

논문 2008-5-19

## 이동체 데이터 스트림을 위한 시공간 DSMS의 설계 및 구현

## Design and Implementation of the Spatio-Temporal DSMS for Moving Object Data Streams

이기영\*, 김정준\*\*

Ki-Young Lee, Joung-Joon Kim

**요약** 최근 위치 측위 기술, 무선 통신 기술 등이 급속도로 발달하고 이동체 데이터가 널리 활용됨에 따라 이동체 데이터 스트림을 실시간으로 서비스하기 위한 실시간 위치 추적 시스템의 연구 개발이 활발히 진행되고 있다. 그러나 이러한 시스템에서 기반 시스템으로 사용되는 MO(Moving Object) DBMS는 이동체 데이터 스트림 관리에 비효율적이고, 기존의 DSMS(Data Stream Management System)는 시공간 데이터를 효율적으로 처리하지 못하는 문제점이 있다. 따라서, 본 논문에서는 이동체 데이터 스트림의 효율적인 실시간 관리를 위한 시공간 DSMS를 설계 및 구현하였다. 본 논문에서 구현한 시공간 DSMS는 스탠포드 대학의 STREAM(STanford stREam dAta Manager)을 기반으로 이동체 데이터 스트림의 실시간 관리와 시공간 질의 처리 기능 그리고 입력 부하를 줄이기 위한 필터링 기능을 지원하는 시스템이다. 특히, 시공간 DSMS에서 사용하는 시공간 연산자는 호환성을 위해서 OGC에서 제시한 "Simple Feature Specification for SQL" 표준 명세를 확장하여 SQL 형태의 표준 인터페이스를 지원한다. 마지막으로 본 논문에서 구현한 시공간 DSMS를 이동체 데이터 스트림의 실시간 위치 추적 관리가 필요한 실시간 모니터링 분야에 적용해 봄으로써 시스템의 효용성을 입증하였다.

**Abstract** Recently, according to the rapid development of location positioning technology and wireless communications technology and increasement of usage of moving object data, many researches and developments on the real-time locating systems which provides real time service of moving object data stream are under proceeding. However, MO (Moving Object) DBMS used based system in the in these systems is the inefficient management of moving object data streams, and the existing DSMS (Data Stream Management System) has problems that spatio-temporal data are not handled efficiently. Therefore, in this thesis, we designed and implemented spatio-temporal DSMS for efficient real-time management of moving object data stream. This thesis implemented spatio-temporal DSMS based STREAM (STanford stREam dAta Manager) of Stanford University is supporting real-time management of moving object data stream and spatio-temporal query processing and filtering for reduce the input loading. Specifically, spatio-temporal operators of the spatio-temporal DSMS support standard interface of SQL form which extended "Simple Feature Specification for SQL" standard specifications presented by OGC for compatibility. Finally, implemented spatio-temporal DSMS in this thesis, proved the effectiveness of the system that as applied real-time monitoring areas that require real-time locating of object data stream DSMS.

**Key Words** : Data Streams, Moving Object, Spatio-Temporal Data, DSMS

\*정회원, 을지대학교 의료산업학부

\*\*정회원, 건국대학교 컴퓨터공학과

접수일자 2008.8.5, 수정완료 2008.9.3

## 1. 서론

최근 RFID, GPS 등을 통한 위치 측위 기술과 무선 통

신 기술의 발달로 이동체 데이터 수집 수단이 획기적으로 발전하고 그 활용 범위가 다양해지고 있다<sup>[1]</sup>. 또한, 여러 분야에서 이동체 데이터를 이용한 서비스가 활용됨에 따라 이동체의 위치 정보를 실시간으로 서비스하기 위한 실시간 위치 추적 시스템의 연구 개발이 활발히 진행되고 있다<sup>[2]</sup>.

이러한 이동체 위치 정보를 실시간으로 원활히 서비스하기 위해서는 이동체 데이터 스트림을 실시간으로 처리할 수 있는 기반 시스템이 필요하다. 그러나 이러한 기반 시스템으로 MODBMS(Moving Object DBMS)와 같은 이동체 관리 시스템을 사용할 경우 지속적으로 입력되는 데이터 스트림 관리에 비효율적이며, 기존 DSMS(Data Stream Management System)와 같은 데이터 스트림 관리 시스템을 사용할 경우 이동체 데이터 스트림 처리가 비효율적이라는 문제점을 가지고 있다<sup>[3][4]</sup>.

따라서 본 논문에서는 이동체 데이터 스트림의 효율적인 실시간 관리를 위해 시공간 DSMS를 설계 및 구현하였다. 본 논문에서 개발한 시공간 DSMS는 스탠포드 대학의 STREAM(STanford stREam dAtA Manager)<sup>[5][6]</sup>을 기반으로 이동체 데이터 스트림의 실시간 관리와 시공간 질의 처리 기능 그리고 입력 부하를 줄이기 위한 필터링 기능을 지원하는 시스템이다. 특히, 시공간 DSMS에서 사용하는 시공간 연산자는 호환성을 위해서 OGC에서 제시한 “Simple Feature Specification for SQL”<sup>[7]</sup> 표준 명세를 확장하여 SQL 형태의 표준 인터페이스를 지원한다. 마지막으로 본 논문에서 구현한 시공간 DSMS를 이동체 데이터 스트림의 실시간 위치 추적 관리가 필요한 실시간 모니터링 분야에 적용해 봄으로써 시스템의 효율성을 입증하였다.

본 논문의 구성은 다음과 같다. 제 I 장의 서론에 이어, 제 II 장에서는 관련 연구로 STREAM과 OGC의 “Simple Feature Specification for SQL”에 대하여 설명한다. 제 III 장에서는 시스템 전체 구조와 시공간 DSMS의 각 관리자에 대하여 기술하고, 제 IV 장에서는 시공간 DSMS의 구현 내용과 가상 시나리오를 설명한다. 마지막으로, 제 V 장에서는 결론에 대하여 언급한다.

## II. 관련 연구

본 장에서는 시공간 DSMS를 구현하기 위해 사용한

기반 시스템인 STREAM과 OGC의 “Simple Feature Specification for SQL” 표준 명세에 대하여 살펴본다.

### 1. STREAM(STanford stREam dAtA Manager)

STREAM은 지속적으로 끊임없이 입력되는 데이터 스트림을 실시간으로 처리하기 위한 스트림 관리 시스템으로 Stanford 대학에서 개발한 DSMS이며 현재 서버 0.6 버전과 클라이언트 0.3 버전이 소스와 함께 BSD License 형태로 공개되어 있다<sup>[3][4]</sup>. 서버는 Linux 기반에서 C++로 구현되어 있으며 라이브러리로 사용이 가능하고, 데이터 스트림 처리를 위한 연속 질의 언어 CQL을 지원한다. 클라이언트는 운영체제에 독립적인 Java로 구현되어 있으며 데이터 입력 및 질의 입력, 실행 계획 뷰어 등을 제공한다.

그림 1은 STREAM의 동작 방식을 보여주고 있다. 사용자가 등록한 연속 질의는 Select, Project, Join 등의 연산자가 트리 형태로 모여 있는 실행 계획으로 변환되어 저장 및 처리된다. 각 연산자는 필요에 따라 저장소를 생성하게 되며 이 저장소에는 질의 처리의 중간 결과가 저장되어 여러 연산자에 의해 공유된다. 각 연산자는 스템브(stub)라는 상태 저장 관리자를 갖고 있어서 저장소에 대한 참조 값을 보관하며, 연산자들은 큐로 연결되어 있어서 각 연산자는 입력 큐의 데이터들을 처리하여 출력 큐에 저장한다. 이 외에도, 하나의 질의 결과가 다른 질의들에 의해 공유되는 기능이 있어서 질의 처리 속도 및 공간 효율성을 향상시킨다.

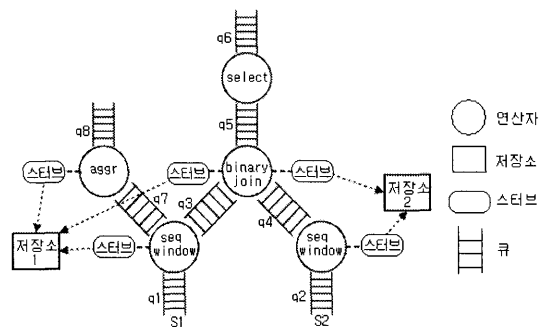


그림 1. STREAM의 동작 방식  
Fig. 1. Operation Method of STREAM

본 논문에서는 이러한 STREAM을 확장하여 공간 DSMS를 개발하기 위해 OGC에서 제시한 “Simple Feature Specification for SQL” 표준 명세에 따라 공간

데이터 타입 및 연산자를 추가하였으며, 공간 연산자의 구현을 위해 GEOS(Geometry Engine Open Source)<sup>[8]</sup>를 사용하였다.

### 2. Simple Feature Specification for SQL

“Simple Feature Specification for SQL” 표준 명세는 ODBC API를 경유하는 심플 공간 피쳐 집합의 저장, 검색, 질의, 갱신을 지원하는 표준 SQL 스키마를 정의하고 있다<sup>[7]</sup>. 표 1은 이 명세에서 정의하고 있는 공간 데이터 타입 중 인스턴스화할 수 있는 타입들의 WKT(Well-Known Text) 표현을 보여준다.

표 1. 공간 데이터의 WKT 표현  
Table 1. WKT Representation of Spatial Data Dype

Geometry	Type SQL Text Literal Representation
Point	' POINT (10 10) '
LineString	' LINESTRING ( 10 10, 20 20, 30 40) '
Polygon	' POLYGON ((10 10, 10 20, 20 20, 20 15, 10 10)) '
Multipoint	' MULTIPPOINT (10 10, 20 20) '
MultiLineString	' MULTILINESTRING ((10 10, 20 20), (15 15, 30 15)) '
MultiPolygon	' MULTIPOLYGON ( ((10 10, 10 20, 20 20, 20 15, 10 10)), ((60 60, 70 70, 80 60, 60 60)) ) '
GeomCollection	' GEOMETRYCOLLECTION (POINT (10 10), POINT (30 30), LINESTRING (15 15, 20 20)) '

“Simple Feature Specification for SQL” 표준 명세에서 정의하고 있는 공간 연산자는 객체간의 관계를 테스트하는 연산자, 공간 분석을 위한 연산자, 그리고 각 타입별 함수로 나눌 수 있다. 표 2는 표준 공간 연산자들을 보여준다.

표 2. 표준 공간 연산자  
Table 2. Standard Spatial Operator

SQL Functions for testing Spatial Relations	SQL Functions on each Type
Equals(Geometry, Geometry)	Dimension(Geometry)
Disjoint(Geometry, Geometry)	GeometryType(Geometry)
Touches(Geometry, Geometry)	AsText(Geometry)
Within(Geometry, Geometry)	AsBinary(Geometry)
Overlaps(Geometry, Geometry)	SRID(Geometry)
Crosses(Geometry, Geometry)	IsEmpty(Geometry)
Intersects(Geometry, Geometry)	ISimple(Geometry)
Contains(Geometry, Geometry)	Boundary(Geometry)
Relate(Geometry, Geometry, PatternMatrix)	Envelope(Geometry)
	X(Point)
	Y(Point)
	StartPoint(Curve)
	EndPoint(Curve)
	ISClosed(Curve)
	IsRing(Curve or MultiCurve)
	Length(Curve or MultiCurve)
	NumPoints(LineString)
	PointN(LineString, Integer)
	Centroid(Surface or MultiSurface)
	PointOnSurface(Surface or MultiSurface)
	Area(Surface or MultiSurface)
	ExteriorRing(Polygon)
	NumInteriorRing(Polygon)
	InteriorRingN(Polygon, Integer)
	NumGeometries(GeomCollection)
	GeometryN(GeomCollection, Integer)
SQL Functions for Spatial Analysis	
Distance(Geometry, Geometry)	
Intersection (Geometry, Geometry)	
Difference (Geometry, Geometry)	
Union (Geometry, Geometry)	
SymDifference(Geometry, Geometry)	
Buffer (Geometry, Double Precision)	
ConvexHull(Geometry)	

## III. 시스템 설계

본 장에서는 시스템 전체 구조에 대해 설명하고, 시공간 DSMS의 각 관리자에 대해 자세히 설명한다.

### 1. 시스템 전체 구조

시공간 DSMS의 시스템 전체 구조는 그림 2와 같다.

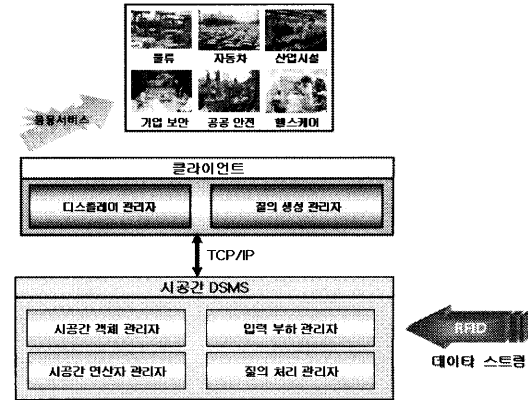


그림 2. 시스템 구조  
Fig. 2. System Architecture

그림 2에서 보는 바와 같이 시공간 DSMS는 질의 처리 관리자, 입력 부하 관리자, 시공간 연산자 관리자, 시공간 객체 관리자로 구성되고 클라이언트는 디스플레이 관리자와 질의 생성 관리자로 구성된다.

### 2. 시공간 DSMS

#### 2.1 질의 처리 관리자

질의 처리 관리자는 클라이언트로부터 전달받은 연속질의를 시공간 DSMS에 등록하고 분석 및 처리할 수 있는 기능을 제공한다. 또한, 대용량의 시공간 데이터 스트림을 실시간으로 처리하기 위하여 SQL과 유사한 질의 언어인 CQL(Continuous Query Language)이라는 연속질의 언어를 사용한다. CQL은 시공간 데이터 스트림과 윈도우(Window)를 타임스탬프(TimeStamp) 형태의 정렬된 테이블로 간주하고 질의에 대한 결과를 전달하기 위한 변환 연산자를 제공한다. 그림 3은 시공간 DSMS에서 제공하는 CQL 연산자를 보여준다.

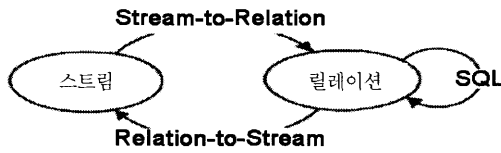


그림 3. CQL의 연산자  
Fig. 3. Operator of CQL

CQL은 스트림을 릴레이션으로 바꿔주는 Stream-to-Relation 연산자, 릴레이션을 다시 스트림으로 바꿔주는 Relation-to-Stream 연산자, 기존 관계형 DBMS에서 SQL로 처리하던 관계 대수의 연산자로 구성된다.

### 2.2 입력 부하 관리자

입력 부하 관리자는 시공간 DSMS에서 처리하지 못할 과도한 데이터 스트림 입력이 들어왔을 때 정확도 손실을 최소화 하면서 입력 데이터 스트림 양을 줄여 과부하 문제를 해결하기 위한 필터링 기능을 제공한다. 그림 4는 시공간 DSMS에서 시공간 데이터가 필터링되는 과정을 보여준다.

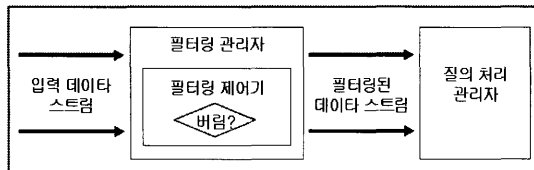


그림 4. 시공간 데이터의 필터링  
Fig. 4. Filtering of Spatio-Temporal Data

시공간 DSMS에서는 필터링을 위해 좌표 거리 차이, 특정 시간 간격, 특정 이동 객체의 아이디 등이 필터링 조건으로 지원된다.

### 2.3 시공간 연산자 관리자

시공간 데이터를 효율적으로 처리하기 위해서는 복잡한 연산 과정이 필요하나 기존 STREAM은 다차원 형태인 시공간 데이터를 지원하지 않는다. 따라서, 본 논문에서는 GEOS에서 제공하는 알고리즘을 사용하여 시공간 연산자를 처리하는 시공간 연산자 관리자를 구현하였다. 표 3은 OGC에서 정의한 공간 연산자를 확장하여 시공간 DSMS가 지원하는 14개의 시공간 연산자를 보여준다.

표 3. 시공간 DSMS의 시공간 연산자

Table 3. Spatio-Temporal Operator of Spatio-Temporal DSMS

종류	이름	설명
시공간 관계 함수	ST_Equals	각 단위 시간마다 두 시공간 객체가 동일하면 참을 반환함.
	ST_Disjoint	각 단위 시간마다 두 시공간 객체가 교집합이 없으면 참을 반환함.
	ST_Touches	각 단위 시간마다 두 시공간 객체의 경계선만 닿았으면 참을 반환함.
	ST_Within	각 단위 시간마다 첫 번째 시공간 객체가 두 번째 시공간 객체를 포함하면 참을 반환함.
	ST_Overlaps	각 단위 시간마다 두 시공간 객체가 같은 차원이면서 겹쳐 있으면 참을 반환함.
	ST_Crosses	각 단위 시간마다 두 시공간 객체가 교차하면서 교차된 영역의 차원이 같거나 줄어들면 참을 반환함.
	ST_Intersects	각 단위 시간마다 두 시공간 객체가 교차하면 참을 반환함.
	ST_Contains	각 단위 시간마다 두 번째 시공간 객체가 첫 번째 시공간 객체를 포함하면 참을 반환함.
	ST_Relate	각 단위 시간마다 두 시공간 객체의 위상관계가 주어진 조건(9-Intersection Matrix)을 만족하는지 여부 반환함.
시공간 분석 함수	ST_Distance	두 시공간 객체간의 거리 변화를 시간 순으로 추출하여 반환함.
	ST_Intersection	두 시공간 객체가 겹치는 부분을 시간 순으로 추출하여 반환함.
	ST_Difference	첫 번째 시공간 객체에서 두 번째 시공간 객체를 뺀 부분을 시간 순으로 추출하여 반환함.
	ST_Union	두 시공간 객체를 합친 시공간 객체를 시간 순으로 추출하여 반환함.
	ST_Buffer	시공간 객체 주변을 주어진 크기만큼 버퍼링 해서 시간 순으로 계산, 추출하여 반환함.

시공간 관계 연산자는 두 공간 객체의 공간상의 관계를 파악하기 위한 연산자들이며, 시공간 분석 연산자는 하나 혹은 두 시공간 객체로부터 새로운 시공간 객체 혹은 의미 있는 값을 얻기 위한 연산자들이다.

### 2.4 시공간 객체 관리자

기존 STREAM은 메모리에 대한 포인터를 지원하지 않기 때문에 저장소가 더 이상 사용하지 않는 튜플을 메모리에서 해제할 때 참조 값이 적혀있는 메모리 영역만을 삭제해 주었다. 그러나 시공간 객체는 Heap 영역에 생성되고 저장소에는 포인터 값만 저장되기 때문에 저장소에 저장되어 있는 포인터 정보가 메모리에서 삭제되고 하더라도 Heap 영역에 있는 시공간 객체가 메모리에서 삭제되지 않는다. 이 때문에 시공간 객체를 별도로 추가/삭제하는 과정이 필요하다. 그리고 Heap 영역에 있는 시공간 객체를 저장소가 직접 삭제하는 방식은 시공간 객체가 여러 저장소에 동시에 존재할 경우 같은 작업을 중복 처리해야 하기 때문에 비효율적이다.

따라서 이러한 문제를 해결하기 위해 여러 저장소에서 참조하는 시공간 객체를 효율적으로 관리하는 시공간

객체 관리자를 개발하였다. 시공간 객체 관리자는 여러 저장소에서 참조하는 시공간 객체의 시공간 객체 아이디와 참조 카운터를 관리함으로써 시공간 객체가 사용하는 메모리 공간의 낭비를 최소화하고 중복되는 메모리 할당 및 해제 작업을 줄인다. 그림 5는 시공간 객체가 여러 저장소에서 공유되는 예를 보여준다.

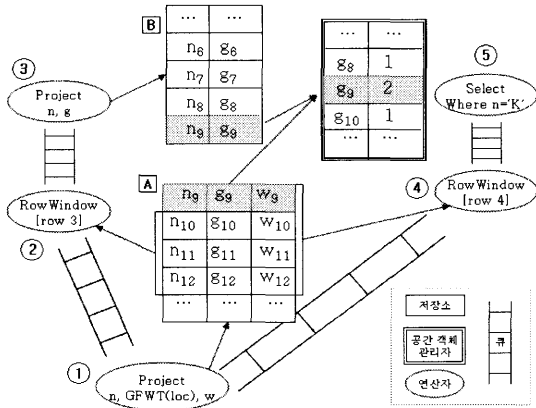


그림 5. 시공간 객체 공유의 예  
Fig. 5. Example of Sharing Spatio-Temporal Object

그림 5를 통하여 시공간 객체 관리자의 기본적인 동작 방식을 살펴보면 다음과 같다. 연산자 1이 GFWT 함수를 통해 시공간 객체를 생성한 뒤 이것을 저장소 A에 저장해 두면 연산자 2와 4가 공유하게 된다. 이때, 연산자 2는 저장소 A에 있는 9번째 튜플을 처리하고 슬라이딩 윈도우를 10부터 12번 튜플로 옮긴 뒤 연산자 3이 이를 저장소 B에 저장하지만, 연산자 4가 9번째 튜플을 아직 처리하지 않아서 저장소 A에는 9번째 튜플이 남아있게 된다. 이러한 경우 저장소 A와 저장소 B 모두에 시공간 객체 g9가 존재하게 되며 시공간 객체 관리자는 g9에 대한 참조 카운터를 2로 늘린다. 연산자 4가 9번째 튜플을 처리하면 저장소 A에서 9번째 튜플이 삭제되는데, 이때 시공간 객체 관리자는 Heap 영역에 있는 시공간 객체를 바로 삭제하지 않고 참조 카운터만 1로 줄인 후, 저장소 B에서도 9번째 튜플이 삭제되면 참조 카운터가 0이 되면서 g9를 삭제한다.

### 3. 클라이언트

클라이언트는 질의 생성 관리자와 디스플레이 관리자로 구성된다.

### 3.1 질의 생성 관리자

질의 생성 관리자는 클라이언트에서 검색하고자 하는 이동체 데이터의 검색 조건을 입력 받아 시공간 DSMS에 전송하기 위한 질의를 생성한다.

### 3.2 디스플레이 관리자

디스플레이 관리자는 시공간 DSMS로부터 전달받은 이동체 데이터의 위치를 사용자가 볼 수 있도록 클라이언트 화면에 점으로 표시한다. 또한, ZOOM IN/OUT과 PAN 기능을 제공한다.

## IV. 시스템 구현

본 장에서는 시스템의 구현 환경과 구현 상세 내용을 설명하고, 가상 시나리오를 통해 시스템의 효율성을 실험한다.

### 1. 구현 환경

시공간 DSMS 개발을 위해 운영체제는 Ubuntu 6.06을 사용하였고, 개발 도구는 g++ 3.2.3 버전을 사용하였다. 그리고 시공간 질의 처리 시 사용되는 시공간 연산자 개발을 위해 GEOS 라이브러리를 사용하였다. 클라이언트는 운영체제로 Windows XP Professional을 사용하였고, 개발 도구는 JAVA 1.5 버전을 사용하였다.

### 2. 시공간 DSMS 구현

본 논문에서는 시공간 질의 처리를 위해 기존 STREAM에 연속 질의를 분석 및 처리하는 질의 처리 관리자, 시공간 DSMS에서 입력 데이터 스트림 양을 줄이기 위한 필터링 기능을 제공하는 입력 부하 관리자, 연속 질의에 포함된 시공간 연산자를 처리하는 시공간 연산자 관리자, 시공간 객체를 효율적으로 관리하기 위한 시공간 객체 관리자를 추가 하였다. 그림 6은 시공간 DSMS와 클라이언트의 실행 화면을 보여준다.

그림 6의 클라이언트내의 다이얼로그 박스는 시공간 DSMS에 연결하기 위해 호스트 IP 주소와 포트를 설정하기 위한 것이다.

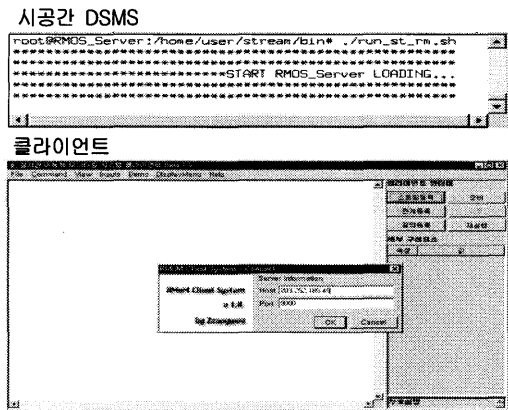


그림 6. 시공간 DSMS와 클라이언트의 실행 화면  
Fig. 6. Execute Screen of Spatio-Temporal DSMS and Client

구현된 시공간 DSMS의 테스트를 위한 질의로 공간 객체 A가 공간 객체 B를 포함하는지 여부를 반환하는 시공간 함수인 ST\_Contains를 사용하여 질의문을 생성하였으며, 그림 7은 질의 처리 결과를 보여준다.

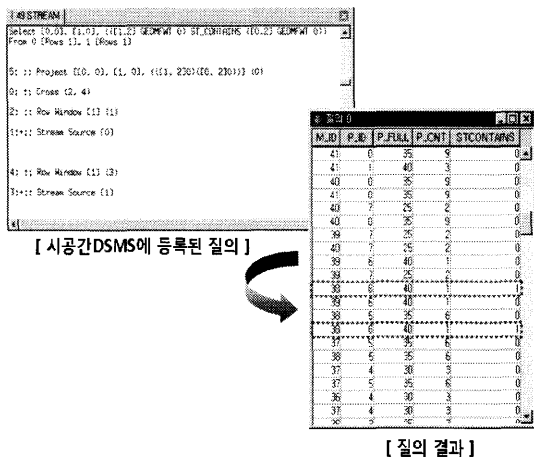


그림 7. 시공간 연산자의 질의 처리 결과  
Fig. 7. Query Processing Result of Spatio-Temporal Operator

그림 7에서는 38번 ID를 가진 이동체가 6번 ID를 가진 공간에 진입하여 ST\_Contains의 결과로 True값을 반환한 것을 확인할 수 있다.

### 3. 가상 시나리오

시공간 DSMS의 효율성을 검증하기 위해 실시간 주차 모니터링 서비스에 시공간 DSMS를 적용하였다. 실

시간 주차 모니터링 서비스에서는 이동 차량과 주차 구역 차량 인식 센서 등을 임의로 생성하여 주차 구역에 진입하는 이동 차량과 주변 주차 구역을 등을 검색한다.

실험을 위해 이동 차량과 주차 구역의 테이블을 생성하였으며, 생성한 이동 차량 (M) 테이블에는 이동 차량 id와 현재 위치 정보가 있고, 주차구역 (P) 테이블에는 주차구역 id와 공간 정보 외에 전체 주차 공간의 수량과 사용 가능한 주차 공간의 수량 정보가 있다.

본 실시간 주차 모니터링에서 사용한 질의 목록은 표 4와 같다.

표 4. 질의 목록  
Table 4. Query List

이름	질의
질의 1 (Geo_m)	Select id, geomFromText( M.loc, 0 ) From M [range 2 seconds];
질의 2 (Geo_p)	Select id, geomFromText( P.loc, 0 ), full, cnt From P [range 2 seconds];
질의 3 (st_cont)	Select M.id, P.id, P.full, P.cnt, ST_Contains(GeomFromText(P.loc, 0), geomFromText(M.loc, 0)) From M, P;
질의 4 (st_dist)	Select M.id, P.id, ST_Distance(GeomFromText(P.loc, 0), geomFromText(M.loc, 0)) From M, P;
질의 5	Select M_id, P_id, From st_dist Where s_dist < 500;

<질의 1>는 이동 차량의 위치를 모니터링 하고, <질의 2>는 근처 주차 구역의 현황을 모니터링 한다. 그리고 <질의 3>은 현재 이동 차량이 어느 주차 구역에 진입했는지를 모니터링 하고, <질의 4>는 주변 주차구역과의 거리를 구하며, <질의 5>는 반경 500m 내에 있는 주변 주차 구역을 검색한다.

실시간 주차 모니터링 서비스는 표 4에서 사용된 질의를 통해 입력 데이터 스트림을 처리하게 된다. 그림 8은 등록된 질의에 대한 질의 처리 결과를 보여준다.

그림 8은 이동 차량과 주차 구역의 거리를 구하는 <질의 4>의 처리 결과와 구해진 거리를 기반으로 이동 차량의 반경 500m 내에 있는 주차구역을 검색해서 출력하는 <질의 5>의 처리 결과를 보여주고 있다.

## V. 결론

본 논문에서는 기존 MODBMS와 같은 이동체 관리 시스템이 갖는 비효율적인 실시간 스트림 관리 문제와 기존 DSMS에서 시공간 데이터를 처리하지 못하는 문제를 해결하기 위해 이동체 데이터 스트림의 효율적인 실시간 처리를 지원하는 시공간 DSMS를 개발하였다.

시공간 DSMS는 이동체 데이터 스트림의 실시간 관리와 시공간 질의 처리 기능 그리고 입력 부하를 줄이기 위한 필터링 기능을 지원하는 시스템이다. 특히, 시공간 DSMS에서 사용하는 시공간 연산자는 호환성을 위해서 OGC에서 제시한 “Simple Feature Specification for SQL”[7] 표준 명세를 확장하여 SQL 형태의 표준 인터페이스를 지원한다. 본 논문에서는 이러한 시공간 DSMS를 실시간 이동체 관리가 필요한 가상 시나리오에 적용해 봄으로써 시스템의 효용성을 검증하였다.

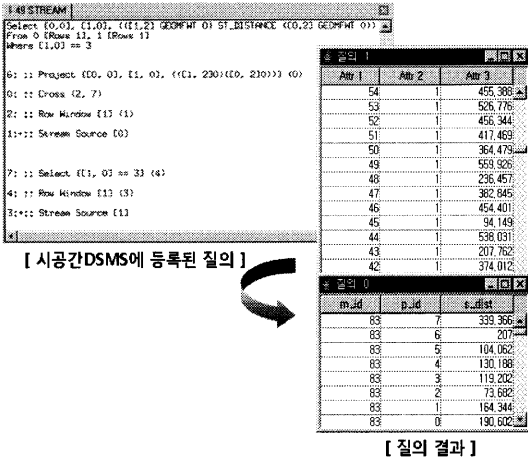


그림 8. 시나리오의 질의 처리 결과  
Fig. 8. Query Processing Result of Scenario

그림 9는 질의 처리 결과에 따른 클라이언트의 디스플레이 관리자 처리 화면이다.



그림 9. 시나리오의 질의 처리 결과 화면  
Fig. 9. Query Processing Result Screen of Scenario

그림 9는 <질의 1>과 <질의 2>의 처리 결과인 이동 차량의 현재 위치와 주차 구역의 위치, 주차 구역내의 전체 주차 공간의 수 및 주차 가능 공간의 수를 디스플레이 관리자를 통해 화면으로 보여주고 있다. 이 때 화면 상의 점은 이동 차량, 사각형은 센서, 점선은 주차 구역을 나타낸다.

## 참고 문헌

- [1] 손해원, 모희숙, 성낙선, “UHF RFID 기술”, 전자통신동향분석, 제20권 제3호, pp.67-80, 2005
- [2] 지능형교통시스템(ITS)전담기구, 교통·전자·통신 등을 통한 실시간 교통정보 시스템, <http://www.itskorea.or.kr/its/definition.asp>, 2006
- [3] 원종호, 이미영, 김명준, “유비쿼터스 컴퓨팅 환경을 위한 RFID 기반 센서 데이터 처리 미들웨어 기술 동향”, 전자통신동향분석, 19권 5호, pp. 21-29, 2004
- [4] Golab L., and Özsu M. T., “Issues in Data Stream Management”, SIGMOD Record, Vol.32, No.2, pp. 5-14, 2003
- [5] Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., Nishizawa, I., Rosenstein, J., and Widom, J., “STREAM: The Stanford Stream Data Manager”, IEEE Data Engineering Bulletin, Vol.26, No.1, pp.19-26, 2003
- [6] Arasu, A., Babu, S., and Widom, J., “The CQL Continuous Query Language: Semantic Foundations and Query Execution”, The VLDB Journal, Vol.15, No.2, pp. 121-142, 2006

- [7] Open Geospatial Consortium Inc., OpenGIS Implementation Specification for Geographic information - Simple Feature Access - Part 2: SQL option, 2005
- [8] Ramsey, P., The State of Open Source GIS, Refrations Research Inc., 2006

### 저자 소개

#### 이 기 영(정회원)



- 1984년 숭실대학교 전자계산학과 학사 졸업.
- 1988년 건국대학교 대학원 컴퓨터공학과 석사 졸업.
- 2005년 건국대학교 대학원 컴퓨터공학과 박사 졸업
- 1984년~1991년 한국해양연구원 연구원

구원

- 1996년~1998년 한국컴퓨터정보학회 이사 및 서울동부지회장
- 1991년~현재 을지대학교 의료산업학부 교수

<주관심분야 : 공간 데이터베이스, GIS, LBS, USN, 텔레매틱스>

#### 김 정 준(정회원)



- 2003년 건국대학교 컴퓨터공학과 학사 졸업.
- 2005년 건국대학교 대학원 컴퓨터공학과 석사 졸업.
- 2005년~현재 건국대학교 대학원 컴퓨터공학과 박사 수료.

<주관심분야 : 공간 메인 메모리 데이터베이스, GIS, LBS, 텔레매틱스, USN>