

# 시공간 질의 처리 시스템의 설계 및 구현

이 성 종<sup>†</sup> · 김 동 호<sup>††</sup> · 류 근 호<sup>†††</sup>

## 요 약

시공간 데이터베이스는 현실세계에 존재하는 다양한 유형의 객체에 대한 공간 관리와 이력 정보를 동시에 제공함으로써 지리정보시스템, 도시계획시스템, 자동주행시스템 등과 같이 광범위한 응용에서 사용된다. 하지만 기존의 데이터베이스에서 사용된 질의 언어만으로는 공간 객체를 대상으로 효율적인 이력 연산을 질의에서 표현하기 어려운 문제점을 갖는다. 또한 질의 처리를 위한 비용의 관점에서 기존의 질의 표현 자체에 포함된 구문적 한계로 인해 효율성이 저하되는 문제점이 있다. 따라서 이 논문에서는 관계형 데이터베이스에서 가장 널리 사용되는 언어인 SQL을 기반으로 확장된 시공간 데이터베이스 질의 언어인 STQL을 소개한다. 아울러 STQL로 작성된 질의를 입력받아 주기억장치에서 처리하는 시공간 질의 처리 시스템을 구현하고 평가한다.

## Design and Implementation of Spatiotemporal Query Processing Systems

SeongJong Lee<sup>†</sup> · DongHo Kim<sup>††</sup> · KeunHo Ryu<sup>†††</sup>

## ABSTRACT

The spatiotemporal databases support a historical informations as well as spatial managements for various kinds of objects in the real world, and can be efficiently used in many applications such as geographic information system, urban plan system, car navigation system. However it is difficult to represent efficiently historical operations with conventional database query language for spatial objects. In terms of cost for query processing, it also degenerates performance of query processing because of syntactic limitations which is innate in conventional query representation. So in this paper, we introduce a new query language, entitled as STQL, which has been extended on the basis of the most popular relational database query language SQL. And we implement as well as evaluate a spatiotemporal query processing system that get a query written by STQL and then process it in a main memory.

### 1. 서 론

시공간 데이터베이스(spatiotemporal databases)는 현실세계에 존재하는 다양하고 복잡한 객체에 대하여

공간 관리(spatial management)와 이력 정보(historical information)를 동시에 제공함으로써, 지리정보시스템, 도시관리시스템, 환경관리시스템, 자동주행시스템 등과 같이 광범위한 응용분야에서 적용된다[8,13].

지난 30여 년간 시간지원 데이터베이스(temporal databases)[22] 및 공간 데이터베이스(spatial databases)[6]와 관련된 연구가 수행되었다. 하지만 이들 연구는 서로를 고려하지 않고 각자 독립적인 영역으로 진행되어왔기 때문에 이들을 직접적으로 시공간 데이

※ 이 연구는 한국통신의 '99년 정보통신 기초연구비 지원과 정보통신부의 정보통신 우수시범학교 지원 사업에 의하여 수행된 것입니다.  
† 준 회원 : 충북대학교 대학원 전자계산학과  
†† 정 회원 : 한국전자통신연구원 선임연구원  
††† 종신회원 : 충북대학교 컴퓨터학과 교수  
논문접수: 1998년 10월 1일, 심사완료: 1999년 2월 2일

터베이스를 위한 응용에 적용하기 어려운 문제점을 갖는다[8]. 최근 시간지원 데이터베이스와 공간 데이터베이스의 연구가 성숙단계로 접어들면서 시공간 데이터베이스에 관한 관심이 급속도로 증가하였으며, 주로 선진국에서 시공간 데이터베이스 연산 및 모델에 대한 연구들을 중심으로 하는 초창기 연구가 진행되었다. 그러나 이들 연구도 개념적인 관점에서 공간 객체가 시간의 흐름에 따라 변화하는 정적인 구조만을 다루었을 뿐, 실질적인 데이터베이스 관점에서 이력 정보와 공간 관리를 동시에 효율적으로 관리하기 위한 연구는 미비한 실정이다. 더구나 기존의 데이터베이스에서 사용된 질의 언어를 사용하여 객체를 대상으로 하는 이력 정보와 공간 연산을 질의에서 표현하기 어려울 뿐만 아니라 처리를 위한 비용의 관점에서 효율성이 저하되는 문제점이 있다.

따라서 이 논문에서는 기존의 관계형 데이터베이스의 표준 질의 언어인 SQL을 기반으로 이력 정보와 공간 관리 기능을 부가한 시공간 데이터베이스 질의 언어인 STQL을 설계하고, 이를 효율적으로 처리하기 위한 방안으로서 STQL로 작성된 질의를 처리하는 시공간 질의 처리 시스템을 구현한다.

이 논문의 구성은 다음과 같다. 2장에서는 시간지원 데이터베이스 질의 언어와 공간 데이터베이스 질의 언어들의 특징과 시공간 데이터베이스의 관련된 이전 연구들을 정리한다. 3장에서는 이원시간을 지원하는 삼차원 시공간 데이터 모델과 관계형 시공간 질의 언어인 STQL을 설명하며, 4장에서는 효율적인 시공간 질의 처리 방안을 기술하고, 5장에서는 이를 바탕으로 구현된 시공간 질의 처리 시스템의 구조와 세부 구성 요소들에 대하여 설명한다. 6장에서는 제안된 시공간 질의 처리 시스템에서의 간단한 질의 처리 예를 보이며, 마지막으로 7장에서는 이 연구의 요약 및 앞으로의 연구 과제를 제시한다.

## 2. 관련 연구

시공간 데이터베이스와 관련된 연구에는 대표적으로 시간지원 데이터베이스 연구와 공간 데이터베이스 연구로 나누어 진다. 시간지원 데이터베이스 질의 언어에는 대표적으로 TQuel[17], TSQL2[18], HQuel[21] 등이 있다. TQuel은 관계 해석(relational calculus)을 기반으로 하는 Quel에 대하여 시간차원으로 확장한 언

어로서 구문상의 명료함과 시간 연산에 대한 수학적 증명용의 용이함의 장점을 갖는 언어이며, TSQL2는 시간지원 데이터베이스 표준 질의어로 목표를 두어 연구된 언어로서 이전의 질의언어들이 제공하는 대부분의 기능을 제공하고 이들과 호환되는 장점을 갖는다. 하지만 이들 질의 언어는 다양한 유형의 이력연산을 제공할 수 있지만 기존의 데이터베이스에서 정의된 타입만을 대상으로 이력을 지원하기 때문에 공간 속성을 처리하기 위한 확장 방안이 매우 미비한 문제점을 갖는다. 그리고 공간 데이터베이스 질의 언어에는 Spatial SQL[4], PSQL[16], GeoSAL[20] 등이 있지만, 이들 질의 언어는 오직 현재 시점에서 유효한 공간 정보만을 대상으로 연산이 이루어지기 때문에 이력을 제공하기 위한 질의 작성성이 매우 어려운 문제점을 갖는다.

지금까지 시공간 데이터베이스와 관련된 연구들은 추상적으로 데이터 모델에 치중하여 진행되었다. 세부적으로 시간지원 데이터베이스와 공간 데이터베이스에 관련된 연구조사[1], 시공간 데이터베이스 모델링[3,8,12,14,23] 및 시공간 연산자[13]등에 대해 연구되었다. 개념적인 시공간 데이터베이스 모델링에는 시간분할 스냅모델(time-slice snapshot model)[12], 기본상태와 개정모델(base state with amendments model)[14], 공간-시간 복합 모델(space-time composite model) [10], Tempest 삼원 모델(tempest triad model)[15] 등이 있다. 그리고 최근에는 기존의 SDE[5] 또는 고딕(Gothic)과 같은 공간 데이터베이스를 기반으로 그래픽 사용자 접속기상에서 시간 연산을 위한 간단한 메뉴를 통해 내부 함수를 호출하는 형태를 제공함으로써 초보 사용자에게는 비교적 사용이 용이한 장점을 갖는 일부 연구[10]도 수행되었다.

하지만 이들 연구는 공간 객체에 대한 이력 표현이 상당히 제한되어 다양한 유형의 시공간 질의를 표현하기가 어려울 뿐만 아니라 작성된 질의를 처리하기 위한 비용의 관점에서 하위 시스템으로 수행되는 거대한 규모의 기존의 공간 관리 시스템과 상위의 이력 처리를 위한 부가적인 시스템간의 연동으로 인해 과도한 비용과 불필요한 과정이 야기되는 문제점을 갖는다.

따라서 이 논문에서는 이전 연구를 바탕으로 현실 세계에 존재하는 객체에 대한 시공간 정보를 데이터베이스에서 관리하기 위한 방법과 기존의 SQL를 기반으로 시간지원 연산과 공간 연산을 표현할 수 있도록 확장한 STQL을 설명하며, 이를 기반으로 작성된 질의에

대한 효율적인 시공간 질의 처리 시스템을 구현한다.

### 3. 시공간 데이터 모델과 질의어

#### 3.1 데이터 모델

현실 세계에 존재하는 객체는 기존의 데이터베이스에서 정의된 일반적인 정보와 함께 이차원의 공간 정보와 일차원의 시간정보로 구성된 삼차원 형태를 갖는다. 즉, 시간 영역의 관점에서 객체에 대한 시간 정보는 해당 객체가 현실 세계에서 유효한 시간 값을 가리키는 시점(event) 또는 간격(period)의 형태의 유효 시간(valid time)으로 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 시간(transaction time)이라는 추가적인 시간 영역이 추가되어야만 정확한 이력관리가 가능하다. 이것은 시간지원 데이터베이스에서 설명되는 시간 영역의 고유한 특성에서 기인한다[17,18]. 따라서 시공간 데이터 개념 모델은 (식 1)과 같이 거래시간을 기반으로 하여 삼차원 시공간 객체 집합으로 구성되는 시공간 영역( $\lambda$ )을 표현한다.

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

(식 1)에서 선형 일차원의 거래 시간 영역(TT)에서 이차원의 공간 영역(x, y)과 선형 일차원의 유효 시간 영역(VT)으로 구성된 삼차원 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 없기 때문이다.

시공간 데이터베이스 논리 모델 중에서 관계형 모델을 기반으로 하는 구조는 하나의 테이블에는 사용자가 정의하는 일반 속성과 공간 속성 및 유효 시간 속성, 그리고 시스템에 의해 관리되는 거래 시간 속성으로 구성된다. 여기서 공간 속성은 하나 이상의 원자 값(atomic value)으로 구성됨을 가정하는데, POINT 속성 타입은 하나의 공간 좌표 쌍을 가지며, LINE 속성

타입은 시작점과 끝점을 의미하는 두 개의 공간 좌표 쌍으로 구성된다. 그리고 POLYGON 속성 타입은 연속적인 점들의 집합을 의미하는 두 개 이상의 공간 좌표 쌍으로 구성된다. 아울러 테이블에 대하여 속성 단위의 유효 시간과 튜플 단위의 거래 시간을 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. 논리적인 모델의 관점에서 사용자는 기존의 관계형 시간지원 데이터베이스에서 정의된 테이블 구조와 동일한 개념을 그대로 적용할 수 있으며, 이를 위해 논리적인 시공간 데이터베이스 모델에서는 공간 속성에 대하여 확장된 제1정규형을 제공해야 한다. 즉, 공간 속성을 구성하는 값은 하나의 공간 좌표 값에서 일련의 공간 좌표 값 범위로 정의되는데, 일반 속성과 달리 이들 값을 그 자체로서 원자 값으로 처리함을 의미한다.

주기	일반속성 (문자, 정수, 실수, 무중)	공간속성 (점, 선, 다각형)	시간속성 (유효시간, 거래시간)
----	--------------------------	---------------------	----------------------

(그림 1) 시공간 데이터베이스 논리 모델 : 관계형 구조

#### 3.2 질의어

기존의 질의언어와 마찬가지로 시공간 질의 언어(STQL)[11]는 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 속성뿐만 아니라 시간 속성 및 공간 속성을 연산의 구성요소로 사용한다.

시공간 테이블 생성문의 구분에서 사용되는 속성 타입에는 CHAR, INTEGER, FLOAT 등의 기본적인 속성 타입뿐만 아니라 POINT, LINE, LINES, TRIANGLE, RECTANGLE, CIRCLE, 그리고 POLYGON 등의 공간 속성 타입이 있다. 또한 선택적으로 AS절에서 사용되는 VALID 구분은 해당 테이블이 유효시간을 지원함을 의미하며, 특별히 STATE 구분은 시점이 아닌 시작 시간과 종료시간으로 구성된 간격 형태를 의미한다. 그리고 TRANSACTION 구분은 해당 테이블이 거

래시간을 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 공통적으로 기존의 일반 조건뿐만 아니라 시간 연산자와 공간 연산자를 조건절에서 사용할 수 있으며, 저장될 자료에 대한 유효 시간 정보를 표현하는 구문을 추가로 갖는다. 즉, 시공간 데이터베이스 질의언어에서 WHERE절은 일반 속성과 공간 속성을 대상으로 하는 조건 수식이 기술되며, WHEN절은 유효 시간 속성을 대상으로 하는 조건 수식이 선택적으로 기술된다.

시공간 삽입문(spatiotemporal insert statement)의 구문은 기존의 데이터베이스 질의어와 달리 공간 속성에 대한 이력 정보를 삽입하기 때문에 다소 복잡한 구조를 갖는다. VALUE절은 삽입될 자료의 일반 속성과 공간 속성의 실제 값을 지정하는 기능을 제공하며, VALID절은 해당 객체의 자료에 대한 유효시간을 표현하기 위하여 사용된다.

시공간 삭제문(spatiotemporal delete statement)은 튜플을 기본 단위로 처리하며 시간지원 연산의 특성상 거래 시간은 연산 대상에서 제외되는 중요한 특징이 있다. 시공간 삭제문에서 사용되는 VALID절과 WHERE는 각각 삭제될 시공간 객체의 유효 시간 속성 값과 일반 속성 및 공간 속성 값을 표현한다.

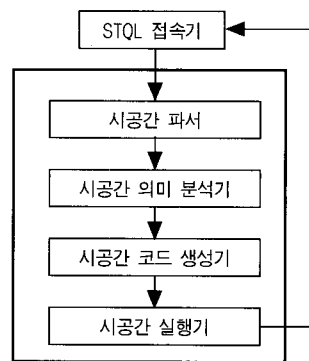
시공간 변경문(spatiotemporal update statement)에서 SET절에는 일반 속성 값과 공간 속성 값을 포함하는 변경될 새로운 속성 값들이 기술되며, WHEN절에서는 삭제될 객체가 갖는 유효 시간을 지정한다. 또한 VALID절에서 기술되는 유효시간 값은 객체의 삭제될 유효 시간 값과 동시에 새로이 추가될 객체의 유효 시간 값을 의미한다.

시공간 검색문(spatiotemporal select statement)은 데이터베이스에 저장된 객체에 대하여 공간 조건과 시간 조건을 동시에 지정함으로써 과거로부터 현재에 이르는 객체에 대한 공간 정보를 사용자에게 제공한다. 시공간 검색문의 목표 절(target clause)에는 일반 속성과 공간 속성이 사용되며, FROM절에서는 지정된 테이블에 대한 별명을 사용함으로써 2개 이상의 테이블로 구성된 복잡한 질의 구조를 보다 간단하게 표현

해준다. 그리고 WHEN절과 WHERE절을 통해 검색될 객체의 유효 시간 값과 일반 속성 및 공간 속성 값에 대한 조건을 기술한다.

#### 4. 시공간 질의 처리

이 장에서는 3장에서 설명한 시공간 데이터베이스 질의언어인 STQL로 작성된 질의가 입력되는 경우, 시공간 데이터베이스를 대상으로 효율적으로 처리하는 시공간 질의 처리 시스템에서 정의되는 데이터베이스 구조와 세부적인 처리 과정에 알고리즘을 설명한다. 시공간 질의 처리는 이전의 시간지원 데이터베이스 관리 시스템 주기억장치 프로토타입[7]을 기반으로 일반 속성뿐만 아니라 공간 속성에 대한 이력 정보를 지원하도록 확장되었다. (그림 2)에서 표현된 바와 같이, STQL로 작성된 사용자 질의가 입력되면, 시공간 파서로부터 시공간 실행기로 구성된 시공간 질의 처리 시스템에 의해 지정된 시공간 데이터베이스를 대상으로 일련의 연산이 이루어진 다음 결과가 반환된다. 즉, 기존의 객체에 대한 이력을 지원하기 위한 시간지원 질의 처리 전략에서 공간 속성을 데이터베이스에 포함하고 이를 대상으로 하는 공간 연산을 추가적으로 지원하도록 질의 처리 구조를 확장하였다.



(그림 2) 시공간 질의 처리

시공간 질의에서 공통적으로 사용되는 WHERE절과 WHEN절에서 기술되는 시공간 연산에 대하여 <표 1>과 같이 연산자와 함수의 형태로 분류되며, 이들 모두는 시스템에서 기본적인 함수로 직접 연결되거나 이들 함수의 일련의 실행 형태로 연결되어 처리된다. <표 1>에서 LENGTH는 주어진 공간 객체의 선 정보에

대하여 기하적인 길이값을 계산하여 정수값 또는 실수값을 반환하며, 사용자 질의에서는 함수의 형태인 LENGTH()로 사용됨을 의미한다. 반면 PRECEDE는 주어진 공간 객체들의 유효 시간값에 대하여 임의의 지정된 유효 시간값 보다 작은 값 또는 큰 값을 갖는 경우 각각 참 또는 거짓을 반환하며, 사용자 질의에서는 이항 연산자 형태인 VALID(o) PRECEDE PERIOD (Jan 1 1995, Dec 31 1995)로 사용됨을 의미한다. 이와 같이 시공간 연산에 대하여 개별적인 의미를 중심으로 연산자 또는 함수의 형태로 구분하여 정의되었다.

〈표 1〉 시공간 연산자 및 함수

분류	이름	반환 값의 유형	
공간	연산자	CROSS, DISJOINT, CONTACT, ENCLOSE, EAST, WEST, SOUTH, NORTH	참/거짓
	함 수	LENGTH, AREA, DISTANCE	정수/실수
시간	연산자	PRECEDE, OVERLAP, CONTAIN, ADJACENT	참/거짓
	함 수	VALID, TRANSACTION, EXTEND	시간 값

시공간 질의를 구성하는 네 가지 유형의 질의문에 대한 처리는 다음과 같다. 시공간 테이블 생성문인 'create' 명령은 데이터베이스에 테이블을 만들기 위해 사용되는 명령으로서 질의의 형태에 따라 네 가지 종류의 테이블을 만들 수 있다. 테이블의 생성은 의미분석기에서 새로 구성하고자 하는 테이블이 기존의 테이블과 중복이 되는지 여부 검사와 테이블을 구성하는 속성들이 타당한가의 검사를 한 후, 이상이 발생하지 않을 경우 의미 분석기에서 테이블을 구축하게 된다. 그러므로 테이블의 생성은 의미 분석기 단계에서 대부분의 작업이 완료된다.

시공간 데이터 삽입문인 'insert'명령은 생성된 테이블에 튜플을 추가하기 위한 명령으로 질의의 형태에 따라 한 레코드 단위로 삽입하는 방법과 이미 존재하는 다른 테이블을 통해 하나의 질의문으로 여러 개의 레코드 정보를 한꺼번에 테이블에 삽입하는 방법이 있다.

시공간 데이터 삭제문인 'delete'명령은 테이블로부터 임의의 튜플을 제거한다. 시공간 데이터베이스에서는 삭제 연산에 대하여 오직 논리적인 삭제만을 허용하기 때문에 삭제되는 튜플은 현재 버전에서 이력 버

전으로 전이가 이루어진다. 이렇게 삭제된 튜플은 논리적인 데이터의 연결이 제거되므로 제거된 자료의 검색을 위해서는 WHEN절에서 TRANSACTION() 연산을 통해 지정된 시간 범위를 만족하는 롤백 테이블(rollback table)로 이동시켜 후에 참조가 가능하다. 여기서 TRANSACTION() 연산이 수행 가능한 테이블에는 거래시간이 지원되는 거래 시간 테이블과 이원시간 테이블에서만 가능하다. 반면 스냅 테이블은 기존의 데이터베이스에서와 동일하게 논리적인 연결과 물리적인 연결이 모두 제거된다.

시공간 데이터 변경문인 'update'명령은 지정된 시간 및 공간을 만족하는 일반 속성, 공간 속성 및 유효 시간 속성의 값을 새로운 값으로 대체하는 기능을 가지며, 시공간 데이터 삽입문과 삭제문의 조합으로 구성된다. 기존의 데이터베이스에서는 조건을 만족하는 튜플을 선정하여 해당 속성 값을 지정된 값으로 변경함으로써 간단하게 변경 연산을 수행하지만, 시공간 데이터베이스에서는 논리적인 삭제만을 허용하기 때문에 변경될 튜플에 대하여 최소 하나 이상의 새로운 튜플이 추가되는 연산으로 대체된다.

시공간 데이터 검색문인 'select'명령을 입력하면 시공간 구문 분석기는 입력된 질의에 대한 구문 검사를 하여 정확한 구문일 경우 시공간 의미 분석기로 전달한다. 시공간 의미 분석기는 시공간 구문 분석기로부터 전달받은 질의를 검토하여 질의에 작성한 테이블, 속성들, 상수값 등이 시스템 스키마에 기록된 정보와 일치할 경우 시공간 코드 생성기로 입력된 질의를 전달하여 검색문에 대한 실행 트리를 생성하고 생성된 실행 트리는 시공간 실행기로 전달되어 실제 데이터베이스를 접근하여 원하는 정보를 찾게 된다.

### 5. 구 현

이 논문에서 구현된 시공간 질의 처리 시스템은 SUN Sparc 워크스테이션에서 Solaris 5.1.3 운영체제를 기반으로 IDL 도구[19]와 C언어를 사용한 X-윈도우 모티프 프로그래밍을 통해 대용량의 주기억장치를 기반으로 처리될 자료와 연산을 모두 주기억장치에 적재한 후 실행하는 주기억장치 처리 기법을 적용하였다. 자료 구조를 위해 IDL구문으로 작성된 프로그램과 Lex와 Yacc 프로그램, C언어로 작성된 코드 프로그램, 그리고 사용자 접속기를 위한 모티프 프로그램을 포함

하는 전체 원시 프로그램의 크기는 대략 20,000라인이며, 실행 파일의 크기는 대략 1.4MB이다.

5.1 자료 구조

5.1.1 데이터베이스

시공간 데이터베이스는 일반 속성 값뿐만 아니라 공간 속성 값을 저장할 수 있어야 하고 아울러 자료 값에 대하여 효율적인 이력 정보를 표현할 수 있어야 하는데, 기본적인 구조는 기존의 데이터베이스와 동일한 관계형 모델의 형태를 갖는다. 또한 시공간 데이터베이스 관리 시스템에서 생성되는 모든 유형의 데이터베이스는 반드시 내부에 스키마를 갖게 되며, 스키마 정보를 통해 입력된 질의에 대한 의미 검토를 수행하게 된다. 시공간 스키마 데이터 구조는 수행된 트랜잭션 번호와 데이터베이스 내에서 정의된 속성들로 구성되며 속성은 이름과 타입을 갖는다. 각 스키마의 속성들은 자료의 기본형인 정수, 실수, 문자열, 불린과 유효 시간 및 점, 선, 다각형 등의 공간 자료형이 되며, 각각은 데이터 값들의 최대 크기를 속성 내에 포함시킬 수 있다.

Structure	Database	From Schema Is
Database	=>	syn_transNumber : Integer;
Table	=>	syn_states : Seq Of State;
State	=>	syn_tuples : Seq Of Tuple,
		syn_transNumber : Integer;
Tuple	=>	syn_attrValues : Seq Of AttrValue,
		syn_start : Integer,
		syn_stop : Integer;
AttrValue	=>	syn_attrPtr : Attribute,
		syn_represType : RepresType,
		syn_validPeriods : Seq Of Period;
RepresType	::=	RepInteger   RepShort
		RepLong   RepFloat
		RepDouble   RepString
		RepBoolean   RepPoint
		RepLine   RepLines
		RepTriangle   RepRectangle
		RepCircle   RepPolygon;

(그림 3) 시공간 데이터베이스 구조에 대한 IDL 표현

5.1.2 질의 구문

사용자가 입력하는 시공간 질의의 구문을 저장하기 위한 IDL 자료표현은 (그림 4)와 같다. 시공간 질의는 문장의 유형에 따라서 FROM절, VALID절, WHERE절, 그리고 WHEN절 등의 요소로서 구성된다. 의미 분석기에 데이터 구조에서는 튜플, 속성과 테이블에

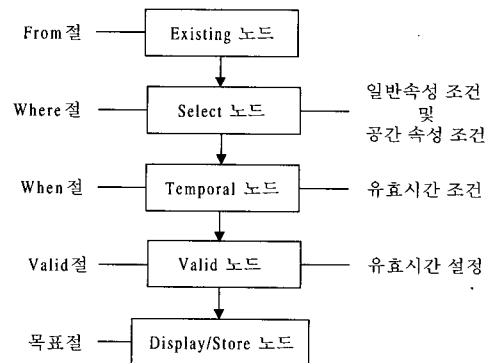
대한 참조와 시간 및 공간 속성에 대한 참조와 수식에 대한 검사 등을 위한 참조로 구성된다.

DMLStmt	::=	SelectStmt   InsertStmt
		DeleteStmt   UpdateStmt;
DMLStmt	=>	syn_fromClause : FromClauseORvoid,
		syn_validClause : ValidClauseORvoid,
		syn_whereClause : WhereClauseORvoid,
		syn_whenClause : WhenClauseORvoid;
FromClauseORvoid	::=	FromClause   void;
FromClause	=>	syn_tables : Seq Of TableList;
ValidClauseORvoid	::=	ValidClause   void;
ValidClause	=>	syn_validExpr : TemporalExpr;
WhereClauseORvoid	::=	WhereClause   void;
WhereClause	=>	syn_whereExpr : WhereExpr;
WhenClauseORvoid	::=	WhenClause   void;
WhenClause	=>	syn_whenExpr : TemporalExpr;

(그림 4) 시공간 질의 구문을 위한 IDL 표현

5.1.3 시공간 실행 트리

시공간 실행 트리는 입력된 시공간 질의에 대한 처리 계획을 트리 형태로 표현하며, 시공간 코드 생성기에 의해 (그림 5)와 같은 형태로 생성된다. 일반적으로 입력된 시공간 질의에 대응하는 시공간 실행 트리의 구성은 Existing 노드, Cartesian product 노드 또는 Join노드, 그리고 Select 노드, Temporal 노드, Valid 노드, Project 노드, Display 노드 혹은 Store 노드의 순서를 갖는다. 이들 노드는 개별적인 자료구조를 갖는데, 각각은 (그림 6)에서와 같이 IDL을 사용하여 작성하였다.



(그림 5) 시공간 실행 트리 구조

Existing 노드는 데이터베이스로부터 생성된 일종의 뷰(view)를 의미하며 테이블 자체 또는 일부를 가리킨

다. 2개 이상의 테이블에 대한 연산이 요구되는 경우 Cartesian product 노드가 생성되며 그외에는 Select 노드가 생성된다. 또한 그들은 자손 노드로서 Temporal 노드와 Valid 노드 및 Project 노드가 생성되며, 최종적으로 Store 노드 또는 Display노드가 생성된다. 실행 트리내 각 대수 연산 노드들 간에는 포인터로 연결된다. 특히 Select 노드에는 일반 속성뿐만 아니라 공간 속성에 대한 조건을 처리하기 위한 자료 구조와 해당 루틴이 연결되어 처리된다.

---

Roots	=>	syn_tables	:	Seq Of Existing;
Existing	=>	syn_where	:	Location,
		syn_description	:	Table,
		syn_descendants	:	Set Of DescendentInfo;
Node	::=	DisplayNode		NonterminalNode;
Node	=>	syn_nodeId	:	Integer,
		syn_leftInputArc	:	DescendentInfo;
NonterminalNode	::=	UnaryNode		BinaryNode;
NonterminalNode	=>	syn_descendants	:	Set Of DescendentInfo;
UnaryNode	::=	SelectNode		TemporalNode
		SpatialNode		ProjectNode
		StoreNode		ValidNode;
SelectNode	=>	syn_whereExpr	:	WhereExpr;
TemporalNode	=>	syn_domainPtrs	:	Seq Of DomainPtr,
		syn_temporalExpr	:	TemporalExpr;
ValidNode	=>	syn_domainPtrs	:	Seq Of DomainPtr,
		syn_temporalExpr	:	TemporalExpr;
ProjectNode	=>	syn_targetElements	:	Seq Of TargetElem,
		syn_allTargetRef	:	Boolean;
DisplayNode	=>	syn_attributes	:	Seq Of AttributeRef;
StoreNode	=>	syn_addToDB	:	Boolean,
		syn_targetTable	:	TableRef;

---

(그림 6) 시공간 실행 트리의 자료 구조에 대한 IDL표현

### 5.1.4 실행기

시공간 실행기는 시공간 코드 생성기가 처리한 실행 트리의 각 노드를 방문하면서 실제적인 연산을 호출하고, 다음의 3단계 처리기법을 사용하여 질의를 계산한다. 첫 번째 단계에서는 해당 노드를 초기화하며, 두 번째 단계에서는 해당노드에 대한 갱신 연산을 수행함과 동시에 하위노드에 대한 초기화 연산이 수행된다. 세 번째 단계에서는 하위 노드로의 전파 연산이 수행되며, 아울러 하위 노드에서는 갱신연산이 이루어지고, 그 하위 노드의 하위 노드에서는 초기화 연산이 이루어진다. 이와 같은 시공간 질의 처리 전략에서는 한 수행 트리의 실제적인 계산은 트리의 루트 노드인 Existing 노드로부터 터미널 노드인 Display노드 혹은 Store노드로 연속적으로 호출되어지는 과정을 거치고, 터미널 노드에서 수행이 완료되면 차례대로 루트노드

로 제어가 반환된다.

---

Command	::=	Initialize		Propagate		Update;
Command	=>	syn_relationRef	:	Table;		
Initialize	=>					
Propagate	=>					
Update	=>	syn_updateSpec	:	Seq Of TuplePair;		
TuplePair	=>	syn_beforeImage	:	Tuple,		
		syn_afterImage	:	Tuple;		
Node	=>	syn_leftTransFunc	:	TransFunction,		
		syn_leftTuples	:	Seq Of Tuple;		
BinaryNode	=>	syn_rightTransFunc	:	TransFunction,		
		syn_rightTuples	:	Seq Of Tuple;		
DisplayNode	=>	syn_displayTuples	:	Seq Of DisplayTuple,		
		syn_displayFile	:	FilePtr;		

---

(그림 7) 시공간 실행기의 자료 구조에 대한 IDL표현

## 5.2 알고리즘

### 5.2.1 시공간 파서와 의미 분석기

시공간 파서(spatiotemporal parser)는 사용자 질의에 대하여 Lex를 이용하여 어휘 분석을 수행하며 Yacc를 이용하여 구문 분석을 수행한다. 이들의 역할은 질의에 기술된 연산에 필요한 자료구조의 할당 및 초기화를 수행한다. 시간 조건에 대한 의미와 일반 조건, 그리고 공간 조건에 대한 의미는 C언어로 작성된 프로그램으로 수행된다. 시공간 어휘 분석기를 위해 시공간 데이터베이스에서 사용하는 '키워드', '사용자 정의어'의 형태, '상수 정의' 등을 Lex 형식에 맞게 작성하였으며, C코드로 변형된 어휘 분석기는 토큰 검사를 위한 각종 테이블과 메모리 설정을 해 준다. Yacc로 작성된 코드는 Yacc 컴파일러에 의해 C코드로 변형되며, 구문 분석기는 사용자가 입력한 질의를 어휘분석기와 연동하여 LALR파싱을 통해 구문의 정확성을 검사하는 기능을 수행한다.

시공간 의미 분석기(spatiotemporal semantic analyzer)는 입력된 질의의 파서 트리와 질의 대상이 되는 데이터베이스에 대한 정보를 입력받아 질의가 시공간 자료 조작문일 경우에 질의에서 사용한 테이블의 속성이 스키마 정보에 기록된 속성들과 일치하는지의 여부와 함께 데이터베이스를 사용하고자 하는 속성의 타입 등을 검사한다. 또한, 입력된 질의가 시공간 자료 정의문인 경우는 데이터 정의와 관련된 수행 절차의 대부분은 이 루틴에서 처리된다. 예를 들어, 테이블을 생성하는 질의가 들어올 경우 해당 테이블의 속성과 테이블을 구성하는 속성의 수와 그들의 개별적인 자료형 및 키 속성 정의 등을 기존의 데이터베이스에 정의된

테이블과의 중복 여부를 검사하여 중복이 없을 경우 생성하고자 하는 테이블과 관련된 정보를 스키마에 등록하는 작업이 이 단계에 이루어진다.

### 5.2.2 시공간 코드 생성기

시공간 코드 생성기(spatiotemporal code generator)는 입력된 질의에 대하여 데이터베이스 내부에서 처리하기 위한 표현에 해당하는 관계 대수로서 구성되는 시공간 실행 트리를 생성한다. 즉, 시공간 실행 트리는 입력된 질의에 대하여 내부적으로 처리 가능한 형태로 변환이 완료된 관계 대수들로 구성되는데, 실행 트리 상에 존재하는 노드에는 Existing 노드로부터 Display 노드 등이 있다. Existing 노드는 질의에서 기술된 테이블을 데이터베이스로부터 찾기 위한 정보를 갖고, Select 노드는 질의의 WHERE절에서 기술된 일반 조건 및 공간 조건을 처리한다. Temporal 노드와 Valid 노드는 각각 WHEN절 VALID절에서 기술된 시간 조건 및 시간 지정 값을 담당한다. Project 노드는 목표 리스트(target list)에서 서술된 속성들을 바탕으로 결과 테이블을 생성하는 역할을 한다. 마지막으로 Display 노드 또는 Store 노드는 상위의 Project 노드의 수행 결과를 화면으로 출력하거나 새로운 테이블로 저장하는 기능을 수행한다. (그림 8)에서는 입력된 질의에 대하여 시공간 실행 트리를 생성하는 시공간 코드 생성기의 작동과정을 보여준다.

```

STQP_CodeGenerator()
Input : Statement, Roots of tree, Node identifier, Tuple variables
Output : Execution tree
{
    Extract tuple variables from input statement;
    Count a number of tuple variable;
    if (total number of tuple equal to 1) {
        Build Existing_Node and then attach it to tree as a root;
        Build Select_Node and then attach it to Existing_Node as a child;
    }
    else {
        Build Cartesian_Subtree and then attach it to tree as a root;
        Build Select_Node and then attach it to Existing_Node as a child;
    }
    Build Temporal_Node and then attach it to Select_Node as a child;
    Build Valid_Node and then attach it to Temporal_Node as a child;
    Build Project_Node and then attach it to Valid_Node as a child;

    if (type of input statement is equal to Select statement)
        Build Display_Node and then attach it to Project_Node as a child;
    else
        Build Store_Node and then attach it to Project_Node as a child;
}
    
```

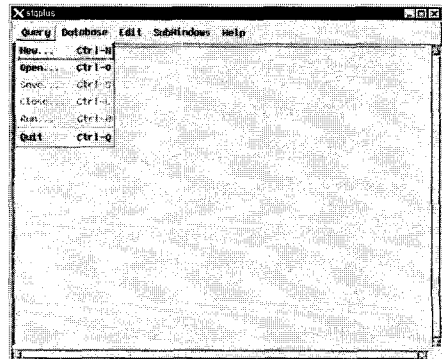
(그림 8) 시공간 코드 생성기 알고리즘

### 5.2.3 시공간 실행기

시공간 실행기는 시공간 코드 생성기에 의해 생성된 시공간 실행 트리를 루트 노드로부터 최하위 노드까지 차례로 방문하면서 해당 연산자를 구동시켜 그 결과를 다시 하위 노드로 전달하는 과정을 통해 최종 결과를 출력한다. 시공간 실행기는 시공간 코드 생성기가 생성한 시공간 실행 트리의 각 노드를 방문하여 해당 노드에서 정의된 연산을 수행한 다음, 후속노드로 제어 및 수행 중간 결과가 전달되는 기법을 사용한다. 또한 해당 튜플의 차분 정보를 관리하는 기능도 포함된다. 시공간 실행기에서 사용하는 기법으로는 스트림 처리(stream processing)로서 초기화 단계, 갱신 단계, 그리고 전파 단계로 세분화하여 상위 노드로부터 자신의 노드로 제어를 전달받아 처리한 다음 하위 노드로 제어를 전달하며, 이러한 과정이 루트 노드로부터 최하위 노드에 이르기까지 재귀적으로 수행된다.

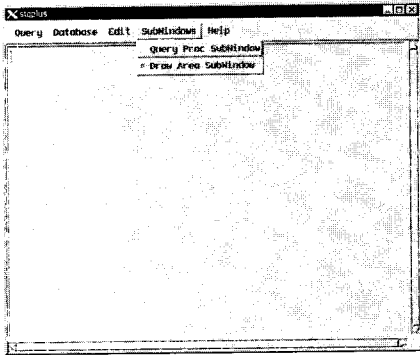
### 5.3 사용자 접속기

시공간 질의 처리 시스템에서의 사용자 접속기는 X-윈도우 모티프 프로그램을 사용하여 (그림 9)와 같이 구현되었다. 주 화면은 시공간 데이터베이스 생성과 연결, 사용자 질의의 작성 및 저장을 담당한다. 사용자 질의의 수행을 위해서는 시공간 질의 처리 시스템에서 처리된 결과를 출력하는 화면과 벡터 정보를 그리기 위한 별도의 화면이 활성화되어야 하는데, (그림 9b)의 subWindows화면에서 설정하도록 하였다.



(a) 시공간 질의 관리 메뉴





(b) 질의 처리 결과와 벡터 정보 출력을 위한 윈도우 활성 메뉴

(그림 9) 시공간 질의 처리 시스템의 사용자 접속기 예

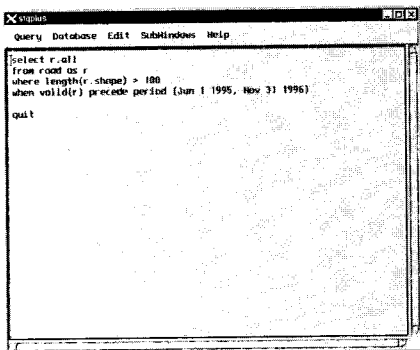
## 6. 수행 예와 평가

### 6.1 수행 예

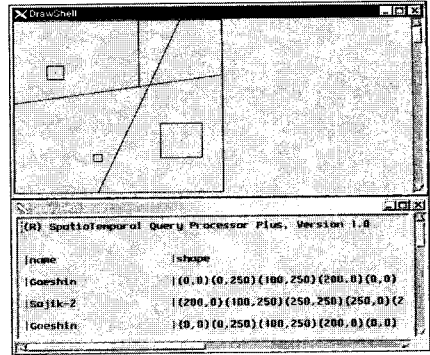
아래의 [예 1]은 주어진 도로 테이블에 저장된 각 튜플에 대하여 공간 속성에 대하여 공간 함수인 LENGTH()와 유효 시간에 대한 연산자인 PRECEDE를 동시에 적용한 형태이며, (그림 10)에서는 시공간 질의 처리 시스템에서 수행한 결과로서 (그림 10b)에서는 주어진 도로들로부터 설정된 조건을 만족하는 튜플인 “개신로”의 벡터 정보만이 구별되어 출력됨을 알 수 있다.

[예 1] 도로 테이블로부터 길이가 100미터를 초과하며, 1995년 6월 1일부터 1996년 12월 31일까지 시간 범위의 이전에 완공된 도로를 출력하라.

```
select r.all
from road as r
where length(r.shape) > 100
when valid(r) precede period (Jun 1 1995, Nov 31 1996)
```



(a) 시공간 검색문



(b) 시공간 검색문 실행 결과

(그림 10) 시공간 질의 처리 시스템의 실행 예

### 6.2 시스템 평가

현재 구현된 시공간 질의 처리 시스템은 주기억장치 처리 기법을 기반으로 전체 시공간 데이터베이스와 프로그램 코드를 주기억장치에 적재된 후 처리된다. 이것은 아직까지 시공간 데이터베이스를 위한 하부 저장 시스템의 연구가 진행되지 못한 이유에서 비롯된다. 따라서 현재 구현된 시공간 질의 처리 시스템의 처리 속도에 대한 평가는 다음과 같이 정리될 수 있다.

- 간단한 사용자 질의에 대한 처리는 디스크 접근 연산이 없는 관계로 매우 빠르게 이루어 질 수 있다.
- 사용자 질의가 비교적 큰 규모로 입력되면 모든 중간 결과가 주기억장치에서 처리될 수 없기 때문에 운영체제의 메모리 관리 기법에 의해 빈번한 디스크 연산이 발생하여 성능이 저하될 수 있으므로 이에 대한 고려가 필요하다.
- 공간 데이터의 경우는 벡터 정보만을 다루고 있기 때문에 위성 사진과 같은 래스터 정보의 처리는 향후 연구에서 제공해야 한다.

그러나 현재 구현된 시공간 질의 처리 시스템이 여러 가지 성능 관점에서 개선의 여지가 남아있음에도 불구하고, 이전의 연구에서 구현된 메뉴 방식에 의한 제한적인 공간 데이터의 이력 정보 제공에 비할 때 국내외적으로 최초의 시공간 질의를 처리하기 위한 시스템으로서 이 분야의 앞으로의 연구에 전환점 역할을 하리라 생각한다.

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 언어 및 질의 처리 시스템의 연구는 매우 미비한 실정이다.

따라서 이 논문에서는 이전 연구를 바탕으로 현실 세계에 존재하는 객체에 대한 시공간 정보를 데이터베이스에서 관리하기 위한 모델과 기존의 SQL를 기반으로 시간지원 연산과 공간 연산을 표현할 수 있도록 확장한 STQL에 대하여 설명하고, STQL로 작성된 시공간 질의에 대한 처리를 위한 시공간 질의 처리시스템의 설계 내용과 구현된 시스템의 특징 및 장단점을 설명하였다.

즉, 시공간 데이터 표현을 위하여 IDL을 이용한 시공간 데이터 구조를 정의하고 이를 이용한 STQL 구문 및 시공간 실행 트리를 설계 구현 하였으며, 구성된 실행 트리는 시공간 실행기에 의하여 수행되도록 하였다. 또한 STQP의 요소로는 시공간 파서와 의미 분석기, 시공간 코드 생성기 등을 구성하였다. 구현된 STQP는 기존의 관계형 질의 처리 시스템을 기반으로 하여 공간 객체에 대한 벡터 정보를 처리할 수 있도록 확장된 것이다. 이러한 STQP는 주기억장치 프로토타입으로서 질의상의 이력 정보와 공간연산을 효율적으로 처리한다.

앞으로 이 논문에서 구현된 시공간 질의 처리 시스템과 연동할 효율적인 하부 저장 시스템을 연구하고, 시공간 데이터베이스 관리 시스템의 성능 평가와 개선을 위한 질의 최적화와 시공간 집계함수에 대한 연구가 진행될 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 *SDE™*," 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, K.W. Jeon, K.J. Jeong, K.J. Kim, and K.H. Ryu, "A Temporal Datanbase Management Testbed," Transaction of KIPS, Vol.1, No.1, May, 1994.

[8] D.H. Kim, K.W. Nam, K.H. Ryu, "The Spatiotemporal Datbase 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] D. H. Kim, J. Y. Lee, Y.D. Joo, and K.H. Ryu, "Spatiotemporal Data Model and Extention of Their Operations for a Layered Temporal Geographic Information System," Trnasaction of KIPS, Vol.5, No.5, May, 1998.

[11] D. H. Kim, K. H. Ryu, "A Relational Spatiotemporal Database Query Language and Their Operation," Trnasaction of KIPS, Vol.5, No.10, Oct., 1998.

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

[13] K.W. Nam, D.H. Kim, and K.H. Ryu, "The Spatiotemporal Realtionship Operator," ITC-CSCC '96, International Technical Conference on Circuits/Systems, Computers, and Communications, Jul., 1996.

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

[15] 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.

[16] 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.

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

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

[19] R. Snodgrass, "The Interface Description Language: Definition and Use," Rockville, MD: Computer Science Press, 1989.

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

[21] 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.

[22] V. Tsotras, and A. Kumar, "Temporal Database Bibliography Update," ACM SIGMOD RECORD, Vol.25, No.1, Mar., 1996.

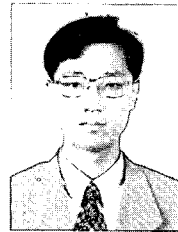
[23] M. Worboys, "A Unified Model for Spatial and Temporal Information," The Computer Journal, Vol.37, No.1, 1994.



**이성종**

e-mail : sjlee@dblab.chungbuk.ac.kr  
 1994년 충북대학교 컴퓨터학과 졸업(이학사)  
 1996년 충북대학교 대학원 전자계산학과 졸업(이학석사)

1996년~현재 충북대학교 대학원 전자계산학과 박사과정  
 관심분야 : 시간 데이터베이스, 시공간 데이터베이스, 객체지향 데이터베이스, 지리정보시스템 등



**김동호**

e-mail : kdh@etri.re.kr  
 1993년 충북대학교 전자계산학과 졸업(이학사)  
 1995년 충북대학교 대학원 전자계산학과 졸업(이학석사)  
 1999년 충북대학교 대학원 전자계산학과 졸업(이학박사)

1997년 한국전자통신연구원(위축연구원)  
 1999년~현재 한국전자통신연구원(선임연구원)  
 관심분야 : 시간 데이터베이스, 시공간 데이터베이스, 멀티미디어 시스템, 객체지향 데이터베이스, 루팅 제어 등



**류근호**

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  
 관심분야 : 시간 데이터베이스, 시공간 데이터베이스, 지식기반 정보검색, 객체 및 지식베이스 시스템