

# Oracle 기반의 OpenGIS 공간 인터페이스의 개발<sup>†</sup>

## Development of an OpenGIS Spatial Interface based on Oracle

박춘걸\* / Chun-Geol Park    박희현\*\* / Hee-Hyun Park  
 강홍구\*\*\* / Hong-Koo Kang    한기준\*\*\*\* / Ki-Joon Han

### 요약

최근에 공간 데이터의 수집 방법이 발전하면서 다양한 산업과 연구 분야에서 공간 데이터가 생성되고 유통 및 활용되고 있다. 이러한 대용량의 공간 데이터를 효율적으로 처리하기 위하여 ESRI의 ArcSDE, Oracle의 Oracle Spatial과 같은 기존 DBMS를 확장하는 연구들이 활발히 진행되고 있다. 그러나 이러한 상용 DBMS의 확장을 사용 시에는 추가적인 비용이 발생하고, 사용되는 공간 데이터 타입과 공간 연산자들이 서로 상이함으로 인해 상호운용성의 문제를 초래하고 있다.

따라서 본 논문에서는 OGC(Open Geospatial Consortium)에서 제시한 “SQL을 위한 심플 피쳐 명세(Simple Features Specification for SQL)”의 표준 인터페이스를 지원하는 OpenGIS 공간 인터페이스를 Oracle 상에서 개발하였다. OpenGIS 공간 인터페이스는 “SQL을 위한 심플 피쳐 명세”에서 제시한 모든 데이터 타입과 공간 연산자를 지원하기 때문에 사용자는 추가적인 비용 없이 본 논문에서 개발한 OpenGIS 공간 인터페이스를 사용하여 Oracle의 대용량 공간 데이터를 효율적으로 처리할 수 있다. 또한, 성능 평가를 통해 OpenGIS 공간 인터페이스가 Oracle Spatial보다 응답 시간 측면에서 우수한 성능을 보임을 입증하였다.

### Abstract

Recently, with the development of collecting methods of spatial data, the spatial data is produced, circulated, and used in various fields of industry and research. To manage the mass spatial data efficiently, the researches on extension of the existing commercial DBMS, such as ESRI's ArcSDE or Oracle's Oracle Spatial, is making progress actively. However, the usage of the extension of the commercial DBMS incurs an additional expense and causes an interoperability problem due to differences in spatial data types and spatial operators.

Therefore, in this paper, we developed an OpenGIS Spatial Interface for Oracle, which supports a standard interface by following the “Simple Features Specification for SQL” proposed by OGC(Open Geospatial Consortium). Since the OpenGIS Spatial Interface provides all spatial data types and spatial operators proposed in “Simple Features Specification for SQL”, users can manage mass spatial data of Oracle efficiently by using

<sup>†</sup> 본 연구는 서울시 산학연 협력사업의 신기술 연구개발 지원사업의 지원으로 수행되었음.

- 논문접수 : 2007.2.9      ■ 심사완료 : 2007.4.25
- \* 교신저자 건국대학교 컴퓨터공학과 박사과정(cgpark@db.konkuk.ac.kr)
- \*\* 건국대학교 컴퓨터공학과 석사(hhpark@db.konkuk.ac.kr)
- \*\*\* 건국대학교 컴퓨터공학과 박사과정(hkkang@db.konkuk.ac.kr)
- \*\*\*\* 건국대학교 컴퓨터공학부 교수(kjhan@db.konkuk.ac.kr)

the standard interface without additional expense. In addition, we proved that the OpenGIS Spatial Interface is superior to the Oracle Spatial in the response time through the performance evaluation.

**주요어** : 공간 데이터, OpenGIS 공간 인터페이스, OGC, 상호운용성, Oracle Spatial

**Keyword** : Spatial Data, OpenGIS Spatial Interface, OGC, Interoperability, Oracle Spatial

## 1. 서론

오늘날 GIS(Geographic Information System)나 LBS(Location Based Service)와 같이 공간 데이터를 활용하는 서비스가 일반화되면서 다양한 분야에서 공간 데이터가 활용되고 있다[1,2]. 이처럼 공간 데이터의 활용 범위가 확대됨에 따라 대용량의 공간 데이터를 보다 효율적으로 처리하기 위하여 복잡한 공간 데이터의 저장과 관리[3,4], 다양한 공간 데이터 타입의 지원[5,6,7], 공간 데이터에 대한 빠른 접근[1,4,8], 공간 데이터 간의 위상 연산 지원[9,10], 그리고 고비용의 공간 연산 처리를 위한 질의 처리 기법[2,11]과 같은 기술들이 요구되고 있다.

이러한 요구 사항에 의해 최근 기존의 DBMS(DataBase Management System)를 확장하는 연구들이 활발히 진행되고 있다. 상용 DBMS인 Oracle에서도 공간 데이터 처리를 위한 확장이 이루어지고 있는데 [2,12], 대표적인 것이 ESRI에서 개발한 ArcSDE[13]와 Oracle 자체에서 개발한 Oracle Spatial[14]이다. ArcSDE는 Oracle, MS-SQL, MySQL 등의 상용 DBMS에 공간 데이터 타입과 공간 연산자를 지원하고, Oracle Spatial은 Oracle에 대해서만 공간 데이터 타입과 공간 연산자를 지원한다. 그러나 이러한 ArcSDE나 Oracle Spatial을 사용하기 위해서는 제품의 라이선스를 구입해야 하는 추가적인 비용이 발생하며, 또한 이들 시스템에서 사용되는 공간 데이터 타입과 공간 연산자들이 서로 다르기 때문에 상호운용성 문제를 야기하고 있다[1,2].

따라서 본 논문에서는 OGC(Open Geospatial Consortium)에서 제시한 “SQL을 위한 심플 피쳐

명세” [5,6,7]의 표준 인터페이스를 지원하는 OpenGIS 공간 인터페이스를 개발하였다. 본 논문에서 개발한 OpenGIS 공간 인터페이스는 국제적으로 시장 점유율이 가장 높고 안정적인 상용 DBMS인 Oracle을 기반으로 공간 객체 모델과 기본적인 Geometry 함수를 제공하는 GEOS(Geometry Engine Open Source)[15]의 API를 사용하여 개발되었다. GEOS는 “SQL을 위한 심플 피쳐 명세”에서 제시하고 있는 모든 데이터 타입과 공간 연산자들을 지원하고 있는 공개 API이다. 따라서 본 논문에서 개발한 OpenGIS 공간 인터페이스를 통하여 사용자는 추가적인 비용 없이 “SQL을 위한 심플 피쳐 명세”에서 제시한 모든 공간 데이터 타입과 공간 연산자를 사용하여 Oracle에 저장되어 있는 공간 데이터를 효율적으로 처리할 수 있다.

본 논문에서 개발한 OpenGIS 공간 인터페이스에서 지원하는 공간 함수들이 Oracle에 의해 호출되면 Oracle에서 지원하는 Create 구문을 이용하여 PL/SQL 부프로그램으로 등록되어야 한다. 이 PL/SQL 부프로그램은 디스크에 저장되어 있다가 사용자의 질의가 있을 때, Oracle의 공유 풀(shared pool)로 적재되어 사용된다. 그러나 시간이 흐름에 따라 공유 풀에서 디스크로 페이지 아웃될 수 있는데, 이 경우 다음 호출 시 다시 디스크를 검색하여 공유 풀로 적재해야 하므로 지연이 발생한다. 그러므로 본 논문에서는 이러한 지연을 예방하고자 빈번히 사용되는 PL/SQL 부프로그램을 Oracle의 공유 풀에 상주시키는 피닝(pining)을 사용하였다[16]. 즉, 본 논문에서 개발한 OpenGIS 공간 인터페이스는 제공되는 모든 함수들을 패키지로 캡슐화하고 Oracle 데이터베이스의 구동 시 피닝하여 디스크 입/출력으로 인한 지연을 최소화하였다.

지금까지 설명한 바와 같이 본 논문에서는 대용량의 공간 데이터를 처리하기 위한 기존의 상용 DBMS의 확장이 초래하는 추가적인 비용 문제와 상호운용성 문제를 해결하고자 GEOS를 사용하여 “SQL을 위한 심플 피쳐 명세”에서 제시한 모든 공간 데이터 타입과 공간 연산자를 지원하는 OpenGIS 공간 인터페이스를 개발하였다. 또한 성능 평가를 통해 본 논문에서 개발한 OpenGIS 공간 인터페이스가 Oracle Spatial 보다 응답 시간에서 우수한 성능을 갖는다는 것도 입증하였다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 관련 연구로 OGC에서 제시한 “SQL을 위한 심플 피쳐 명세”와 Oracle Spatial, 그리고 GEOS에 대해 살펴본다. 제 3 장과 제 4 장에서는 각각 OpenGIS 공간 인터페이스의 설계 및 구현에 대해 상세히 설명하고, 제 5 장에서는 성능 평가 결과를 기술한다. 마지막으로, 제 6 장에서는 결론 및 향후 연구 과제를 언급한다.

## 2. 관련 연구

본 장에서는 표준 인터페이스를 정의하고 있는 “SQL을 위한 심플 피쳐 명세”를 분석하고, 본 논문에서 개발한 OpenGIS 공간 인터페이스와 비교 대상이 되는 Oracle Spatial을 설명한다. 그리고 본 논문에서 OpenGIS 공간 인터페이스 개발에 사용된 GEOS에 대해서 살펴본다.

### 2.1 SQL을 위한 심플 피쳐 명세

OGC에서 제안한 “SQL을 위한 심플 피쳐 명세”는 ODBC API를 통해 전달되는 심플 공간 피쳐 집합의 저장, 검색, 질의, 그리고 갱신을 지원하는 표준 SQL 스키마를 정의하고 있다[5,6,7]. 심플 피쳐는 공간과 비공간 속성을 모두 지원하기 위해 OGC의 “OpenGIS 추상 명세(OpenGIS Abstract Specification)” [17]에 의해 정의되고 있는데, 심플 피쳐는 점 사이의 선형 보간법을 이용한 2차원 공간 데이터를 기반으로 한다. <표 1>은 “SQL을

위한 심플 피쳐 명세”에서 정의하고 있는 공간 데이터 타입 중 인스턴스화 할 수 있는 타입들의 WKT(Well-Known Text) 표현을 보여준다.

<표 1> 공간 데이터 타입의 WKT 표현

공간 데이터 타입	WKT 표현
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	MULTIPOINT (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)) )
GeometryCollection	GEOMETRYCOLLECTION ( POINT (10 10), POINT (30 30), LINestring (15 15, 20 20) )

“SQL을 위한 심플 피쳐 명세”에서 정의하고 있는 공간 함수는 객체 간의 공간 관계를 검사하기 위한 SQL 함수, 공간 분석을 위한 SQL 함수, 그리고 각 타입에 대한 SQL 함수로 나눌 수 있다. <표 2>는 “SQL을 위한 심플 피쳐 명세”에서 정의하고 있는 표준 공간 함수들을 보여준다.

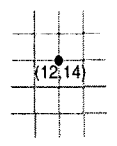
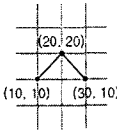
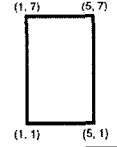
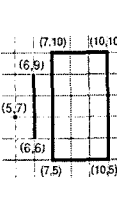
<표 2> 표준 공간 함수

공간 관계 검사를 위한 SQL 함수	각 타입에 대한 SQL 함수
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)	IsSimple(Geometry)
Contains(Geometry, Geometry)	Boundary(Geometry)
Relate(Geometry, Geometry, PatternMatrix)	Envelope(Geometry)
	X(Point)
	Y(Point)
	StartPoint(Curve)
	EndPoint(Curve)
	IsClosed(Curve)
	IsRing(Curve)
	Length(Curve)
	NumPoints(LineString)
	PointN(LineString, Integer)
	Centroid(Surface)
	PointOnSurface(Surface)
	Area(Surface)
	ExteriorRing(Polygon)
	NumInteriorRing(Polygon)
	InteriorRingN(Polygon, Integer)
	NumGeometries(GeometryCollection)
	GeometryN(GeometryCollection, Integer)
	IsRing(MultiCurve)
	Length(MultiCurve)
	Centroid(MultiSurface)
	PointOnSurface(MultiSurface)
	Area(MultiSurface)
공간 분석을 위한 SQL 함수	
Distance(Geometry, Geometry)	
Intersection(Geometry, Geometry)	
Difference(Geometry, Geometry)	
Union(Geometry, Geometry)	
SymDifference(Geometry, Geometry)	
Buffer(Geometry, Double Precision)	
ConvexHull(Geometry)	

## 2.2 Oracle Spatial

Oracle Spatial은 Oracle 데이터베이스에서 공간 데이터를 저장, 접근 및 분석할 수 있도록 하는 함수와 프로시저의 통합된 집합이다[14]. Oracle Spatial에서 지원하는 공간 데이터 타입은 Point, LineString, Polygon 등의 기본적인 타입과 이들 타입의 집합으로 구성된 MultiPoint, MultiLineString, MultiPolygon, GeometryCollection 등의 타입을 지원한다. 또한 점 사이를 곡선으로 연결하는 Arc LineString, Arc Polygon, Circle 등과 같은 타입도 지원한다. 그러나 Oracle Spatial에서는 지원하는 공간 데이터 타입이 많아지면서 이를 표현하는 방법이 복잡해졌다. <표 3>은 기본적인 공간 데이터 타입인 Point, LineString, Polygon과 이들 타입의 집합으로 구성된 GeometryCollection 타입의 WKT 표현과 Oracle Spatial 표현을 보여준다.

<표 3> 공간 데이터의 WKT와 Oracle Spatial 표현

	WKT Point(12 14)	Oracle Spatial SDO_GEOMETRY( 2001, NULL, SDO_POINT_TYPE(12, 14, NULL) NULL, NULL, NULL )
	WKT LINESTRING(10 10, 20 20, 30 10)	Oracle Spatial SDO_GEOMETRY( 2002, NULL, NULL, SDO_ELEM_INFO_ARRAY(1, 2, 1), SDO_ORDINATE_ARRAY(10, 10, 20, 20, 30, 10) )
	WKT POLYGON((1 7, 5 7, 5 1, 1 1, 1 7))	Oracle Spatial SDO_GEOMETRY( 2003, NULL, NULL, SDO_ELEM_INFO_ARRAY(1, 1003, 1), SDO_ORDINATE_ARRAY(1, 7, 5, 7, 5, 1, 1, 1, 7) )
	WKT GEOMETRYCOLLECTION( POINT (5 7), LINESTRING (6 9, 6 6), POLYGON ((7 10, 10 10, 10 5, 7 5, 7 10)) )	Oracle Spatial SDO_GEOMETRY( 2004, NULL, NULL, SDO_ELEM_INFO_ARRAY(1, 1, 1, 3, 2, 1, 7, 1003, 1), SDO_ORDINATE_ARRAY(5, 7, 6, 9, 6, 6, 7, 10, 10, 10, 10, 5, 7, 5, 7, 10) )

Oracle Spatial에서 공간 객체는 SDO\_GEOMETRY 타입으로 저장되는데 SDO\_GEOMETRY 타입의 구조는 <표 3>에서 볼 수 있듯이 5개의 속성을 갖는다. 첫 번째 속성은 데이터 타입을 나타내고, 두 번째 속성은 SRID(Spatial Reference system Identifier)를 나타내며, 세 번째 속성은 Point 타입을 정의할 때 사용된다. 그리고 네 번째 속성은 다섯 번째 속성으로 오는 가변 길이 좌표의 객체 별 시작 위치와 타입을 나타내는데 사용되고, 다섯 번째 속성은 공간 객체의 좌표값을 나타낸다.

Oracle Spatial PL/SQL API는 공간 연산자, 프로시저, 그리고 함수 형태로 공간 연산을 지원한다. 그러나 Oracle Spatial은 “SQL을 위한 심플 피쳐 명세”에서 정의하고 있는 표준 공간 함수들을 모두 지원하지 않고 있다. Oracle Spatial은 공간 분석을 위한 SQL 함수에서는 표준 공간 함수들을 모두 지원하지만 객체 간의 공간 관계를 검사하는 함수에서는 Overlaps와 Crosses를 지원하지 않는다. 그리고 Oracle Spatial은 타입 별 공간 함수의 경우 총 31개의 표준 공간 함수 중 14개의 공간 함수만을 지원하고 있다. Oracle Spatial이 지원하지 않는 공간 함수는 사용자가 PL/SQL을 이용하여 외부 프로시저 형태로 생성될 수 있다.

## 2.3 GEOS

GEOS는 JTS(Java Topology Suite)를 C++ 언어로 포팅한 공개 소스이다[15]. JTS는 공간 객체 모델과 기본적인 기하(geometric) 함수를 제공하는 JAVA로 작성된 API이다. GEOS는 이와 같은 JTS의 공간 객체 모델과 모든 함수를 완벽히 지원하는 것을 목표로 한다. 이러한 GEOS는 OGC의 “SQL을 위한 심플 피쳐 명세”에서 정의한 Geometry 모델을 기반으로 하는 데이터 타입들과 DE-9IM의 패턴 매트릭스를 기반으로 하는 공간 Predicates, Overlay 함수(Intersection, Difference, Union, Symmetric Difference), Buffer, Convex Hull, 면적과 거리 함수(Area, Distance), 그리고 위상적 평가(topological validity) 함수 등을 포함

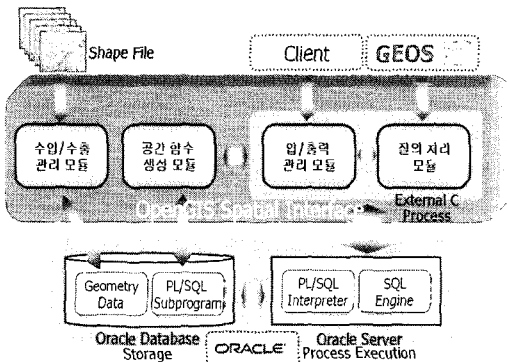
하고 있다. GEOS는 공유 라이브러리 형태로 컴파일되고, 현재 PostgreSQL을 공간 데이터 처리를 위해 확장한 PostGIS의 Geometry 엔진으로 사용되고 있다.

### 3. OpenGIS 공간 인터페이스의 설계

본 장에서는 Oracle 상에서 GEOS를 사용하는 OpenGIS 공간 인터페이스의 전체 구조를 살펴보고, OpenGIS 공간 인터페이스를 구성하고 있는 입/출력 관리 모듈, 질의 처리 모듈, 공간 함수 생성 모듈, 수입/수출 관리 모듈의 설계에 대하여 설명한다.

#### 3.1 전체 구조

OpenGIS 공간 인터페이스의 전체 구조는 <그림 1>과 같다.



<그림 1> OpenGIS 공간 인터페이스의 전체 구조

<그림 1>에서 볼 수 있듯이 OpenGIS 공간 인터페이스는 상용 DBMS인 Oracle을 기반으로 사용자로부터의 질의 입력과 결과 출력을 담당하는 입/출력 관리 모듈, 실제적인 공간 연산을 처리하는 질의 처리 모듈, 공간 함수를 Oracle에 등록하는 공간 함수 생성 모듈, 그리고 Oracle과 Shape 파일 간의 입/출력을 담당하는 수입/수출 관리 모듈로 구성되어 있다.

#### 3.2 입/출력 관리 모듈

입/출력 관리 모듈의 역할은 크게 사용자로부터 입력된 질의 구문의 유효성 검사, 유효한 질의에 대하여 대응되는 함수 호출, 그리고 사용자에게 호출된 함수의 결과 값 반환으로 나눌 수 있다. 먼저 질의 구문의 유효성을 검사하기 위하여 질의 구문을 파싱하고 올바르지 않은 토큰(token)에 대한 예외 처리가 이루어진다. 여기서 사용될 수 있는 데이터 타입명은 “SQL을 위한 심플 피쳐 명세”에서 제시된 인스턴스화 할 수 있는 데이터 타입명인 Point, LineString, Polygon, MultiPoint, Multi-LineString, MultiPolygon, GeometryCollection 이고, 사용 가능한 공간 함수명은 Oracle 예약어인 Overlaps와 Union을 제외하고 “SQL을 위한 심플 피쳐 명세”에 제시된 함수명과 동일하다. Overlaps와 Union은 External Procedure를 의미하는 “Ex\_”를 접두어로 추가하여 Ex\_Overlaps, Ex\_Union으로 사용하였다. 이와 같이 입/출력 관리 모듈은 질의 구문에서 사용된 공간 함수의 유효성을 검사한 후 질의를 질의 처리 모듈에 전달한다.

#### 3.3 질의 처리 모듈

질의 처리 모듈은 입/출력 관리 모듈에서 전달받은 질의를 처리하기 위해 실제적인 공간 연산을 수행하는 모듈이다. 이 모듈은 요청된 질의에 따라 공간 연산을 수행한 후 결과를 다시 입/출력 관리 모듈로 전달한다.

질의 처리 모듈은 “SQL을 위한 심플 피쳐 명세”에 제시된 모든 공간 함수를 지원하도록 설계되었다. 따라서 이 모듈에서 지원하는 함수들은 표준 공간 함수의 분류와 같이 공간 관계 함수, 공간 분석 함수, 그리고 각 타입 별 함수로 나눌 수 있다. 이때 공간 관계 함수와 공간 분석 함수의 매개 변수로 사용되는 데이터는 “SQL을 위한 심플 피쳐 명세”에서 명시된 SQL Geometry 데이터 모델의 최상위 클래스인 Geometry 타입으로 변환되어야 한다. 그리고 각 타입 별 함수의 매개 변수는 각 타입

에 맞도록 변환되어야 한다. 이러한 과정을 거친 후 질의 처리 모듈은 실제적인 연산을 수행할 때 공간 객체 모델과 기본적인 Geometry 함수를 제공하는 GEOS API를 이용한다. 마지막으로 질의 처리 모듈은 공간 연산을 수행한 결과를 입/출력 관리 모듈로 반환한다.

### 3.4 공간 함수 생성 모듈

공간 함수 생성 모듈은 입/출력 관리 모듈과 질의 처리 모듈을 Oracle이 호출할 수 있도록 PL/SQL 부프로그램으로 등록하는 모듈이다. <그림 1>의 OpenGIS 공간 인터페이스의 전체 구조에서 볼 수 있듯이 입/출력 관리 모듈과 질의 처리 모듈은 외부 C 프로세스 형태로 설계되었다. 공간 함수 생성 모듈은 이들 모듈을 Oracle에서 지원하는 Create 구문을 이용해 PL/SQL 부프로그램으로 등록하고, 등록 정보를 데이터 사전(data dictionary)에 저장한다.

등록된 PL/SQL 부프로그램은 Oracle 데이터베이스 저장소(디스크)에 저장되어 있다가 SQL문을 통한 사용자의 호출이 있을 때 Oracle 공유 풀(메모리)에 적재된다. Oracle 공유 풀은 LRU(Least Recently Used) 알고리즘을 사용하므로 시간이 흐름에 따라 PL/SQL 부프로그램이 페이지 아웃될 수 있다. 공간 함수 생성 모듈은 공유 풀에서 페이지ング으로 인한 디스크 입/출력과 메모리 공간의 단편화 또는 비연속적인 메모리 할당에 의한 에러를 피하기 위하여 등록된 PL/SQL 부프로그램을 Oracle 공유 풀에 상주시키는 피닝을 사용한다.

### 3.5 수입/수출 관리 모듈

수입/수출 관리 모듈은 Oracle 데이터베이스 외부에 위치한 Shape 파일로부터 공간 데이터를 추출하여 Oracle 내부에 저장하고, 다시 Oracle 내부에서 공간 데이터를 추출하여 Shape 파일로 저장하는 모듈이다. Shape 파일은 공간 피쳐를 위한 비위상적 Geometry 정보와 속성 정보를 저장하고 있다. 수입/수출 관리 모듈은 Shape 파일을

Oracle에 저장할 때 공간 데이터뿐만 아니라 공간 데이터의 빠른 검색을 위한 MBR(Minimum Boundary Rectangle) 정보도 저장한다.

## 4. OpenGIS 공간 인터페이스의 구현

본 장에서는 Oracle 상에서 GEOS를 사용하는 OpenGIS 공간 인터페이스의 구현 환경을 살펴보고, OpenGIS 공간 인터페이스를 구성하고 있는 입/출력 관리 모듈, 질의 처리 모듈, 공간 함수 생성 모듈, 수입/수출 관리 모듈의 구현에 대하여 설명한다.

### 4.1 구현 환경

본 논문에서 개발한 OpenGIS 공간 인터페이스의 구현을 위하여 운영체제는 레드햇 엔터프라이즈 리눅스 v3, 상용 DBMS는 리눅스 버전의 Oracle 10g(10.2.0.1.0), 개발 도구는 리눅스에서 제공하는 g++ 3.2.3 버전과 Oracle이 제공하는 절차적 언어인 PL/SQL을 사용하였다. 그리고 공간 데이터 처리를 위한 API를 제공하는 GEOS는 2.2.3 버전을 사용하였다.

### 4.2 입/출력 관리 모듈

입/출력 관리 모듈은 사용자로부터 입력된 구문이 올바른지를 판단하기 위해 구문 분석을 수행한다. 구문 분석에서 사용되는 토큰은 인스턴스화 할 수 있는 모든 공간 데이터 타입명과 Ex\_Overlaps 와 Ex\_Union을 포함한 모든 표준 공간 함수명이다. 구문 분석에서 에러가 없을 경우 제어를 질의 처리 모듈로 넘기게 된다. 이와 같은 작업은 GEOS의 WKT 파서 클래스와 WKB 파서 클래스를 이용하였으며 관련 API는 <표 4>와 같다.

<표 4>의 파서 클래스에서 제공하는 API는 Geometry 타입의 매개 변수를 갖는 공간 함수에서 사용된다. 이 API는 사용자로부터의 입력 문자열을 GEOS가 사용하는 Geometry 타입으로 변환해준다.

<표 4> 입/출력 관련 GEOS API

클래스	멤버 함수	설명
geos::WKTReader	WKTReader (const GeometryFactory *gf)	GeometryFactory로 파서를 초기화
	Geometry * read (string Well-Known Text)	WKT 문자열을 파싱하여 Geometry로 변환
geos::WKTWriter	string write (const Geometry *geometry)	Geometry를 WKT 문자열로 변환
geos::WKBReader	Geometry * read (istream &is)	Input stream으로부터 Geometry를 추출
	ostream & printHEX(istream &is, ostream &os)	Input stream을 HEX 코드로 변환하여 출력
geos::WKBWriter	void write (const Geometry &g, ostream &os)	Geometry를 out stream으로 출력

입/출력 관리 모듈에서는 Geometry 타입을 매개 변수로 갖는 공간 함수마다 지원하는 공간 데이터 타입이 다르기 때문에 <표 4>의 파서 클래스에서 제공하는 API를 이용하여 입력된 문자열로부터 인스턴스화 할 수 있는 공간 데이터 타입을 추출하고 현재 공간 함수가 추출한 공간 데이터 타입을 지원하는지 여부를 판단한다. 그리하여, 공간 함수가 추출한 공간 데이터 타입을 지원하면 적합한 공간 데이터 타입으로 변환하고 변환된 공간 데이터 타입을 질의 처리 모듈로 전달한다.

### 4.3 질의 처리 모듈

질의 처리 모듈은 입/출력 관리 모듈에서 넘어온 Geometry 또는 Point, LineString, Polygon 등의 공간 데이터 타입으로 변환된 공간 데이터를 받아 실제적으로 사용자로부터 요청된 연산을 수행하고 연산 결과를 다시 입/출력 관리 모듈로 전달하는 모듈이다. 또한, 질의 처리 모듈은 연산 수행 과정에서 발생하는 에러를 처리한다. 질의 처리 모듈에서 사용된 GEOS API는 <표 5>와 같다.

질의 처리 모듈에서 공간 함수는 반환 값에 따라 처리가 다르다. 먼저 Contains 함수와 같이 True/False를 반환하는 공간 Predicate는 정수 (True일 때 1, False일 때 0, Unknown일 때 -1)를 반환하기 때문에 반환 값을 입/출력 관리 모듈에 바로 전달할 수 있다. 그러나 Union 함수나 Overlay 함수는 Geometry 객체를 반환하기 때문에 반환 값을 입/출력 관리 모듈로 전달하기 위해 Oracle이 처리할 수 있는 데이터 타입으로 변환해야 한다.

<표 5> 공간 함수 관련 GEOS API

공간 관계 검사를 위한 SQL 함수	각 타입에 대한 SQL 함수
Geometry::equals(Geometry) Geometry::disjoint(Geometry) Geometry::touches(Geometry) Geometry::within(Geometry) Geometry::overlaps(Geometry) Geometry::crosses(Geometry) Geometry::intersects(Geometry) Geometry::contains(Geometry) Geometry::relate(Geometry, PatternMatrix)	Geometry::getDimension() Geometry::GeometryType() read(string)/write(Geometry) read(istream)/write(Geometry, ostream) Geometry::getSRID() Geometry::isEmpty() Geometry::isSimple() Geometry::getBoundary() Geometry::getEnvelope() Point::getX(Point) Point::getY(Point) LineString::getStartPoint() LineString::getEndPoint() LineString::isClosed() LineString::isRing()
공간 분석을 위한 SQL 함수	Geometry::getLength() LineString::getNumPoints(LineString) LineString::getPointN(Integer) Geometry::getCentroid() Geometry::getInteriorPoint() Geometry::getArea() Polygon::getExteriorRing() Polygon::getNumInteriorRing() Polygon::getInteriorRingN(Integer) GeometryCollection::getNumGeometries() GeometryCollection::getGeometryN(Integer) MultiLineString::isClosed()
Geometry::distance(Geometry) Geometry::Intersection(Geometry) Geometry::Difference(Geometry) Geometry::Union(Geometry) Geometry::SymDifference(Geometry) Geometry::Buffer(Integer) Geometry::ConvexHull()	

### 4.4 공간 함수 생성 모듈

공간 함수 생성 모듈은 공유 라이브러리 형태로 개발된 입/출력 관리 모듈과 질의 처리 모듈을 Oracle에서 호출할 수 있도록 하는 PL/SQL 프로시저들로 구성된다. 일반적으로 Oracle에서 공간 함수를 생성되기 위해서 Create Function 구문을 이용하지만 공간 함수의 효율적인 관리와 메모리 공간 확보를 위해 Create Package 구문을 이용하여 패키지 형태로 공간 함수들을 캡슐화하였다. <그림 2>는 공간 함수 생성 모듈에서 사용된 Create Package 구문을 보여준다.

```

CREATE OR REPLACE PACKAGE OSI AS
FUNCTION Equals (input1 VARCHAR2, input2 VARCHAR2) RETURN BINARY_INTEGER;
FUNCTION Contains (input1 VARCHAR2, input2 VARCHAR2) RETURN BINARY_INTEGER;
:
:
FUNCTION NumGeometries (input VARCHAR2) RETURN BINARY_INTEGER;
FUNCTION GeometryN (input VARCHAR2, N BINARY_INTEGER) RETURN VARCHAR2;
END OSI;
    
```

<그림 2> Create Package 구문

<그림 2>에서 보는 것과 같이 Create Package 구문은 패키지에 담고자하는 프로시저, 함수, 그리고 다른 프로그램 객체들을 선언한다. 그리고 선언된 것들은 Create Package Body 구문을 이용하여

여 패키지 몸체(body)에서 정의된다. <그림 3>은 패키지 생성 구문에서 선언된 함수들 중 일부에 대한 Create Package Body 구문이다.

```
CREATE OR REPLACE PACKAGE BODY OSI AS

FUNCTION Equals(input1 IN VARCHAR2, input2 IN VARCHAR2) RETURN BINARY_INTEGER
AS EXTERNAL LIBRARY myOperators
NAME 'wkiEquals'
LANGUAGE C
PARAMETERS(input1, input2);

.

.

FUNCTION GeometryN(input IN VARCHAR2, N IN BINARY_INTEGER) RETURN VARCHAR2
AS EXTERNAL LIBRARY myOPERATORS
NAME 'wkiGeometryN'
LANGUAGE C
PARAMETERS(input, N);

END OSI;
```

<그림 3> Create Package Body 구문

공간 함수 생성 모듈에서 생성된 패키지의 이름은 패키지에 정의된 공간 함수를 호출할 때에 접두어로 사용된다. 예를 들어, <그림 2>와 <그림 3>에서 생성된 패키지의 이름이 OSI이면 OSI 패키지에 정의된 Contain() 함수를 사용하려면 OSI.Contains()와 같이 호출한다. 공간 함수 생성 모듈에서 생성된 패키지는 Oracle의 시스템 카탈로그에 등록되고 호출되면 공유 풀로 적재된다. 하지만 공유 풀은 LRU 알고리즘을 이용하기 때문에 시간이 지남에 따라 패키지가 페이지 아웃될 수 있다. 본 논문에서는 이러한 페이지 교환으로 인한 디스크 입/출력을 최소화하고 메모리 단편화와 비연속적인 메모리 할당에 의한 에러를 피하기 위해 패키지를 공유 풀에 피닝되도록 구현하였다.

#### 4.5 수입/수출 관리 모듈

수입/수출 관리 모듈은 공간 데이터를 저장하고 있는 Shape 파일에서 Geometry 속성을 추출하여 Oracle로 입력하는 로더 형태로 개발되었다. 먼저 Shape 파일에서 데이터 타입을 확인한 후 가변 길이의 좌표값들을 조합하여 WKT 또는 WKB 형태의 Geometry 데이터를 생성한다. 이러한 공간 데이터를 Oracle 데이터베이스에 저장하기 위한 테이블 스키마는 <그림 4>와 같다.

```
CREATE TABLE <table name> (
    GID NUMBER NOT NULL PRIMARY KEY,
    XMIN <ordinate type>,
    YMIN <ordinate type>,
    XMAX <ordinate type>,
    YMAX <ordinate type>,
    WKB_GEOMETRY VARBINARY,
    <attribute name> <attribute type>
)
```

<그림 4> Geometry 테이블 스키마

Shape 파일은 기본적으로 공간 속성을 담고 있는 shp 파일과 공간 속성에 대한 인덱스 정보를 담고 있는 shx 파일, 그리고 비공간 속성을 담고 있는 dbf 파일로 구성되어 있다. 수입/수출 관리 모듈은 <그림 4>의 Geometry 테이블 스키마에 정의된 공간 속성들을 shx 파일에서 추출하여 Oracle 데이터베이스에 저장한다. 반면에 dbf 파일에 저장되어 있는 비공간 속성들은 제공되는 Shape 파일마다 종류와 개수가 다를 수 있기 때문에 Geometry 테이블 스키마도 달라질 수 있다. 따라서 수입/수출 관리 모듈은 dbf 파일의 헤더에서 비공간 속성의 종류와 개수를 확인하고 이를 Geometry 테이블 스키마에 반영한 후 Oracle 데이터베이스에 저장한다.

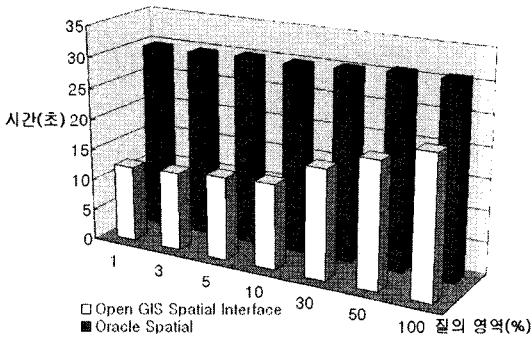
### 5. 성능 평가

본 장에서는 OpenGIS 공간 인터페이스의 정확도와 성능을 검증하기 위해 Oracle Spatial과 성능 평가를 실시한 결과를 보여준다. 성능 평가에 사용된 공간 데이터는 서울시 강동구 건물 정보를 담고 있는 Shape 파일을 Oracle Spatial에서 지원하는 데이터 타입과 WKT 표현으로 각각 변환하여 사용하였다.

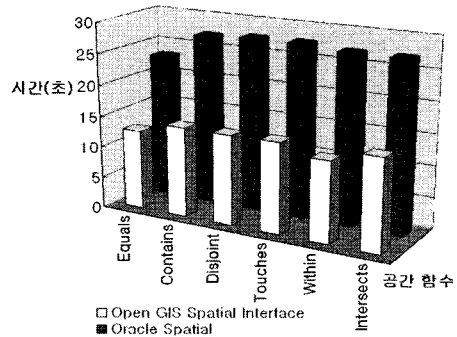
먼저 GIS 분야에서 가장 사용 빈도가 높은 Contains 함수의 성능 평가를 실시하였다. 24,774개의 Polygon 데이터에 대해서 각각 1%, 3%, 5%, 10%, 30%, 50%, 100%의 사각형 질의 영역에 대한 Contains 관계를 묻는 질의의 응답 시간을 비교한 결과는 <그림 5>와 같다.

<그림 5>의 결과와 같이 Oracle Spatial은 질





<그림 5> Contains 함수의 질의 영역별 응답 시간 비교



<그림 6> 공간 관계 테스트 함수의 응답 시간 비교

의 영역과 상관없이 거의 일정한 응답 시간이 소요 되는 반면, OpenGIS 공간 인터페이스를 사용한 결과는 질의 영역이 작을수록 빠른 응답 시간을 보이고 있다. 그리고 평균적으로 OpenGIS 공간 인터페이스는 Contains 연산에서 Oracle Spatial과 비교하여 약 2배의 빠른 응답 시간을 보였다.

Oracle Spatial에서 지원하지 않는 Overlaps와 Crosses를 제외한 나머지 공간 관계를 테스트하는 함수들의 성능 평가는 Contains 함수의 테스트와 동일한 조건에서 10%의 사각형 질의 영역에 대한 각각의 공간 관계를 만족하는 객체의 개수를 알아 보는 질의를 사용하였다. 수행 결과의 응답 시간을 비교하면 <그림 6>과 같다.

<그림 6>과 같이 성능 평가가 가능한 여섯 개의 공간 관계를 테스트하는 함수의 응답 시간은 Contains 함수의 성능 평가와 마찬가지로 OpenGIS 공간 인터페이스를 사용한 경우가 Oracle Spatial을 사용한 경우보다 약 2배 빠른 응답 시간을 얻을 수 있었다.

이러한 결과가 나온 원인들은 다음과 같이 분석되었다. 먼저 Oracle Spatial이 지원하는 공간 데이터 타입들은 <표 3>에서처럼 데이터를 표현하기 위한 속성들이 다소 복잡하고 많은데 반해 OpenGIS 공간 인터페이스의 공간 데이터 타입들은 WKT 표현을 그대로 사용함으로써 데이터 처리량 및 메모리 저장 공간을 줄일 수 있었다. 그리고 공간 데이터를 대상으로 한 공간 연산에서 Oracle

Spatial의 공간 함수들은 대상 객체에 대해 어떠한 필터링도 수행하지 않는데 반해 OpenGIS 공간 인터페이스의 공간 함수들은 객체의 MBR 연산을 통한 필터링을 수행하기 때문에 보다 빠른 응답 시간을 얻을 수 있었다. 또한, OpenGIS 공간 인터페이스는 공간 함수 패키지 크기를 Oracle Spatial의 공간 함수 패키지의 1/5로 줄임으로써 PL/SQL 부 프로그램을 공유 풀에 적재하는 시간과 응답 시간을 빠르게 하였고 공유 풀에 적재된 패키지를 피닝 함으로써 디스크 입/출력을 크게 줄일 수 있었다.

## 6. 결론 및 향후 연구 과제

오늘날 공간 데이터 수집 수단이 획기적으로 발전하고 그 활용 범위가 다양해지면서 여러 분야에서 공간 데이터가 활용되고 있다. 또한 이러한 대용량의 공간 데이터를 효율적으로 처리하기 위해 기존의 DBMS를 확장하는 연구가 활발히 진행되고 있다. 그러나 기존 상용 DBMS 상에서 공간 데이터 처리를 위한 확장을 사용하려면 추가적인 비용이 소요되고, 이들 시스템에서 사용되는 공간 데이터 타입과 공간 연산자의 종류가 서로 다르기 때문에 상호운용성 문제를 초래하고 있다.

따라서 본 논문에서는 이러한 기존의 상용 DBMS의 확장에서의 추가적인 비용 문제와 상호운용성 문제를 해결하기 위해 Oracle 상에서 “SQL을 위한 심플 피쳐 명세”의 표준 인터페이스를 지

원하는 OpenGIS 공간 인터페이스를 개발하였다. 본 논문에서 개발한 OpenGIS 공간 인터페이스는 공간 객체 모델과 기본적인 Geometry 함수를 제공하는 GEOS API를 사용하였으며, 디스크 입/출력으로 인한 수행 지연을 막기 위해 공간 함수를 OSI 패키지로 캡슐화하여 Oracle의 공유 풀에 피닝하였다. 또한, OpenGIS 공간 인터페이스는 Oracle Spatial과의 성능 평가를 통해 응답 시간에서 약 2배의 향상된 성능을 보임을 입증하였다. 앞으로는 OpenGIS 공간 인터페이스에서 빠른 검색을 위한 공간 인덱스의 적용 방안에 대한 연구가 필요하다.

### 참고문헌

1. 김정준, 강홍구, 윤재관, 한기준, “모바일 장치를 위한 공간 MMDBMS의 설계 및 구현,” 한국정보과학회 학술발표논문집, 31권2호, 2004, pp.73-75.
2. 신중수, 김동오, 강홍구, 박춘결, 한기준, “Oracle 기반의 대용량 지도 서비스를 위한 XServer의 설계 및 구현,” 한국공간정보시스템학회 추계학술대회논문집, 7권3호, 2005, pp.47-54.
3. Guting, R., “An Introduction to Spatial Database Systems,” Journal of VLDB, Vol.3, No.4, 1994, pp.357-399.
4. 엄정호, 장재우, “공간 네트워크 데이터베이스를 위한 저장 및 색인 구조의 구현 및 성능 평가,” 한국정보과학회 학술발표논문집, 32권1호, 2005, pp.121-123.
5. OpenGIS Consortium, Inc., *Simple Features Specification For SQL 1.1*, 1999.
6. Ryden, K., *OpenGIS Implementation Specification for Geographic Information—Simple Feature Access—Part 1: Common Architecture*, Open Geospatial Consortium, Inc., 2005.
7. Ryden, K., *OpenGIS Implementation Specification for Geographic Information—Simple Feature Access—Part 2: SQL Option*, Open Geospatial Consortium, Inc., 2005.
8. Cha, S., Kim, K., Lee, B., Song, C., Hwang, S., and Kwon, Y., “MEADOW: A Middleware for Efficient Access to Multiple Geographic Databases through OpenGIS Wrappers,” Journal of Software – Practice and Experience, Vol.32, No.4, 2002, pp.377-402.
9. Schneider, M., and Behr, T., “Topological Relationships between Complex Spatial Objects,” ACM Transactions on Database Systems, Vol.31, No.1, 2006, pp.39-81.
10. 김복원, 남광우, 이종훈, 류근호, “Plane-sweep 알고리즘을 이용한 OpenGIS 공간 연산자의 설계 및 구현,” 한국정보과학회 춘계학술발표논문집, 26권1호, 1999, pp.107-109.
11. Clementini, E., Sharma, J., and Egenhofer, M., “Modeling Topological Spatial Relations: Strategies for Query Processing,” Journal of Computer and Graphics, Vol.18, No.6, 1994, pp.815-822.
12. Kothuri, R., Ravada, S., and Xu, W., “Spatial Processing using Oracle Table Functions,” Proc. of the International Conference on Data Engineering, 2003, pp.851-856.
13. Yingcheng, L., and Ling, L., “Research on Spatial Database Design and Tuning Based on Oracle and ArcSDE,” Proc. of the International Society for Photogrammetry and Remote Sensing, Vol.35, No.4, 2004, pp.370-375.
14. Murray, C., *Oracle Spatial User’s Guide and Reference 10g Release 2 (10.2)*, Oracle, 2005.

15. Ramsey, P., *The State of Open Source GIS*, Refrations Research, Inc., 2006.
16. Ashdown, L., *Oracle Database Application Developer' s Guide – Fundamentals 10g Release 2 (10.2)*, Oracle, 2005.
17. OpenGIS Consortium, Inc., *The OpenGIS Abstract Specification Topic 5: Features Version 4*, 1999.

**박춘걸**

2002년 중국 칭화대학교 기계공학과(공학사)  
 2005년 대구대학교 대학원 정보통신공학과(공학석사)  
 2005년~현재 건국대학교 대학원 컴퓨터공학과 박사  
 과정  
 관심분야 : 공간 데이터베이스, GIS, LBS, 유비쿼터스 컴퓨팅

**박희현**

2005년 건국대학교 컴퓨터공학과(공학사)  
 2007년 건국대학교 대학원 컴퓨터공학과(공학석사)  
 관심분야 : 공간 데이터베이스, GIS, LBS

**강홍구**

2002년 건국대학교 컴퓨터공학과(공학사)  
 2004년 건국대학교 대학원 컴퓨터공학과(공학석사)  
 2004년~현재 건국대학교 대학원 컴퓨터공학과 박사  
 과정  
 관심분야 : 공간 데이터베이스, GIS, LBS, USN, 센서 데이터베이스

**한기준**

1979년 서울대학교 수학교육학과(이학사)  
 1981년 KAIST 전산학과(공학석사)  
 1985년 KAIST 전산학과(공학박사)  
 1990년 Stanford 대학 전산학과 Visiting Scholar  
 2000년~2002년 한국정보과학회 데이터베이스연구회  
 운영위원장  
 2004년~2006년 한국공간정보시스템학회 회장  
 2004년~현재 한국정보시스템감리사협회 회장  
 1985년~현재 건국대학교 컴퓨터공학부 교수  
 관심분야 : 공간 데이터베이스, GIS, LBS, 텔레매틱스, 정보시스템 감리