은 좌표값을 의미하는 벡터 정보로서 1간격을 1m로 간주하여 처리함을 가정하였다. 또한 그림 7(b)에서 출력된 질의결과를 살펴보면, 주어진 테이블에 대하여 거래 종료 시간이 UC(Until Changed)이며 유효시작 시간이 지정된 타임스탬프보다 적어도 같거나 작아야 함을 알 수 있다. 이때 거래 종료 시간이 UC 아닌 튜플들은 테이블에서 논리적으로 삭제되었음을 가리키는 데, 질의 시점이 거래종료 시간보다 항상 크기 때문에 검색되지 못한다. 이를 검색하기 위해서는 질의에서 반드시 TRANSACTION 구문을 사용하여 지정하여야 하며, 이를 가리켜 롤백 연산(rollback operation)이라 한다.

name	owner	price	address	location	shape	VF	VT	TS	TE
우리서점	홍길동	10,000	개신동 25	(100,100)	((90,90), (90,110), (110,110), (110,90))	80/07/01	Forever	80/07/01	UC
밤샘야식	변하도	5,000	개신동 15	(50,50)	((40,40), (40,60), (60,60), (60,40))	80/07/01	Forever	80/07/01	85/03/03
푸짐야식	윤지혜	7,000				80/07/01	Forever	85/03/03	UC
영재서점	홍길동	7,500	개신동 35	(200,200)	((190,190), (190,210), (210,210), (210,190))	87/01/01	Forever	87/01/01	87/01/01
			개신동 55	(300,300)	((290,290), (290,310), (310,310), (310,290))	87/01/01	Forever	87/01/01	88/10/10
						87/01/01	Forever	88/10/10	UC
						88/10/10	Forever	88/10/10	UC

(a) Building 테이블

name	location	shape	VF	VT
우리서점	(100,100)	((90,90), (90,110), (110,110), (110,90))	80/07/01	Forever
밤샘야식	(50,50)	((40,40), (40,60), (60,60), (60,40))	80/07/01	85/03/03
푸짐야식	(50,50)	((40,40), (40,60), (60,60), (60,40))	85/03/01	Forever

(b) 시공간 검색질의 수행 결과

(그림 7) 시공간 검색 문장의 실행 예
(Fig. 7) Example of Spatiotemporal Select Statement

5.3 시공간 삽입문

[질의 3]에서는 그림 7(a)를 대상으로 공간 속성을 포함하는 객체에 대한 이력 정보를 데이터베이스에 삽입하는 간단한 구문을 보여준다. [질의 3]에서 VALUES 구의 공간 속성 값은 POLYGON타입의 일련의 순서쌍으로 구성된 값으로 할당된다. 만일 LINE타입인 경우에는 두 개의 순서쌍으로 구성된 값이 지정되며, POINT 타입인 경우에는 하나의 순서쌍으로 구성된 공간 좌표 값이 사용되었으며, 질의 수행 결과는 그림 8과 같다.

[질의 3] "1990년 3월 1일 하늘이 소유의 대학서점이 12,000킬원의 비용으로 개신동 50번지의 위치에 준공되었다."

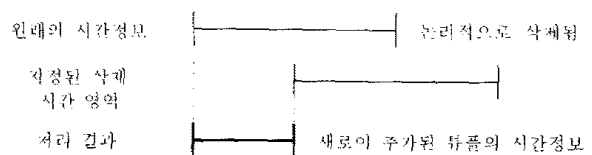
```
INSERT INTO building
VALUES ('대학서점', '하늘이', 12000, '개신동 50',
(70,70), ((55,55),(55,85),(85,85),(85,55)))
VALID PERIOD (1990-03-01, NOW)
```

name	owner	price	address	location	shape	VF	VT	TS	TE
우리서점	홍길동	10,000	개신동 25	(100,100)	((90,90), (90,110), (110,110), (110,90))	80/07/01	Forever	80/07/01	UC
밤샘야식	변하도	5,000	개신동 15	(50,50)	((40,40), (40,60), (60,60), (60,40))	80/07/01	Forever	80/07/01	85/03/03
푸짐야식	윤지혜	7,000				80/07/01	Forever	85/03/03	UC
영재서점	홍길동	7,500	개신동 35	(200,200)	((190,190), (190,210), (210,210), (210,190))	87/01/01	Forever	87/01/01	87/01/01
			개신동 55	(300,300)	((290,290), (290,310), (310,310), (310,290))	87/01/01	Forever	87/01/01	88/10/10
						87/01/01	Forever	88/10/10	UC
대학서점	하늘이	12,000	개신동 50	(70,70)	((55,55), (55,85), (85,85), (85,55))	90/03/01	Forever	90/03/01	UC

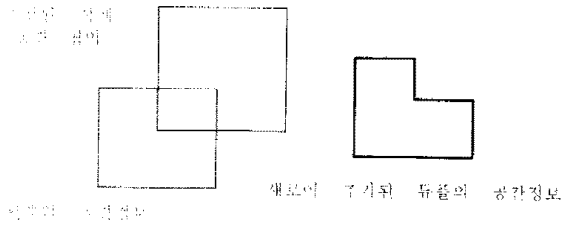
(그림 8) 시공간 삽입질의 실행 예
(Fig. 8) Example of Spatiotemporal Insert Statement

5.4 시공간 삭제문

시공간 데이터베이스에 저장된 객체에 대하여 삭제되는 정보는 객체 전체, 또는 객체의 일부 시간값이었다. 그림 9와 같이 일반적으로 하나의 시공간 객체에 대하여 삭제문이 수행될 경우 하나 이상의 새로운 객체가 생성되는 특징을 갖는데, 이것은 시간시원 연산의 내부적인 특성인 물리적인 삭제 연산의 논리적인 삽입 연산의 대체로부터 기인한다.



(a) 시간정보에 대한 삭제 연산



(b) 공간 정보에 대한 삭제 연산

(그림 9) 시공간 삭제 연산
(Fig. 9) Spatiotemporal Delete Operation

[질의 4]와 같이 데이터베이스에 저장된 객체에 대한 유효시간 속성에 의한 삭제는 그림 9와 같이 새로운 객체를 생성하는 연산으로 대체되는데, 1990년 3월 1일부터 NOW까지의 유효시간을 갖는 객체는 논리적으로 삭제되며, 1990년 3월 1일부터 1996년 2월 29일까지의 유효시간을 갖는 새로운 객체가 데이터베이스에 생성된다. 이후 1990년 3월 1일부터 NOW까지의 유효시간을 갖는 객체는 일반적인 검색문에 의해서는 검색되지 않으며, 앞서 설명한 바와 같이 TRANSACTION 구문을 적용한 특별한 경우에만 검색할 수 있다.

[질의 4] "1996년 3월 1일자로 홍길동이 소유한 "우리서점"의 전체 지분을 매각하였다."

```
DELETE
VALID PERIOD (1996-03-01, NOW)
FROM map AS M
WHAT name='우리서점' AND owner='홍길동'
WHEN VALID(M) OVERLAP
PERIOD (1996-03-01, NOW)
```

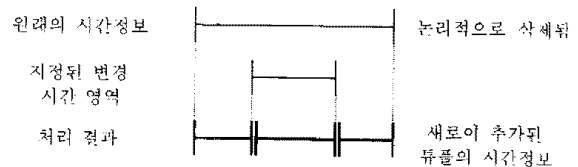
다시 말하면, 그림 10에서 첫 번째 튜플의 거래종료 시간값이 UC가 아닌 타임스탬프 '96-03-01'을 갖는 것은 해당 튜플이 논리적으로 삭제되었음을 나타내는 이력 정보이며, 마지막 튜플은 삭제 연산에 의해 새로이 추가된 현재에서 유효한 정보이다. 중요한 사항은 이력 정보와 현재 시점에서 유효한 정보의 차이로서 이력 정보는 삭제 연산에 의해 논리적으로 삭제된 정보로서 일반적인 질의 연산으로는 검색할 수 없으며, 반드시 TRANSACTION 구문을 사용하여 해당 시점으로 테이블의 상태를 이동시켜야 한다.

name	owner	price	address	location	shape	VF	VT	FS	TE
우리서점	홍길동	10,000	개신동 25	(100,100)	□(100,100,110,110,110,200,100,200,100,100)	00/03/93	Forever	90/03/91	96/03/93
말뚝아시	백희노	5,000	개신동 15	(50,50)	□(50,50,60,60,60,60,50,50)	90/03/90	Forever	88/03/90	95/03/90
말뚝아시	백희노	5,000			□(50,50,60,60,60,60,50,50)	88/03/90	95/03/93	95/03/90	11'
두식아시	임지배	7,000			□(50,50,60,60,60,60,50,50)	85/03/91	Forever	85/03/93	11'
영대서점	홍길동	7,500	개신동 35	(200,200)	□(200,200,210,210,210,300,200,300,200,200)	87/03/91	Forever	87/03/93	87/03/94
			개신동 55	(300,300)	□(300,300,310,310,310,400,300,400,300,300)	87/03/91	Forever	87/03/94	86/03/94
					□(300,300,310,310,310,400,300,400,300,300)	87/03/91	88/03/93	88/03/94	11'
영대서점	홍길동	7,500	개신동 55	(300,300)	□(300,300,310,310,310,400,300,400,300,300)	87/03/91	Forever	88/03/93	11'
말뚝아시	최노이	12,000	개신동 50	(70,70)	□(70,70,80,80,80,80,70,70)	90/03/91	Forever	90/03/91	11'
우리서점	홍길동	10,000	개신동 25	(100,100)	□(100,100,110,110,110,200,100,200,100,100)	90/03/91	96/03/91	96/03/91	11'

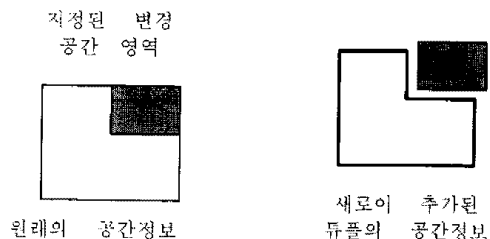
(그림 10) 시공간 삭제질의의 실행 예
(Fig. 10) Example of Spatiotemporal Delete Statement

5.5 시공간 변경문

시공간 변경문(spatiotemporal update statement)에서 객체에 대하여 변경가능한 속성은 일반 속성, 공간 속성 및 시간 속성이 된다. 또한 그림 11에서 처럼 시공간 삭제문과 유사하게 시공간 변경문은 객체의 삭제와 하나 이상의 새로운 객체의 삽입 연산으로 대체된다.



(a) 시간정보에 대한 변경 연산



(b) 공간 정보에 대한 변경 연산

(그림 11) 시공간 변경 연산
(Fig. 11) Spatiotemporal Update Operation

[질의 5]에서는 데이터베이스내의 객체에 대하여 공간 속성의 일부를 변경하는 예로서, 그림 12와 같이

두 개의 새로운 튜플이 삽입된다. 첫 번째 튜플은 변경 대상이 되는 튜플이 논리적으로 삭제되어 해당 튜플을 수정한 내용으로서 1990년 3월 1일부터 1996년 12월 31일까지는 이전 정보를 포함하며, 두 번째 튜플은 삽입 연산에 의한 결과로서 1997년 1월 1일부터 NOW까지 새로이 정의된 공간 속성을 포함하는 새로운 객체가 삽입된다.

[질의 5] “1997년 1월 1일자로 홍길동이 소유한 영채서당의 일부 지분이 반노올에게 양도되었다.”

```
UPDATE building
SET owner TO '반노올' AND shape TO
POLYGON((190,190),(190,300),(300,300),(300,190))
VALID PERIOD (1997-01-01, NOW)
WHERE name='영채서당' AND owner='홍길동'
```

name	owner	price	address	location	shape	VF	VT	TS	TE
우리서당	홍길동	10,000	개신동 25	(100,100)	POLYGON((100,100),(100,200),(200,200),(200,100))	86/01/01	Forever	86/01/01	86/01/01
명문서당	반노올	5,000	개신동 15	(150,50)	POLYGON((150,50),(150,150),(250,150),(250,50))	86/01/01	Forever	86/01/01	86/01/01
밤샘서당	반노올	5,000			POLYGON((150,50),(150,150),(250,150),(250,50))	86/01/01	86/01/01	86/01/01	86/01/01
우리서당	반노올	7,000			POLYGON((150,50),(150,150),(250,150),(250,50))	86/01/01	Forever	86/01/01	86/01/01
영채서당	홍길동	7,500	개신동 35	(200,200)	POLYGON((200,200),(200,300),(300,300),(300,200))	87/01/01	Forever	87/01/01	87/01/01
			개신동 55	(300,200)	POLYGON((300,200),(300,300),(400,300),(400,200))	87/01/01	86/01/01	87/01/01	86/01/01
					POLYGON((100,100),(100,200),(200,200),(200,100))	86/01/01	86/01/01	86/01/01	86/01/01
					POLYGON((150,50),(150,150),(250,150),(250,50))	86/01/01	Forever	86/01/01	86/01/01
					POLYGON((150,50),(150,150),(250,150),(250,50))	87/01/01	Forever	87/01/01	87/01/01
	반노올				POLYGON((150,50),(150,150),(250,150),(250,50))	87/01/01	Forever	87/01/01	87/01/01
대형서당	채노이	12,000	개신동 50	(170,70)	POLYGON((170,70),(170,270),(270,270),(270,70))	86/01/01	Forever	86/01/01	86/01/01

(그림 12) 시공간 변경질의의 실행 예
(Fig. 12) Example of Spatiotemporal Update Statement

5.6 시공간 조인

이 논문에서 제안된 관계형 시공간 데이터베이스에서 조인 연산은 주제(theme)에 의해 개별적인 레이어(layer)에 저장된 공간 정보를 각각 하나의 테이블로 사상한 후 레이어간 정보 결합에 의해 새로운 정보를 생성할 때 발생한다. 즉, 빌딩 레이어와 통신선 레이어간의 결합을 통해 지정된 건물로부터 가장 근접한 맨

홀(manhole)을 검색하는 경우에는 빌딩 테이블의 인의의 튜플에 대하여 통신선 레이어의 모든 튜플과 시간 정보 및 공간 정보에 의한 조인 연산이 이루어져야 한다. 기존의 조인 연산과 마찬가지로 시공간 조인은 데이터베이스 연산에서 매우 비용이 큰 중요한 연산이다. [질의 6]에서는 시공간 조인 연산을 포함하는 예를 보여주고 있으며, 그림 13(b)에서는 그림 6(a)의 건물 테이블과 그림 13(a)의 통신선 테이블간 시공간 조인의 결과를 보여준다.

[질의 6] 건물 테이블과 통신선 테이블로부터 “1986년 3월 1일 이전에 준공된 50 m²이상의 규모를 갖는 건물을 관통하는 선로부터 1km이내의 맨홀 중, 지금부터 1년 이전 준공한 맨홀을 검색하시오.”

```
SELECT C.manhole
FROM building AS B, cable AS C
WHEN BEGIN(VALID(B)) PRECEDE TIMESTAMP
1986-03-01 AND BEGIN(VALID(C))
PRECEDE BIND(NOW - 1 Year)
WHERE AREA(B.shape) > 50 squaremeter AND
B.shape CROSS C.section AND
DISTANCE(C.section, C.manhole) <=
1 kilometer
```

name	manhole	section	VF	VT	TS	TE
개신선 -1	(120,80)	POLYGON((100,100),(100,200),(200,200),(200,100))	86/01/01	Forever	86/01/01	86/01/01
			86/01/01	85/12/01	86/01/01	86/01/01
			86/01/01	Forever	86/01/01	86/01/01
개신선 -2	(40,275)	POLYGON((150,50),(150,150),(250,150),(250,50))	87/01/01	Forever	87/01/01	87/01/01

(a) 통신선 테이블

manhole	VF	VT
(120,80)	86/01/01	Forever

(b) 시공간 조인 결과

(그림 13) 시공간 조인 연산의 실행 예
(Fig. 13) Example of Spatiotemporal Join Operation

6. 기존 언어와의 동질성과 시공간 상수

이 장에서는 설계된 질의언어와 기존의 질의언어 사이의 동질성을 위한 고려사항을 설명하고, 시공간 데이터베이스 질의 연산에서 필요한 특수한 시공간 상수의 처리와 조건절에 대한 기정의값(default value)의 처리를 설명한다.

6.1 기존 언어와의 동질성

이 논문에서 제안된 STQL은 기존의 SQL보다 상위의 언어로서, 일반 정보뿐만 아니라 공간 정보 및 시간 정보를 효율적으로 기술하기 위해 기존의 데이터베이스 질의언어에서 정의된 구문을 변경 및 확장하였다. 기존의 질의언어에서는 일반조건만을 대상으로 처리하므로 WHERE절만이 사용되었다. 하지만 제안된 STQL에서는 일반 조건, 공간 조건 및 시간 조건 모두를 조건절에서 사용할 수도 있어야 하므로, 각각의 기능에 따라 조건절을 세분화하였다. 즉, 연산 대상이 되는 테이블을 지정하는 FROM절, 데이터베이스에 저장된 자료의 유효시간 값을 지정하는 VALID절, 검색 대상에 대한 일반 조건을 기술하는 WHAT절, 공간 조건을 기술하는 WHERE절, 그리고 시간 조건을 기술하는 WHEN절 등으로 정의하였다. 특히 VALID절을 가리켜 시간지원 구성자 연산절(temporal constructor operation clause)로 부르며, 나머지 WHAT절, WHERE절과 WHEN절을 각각 일반 조건절(general predicate clause), 공간 조건절(spatial predicate clause), 및 시간 조건절(temporal predicate clause)로 부른다. 따라서 기존의 SQL 질의를 제안된 STQL로 변환하기 위해서는 WHERE절과 WHAT절간의 이동이 요구된다. 또한 제안된 STQL에서 사용된 WHEN절과 VALID절 및 WHERE절은 기존의 SQL로 변환시 제거되어야 한다. 따라서 제안된 시공간 질의 언어를 입력받는 질의 처리 시스템은 기존의 SQL구문과 제안된 STQL을 상호 변환하는 과정이 요구된다.

6.2 시공간 상수

시공간 데이터베이스에서 공간 객체에 대한 이력을 처리할 때 사용할 수 있는 특수한 시간 상수값에는 Now, Until Changed, 그리고 Forever등이 있다. 이들 시간 상수는 미리 정의된 고유의 값을 갖으며, 사용자로부터 하여금 질의를 작성할 때 편리함을 증대시켜주

는 장점을 갖는다. 먼저 Now값은 유효시간에 대하여 적용되며 시스템에서 두 가지 유형으로 인식된다. 첫째는 절대 상수로서 해당 객체의 유효시간을 항상 유효한 값으로 인식토록 해주며, 둘째로 상대 상수로 사용되는 경우는 질의 시점의 시간이 사용된다. 이들 유형들을 서로 구별하기 위해 BIND()연산자를 사용하며, 특별히 지정하지 않는 경우는 절대 상수로 인식된다. 그리고 Forever는 유효시간에 대하여 미래시간에도 유효함을 나타낼 때 사용된다. 또한 Until Changed는 질의에서 UC로 기술되며 특별히 거래시간에 대하여 적용되며 해당 객체의 변경이 발생할 때까지 유효함을 의미한다. 시간 상수와 유사하게 공간 상수로 정의된 값에는 Here와 There가 있으며, 이들 값은 공간 좌표상의 값을 기준으로 해석된다. 하지만 시간 상수와 달리 이들 값은 SET문장을 통해 사용자 질의에서 미리 정의되어야 하며, 미리 정의가 되지 않는 경우에 각각은 해당 스넵의 최하위/상위 공간좌표(minCoordX, minCoordY/ maxCoordX, maxCoordY)로 정의될 수 있다.

또한 시공간 자료 조작 언어에서 사용되는 WHAT절, WHEN절 및 WHERE절은 일반적으로 모든 질의 상에서 반드시 기술되어야 하지만 특수한 경우 생략될 수 있다. 이러한 경우에 적용되는 값이 다음과 같이 정의된다. 먼저 WHAT절의 경우 튜플의 일반 속성에 적용되는 조건이 항상 참으로 적용되어 데이터베이스 내 모든 튜플이 처리 대상이 된다. 그리고 WHEN절의 경우는 현재 시간과 중첩되어지는 튜플만을 처리 대상으로 한정한다. 즉, 테이블 M에 대한 WHEN절의 기정값은 WHEN TIMESTAMP BIND(NOW) OVERLAP VALID(M) AND TIMESTAMP BIND(NOW) OVERLAP TRANSACTION(M)이 된다. WHEN절의 기정값은 기존의 공간 데이터베이스에서와 동일하게 이력 정보를 제외한 오직 현재시점에서 유효한 객체만을 대상으로 한정한다. 그리고 WHERE절의 기정값은 WHAT과 동일하게 모든 튜플에 대한 공간 정보 조건이 항상 참으로 간주하여 처리한다.

7. 결 론

시공간 데이터베이스는 현실세계에 존재하는 다양하고 복잡한 객체에 대하여 효율적인 공간정보와 시간의 흐름에 따라 변화하는 이력을 동시에 효율적으로

관리함으로써 광범위한 응용분야에서 사용될 수 있다. 최근 시공간 데이터베이스에 관한 관심이 증가하면서 주로 선진국을 중심으로 시공간 데이터베이스 연산 및 모델에 대한 연구들이 진행되고 있지만, 이력정보와 공간정보를 동시에 효율적으로 관리하기 위한 시공간 질의 언어의 연구는 미비한 실정이다.

따라서 이 논문에서는 시공간 객체를 데이터베이스화하기 위한 모델을 제안하였으며, 관계형 데이터베이스 질의 언어인 SQL을 기반으로 이력 정보와 공간 관리 기능을 첨가한 시공간 데이터베이스 질의언어인 STQL를 제안하였다. 세부적으로 시공간 테이블 생성문과 시공간 데이터 검색, 삽입, 삭제, 변경문 등의 구문과 연산의 수행 예를 보였으며, 제안된 STQL은 기존의 질의언어의 확장으로서 STQL과 기존의 SQL과의 동질성을 위한 고려사항을 살펴보았다. 앞으로 이 논문에서 제안된 STQL을 효율적으로 처리하는 시공간 질의 처리 시스템의 설계 및 구현에 관한 연구가 이루어질 예정이다.

참 고 문 헌

[1] K. Al-Taha, R. Snodgrass, and M. Soo, Bibliography on Spatiotemporal Databases, ACM SIGMOD RECORD, Vol.22, No.1, Mar., 1993.

[2] J. Allen, Maintaining Knowledge about Temporal Intervals, Communications of the ACM, Vol.26, No.11, Nov., 1983.

[3] C. Claramunt, Managing Time in GIS : An Event-Oriented Approach, Proceedings of the International Workshop on Temporal Databases, Sep., 1995.

[4] M. Egenhofer, Spatial SQL : A Query and presentation Language, IEEE Transaction on Knowledge and Data Engineering, Vol.6, No.1, Feb., 1994.

[5] ESRI, Introduction to SDETM, ESRI Inc., 1996.

[6] O. Guenther, and A. Buchmann, Research Issues in Spatial Databases, ACM SIGMOD Record, Vol.19, No.4, Dec., 1990.

[7] D.H. Kim, I.H. Lee, and K.H. Ryu, The Design and Implementation of Aggregate Function of Temporal Databases in Main Memory, Journal of The KISS, Vol.21, No.8, Aug., 1994.

[8] D.H. Kim, K.W. Nam, K.H. Ryu, The Spatio-temporal Database Model, Proceedings of The 23rd KISS Fall Conference, Vol.23, No.2, Oct., 1996.

[9] D.H. Kim, and K.H. Ryu, A Study on the Spatiotemporal Aggregate Functions, Proceedings of The KISS SIGDB '97 Winter Conference, Feb. 1997.

[10] G. Langran, Time in Geographic Information Systems, Taylor & Francis Inc., 1990.

[11] K.W. Nam, D.H. Kim, and K.H. Ryu, The Spatio-temporal Realltionship Operator, ITC-CSCC '96, International Technical Conference on Circuits/ Systems, Computers, and Communications, Jul., 1996.

[12] D. Peuquet, A Framework for The Representation of Spatiotemporal Processes in Geographic Information Systems, Proceedings of the International Workshop on an Infreastructure for Temporal Databases, Jun., 1993.

[13] D. Peuquet, and N. Duan, An Event-based Spatiotemporal Data Model(ESTDM) for Temporal Analysis of Geographical Data, International Journal Geographical Information Systems, Vol.9, No.1, 1995.

[14] N. Roussopoulos, C. Faloutsos, and T. Sellis, An Efficient Pictorial Database System for PSQL, IEEE Transaction on Software Engineering, Vol. 14, No.5, May, 1988.

[15] R. Snodgrass, The Temporal Query Language TQuel, ACM TODS, Vol.12, No.2, Jun. 1987.

[16] R. Snodgrass, The TSQL2 Temporal Query Language, Kluwer Academic Publishers, 1995.

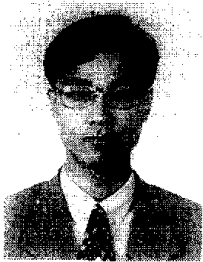
[17] P. Svensson, and Z. Huang, Geo SAL : A Query Language for Spatial Data Analysis, Advances in Spatial Databases, SSD '91, Aug., 1991.

[18] A. Tansel, and M. Arkun, HQuel : A Query Language for Historical Databases, In Proceedings of the 3rd International Workshop on Statistical and Scientific Databases, Jul., 1986.

[19] V. Tsotras, and A. Kumar, Temporal Database Bibliography Update, ACM SIGMOD RECORD,

Vol.25, No.1, Mar., 1996.

[20] M. Worboys, A Unified Model for Spatial and Temporal Information, The Computer Journal, Vol.37, No.1, 1994.



김 동 호

1993년 충북대학교 전자계산학과 졸업(이학사)
1995년 충북대학교 대학원 전자계산학과 졸업(이학석사)
1995년~현재 충북대학교 대학원 전자계산학과 박사수료

1997년 한국전자통신연구원 위촉연구원

관심분야 : 시간지원 데이터베이스, 시공간 데이터베이스, 멀티미디어 시스템, 객체지향 데이터베이스, 루팅제어 등



류 근 호

1976년 숭실대학교 전산학과 졸업(이학사)
1980년 연세대학교 산업대학원 전산전공(공학석사)
1988년 연세대학교 대학원 전산전공(공학박사)

1976년~1986년 육군 군수 지원사 전산실(ROTC장교), 한국전자통신연구소(연구원), 한국방송통신대 전산학과(조교수) 근무

1989년~1991년 Univ. of Arizona Research Staff(Temporal DB) 연구원

1986년~현재 충북대학교 컴퓨터과학과 교수겸 컴퓨터 정보통신 연구소 소장

관심분야 : 시간지원 데이터베이스, 시공간 데이터베이스, DBMS와 정보검색, 객체 및 지식베이스 시스템 등