

1. 튜토리얼

OpenGIS 공간 데이터 모델과 코바 구현 명세

2

1998. 7. 2

홍 봉 희
부 산 대 학 교



Open GIS 공간 데이터 모델과 코바 구현 명세

부산대학교 컴퓨터공학과

98년 7월

홍 봉 희

1

표준화와 기술발전

2

- ✦ **convergence** 대 **divergence**
 - 표준화는 **convergence**
 - 기술 발전은 **divergence**

- ✦ GIS 국제 표준을 통한 GIS 신기술 개발
 - 상호 연동 지원
 - 자바, 인터넷, 코바, OLE/COM 분산 환경을 위한 객체 지향 GIS 응용 S/W 개발

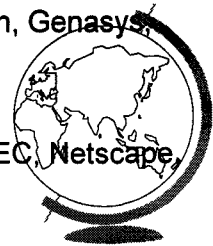
- ✦ GIS 신산업 창출, 새로운 GIS 시장 영역 확대



Open GIS Consortium (OGC)은 무엇인가?

3

- ◆ 상호 연동 지리 정보 처리(interoperable geoprocessing)를 위한 industry-wide architecture 표준 명세 수립
- ◆ GIS SW 공급자 중심의 국제 표준
 - 상호 연동이 가능한 GIS 도구의 시장 영역 확대와 세계 시장의 분점
- ◆ 주요 구성원:
 - 기존 GIS 산업계: ESRI, Intergraph, Laser Scan, Genasys, MapInfo, Smallworld, Bentley
 - DB 산업계: Oracle, Informix,
 - 정보기술 산업계: Autodesk, DEC, HP, IBM, NEC, Netscape, Sun



OGC Mission

4

- ◆ 공급자, integrators, 학계, 정부기관, 표준 기구 포함
 - interoperable geoprocessing 기술 명세의 개발
 - ◆ 코바, 자바, OLE/COM, ODBC 분산 환경을 모두 고려한 미들웨어 인터페이스의 추상 명세 정의
 - 표준 인터페이스 명세를 **superset**으로 정의한 문서
 - ◆ GIS 엔진 또는 대형 공간 DB 엔진을 위한 표준 명세가 아님
 - certifiably interoperable products의 개발 보급
 - ◆ 코바, 자바, OLE/COM, ODBC 분산 환경을 위한 구현 명세 정의
 - 표준 인터페이스인 추상 명세의 **subset**
 - 분산 환경의 미들웨어 인터페이스의 표준화



OGC Mission

5

- ◆ 개방형 시스템, 분산 처리와 componentware frameworks 에 기초한 정보 기술과 geoprocessing 기술의 융합
- ◆ 분산된 geoprocessing을 관련된 산업계 공동 개발을 촉진하기 위한 산업체 forum 제공
- ◆ 우리의 대책:
 - 국내에서 GIS 엔진이나 공간 DB 엔진을 국산 제품화 개발이 쉽지 않음
 - 이미 국내에 기존 legacy GIS 제품이 많이 사용되고 있음
 - 도전 가능한 분야:
 - ◆ 새로운 국제 표준에 맞추어 국산화 개발이 가능하고, 외국에 판매할 수 있는 OpenGIS 미들웨어, GIS 클라이언트 도구의 개발
 - ◆ GIS의 신산업 추세: 인터넷, 자바, 코바 등의 분산 환경의 객체 지향 OpenGIS-compliant S/W의 제품화



OpenGIS 용어 정의

6

- ◆ OpenGIS
 - 개방형 그리고 상호 연동 가능한 지리 정보 처리
 - 분산 환경에서 이질적인 geodata와 geoprocessing 자원의 원격 공유와 실행
- ◆ 상호연동(interoperable)
 - 하나의, 변경되지 않는 운용 인터페이스로 다양한 이질적인 자원을 접근, 공유할 수 있는 것
 - 이질적인 자원에 대한 표준 인터페이스를 지원하는 것
 - OGIS : geodata와 geoprocessing 서비스에 대한 표준 인터페이스 제공
 - ◆ geodata 접근, geodata 교환, 분산된 geoprocessing 연산을 지원



OpenGIS 용어 정의

7

- ◆ OpenGIS Specification ("OGIS"): geodata 공유와 geoprocessing 상호 연동을 가능하게 하는 소프트웨어 명세
- ◆ Open GIS Consortium, Inc.
 - OpenGIS 기술의 개발과 분산된 geoprocessing을 enterprise computing에 통합 기술 개발을 위한 산업체 forum



Why Join OGC?

8

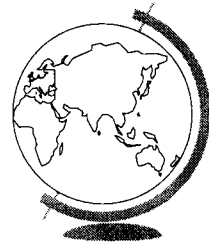
- ◆ 분산된 geoprocessing 응용의 개발을 위한 실제적 표준을 만듦
- ◆ interoperable geoprocessing 제품을 위한 시장 영역 확대
- ◆ interoperability 명세의 표준에 반영
 - 국내 산업계 표준(?)을 국제 표준에 반영
- ◆ 컨소시움의 제품 개발에 참여
 - 국제 표준을 토대로 한 제품의 시장 영역 확대
 - common market growth objectives



기술 개발 개요

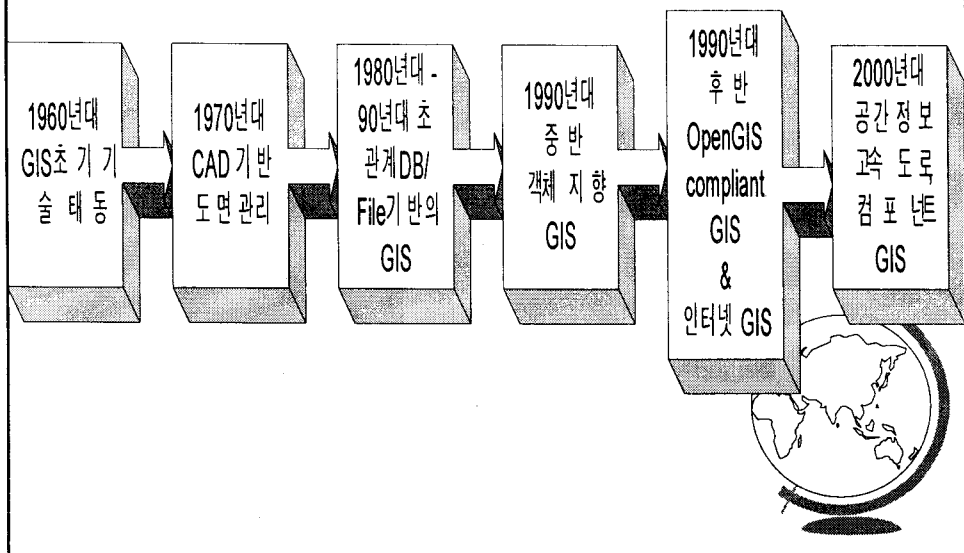
9

- ◆ interoperable geoprocessing을 위한 기술 명세를 정의
 - 특정 computing platforms에 독립적인 **Abstract Specification**을 정의
 - ◆ 포괄적 표준 명세의 정의
 - 포괄적 표준 명세를 특정 computing platforms에 한정된 **Implementation Specifications**로 정의
 - ◆ Abstract Specification이 변경될 수 있음
 - 코바 구현 명세
 - OLE/COM 구현 명세
 - ODBC 구현 명세



GIS의 발전 과정

10

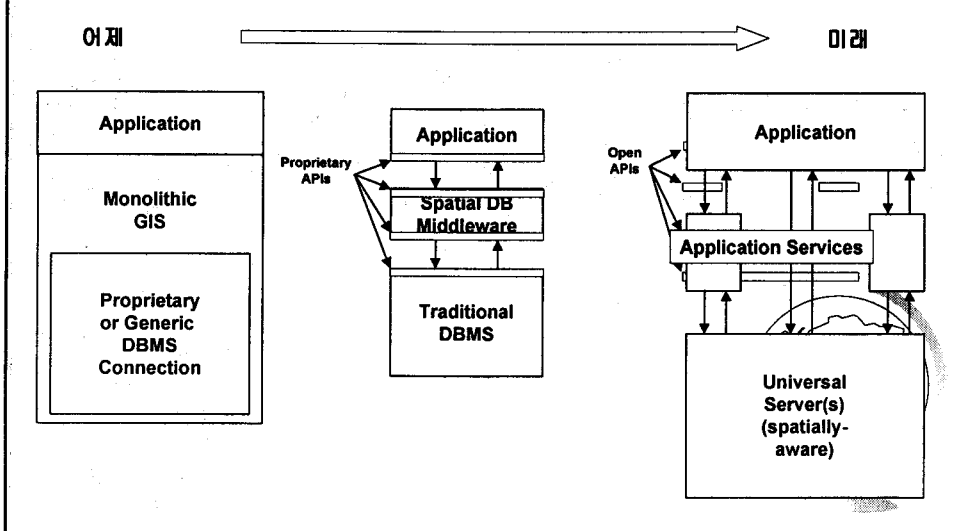


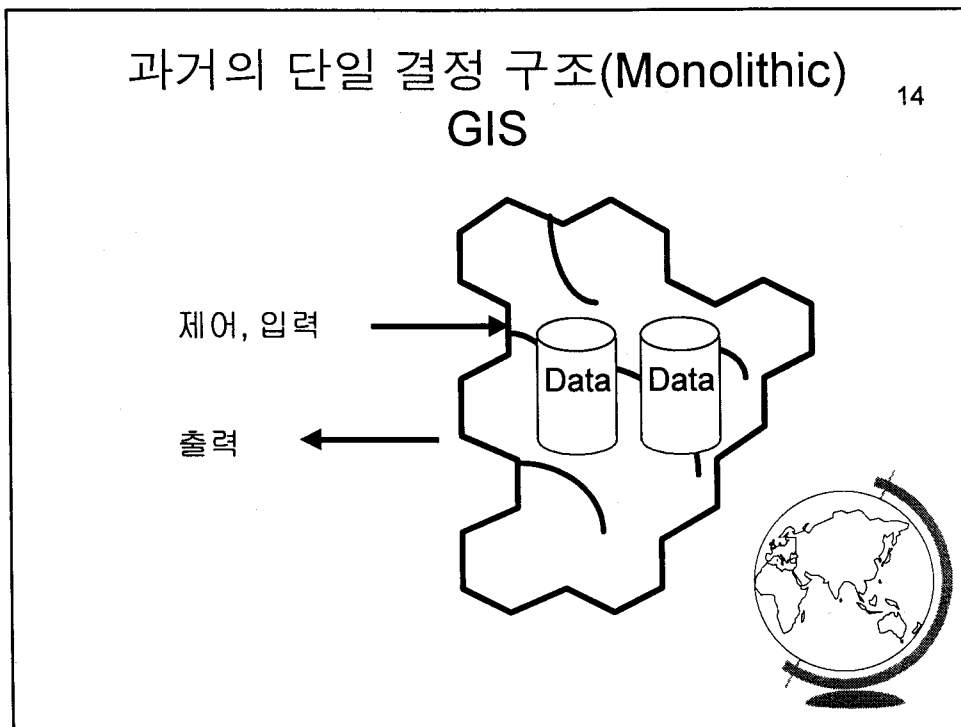
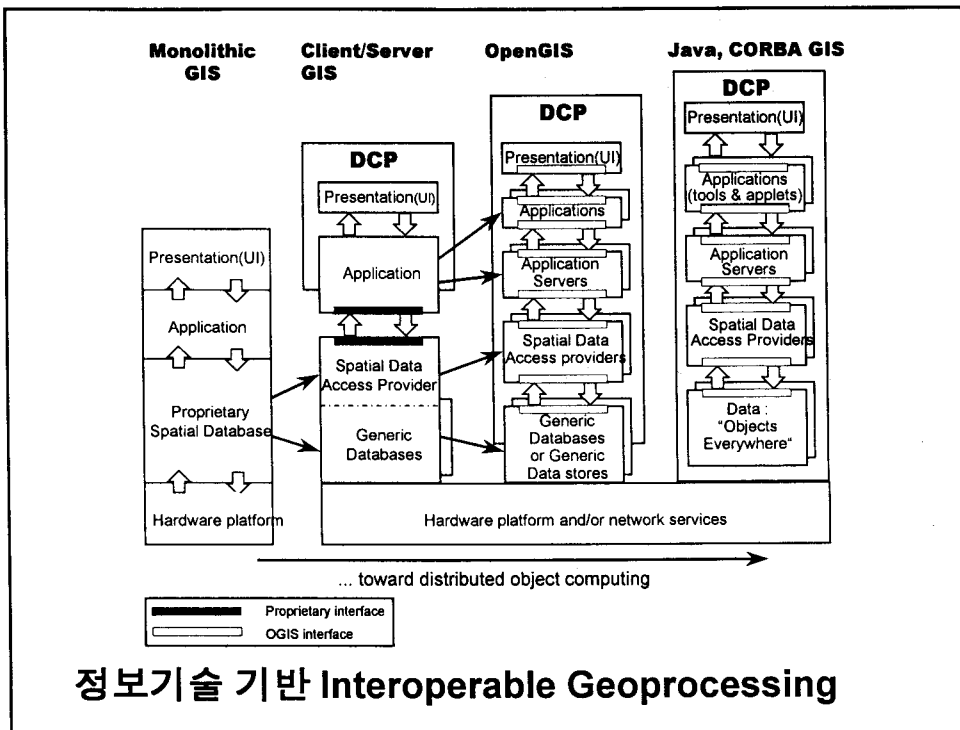
과거 및 현재 GIS 기술

- ◆ 기존의 monolithic GIS
 - 모든 layer가 강결합(tightly coupled)
 - ◆ proprietary spatial DB 사용
 - ◆ application과 spatial DB가 하나의 시스템으로 구성
 - 데이터 전송
 - ◆ 동종의 monolithic system 간에는 데이터 교환
 - ◆ 서로 다른 시스템간에는 데이터 번역
 - ◆ format conversions
 - SDTS, SAIF, GeoTIFF, DXF 등
- ◆ 클라이언트-서버 (two tiers) GIS
 - closed interface (proprietary API) 사용
 - spatial Data Access Provider 사용
 - generic DB 사용



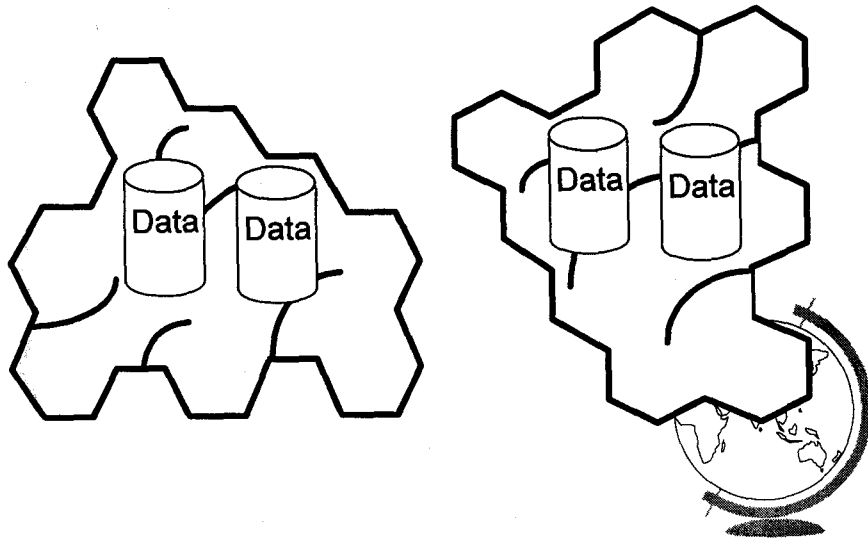
Closed GIS에서 Open GIS 로 이행





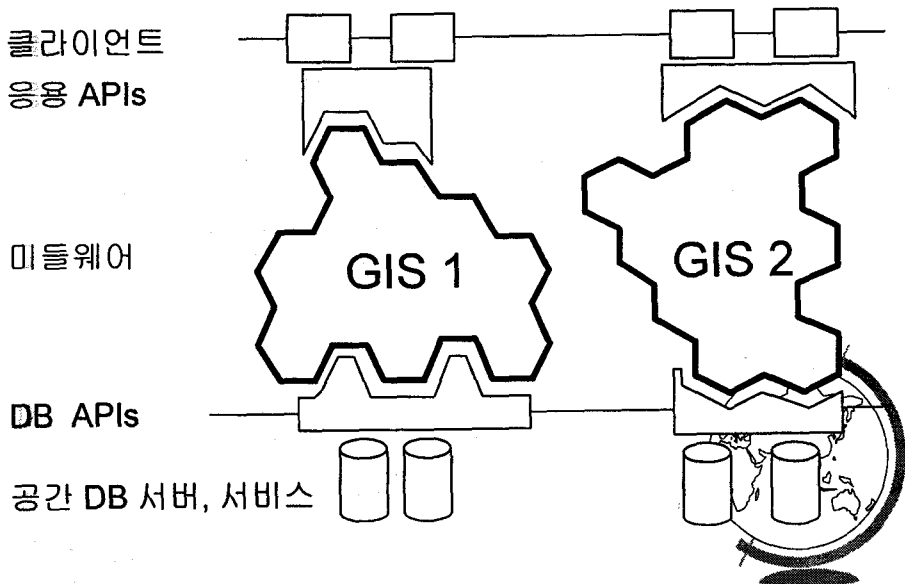
공급자마다 서로 다른 단일 결정 구조의 GIS

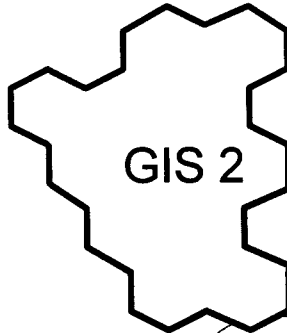
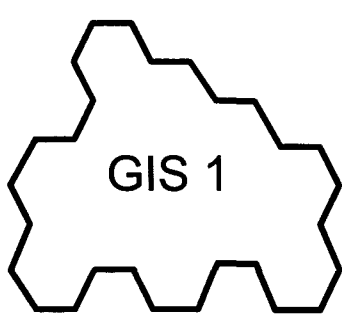
15



미래의 3-Tier(계층) 구조의 GIS

16

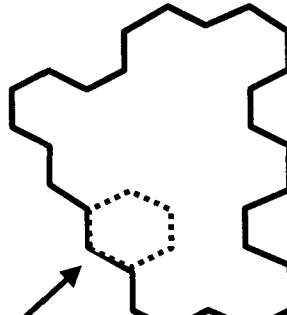
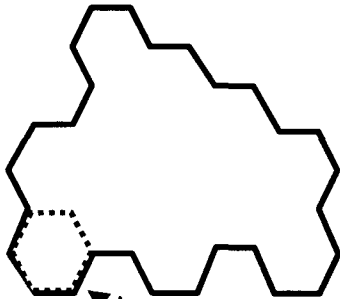




- 아직 상호연동이 지원되지 않음
- 공급자마다 서로 다른 미들웨어



공통된 함수(기능)을 추출

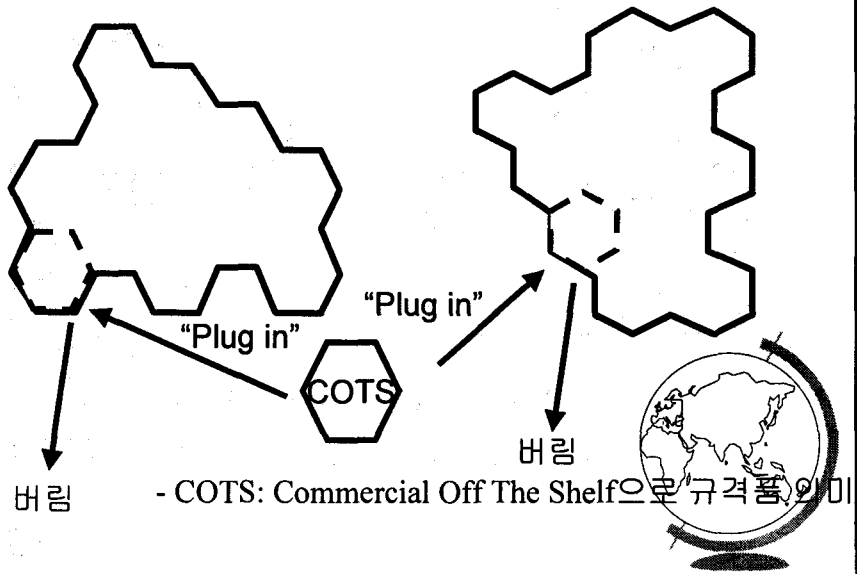


- 공통 모델링 요소(예: Simple Feature)를 추출
- 공통 geoprocessing 함수를 추출



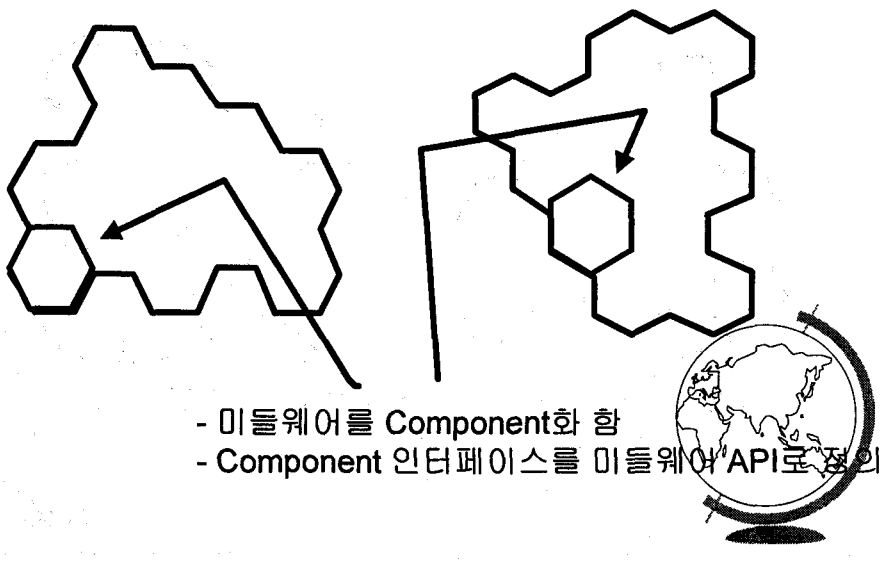
상호 연동 가능 구성 요소

19



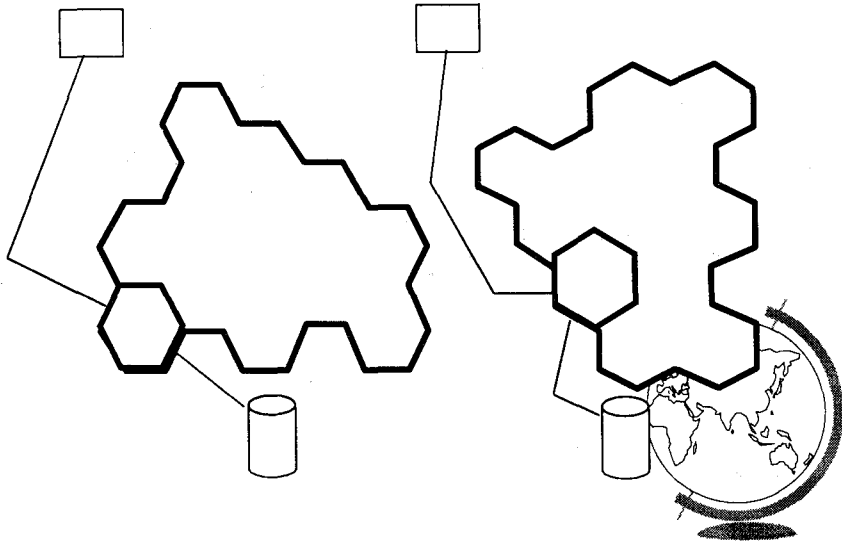
규격품의 인터페이스를 통일

20



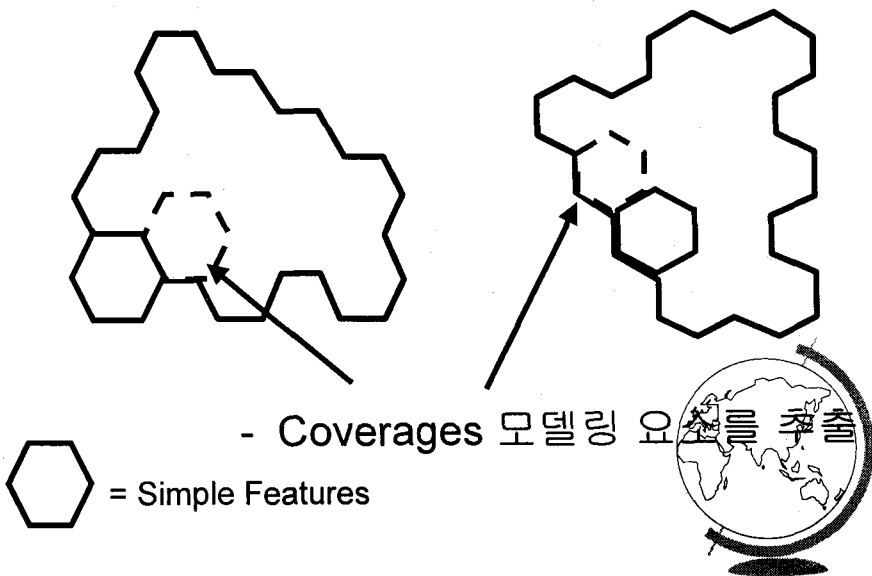
새로운 클라이언트/서버 지원

21



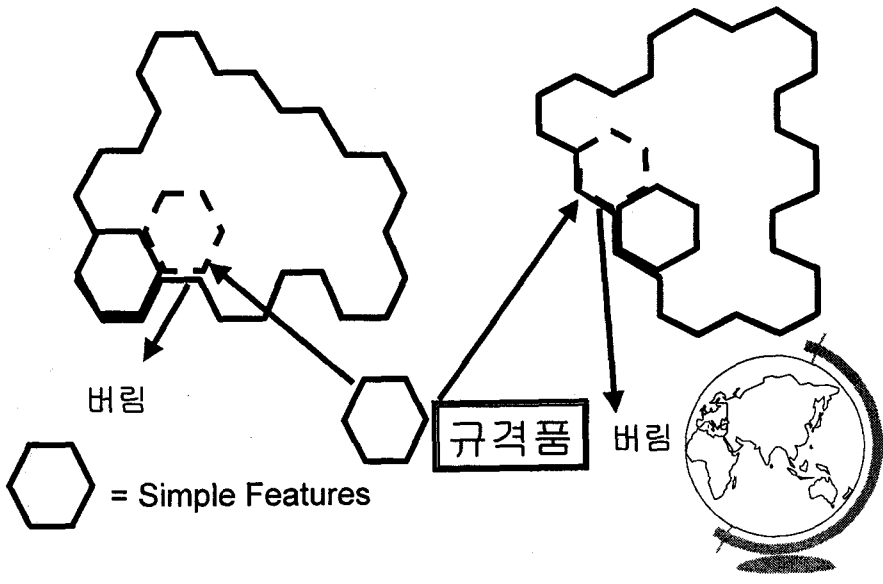
두번째 Component를 만들기

22



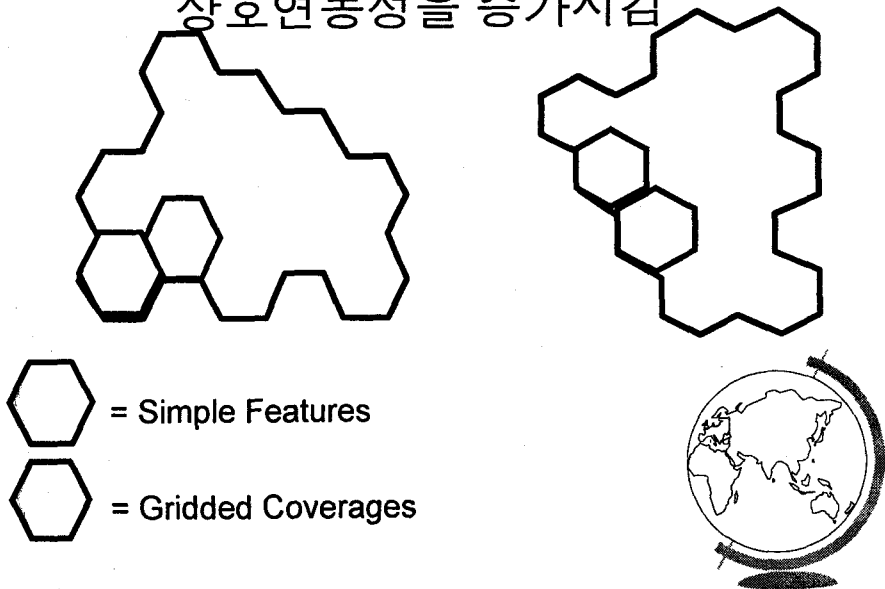
미들웨어를 규격품으로 대체

23



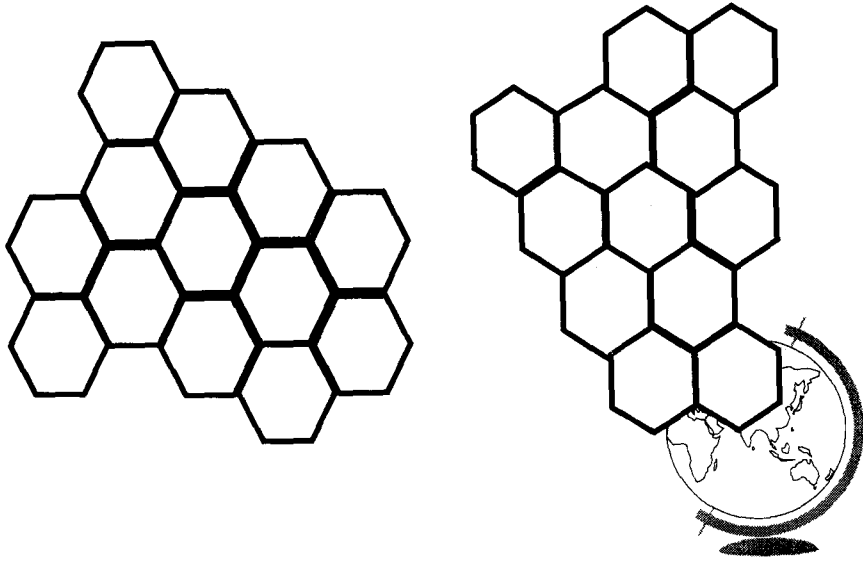
상호연동성을 증가시킴

24



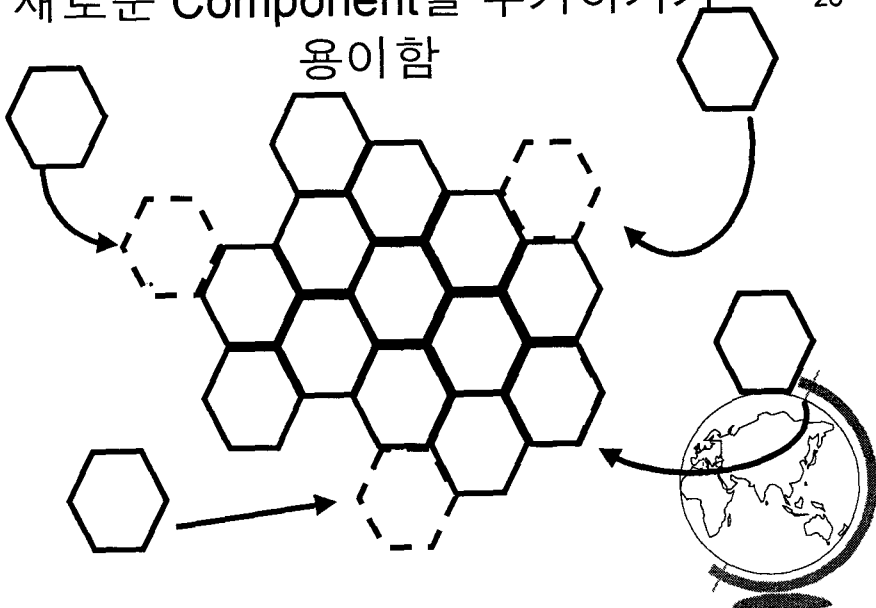
미들웨어를 모두 Component화 함

25



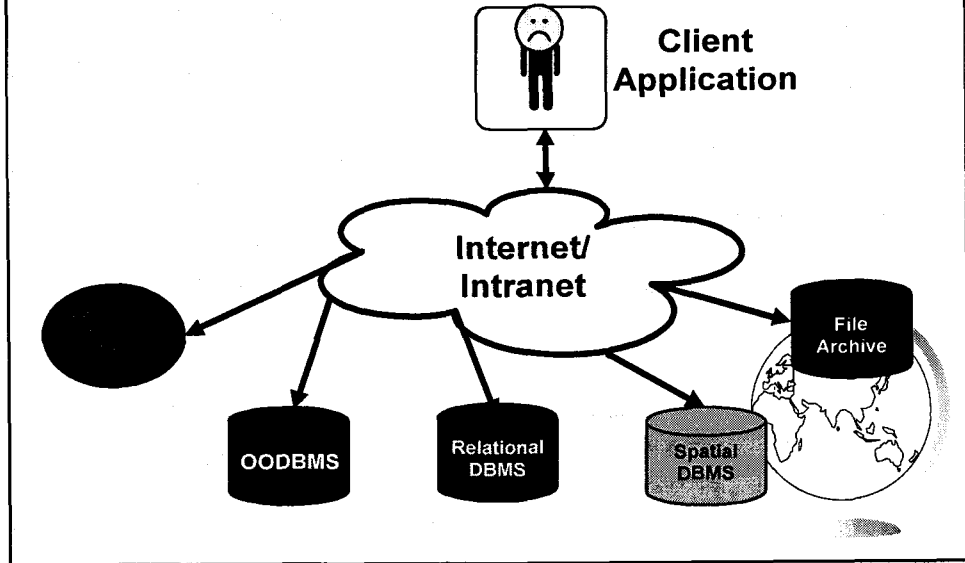
새로운 Component를 추가하기가 용이함

26



기존의 이질적인 Legacy “Enterprise Data”의 통합

27



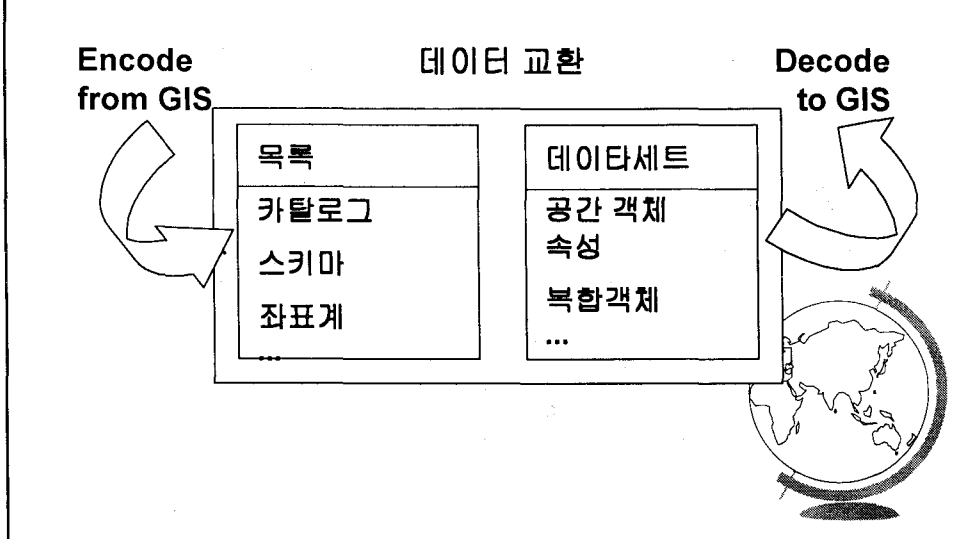
기존 이질적인 공간 DB의 통합 문제

28

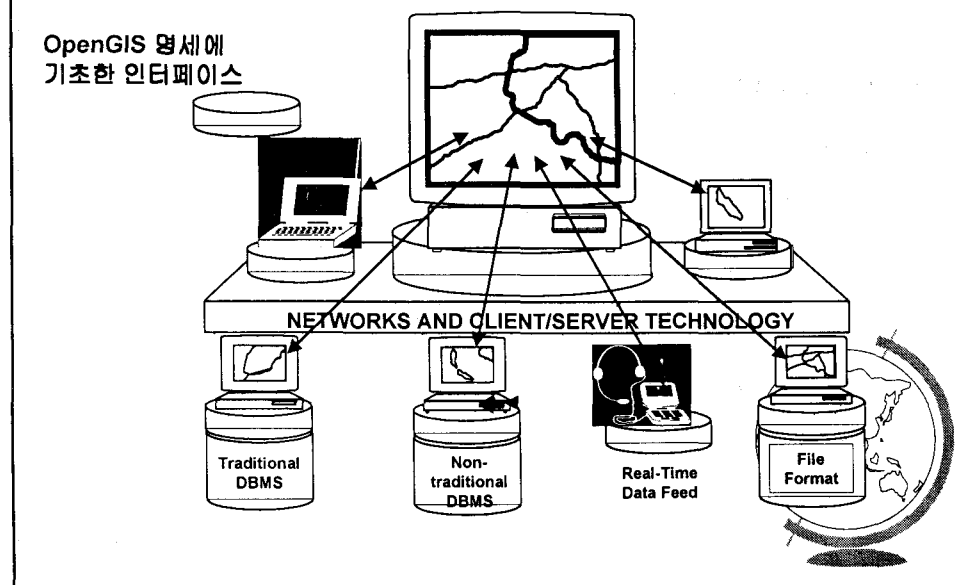
- ◆ 서로 다른 Legacy 데이터 출처
- ◆ 서로 다른 공간 데이터 모델 및 공간 함수
- ◆ 이질적인 platforms
- ◆ 분산된 데이터



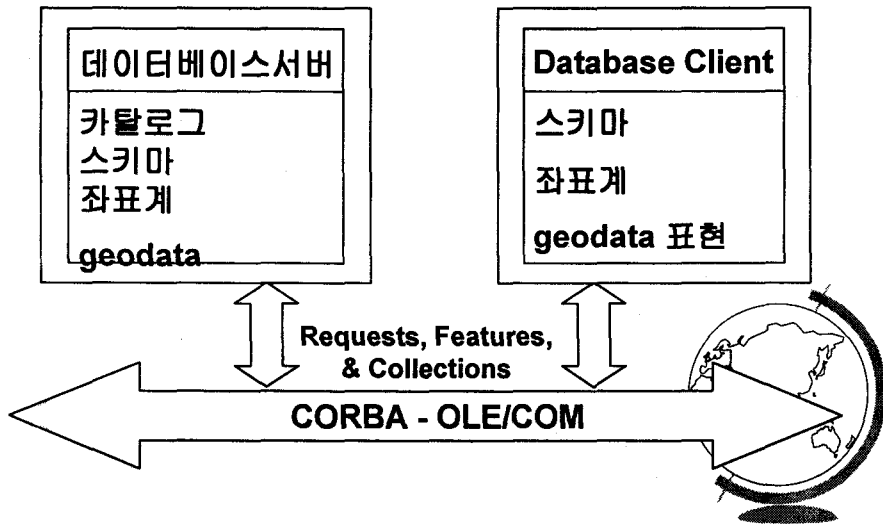
기존 데이터 교환 방법



OpenGIS 명세는 이질적인 Geodata에 대한 투명 접근을 가능하게 함



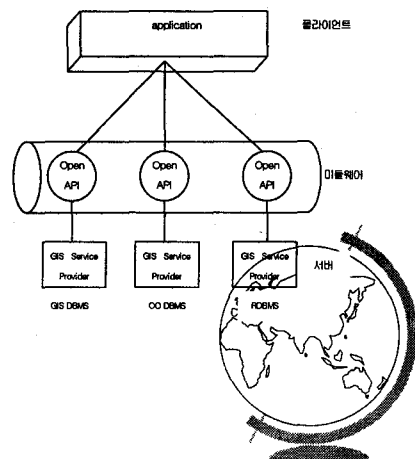
OpenGIS: 동적인 상호 연동을 지원



OpenGIS기반 Open API

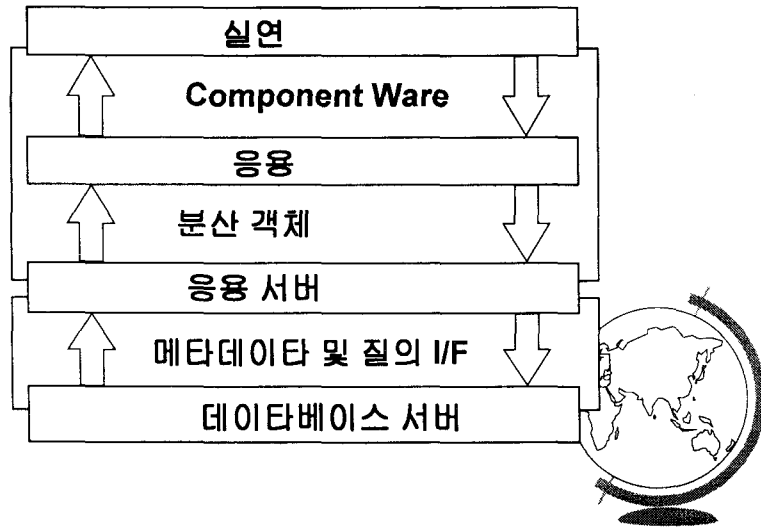
◆ open client/server 구조 설계

- application과 GIS service provider간의 미들웨어로써 표준 인터페이스에 대한 연구 필요
- application은 open API를 토대로 설계 구현



GIS의 레이어 구조

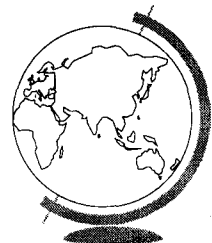
33

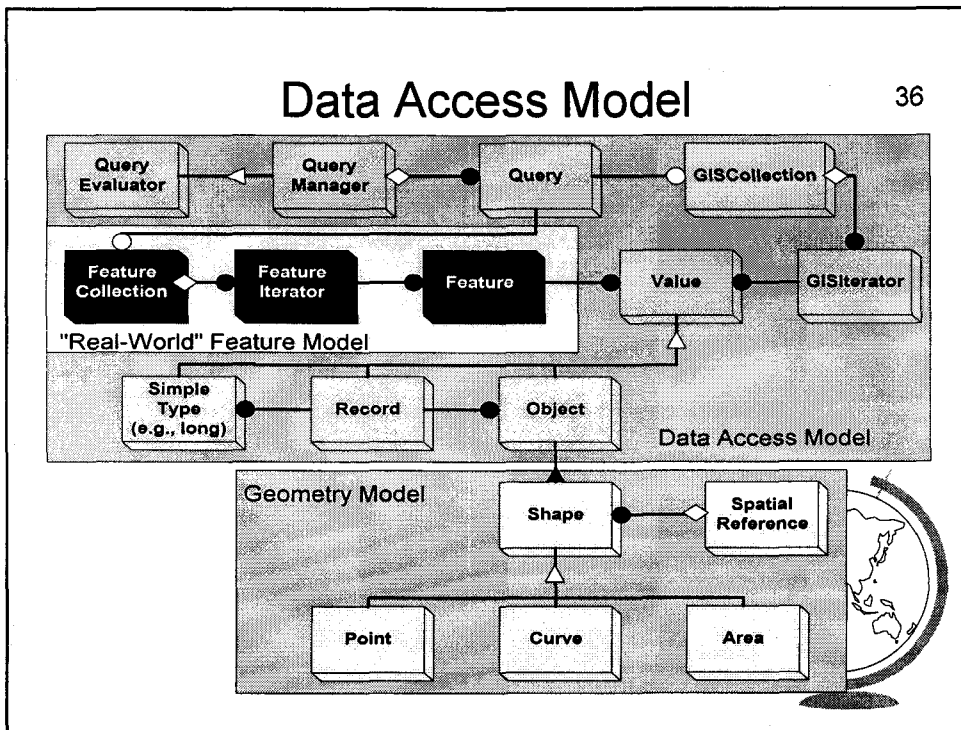
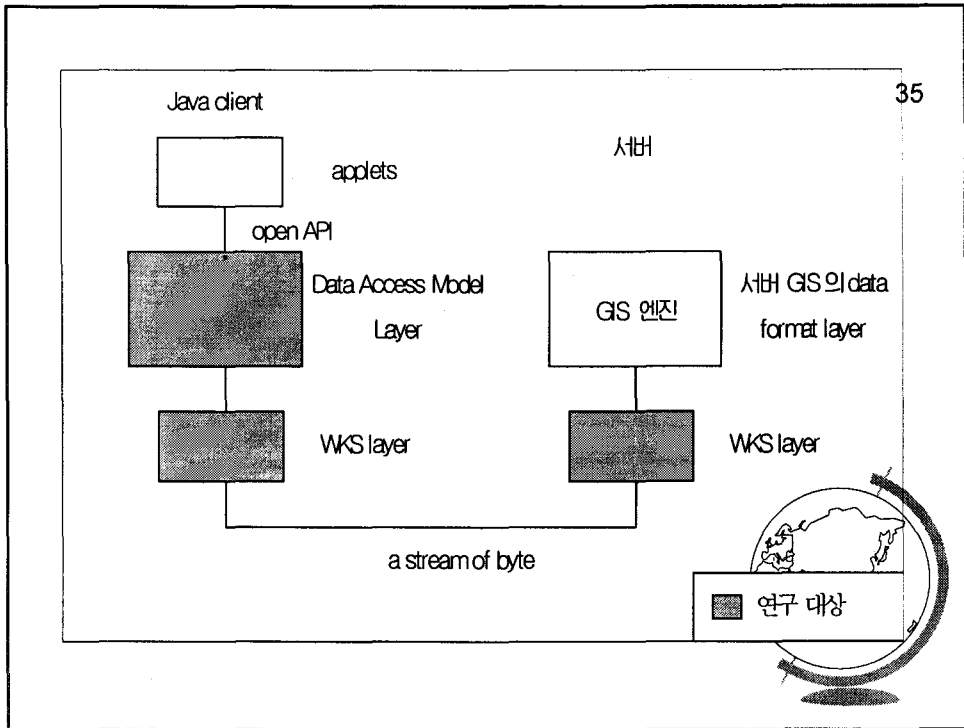


Open API의 표준 데이터 모델

34

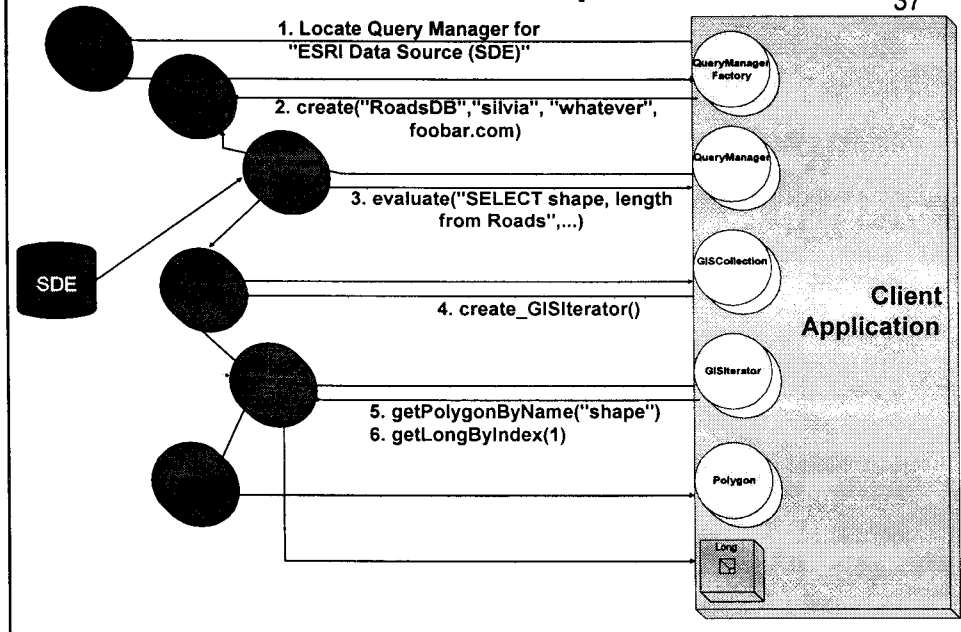
- ◆ 상호 운용을 위한 open API 설계
 - open API의 표준 데이터 접근 모델은 네트워크를 통해 전송되는 stream of byte를 decoding한 결과인 추상 인터페이스 제공
 - open API의 표준 데이터 모델로서 Data Access Model
 - ◆ "Real-World" Feature Model
 - ◆ Geometry Model





클라이언트와 서버 간의 Open API

37



OGIS 기본 골격

38

- ✦ OGIS: Open Geodata Interoperability 명세
 - geodata와 geoprocessing resources에 대한 분산 접근을 위한 GIS 응용 소프트웨어 framework의 포괄적 명세
 - 다른 OGIS-compliant 소프트웨어와 interoperate하는 소프트웨어를 개발하기 위한 공통된 표준 인터페이스 명세
- ✦ 누가 OGIS를 사용하나?
 - OGIS-compliant S/W 개발자
 - ◆ 표준 geodata model과 표준 OGIS 서비스 인터페이스를 사용
 - 공간 정보 유통자: 공간 데이터 provider라고 함
 - ◆ OGIS information community간의 semantic 번역기
 - Integration 개발
 - ◆ interoperable한 응용 개발
 - ◆ 기존 legacy 시스템과의 통합 문제



Interoperable 지리정보처리를 위한 정보기술

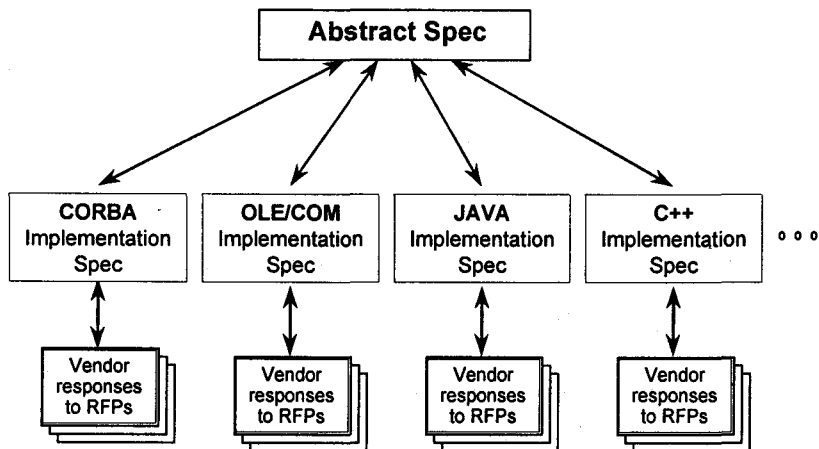
39

- ◆ 정보기술(Information technology) 사용 이유
 - “interoperability through common specification”을 제공하기 위함
 - ◆ 관련 기술: 분산처리기술 + 객체 기술
- ◆ interoperability를 구현하는 방법
 - 표준 specification
 - interoperability를 위한 초석으로서 common geospatial types을 정의
 - ◆ 실현 방법: 객체 기술
 - 클라이언트-서버의 interoperability
 - pluggable components 개념
 - ◆ interoperable components을 이용하여 응용을 개발



OGIS는 DCP 환경과 독립

- ◆ OGIS는 abstract specification임

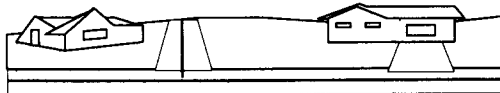


OpenGIS 추상 명세

- ✦ 현재 개요와 14개 주제를 다루고 있음
- ✦ 새로운 주제에 대한 명세 정의 작업중
- ✦ 주제별 명세
 1. Geometry Structures 2. Spatial Reference Systems
 3. Locational Geometry 4. Stored Functions/Interpolation
 5. The OpenGIS Feature and Feature Collections
 6. The Coverage 7. The Earth Image
 8. Feature Relationships 9. Quality
 10. Transfer Technology 11. Metadata
 12. Services Architecture 13. Catalog Service
 14. Semantics and Information Communities



Real World



	Cartographer's View	Tax Assessor's View
Conceptual Views		
OGM Views		

Conceptual Views and OGM views

3단계 객체 지향 설계

43

- ◆ essential model
 - 현실 세계를 기술
 - ◆ the Earth model
 - essential model의 구성 요소 : 객체 타입과 event 타입
- ◆ specification model(추상 명세에 해당)
 - 매우 높은 추상화 단계에서 소프트웨어를 기술
 - ◆ 특정 DCP에 관계없이 문서로 기술한 명세
 - OGIS가 a specification model임
- ◆ implementation model
 - 특정 DCP(예: OLE/COM, 코바)를 위한 구현 명세
 - ◆ 특정 DCP에서 구현된 소프트웨어가 아님
 - ◆ 특정 DCP에서 구현을 위한 인터페이스를 정의한 것
 - CORBA IDL(Interface Description Language)



OGIS를 위한 Earth Model

44

- ◆ 지구의 표현
- ◆ 모델링되는 지리 정보 요소
 - 엔티티(entities)
 - ◆ 비교적 잘 정의된 경계나 spatial extent를 갖는 인식 가능한 객체
 - geometry, 속성, 메타데이터로 기술
 - 현상(phenomena)
 - ◆ 특정 크기를 갖는 것이 아니라 공간상에서 연속적으로 변하는 성질
 - ◆ 공간 상의 특정 점에서 값을 가짐
 - 온도, soil composition, topography



Location: place and time

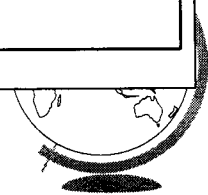
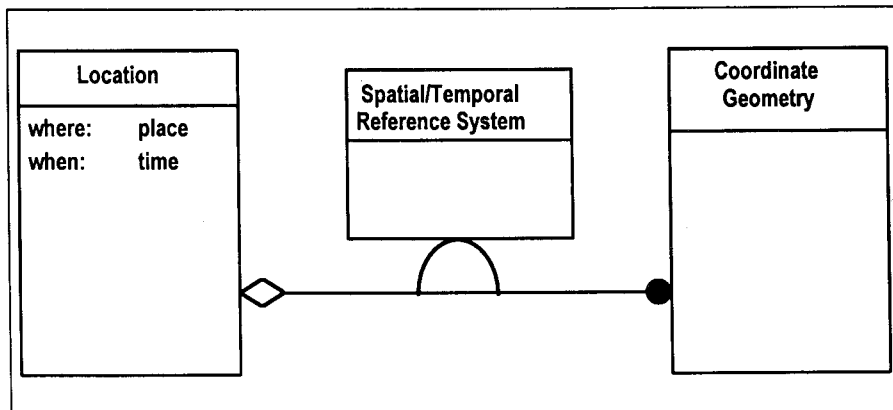
45

- ✦ 모든 entities와 phenomena는 지구상의 위치와 시간에 관계 있음
 - spatial/temporal reference systems
- ✦ 지구 측정과 좌표계는 계속 발전함
 - 지구상의 공간 좌표를 기술하는 방법은 여러가지가 있음
 - 서로 다른 시공간 참조 시스템(좌표계)간에는 변환
- ✦ place와 time을 표현하기 위해 "location"을 사용
 - 모든 entities와 phenomena는 location을 가짐



Location과 Geometry 모델

46



Features와 Coverages

47

◆ 두가지 기본적인 지리정보 타입:

- feature

- ◆ 현실 세계 엔터티의 표현

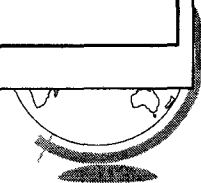
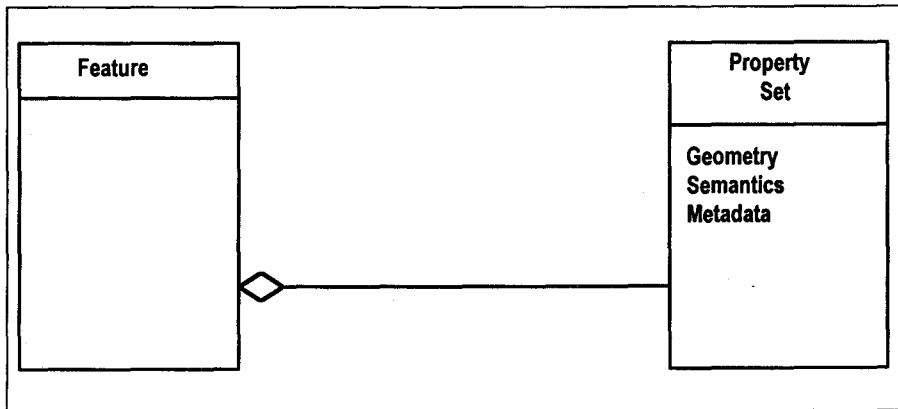
- coverage

- ◆ 시공간 도메인내의 한점에 대한 값을 부여



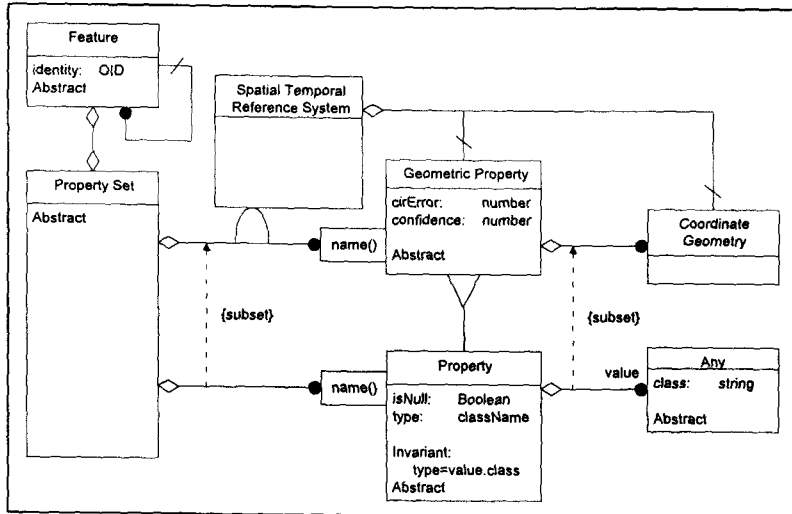
Feature Model

48



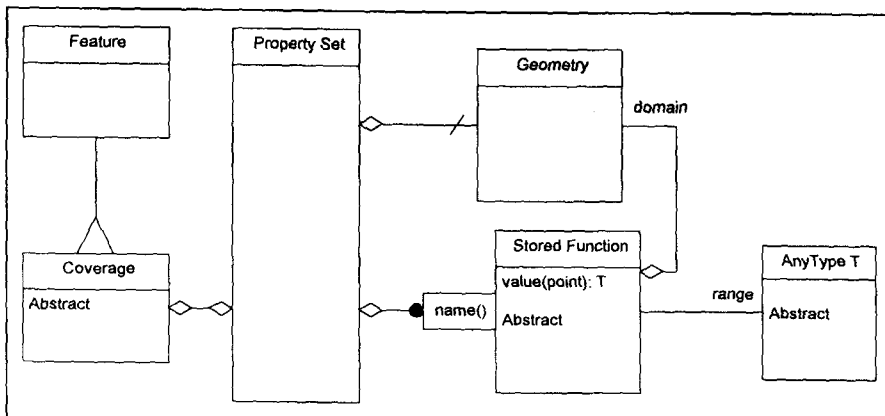
상세한 Feature Model

49



Coverage Model

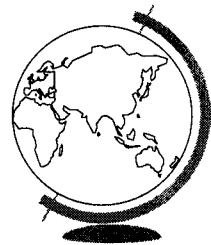
50



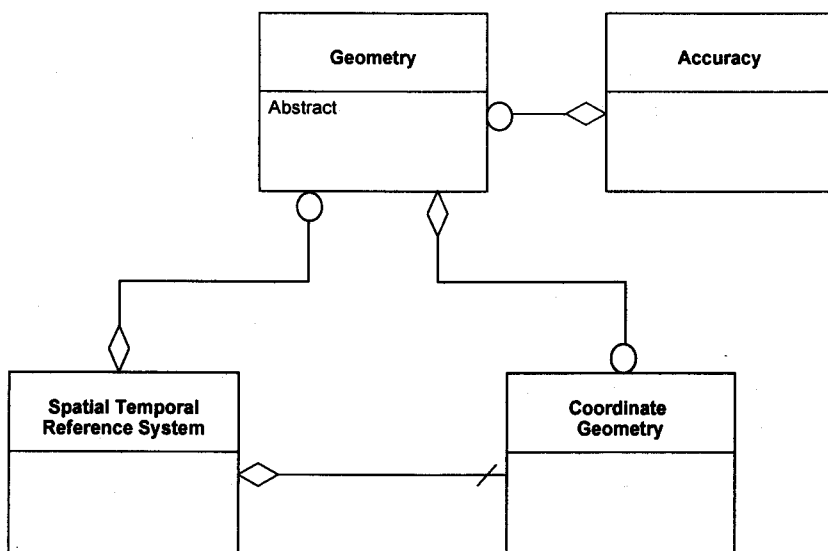
OGIS Features의 Geometries

51

- ◆ Geometry는 좌표 기하(COGO: coordinate geometry)와 참조 시스템(reference system)의 결합
- ◆ 좌표 기하의 구성
 - 동일한 참조 시스템을 갖는 일련의 좌표점들
 - 해석 알고리즘
 - ◆ 시공간 상의 기하 크기를 정의하는 좌표 기하 엔티티(예: MBR)를 구성
 - 시공간 참조 시스템



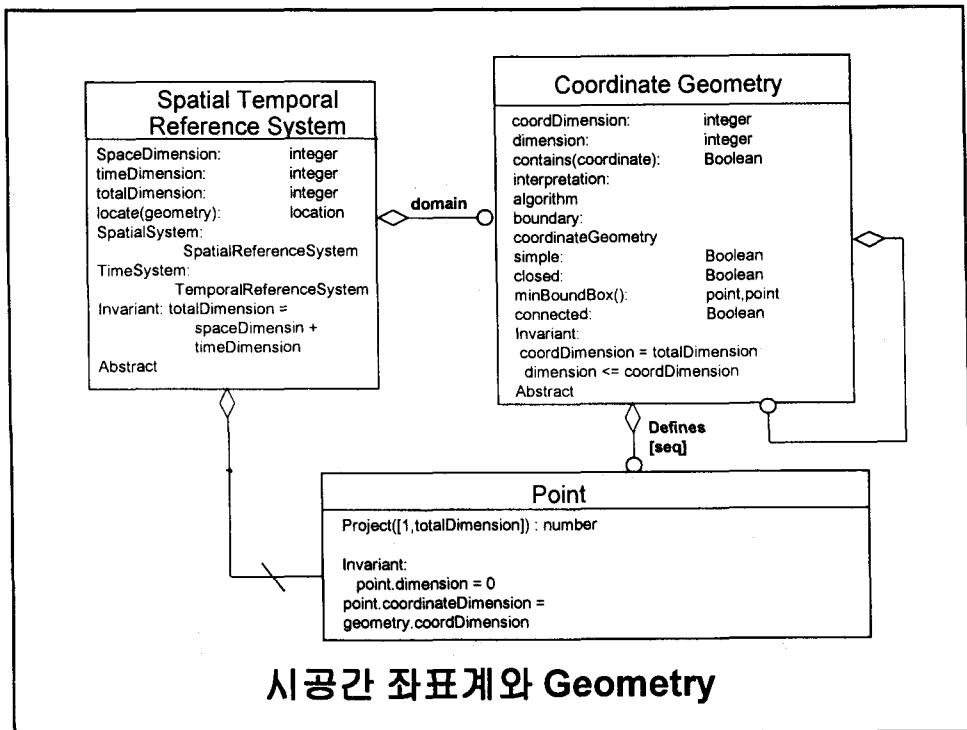
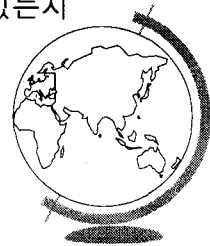
◆ Geometry essential model



Coordinate geometry 속성들

53

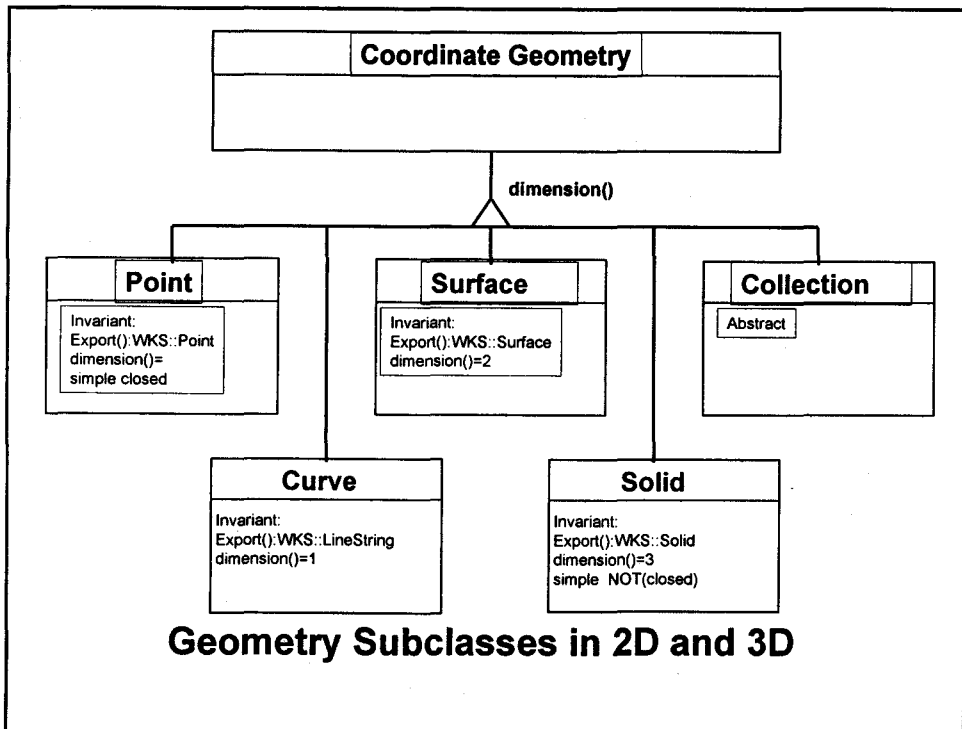
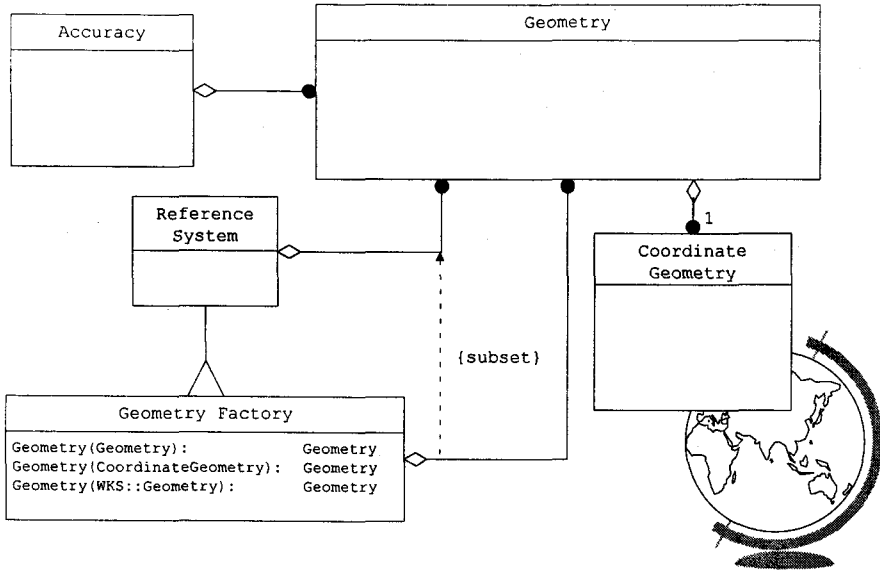
- ◆ 좌표 차원
 - geometry를 정의하는 좌표의 차원
- ◆ 객체 차원
 - 점, 선, 면
 - 좌표 차원보다 낮아야 함
- ◆ 크기(범위) 함수 "is in (coordinate)"
 - 공간 참조 시스템의 한 점이 정의된 기하내에 있는지 결정하는 함수
- ◆ 해석
 - 좌표 기하 엔터티의 크기(범위)를 정의

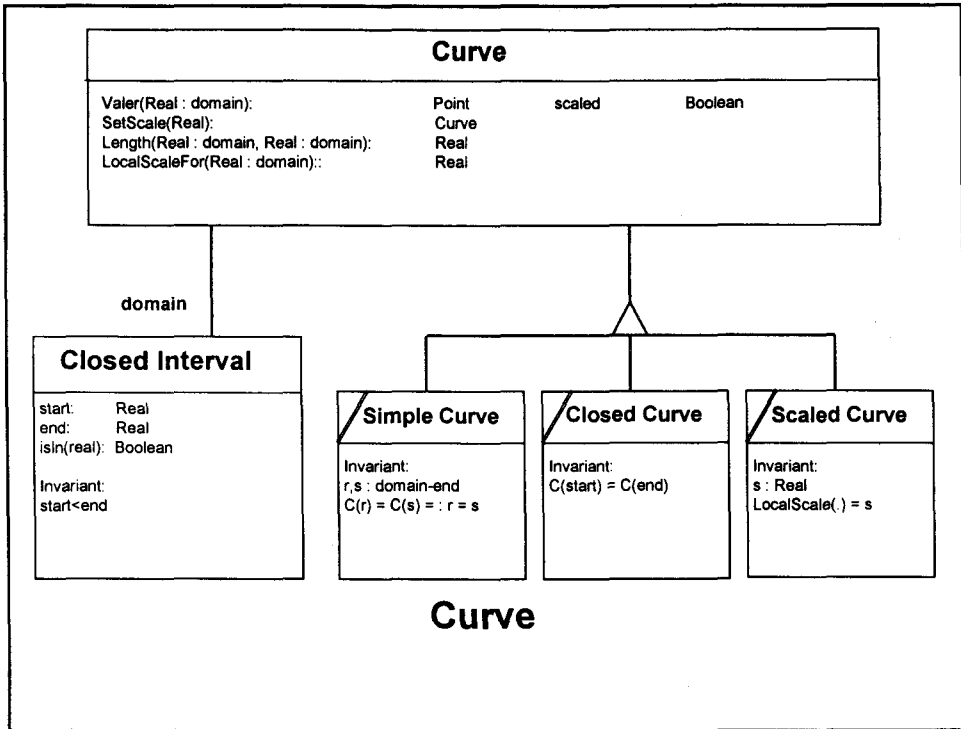


시공간 좌표계와 Geometry

Geometry Factory

55

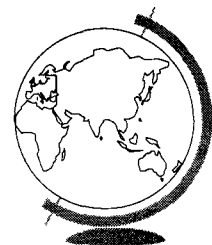




OpenGIS는 인터페이스 표준

58

- ◆ Features와 coverages는 관계 또는 객체 기반 GIS 소프트웨어에 의해 구현될 수 있음
- ◆ OpenGIS-compliant services에 의해 이해(처리)될 수 있는 공통 request 연산들
 - geodata의 접근(get) 및 변경(set)
 - 데이터베이스에서 features의 생성, 변경, 삭제
 - 질의를 사용하여 데이터베이스에서 features의 선택



설계 접근 방법

59

- ◆ simple geographic features를 지원하기 위하여 각각의 distributed computing platform마다 다른 object / data access model을 이용
- ◆ SQL DCP
 - Object model: Relational
 - Data Access Model: ODBC, SQL
- ◆ OLE/COM DCP
 - Object model: COM
 - Data Access Model: OLE DB, DAO
- ◆ OMG
 - Object model: CORBA
 - Data Access Model: Query Services
- ◆ Internet
 - Object model: Java
 - Data Access Model: JDBC



CORBA 구현 명세를 위한 OpenGIS Features API

60

- ◆ 버전 1.0, 98.3.18.
- ◆ 구현 명세 제안 회사
 - Bentley Systems, Inc(중심 역할)
 - ESRI
 - Genasys II, Inc.
 - Oracle Corporation
 - Sun Microsystems, Inc.
 - UCLA(중심 역할)

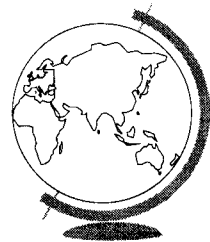


코바 구현 명세 개요

61

✦ CORBA 구현 명세 목적

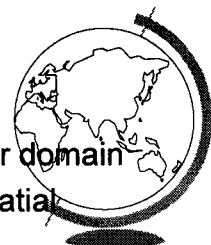
- 'simple' geometry를 갖는 features로 구성된 geodata를 접근, 조작하는 데 필요한 함수를 이용하여 GIS S/W 개발자가 응용 프로그램을 개발할 수 있게 하는 것
- language, operating system, platform, vendor에 독립적인 객체 지향 분산 시스템을 위한 명세 제공



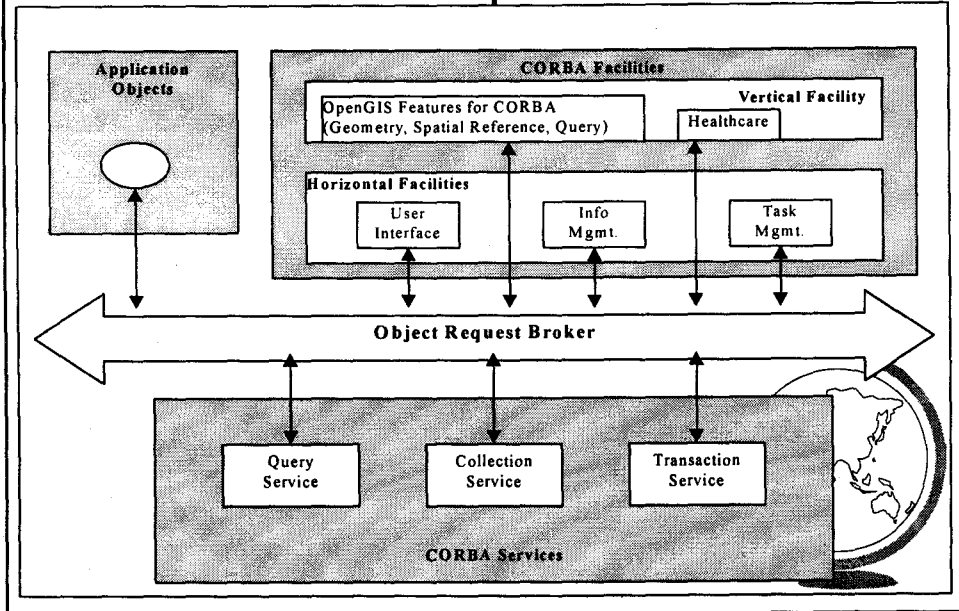
코바 구성

62

- ORB(Object Request Brokers)
- 객체 서비스
 - ◆ Query service
 - ◆ Collection service
 - ◆ Transaction service
- horizontal 서비스
 - ◆ GUI
 - ◆ task management
- vertical 서비스
 - ◆ provide services targeted to a particular domain
 - ◆ vertical CORBA facility covering geospatial information management



CORBA 구조와 OpenGIS Features 63



표준 인터페이스의 정의 64

- ◆ CORBA IDL(Interface Definition Language)로 OpenGIS Constructs를 표현
- ◆ 표준 인터페이스 구성
 - Feature module
 - Geometry module
- ◆ 구현 모델의 고려 사항
 - 기존의(legacy) geospatial 데이터와 응용을 'wrapping' 하여 객체로 표현
 - 새로운 분산된 객체 지향 GIS 응용의 개발



Feature Model Architecture

65

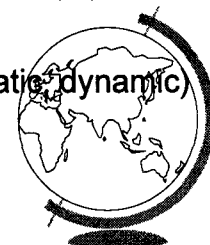
- ◆ Simple geospatial features와 feature collections을 생성, 접근, 질의하기 위한 구현 인터페이스 모델
- ◆ Feature Module
 - feature type
 - ◆ real world entity의 표현
 - property의 집합을 정의
 - feature collection
 - ◆ groups of features
 - ◆ client에게 다양한 feature type을 제공
 - feature schema
 - ◆ collection of feature types
 - ◆ containing feature collection provide it



The Interfaces exposing the OpenGIS Feature Model

66

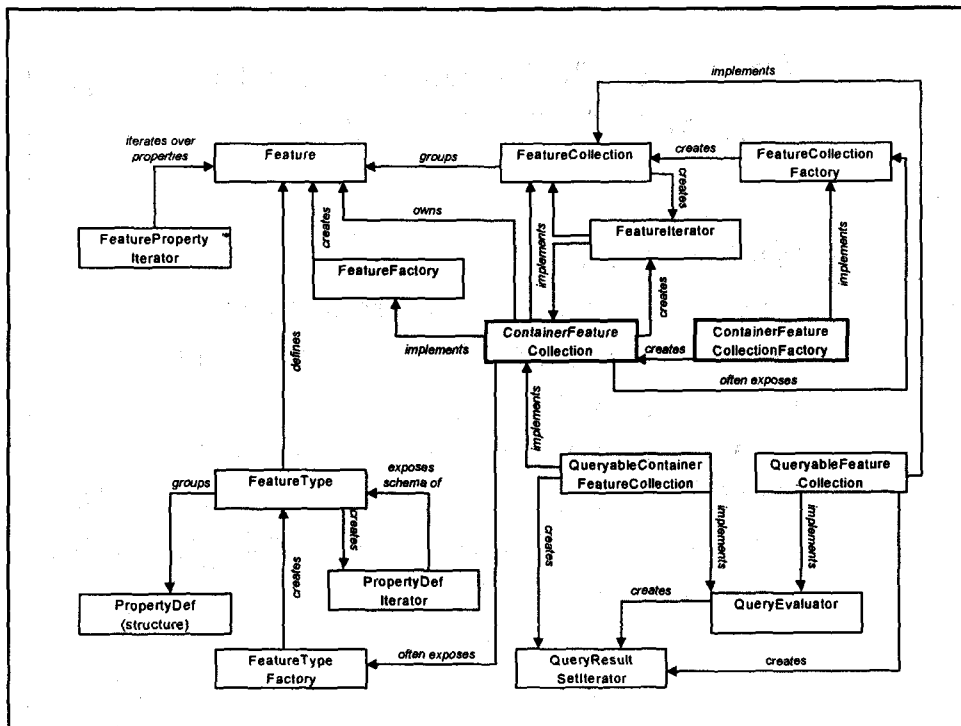
- ◆ OpenGIS constructs
 - Feature, FeatureFactory, FeatureCollection
- ◆ OpenGIS constructs를 클라이언트에게 expose
- ◆ OpenGIS Server 또는 구현은 ContainerFeatureCollection를 클라이언트에게 expose함
 - 코바 naming과 Trader service의 (static/dynamic) binding 메커니즘을 이용



The Interfaces exposing the OpenGIS Feature Model

67

- ◆ ContainerFeatureCollection이 GIS DB에 해당
 - GIS DB는 a set of Features를 포함
 - Features 는 feature interface를 통하여 클라이언트에게 expose됨
 - Features 는 a set of properties를 가짐
- ◆ 클라이언트는 FeatureFactory object를 사용하여 새로운 Feature object를 생성
- ◆ FeatureCollections은 QueryableFeatureCollection interface를 expose함
- ◆ 클라이언트는 SQL, OQL query를 issue, 결과는 QueryResultSetIterator를 통하여 return됨



OpenGIS geometry 인터페이스

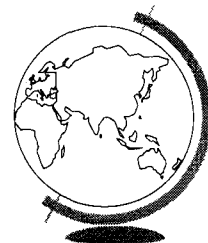
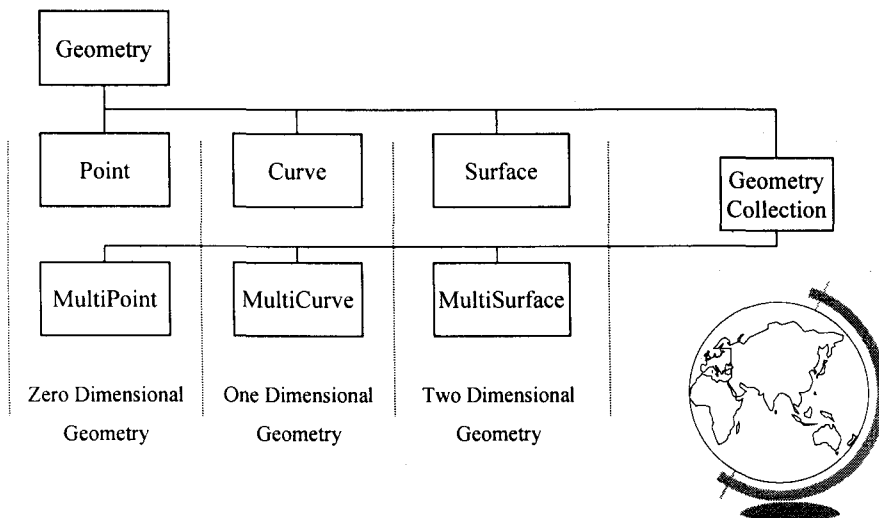
69

- Require the unambiguous exposure of geometric entities
- exposing coordinate geom in the form of WKS



OpenGIS geometry interfaces

70



Feature interfaces

71

- ◆ 클라이언트가 Feature object을 접근할 수 있게 함
 - generic feature interface
 - ◆ feature type에 관계없이 동일한 feature interface를 제공
- ◆ 1. Feature Related Interfaces
 - ◆ Feature interface
 - ◆ FeaturePropertySetIterator interface
 - ◆ FeatureFactory interface
- ◆ 2. Feature Type Related Interfaces
 - ◆ FeatureType interface
 - ◆ PropertyDefIterator interface
 - ◆ FeatureTypeFactory interface



Feature interfaces

72

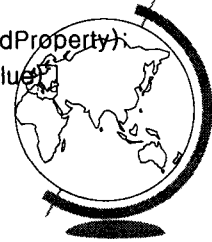
- ◆ 3. Feature Collection Related Interface
 - ◆ FeatureCollection interface ✓
 - ◆ FeatureCollectionFactory interface
 - ◆ FeatureIterator interface
- ◆ 4. Container Feature Collection Interface
 - ◆ ContainerFeatureCollection interface
 - ◆ ContainerFeatureCollectionFactory interface
- ◆ 5. Queryable Interfaces
 - ◆ QueryEvaluator ✓
 - ◆ QueryableFeatureCollection ✓
 - ◆ QueryableContainerFeatureCollection ✓
 - ◆ QueryResultSetIterator ✓
 - ◆ QueryResultSetMetaData



Feature 인터페이스

73

```
Interface Feature {
    .....
    // feature type
    readonly attribute FeatureType feature_type;
    // Geometry
    Geometry get_geometry(in NVPairSeq geometry_context) raises
        (InvalidParams);
    // properties
    boolean property_exists(in Istring name) raises (InvalidProperty);
    any get_property(in Istring name) raises
        (PropertyNotSet, InvalidProperty);
    void set_property(in Istring name, in any value);
    .....
    FeaturePropertySetIterator get_property_iterator();
};
```



Feature의 서버 구현

74

- ◆ 객체 지향 환경에서 GIS features를 객체로 구현
 - feature interface는 feature object에 대한 thin wrapper에 해당됨
 - get_property(), set_property() 연산이 feature object의 internal properties을 검색 변경함

- ◆ RDBMS에서 GIS Features는 관계 테이블의 row로 표현
 - feature interface는 테이블의 row를 접근하는 데 이용됨
 - 테이블 이름과 primary key는 feature의 object reference에 encode됨



Client 시나리오

75

- ◆ generic client
 - a generic 'feature browser' client
 - ◆ features geometric와 non-geometric properties를 디스플레이
- ◆ dedicated client (generic interface)
 - 데이터를 검색하기 위하여 generic interface를 사용
- ◆ dedicated client (type-specific interface)



generic client pseudocode

76

```
Feature          FeatureRef:
FeatureType      featureTypeRef;
CORBA_Any        propertyValue;
PropertyDefSeq   propertySet;
...
//get FeatureType reference
featureTypeRef = featureRef.get_feature_type();

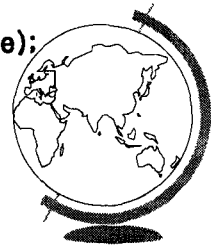
//get Property Names from feature type reference
propertySet = featureTypeRef.get_property_defs();
```



generic client pseudocode

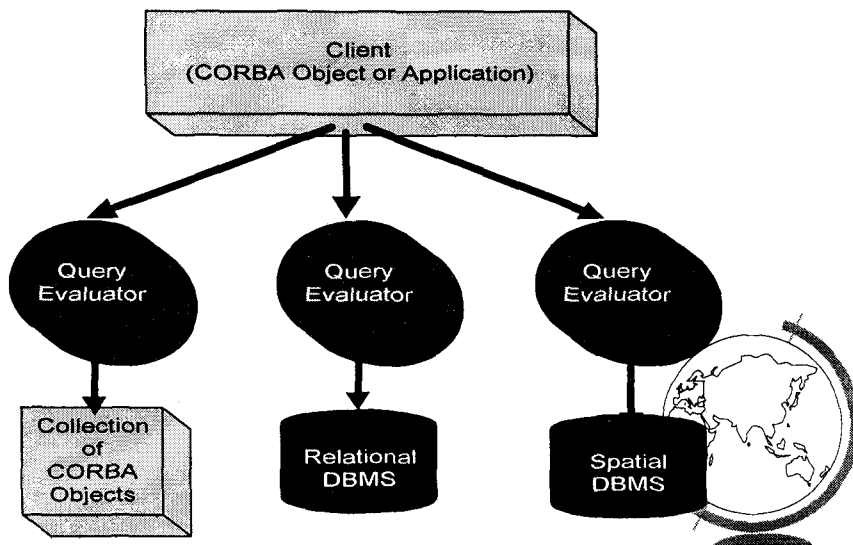
77

```
for (i = 0; i < propertySet.length; i++) {  
  try{  
    propertyValue =  
    featureRef.get_property((propertySet.get(i)).name);  
  
    if(isGeometry(propertyValue.type)  
        drawInGraphicsWindow(propertyValue.value);  
    else .  
        writeInTextWindow(propertyValue.value);  
  }catch(InvalidPropertyName) {  
  }catch(PropertyNotSet){  
  }  
}
```



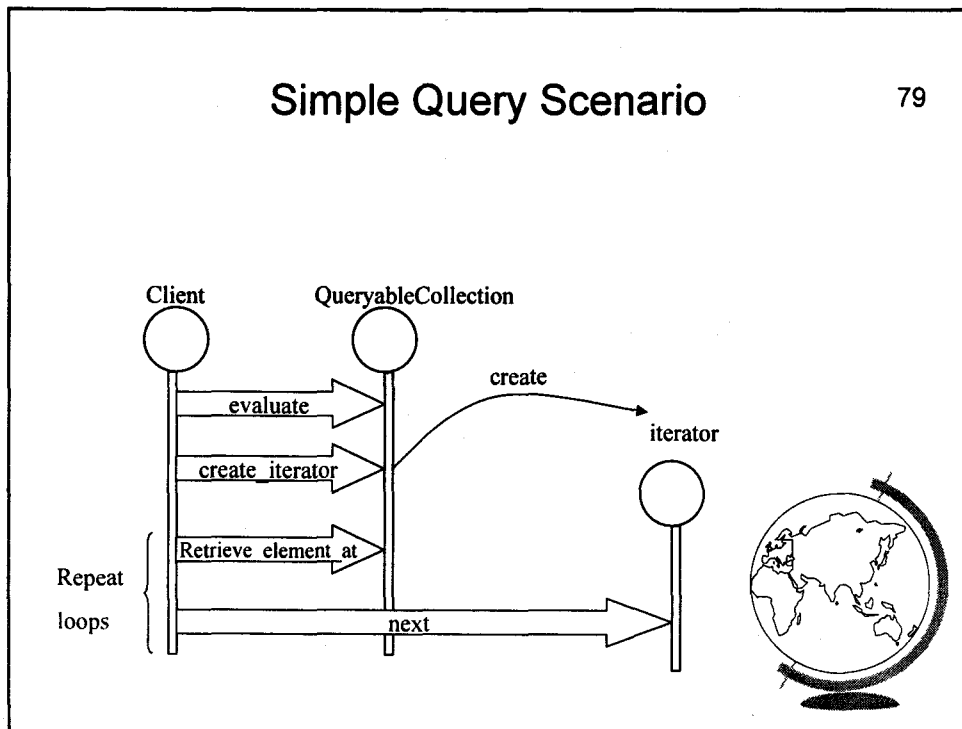
CORBA의 Object Query Service를 Data Access Model로 이용

78



Simple Query Scenario

79



Complex Query 시나리오

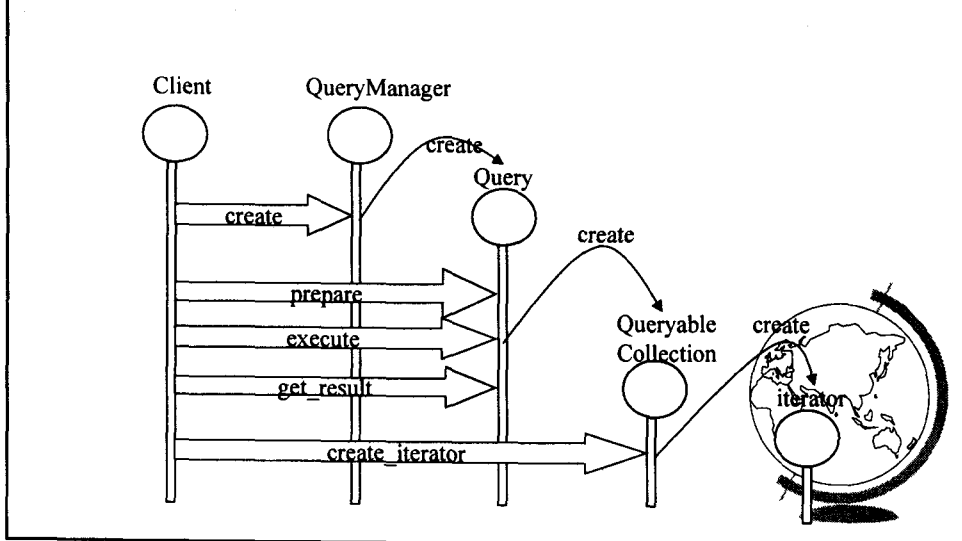
80

- 언제 사용하나?
 - ◆ 나중에 query를 재실행하기 위해 사전에 컴파일
 - ◆ 비동기적으로 query를 실행
 - ◆ query 실행을 중단 또는 계속을 결정하기 위해 상태 정보 검사
- QueryManager 사용
 - ◆ query를 특정 query object으로 할당
 - ◆ query의 실행을 제어하기 위해 query object를 생성



Complex Query Scenario

81



QueryEvaluator Interface

82

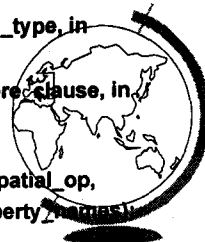
- ◆ Query 문을 표현하고 evaluate하는 것을 지원
- ◆ a query language(SQL, OQL, OQL_93 등)을 선택하고 주어진 변수로 질의문을 합성 생성하는 것을 지원
- ◆ query의 동기식 실행과 결과의 생성을 책임짐
- ◆ a collection of Features에 대한 query를 표현할 수 있는 generic framework를 제공



QueryEvaluator interface

83

```
interface QueryEvaluator{
...
enum QLType {
    SQLQuery, SQL_92Query, OQL, OQLBasic, OQL_93, OQL_93Basic
};
...
enum SpatialOperatorType{
    TouchOp, ContainsOp, WithinOp, DisjointOp, CrossesOp,
    OverlapsOp, IntersectsOp
};
...
QueryResultSetIterator evaluate(in string query, in QLType ql_type, in
    NVPairSeq params);
QueryResultSetIterator query by properties(in string where_clause, in
    QLType ql_type,
    in Geometry geo,
    in SpatialOperatorType spatial_op,
    in IStringSeq return_property_names);
};
```



Query Example

84

```
OGIS::QueryableContainerFeatureCollectionRef countries;

//create the parameter list for query hints
OGIS::NVListSeq params(1);
params.length(1);
params[0].name(CORBA::string_copy("readOnlyFlag"));
params[0].value() <=<= (long) CORBA::B_TRUE;

char* queryString = "SELECT name from Countries;";
OGIS::QueryEvaluator::QLType qlType =
    OGIS::QueryEvaluator::QLType::SQL_92Query;
OGIS::QueryResultSet results;
```



```

//execute the query
results = countries->evaluate(queryString, qlType, params);

OGIS::QueryResultSetIterator iterator;
String country_name;

iterator = countries->create_iterator();

while((more = iterator->advance()) == B_TRUE){
    country_name = get_property_by_name("name");
}
iterator->destroy();
results->destroy();
};

```



Geometry Interface

- ✦ 1. SpatialReferenceSystem Interface
 - Many Entity (figure)
 - SpatialReferenceSystemFactory
 - SpatialReferenceComponentFactory
- ✦ 2. General Geometry Interface
 - Geometry
 - GeometryFactory
 - GeometryCollection
 - GeometryIterator



Geometry Interface

87

```
interface Geometry {
...
readonly attribute      short    dimension;
readonly attribute      Envelope  range_envelope;
    //minBoundingBox
readonly attribute      SpatialReferenceSystem;
    spatial_reference_system;

//geometric characteristics
booleanis_empty();
booleanis_simple();
booleanis_closed();
...
// WKS operators
WKSGeometry export();
WKBGeometry export_WKBGeometry();
```



Geometry Interface

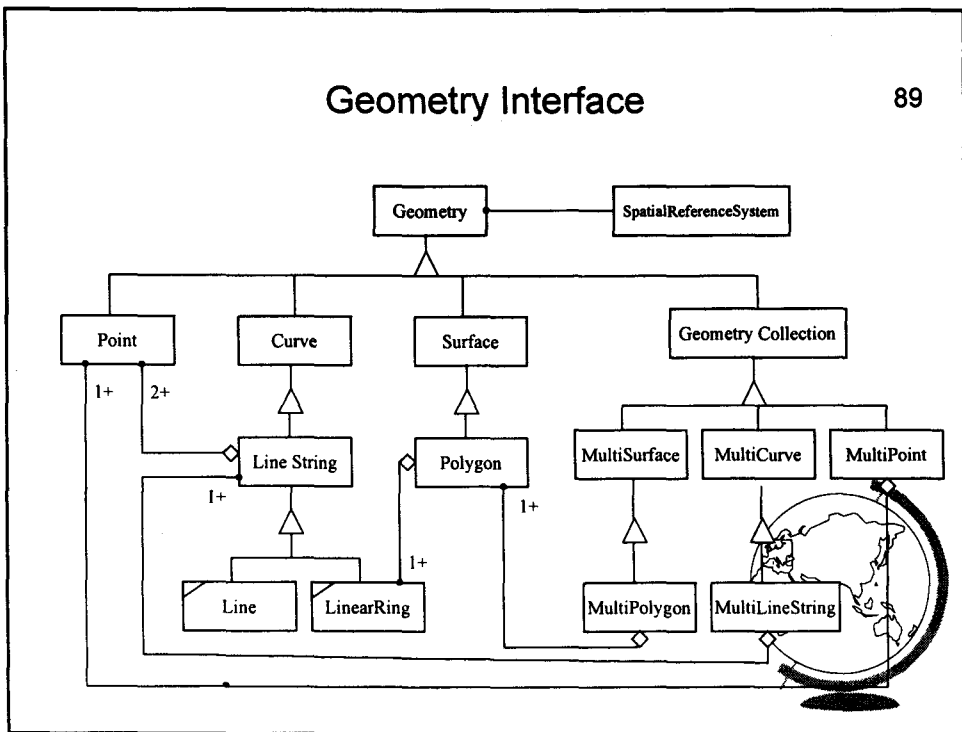
88

```
//relational operators
booleanequals(in Geometry other);
booleantouches(in Geometry other);
booleancontains(in Geometry other);
booleanwithin(in Geometry other);
booleandisjoint(in Geometry other);
booleancrosses(in Geometry other);
booleanoverlaps(in Geometry other);
booleanintersects(in Geometry other);
booleanrelate(in Geometry other);
//metric operators
double distance(in Geometry other);
//set operators
Geometry intersection(in Geometry other);
... };
```



Geometry Interface

89



Point Interface

90

- ◆ 0-차원 기하를 expose 함
- ◆ Geometry interface를 상속받음
- ◆ IDL 명세

```
interface Point: Geometry {
    attribute WKSPoint coordinates;
};
```



참고문헌

91

1. Kurt Buehler and Lance McKee, The OpenGIS Guide: Introduction to Interoperable Geoprocessing and the OpenGIS Specification, 3rd Ed., June 1998, The Open GIS Consortium, Inc.
2. How OGIS Spans Distributed Computing Platforms, <http://www.opengis.org/gisworld/distribu.html>
3. Open GIS Consortium, Inc., The OpenGIS Abstract Specification Model, Version 3, 1998.
4. Open GIS Consortium, Inc., OpenGIS Simple Features Specification for CORBA, Revision 0, Release Date:11 March, 1998.

