

1. 튜토리얼

OpenGIS가 우리에게 주는 의미

1

1998. 7. 2

김 은 형
경 원 대 학 교

OpenGIS가 우리에게 주는 의미

1998. 7. 2

김 은 형
경원 대학교

목 차

1. *OpenGIS* 태동 배경
2. OpenGIS 개요
3. OpenGIS 기술개발 절차
4. OpenGIS vs. SDTS
5. OpenGIS와 컴포넌트
6. Case study: OpenGIS를 이용한 NGIS
기술개발 과제 통합(안)

OpenGIS 태동 배경

- **지리정보처리 환경의 변화**
 - 지리자료 집적량의 급속한 증대, 데이터 포맷과 소프트웨어의 다양성 및 복잡성 증대, 분산처리 네트워크 컴퓨팅 환경의 대두 등
- **지리자료 공유의 필요성**
 - 환경에 대한 관심의 증대
 - 정부와 기업이 받는 업무효율 향상에 대한 압력
 - 유용한 지리자료와 지리정보처리 도구의 급속한 증가
- **데이터 공유에 따른 제문제**
 - 상이한 데이터 포맷
 - 정보수집 방법과 표준에 있어서의 차이
 - 제도적, 경제적, 법률적 장애물

데이터 공유를 위한 노력

- 
- **정보의 섬(Islands of information)**
 - 1970년과 1980년대 초 대부분의 GIS 어플리케이션. 즉, 지리자료가 수직적으로 변환, 저장, 분석, 디스플레이 되는 독립시스템(사람들은 GIS 데이터라고 하지 않고, ARC/INFO 데이터 또는 Intergraph 데이터라고 부른다).
 - **변환의 배(Transfer/conversion Ferries)**
 - 하나의 포맷에서 직접 다른 포맷으로, 또는 중간 포맷을 통해 변환시키는 특정 변환기 이용. 이 두 가지 방법은 모두 완전한 데이터 셋이 변환되고 처리되는 일괄처리지향 솔루션(Batch-Oriented Solution)이다.
 - **접근/질의 다리(Access/query Bridges)**
 - 클라이언트/서버 아키텍처의 개발과 함께, 이러한 접근은 데이터 셋에 직접적인 접근을 허락하고 네트워크를 통한 제한된 검색을 지원. 정보는 검색을 기반으로 선택적으로 다운로드 되고, 정보를 목표 시스템에 전달하기 위해 필요에 따라 처리된다.

상호운용성(Interoperability)

• 정의

- 정보의 이동성(portability) 및 어플리케이션간의 공조 처리 제어(interapplication, cooperative process control)를 제공해주는 시스템 또는 시스템 컴포넌트의 기능

• 특징

- 분산 컴퓨팅 구조에서 "plug and play"를 가능하게 하는 시스템 과 컴포넌트 수용을 위한 첫째 기준
- 분산 데이터 패러다임: 질의 결과가 한 시스템에서 다른 시스템으로 데이터를 담은 용기가 직접 신속하게 전송되는 초고속 운반 시스템
- 네트워크 집적소(Networked Archive) 가능: 공간정보의 온라인 저장소 형태로서 저가의 컴퓨팅 설비를 가진 이용자들의 접근이 언제든지 가능함

개방성(Openness)의 의미

• 자료전송(data transfer) 측면

- 최소한의 의미에서의 개방성
- 노동집약적, 많은 디스크 용량 요구, 데이터 손실 및 파손 위험
- Open API가 구현되면 자료 전송을 위한 데이터 변환의 필요성과 이에 따른 제반 문제가 해결

• 개방형 자료 형식(open data format) 측면

- 자료 형식을 공개함으로써 자료변환이나 교환 표준 없이 그 자료를 읽고 쓸 수 있음
- 개방형 자료 형식이 OGC의 기본 이념

• 개방형 API 측면

- OGIS 사양은 클라이언트와 서버 측에서 구현될 자료 접근 API 집합을 명시
- 자료 변환 없이 여러 공간자료에 접근이 가능

OpenGIS Consortium(OGC) 개요

- **OGC의 설립**

- 1994년 8월, 상호운용 가능한 지리정보처리를 위한 새로운 기술적, 상업적 접근을 촉진하기 위해 조직된 비영리 단체로 세계 각국의 산업계, 정부, 학계가 참여

- **OGC와 타 표준화 기관과의 관계**

- 미국 표준위원회의 GIS 표준위원회(ANSI X3L1)가 OGC의 운영위원회에 참여하고 있으며, 국제표준위원회의 GIS/Geomatics 분과인 ISO TC/211이 OGC와 밀접한 관계를 유지하고 있고, Object Management Group(OMG)과도 밀접한 관계가 있음

OpenGIS Consortium 역할

- 상호운용적 공간정보처리가 가능하도록 필요한 산업 전반적 구조를 새로 만들어 내고 이를 지속적으로 관리
- 상호운영적 공간정보처리에 기반한 사용자 중심의 사업 및 구매 모형을 개발함에 있어 리더십 제공
- **OGC의 비전**
 - 공간정보와 지리정보처리용 자원을 **mainstream computing** 환경 속에서 완전하게 통합시키는 것
 - 정보기반(**information infrastructure**)을 통하여 상호운용적, 상업적 지리정보처리 소프트웨어가 광범위하게 이용되도록 하는 것

목 차

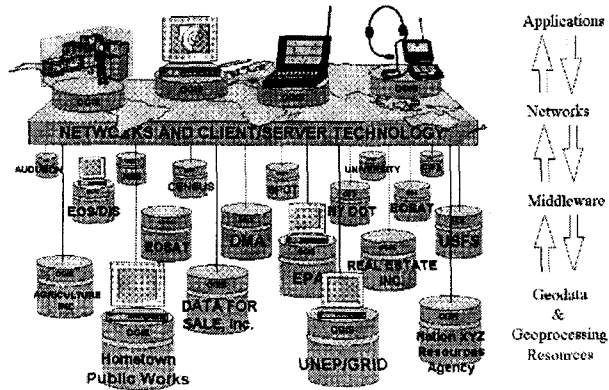
1. OpenGIS 태동 배경
2. **OpenGIS 개요**
3. OpenGIS 기술개발 절차
4. OpenGIS vs. SDTS
5. OpenGIS와 컴포넌트
6. Case study: OpenGIS를 이용한 NGIS
기술개발 과제 통합(안)

OpenGIS의 정의

- **OpenGIS**: 개방적 상호운용적 지리정보처리 또는 서로 이질적인 지리자료와 지리정보처리 자원을 네트워크 환경에서 쉽게 공유할 수 있도록 해주는 기능. “상호운용성 사양의 최고 단계”
- **OpenGIS Specification (“OGIS”)**: 지리자료의 공유와 지리정보처리의 상호운용성을 가능하게 하는 일종의 소프트웨어 사양. 상호운용적 지리정보처리를 위한 인터페이스 표준
- **Open GIS Consortium, Inc.**: OpenGIS 기술 개발과 enterprise computing 환경에의 지리정보처리 부문 통합에 기여하는 회원중심 합의기구

OpenGIS 프로젝트의 목적

- 어플리케이션 개발자들이 단일 환경과 단일 작업 흐름(work flow) 안에서 이루지면서 동시에 네트워크 상에서 활용 가능한 모든 형태의 지리자료나 지리정보처리 기능 및 과정을 사용하기 위한 기술을 규정하는 것



OpenGIS 지리정보처리 기술의 특징

- 전통적인 지리정보처리 기술과는 달리 소프트웨어 개발자들이 OGIS에서 정의된 개방형 지리정보처리 기술을 이용하여 어플리케이션과 컴포넌트를 개발할 경우 다음과 같은 특징을 가짐
 - 상호운용 가능함(Interoperable)
 - 정보 커뮤니티 지원(Supportive of Information Communities)
 - 어디서나 이용 가능(Ubiquitous)
 - 신뢰할 수 있음(Reliable)
 - 사용하기 쉬움(Easy to use)
 - 어떤 환경에서도 사용가능(Portable)
 - 다른 정보기술과 쉽게 결합가능(Cooperative)
 - 규모의 유연성(Scalable)
 - 확장이 가능함(Extensible)
 - 상호 호환성(Compatible)
 - 구현 가능성(Implementable)

OpenGIS Framework

- **개방형 지리자료 모델(Open Geodata Model)**
 - 지구와 지표면의 현상들을 수학적 또는 개념적으로 디지털화하여 표현하는 공통의 수단
- **GIS 서비스 모델**
 - 정보 사회들 사이의 지리자료에 대한 접근, 관리, 조작, 표현, 공유하는 서비스에 대한 공통의 사양 모델(specification model)
- **정보 커뮤니티 모델(Information community model)**
 - 기술적인 상호불운용성의 문제뿐 아니라 제도적인 상호불운용성 문제를 해결하기 위한 개방형 지리자료 모델과 OGIS 서비스 모델

다양한 DCP 환경 고려

- Object Management Group(OMG): Common Object Request Broker Architecture(CORBA)
 - CI Labs consortium: associated OpenDOC component model
 - Microsoft: Object Linking and Embedding/ common Object Model (OLE/COM)
 - Open Software Foundation(OSF): Distributed Computing Environment(DCE)
 - Sunsoft: Java
- ❖ OGC는 이상의 모든 환경에서 구현이 될 수 있도록 사양을 만들 것이지만, 특정 DCP에 국한되지 않도록 하기 위해 추상적 사양(abstract specification)의 형태로 제공하게 됨

OpenGIS의 혜택

• 개발자

- 지리자료 접근을 위한 소프트웨어 개발
- 지리정보처리 자원에 접근하기 위한 소프트웨어 개발
- 특정 목적을 위해 특별히 개발된 어플리케이션
- 개발 환경의 선택
- 다양한 플랫폼으로의 어플리케이션 제공
- 지리정보처리 코드의 재사용

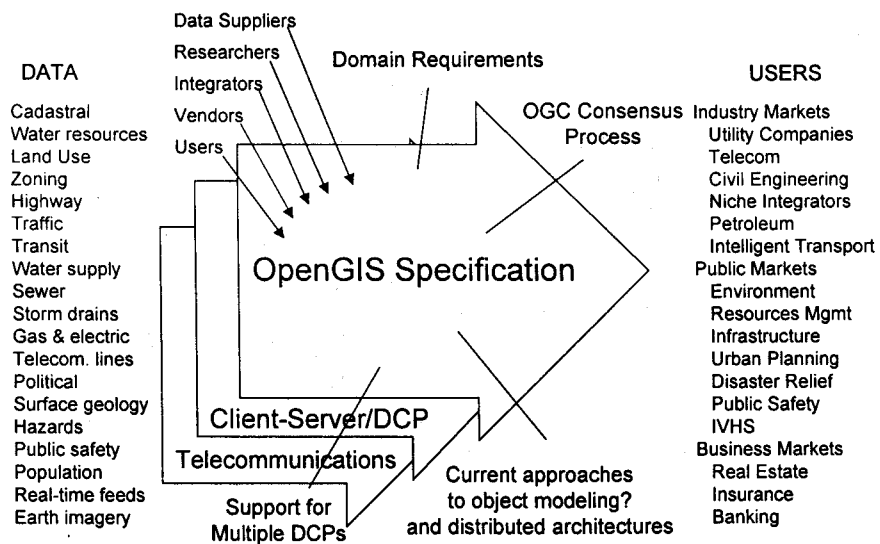
• 정보관리자

- 지리자료의 접근과 유통 촉진
- 클라이언트에 대한 지리정보처리 기능 제공의 유연성 증진
- Corporate Computing Architecture로의 지리자료와 처리기능의 통합
- 적절한 플랫폼(PC 유형, 서버 유형, 분산처리 플랫폼 유형(CORBA, OLE/COM, DCE 등))의 선택에 있어서의 융통성 확대
- 사용자에게 지리정보처리 도구 제공

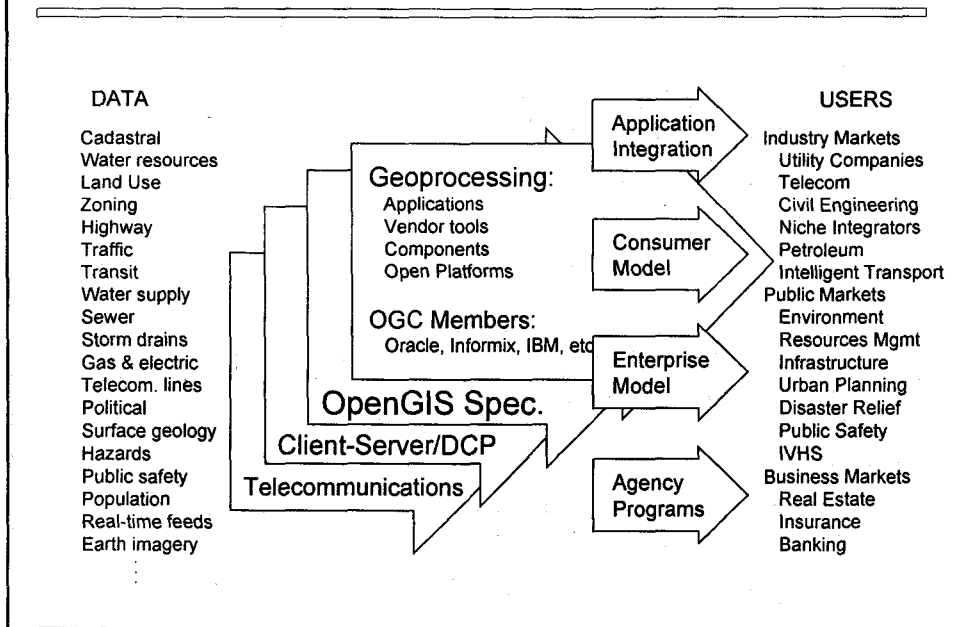
• 최종 사용자

- 현재 접근 가능한 지리자료보다 훨씬 방대한 지리자료의 세계를 실시간으로 접근
- 사용 가능한 지리정보처리 자원의 확대
- 지리정보처리 어플리케이션 범위 확대
- 단일 어플리케이션 환경과 일관된 작업흐름 내에서 다양한 유형의 지리자료와 포맷을 세부적인 기술 문제에 구애 받지 않고도 처리 가능

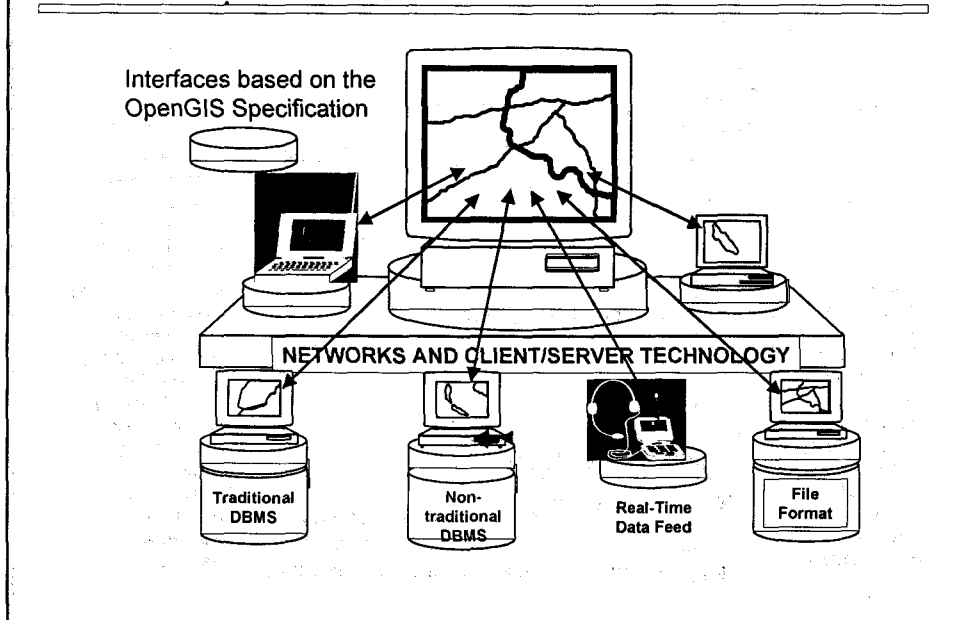
지리정보처리과정에서 OpenGIS의 위치



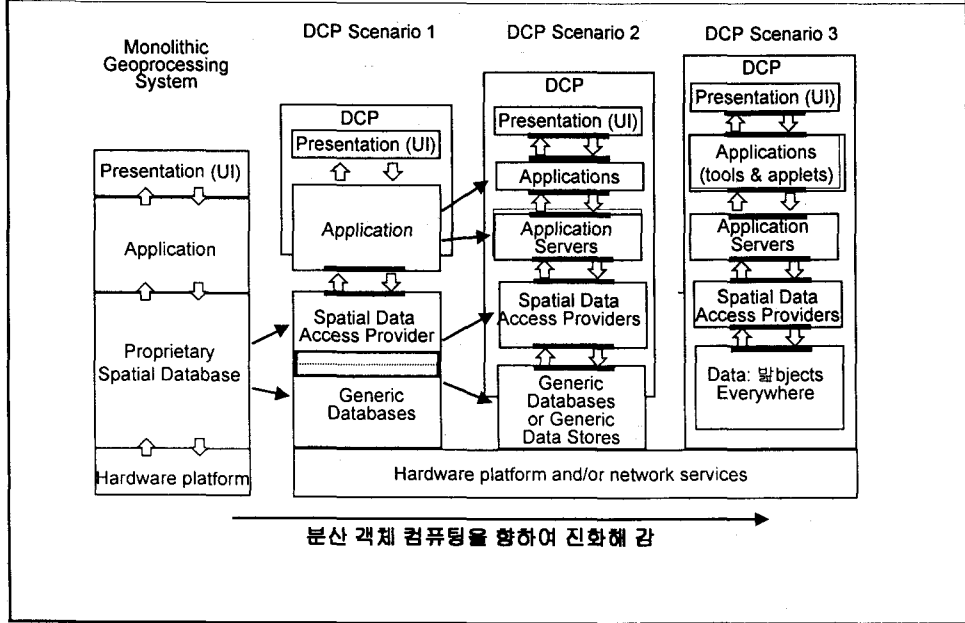
GIS 시장 진입 통로



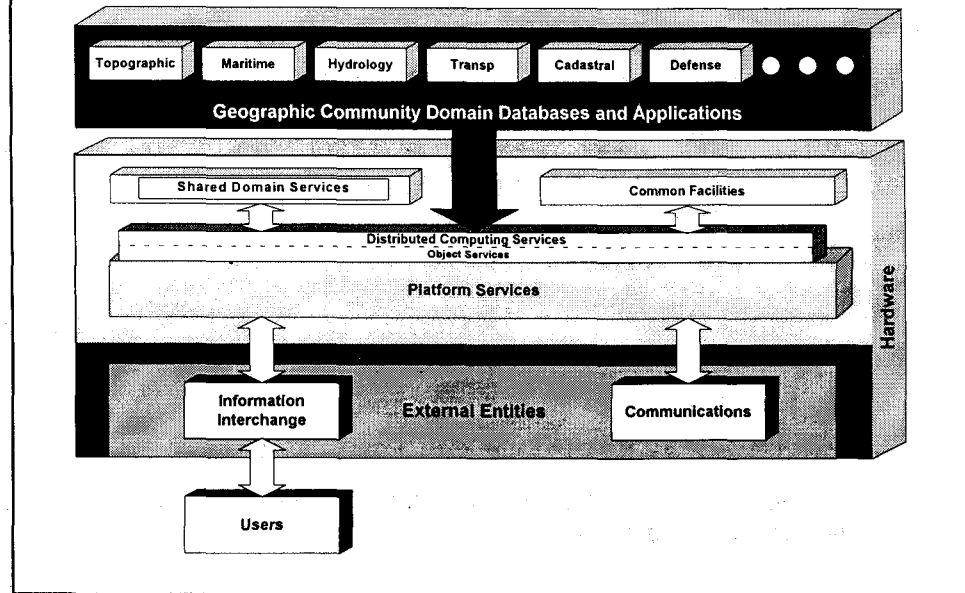
상호이질적인 공간자료에 대한 손쉬운 접근을 추구하는 OpenGIS 사양



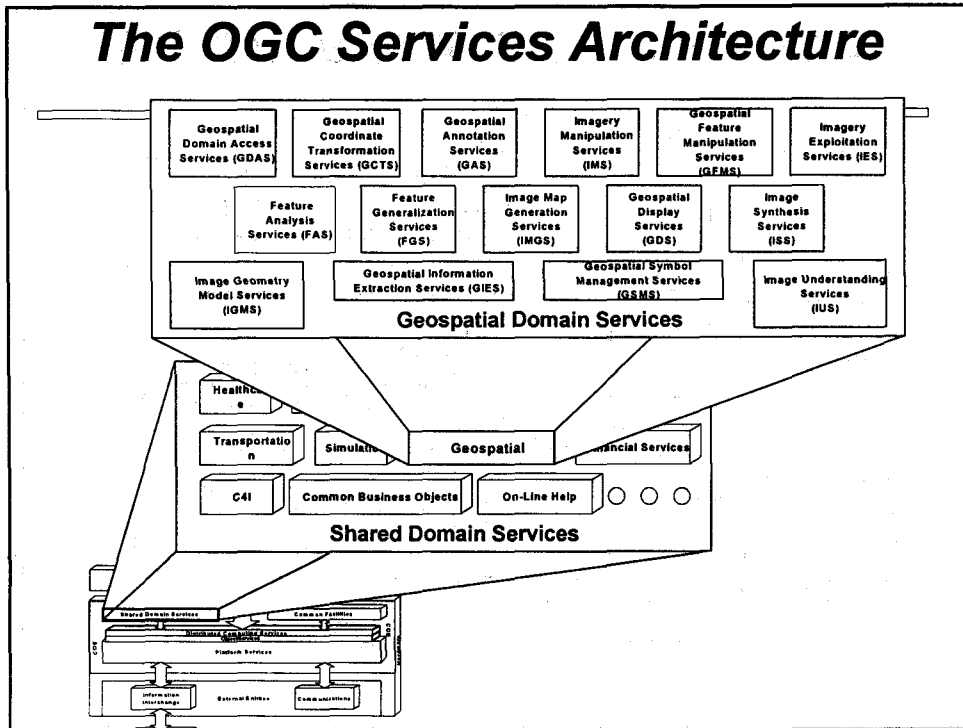
서비스 레이어와 인터페이스의 진화 단계



The Big Picture - OpenGIS Technical Reference Model



The OGC Services Architecture



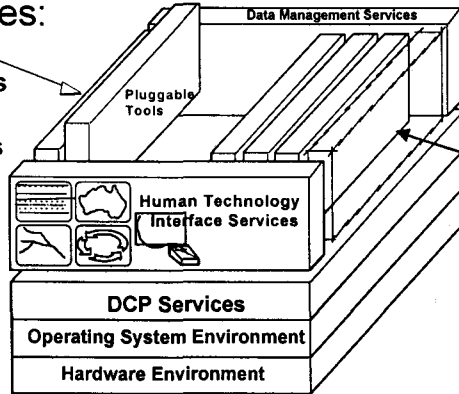
Geospatial Domain Services - examples

- **Geospatial Domain Access Services**
 - A set of interfaces for locating, retrieving and disseminating selected geospatial information from a geospatial information system (GIS), or geospatial library, and for updating the contents of a library (by storing, deleting, or modifying geospatial information)
- **Feature Generalization Services**
 - Services that modify the cartographic characteristics of a feature or feature collection by simplifying its visualization, while maintaining its salient elements – the spatial equivalent of simplification
- **Geospatial Coordinate Transformation Services**
 - Services for converting geospatial coordinates from one reference system to another

플러그블 컴퓨팅 모형

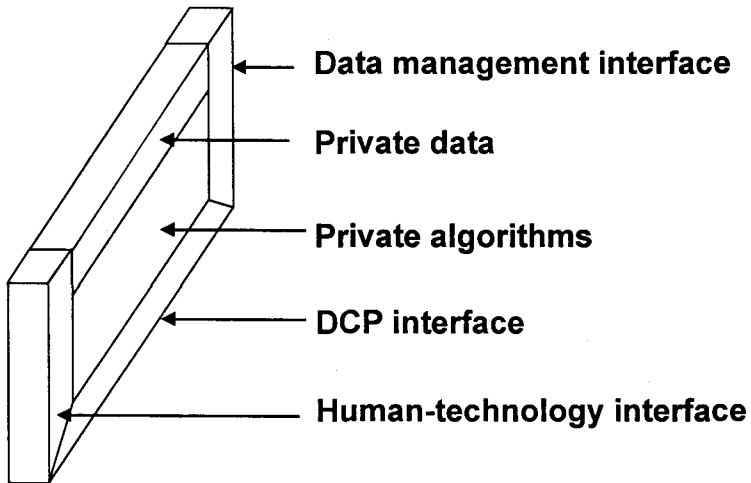
Tool Services:

- GIS Tools
- Imaging Tools
- Expert Tools
- RDBMS Tools



다음 그림
참조

플러그블 Tool



목 차

1. OpenGIS 태동 배경
2. OpenGIS 개요
3. **OpenGIS 기술개발 절차**
4. OpenGIS vs. SDTS
5. OpenGIS와 컴포넌트
6. Case study: OpenGIS를 이용한 NGIS
기술개발 과제 통합(안)

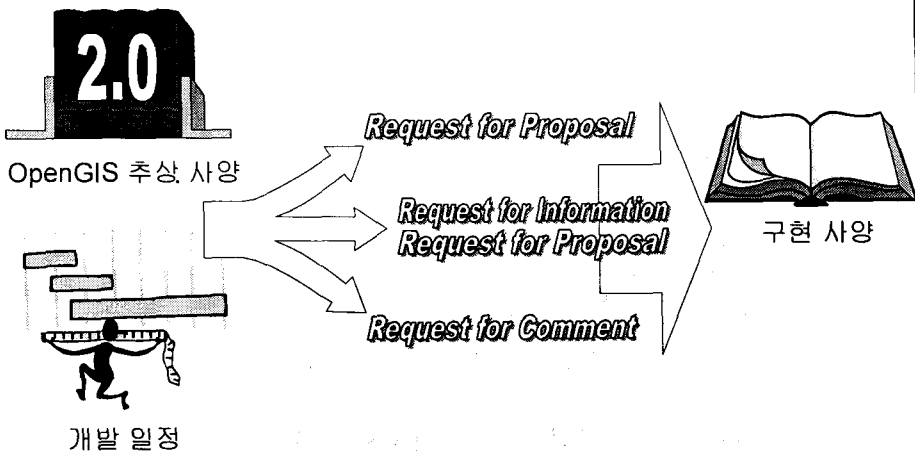
OGIS 기술개발 절차

- 상호운용적 지리정보처리를 위한 기술 사양 개발
 - 컴퓨팅 플랫폼에 종속적이지 않은 추상사양(Abstract Specification) 개발
 - 시간이 지남에 따라 더욱 포괄적으로 변천할 것임
 - 하나의 추상사양을 특정 컴퓨팅 플랫폼에 맞도록 구현사양(Implementation Specifications) 매핑
 - 기능별, 각 단계별로 완성해 감
 - 일부 추상 사양의 변경을 초래할 수도 있음
 - 개발된 사양에 대한 지속적인 관리 및 개선

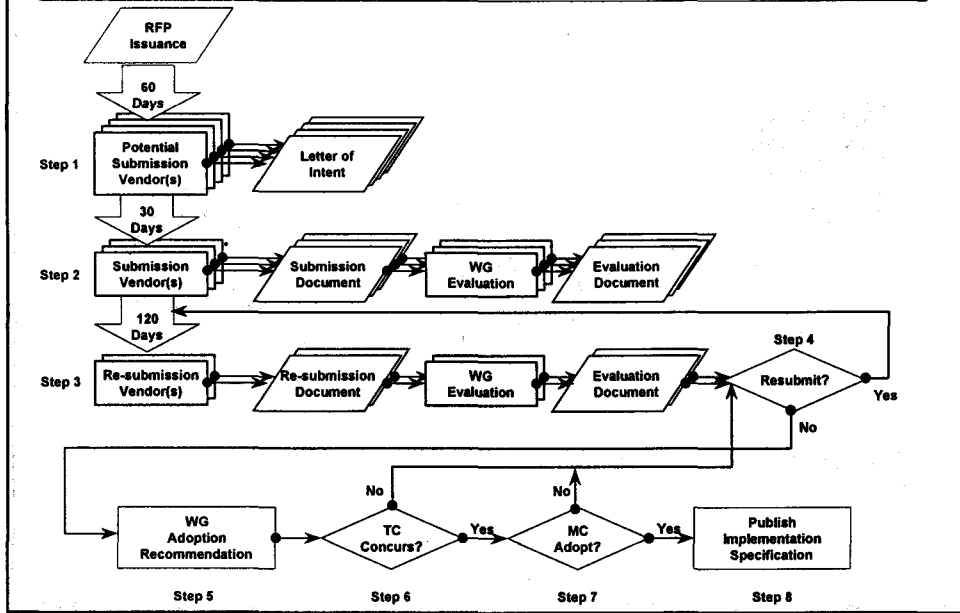
추상사양 변경 과정



구현사양 개발 과정



RFP 제출, 평가 및 채택 과정



OpenGIS 사양 현황

추상사양(Abstract Specification)

- 1998년 6월 현재 14개 주제(Topic)에 대하여 현재 version 3까지 나와 있으며 각 주제는 다음과 같음

- Topic 1: Feature Geometry
- Topic 2: Spatial Reference Systems
- Topic 3: Locational Geometry
- Topic 4: Stored Functions and Interpolation
- Topic 5: The OpenGIS Feature
- Topic 6: The Coverage Type
- Topic 7: Earth Imagery
- Topic 8: Relations Between Features
- Topic 9: Accuracy
- Topic 10: Transfer Technology
- Topic 11: Metadata
- Topic 12: The OpenGIS Service Architecture
- Topic 13: Catalog Services
- Topic 14: Semantics and Information Communities

구현사양(Implementation Specification)

- 1998년 6월 현재 OpenGIS 구현사양은 simple feature에 대해 OLE/COM, CORBA, SQL용 구현사양 revision1 이 발표되어 있음

목 차

1. OpenGIS 태동 배경
2. OpenGIS 개요
3. OpenGIS 기술개발 절차
- 4. OpenGIS vs. SDTS**
5. OpenGIS와 컴포넌트
6. Case study: OpenGIS를 이용한 NGIS
기술개발 과제 통합(안)

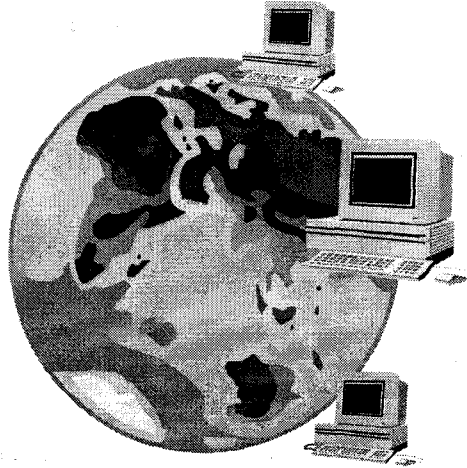
OpenGIS vs. SDTS -1/2

- 유사점
 - 지리자료를 HW, SW 플랫폼, DB, 어플리케이션의 범위를 넘어 공유할 수 있게 하자는 공통된 목적
 - 수많은 공간자료 모델을 포괄해야 한다는 기술적 유사성
- 차이점
 - OGIS는 SDTS에서 지향하는 상이한 공간자료 모델의 수용뿐만 아니라 이런 모델들을 통합할 진정한 프로세스 모델(Process model)
 - 그 목적이 SDTS는 자료의 변환에 그치는데 비해 OGIS는 상호운용성에 있음

OpenGIS vs. SDTS - 2/2

- OpenGIS

- 비용이 많이 드는 자료 변환 작업 없이 이질적인 Client/Server 사이에 runtime 수준에서 통신할 수 있도록 해주는 CORBA 나 OLE/COM의 확장 (extension)



- SDTS

- 디지털 지리정보의 보존과 대용량 전송을 목적으로 하는 벤더 중립적, 정적인 문서(static archive)

차세대 SDTS ?

- OpenGIS가 있는데 SDTS가 필요한가?

- 대용량의 지리정보 전송의 필요성
- OpenGIS access 없이 데이터 전송의 필요성
- OpenGIS Geospatial Information Transfer Service 인터페이스

- 차세대 SDTS 개발을 위한 노력

- 특별 working group 구성과 활동
 - SDTS Enhancements WG
 - Object Profile WG
 - Metadata Harmonization WG

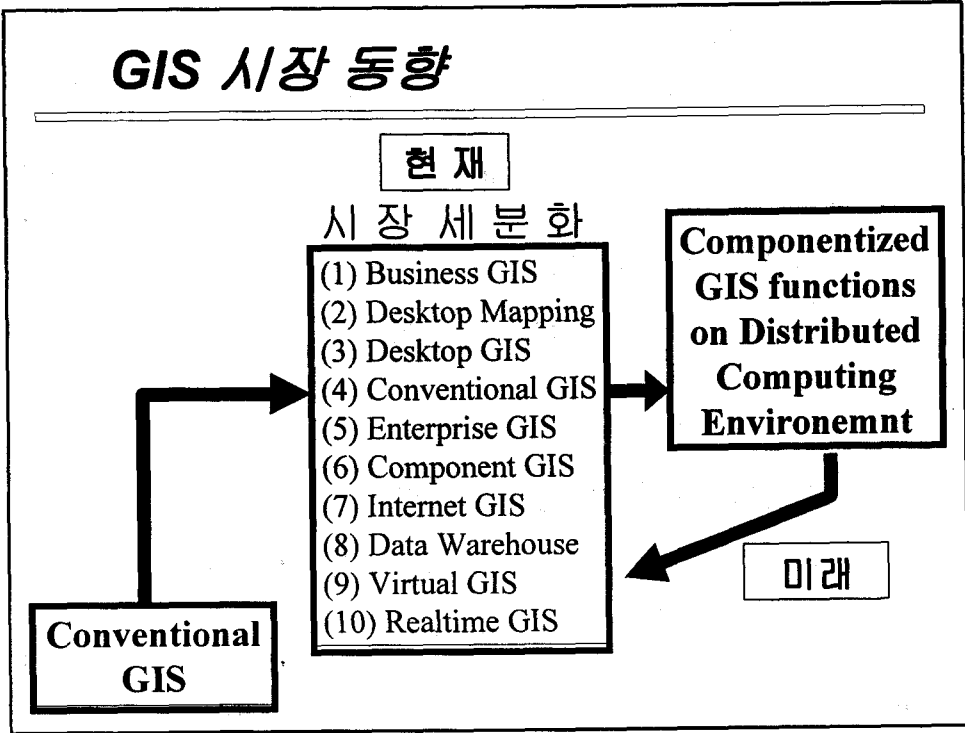
목 차

1. OpenGIS 태동 배경
2. OpenGIS 개요
3. OpenGIS 기술개발 절차
4. OpenGIS vs. SDTS
- 5. OpenGIS와 컴포넌트**
6. Case study: OpenGIS를 이용한 NGIS
기술개발 과제 통합(안)

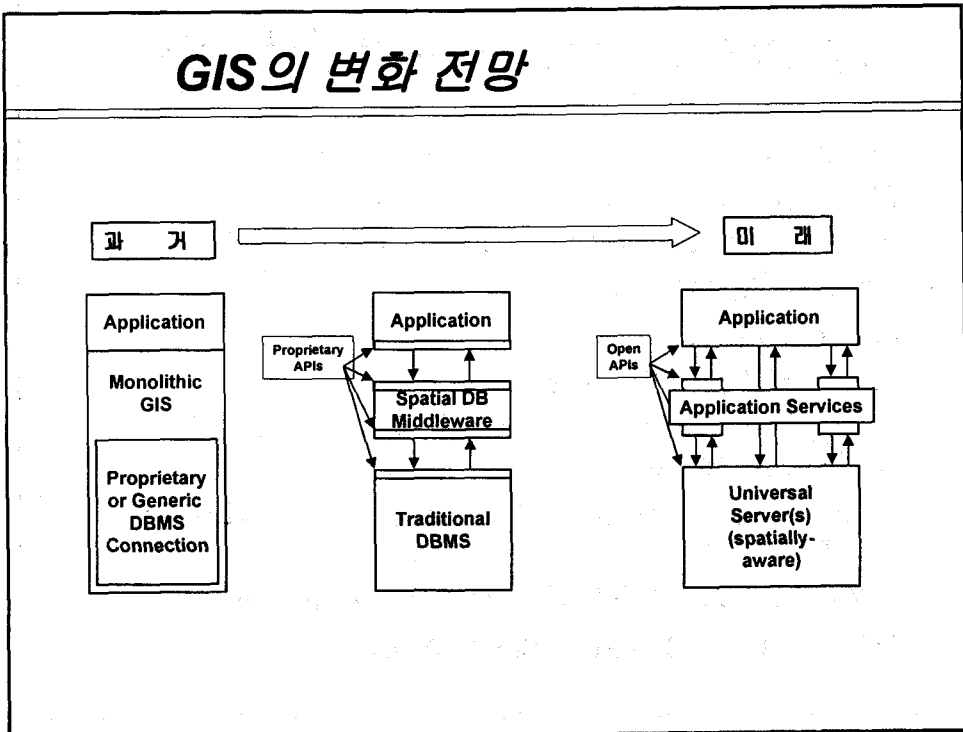
왜 컴포넌트를 이야기 하는가?

- Component: 독립적으로 실행 가능한 objects
- 정보기술에 근거한 “Distributed objects”
- 속도, 품질, 유연성을 요하는 Software engineering
- Craft vs. Construction
- Information appliance
- 개인 어플리케이션의 미래
- 컴포넌트 사이에 호환성을 마련해 주는 표준의 등장:
OLE/COM, CORBA
- Components가 새로운 소프트웨어 개발 패러다임에서
중요한 요소임
- GIS 소프트웨어 컴포넌트 시장이 성장하고 있음
- “all or nothing” → only what you need

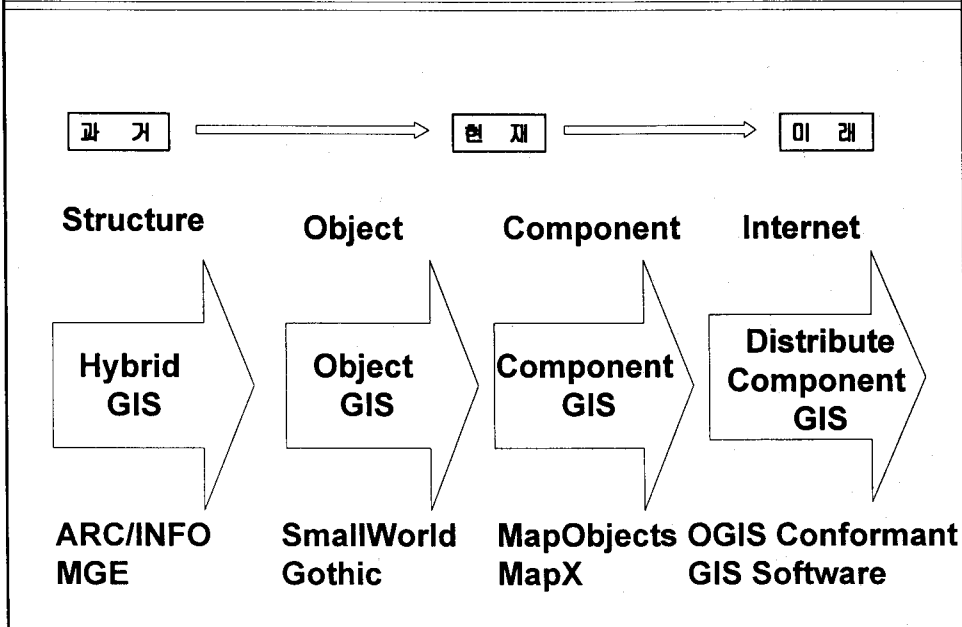
GIS 시장 동향



GIS의 변화 전망



GIS 소프트웨어 발전사



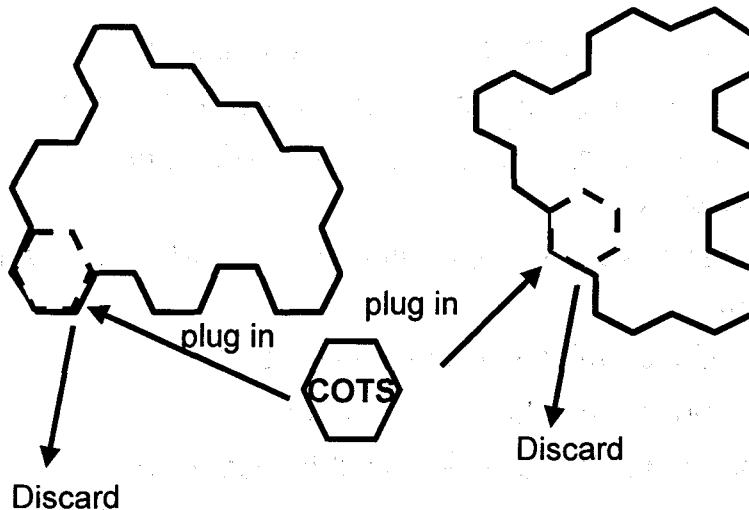
Object의 단점

- Traditional objects lack a consistent interface
- Traditional objects lack a common infrastructure
- The component model includes interface and infrastructure
- Procedure-driven subroutines did not reflect the real world
- Traditional objects are relatively static
- Traditional objects reuse is limited
- Traditional objects are difficult to work with
- Object technology is largely inaccessible to the nonprogrammer

컴포넌트의 장점

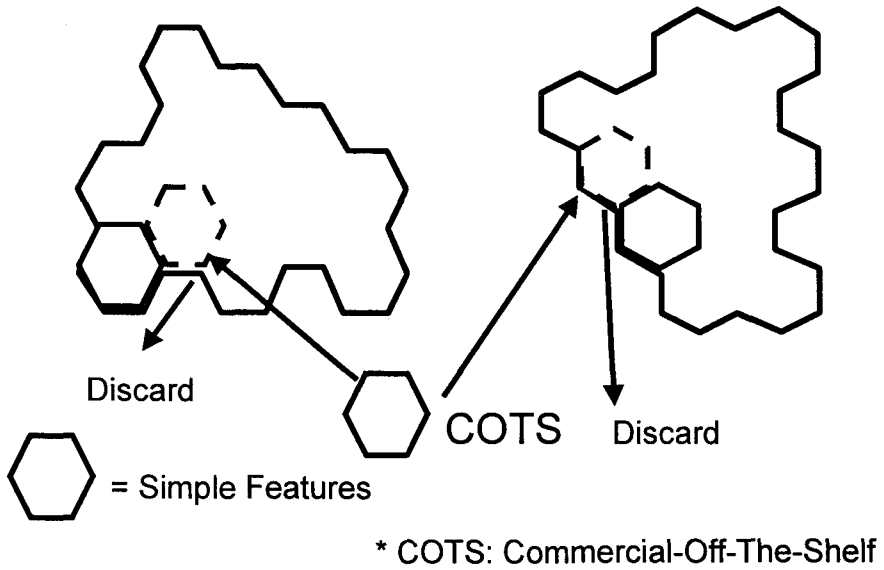
- 비주얼 툴을 이용한 어플리케이션 개발이 용이함
(e.g. Visual Basic etc.)
 - 고품질과 신뢰할 수 있는 어플리케이션의 개발이 가능
 - 개발자들이 문제의 본질인 **business problem**에 매달릴 수 있게 됨
 - 전통적인 개발방법론 보다 저렴
 - 개발 도구 및 환경에 구매를 받지 않음
 - 최종 사용자들이 필요한 모듈을 새로 개발하는 것과 컴포넌트를 구입하는 것 중에 고를 여지를 가짐
- ❖ *Componentware, the promise of assembling quality software from off-the-shelf component, is the only software development technology on the horizon that might have a dramatic enough impact on programmer productivity to eventually address the software shortage.*
- Avron Barr and Shirley Tessler, Stanford Computer Industry Project

상호운용적인 컴포넌트

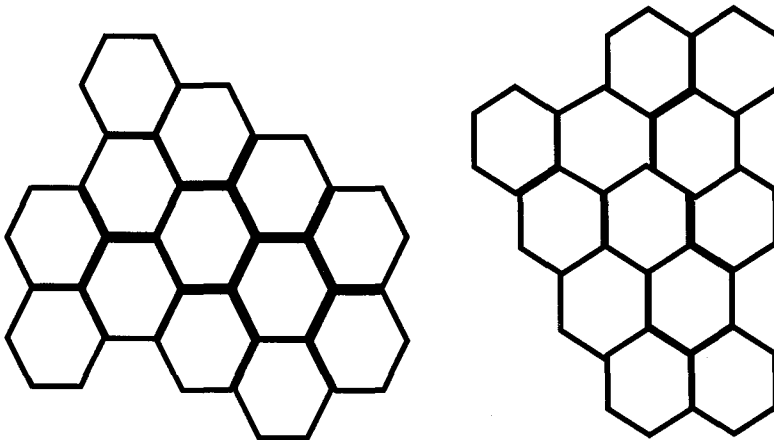


* COTS: Commercial-Off-The-Shelf

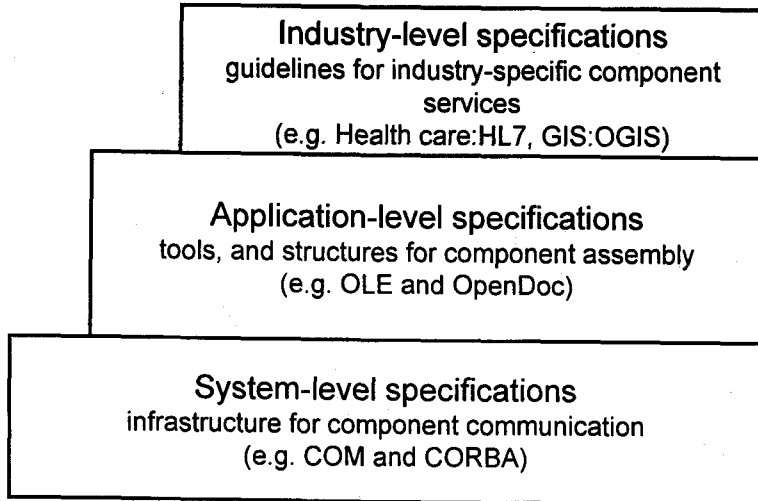
기능을 분리한 COTS-ify



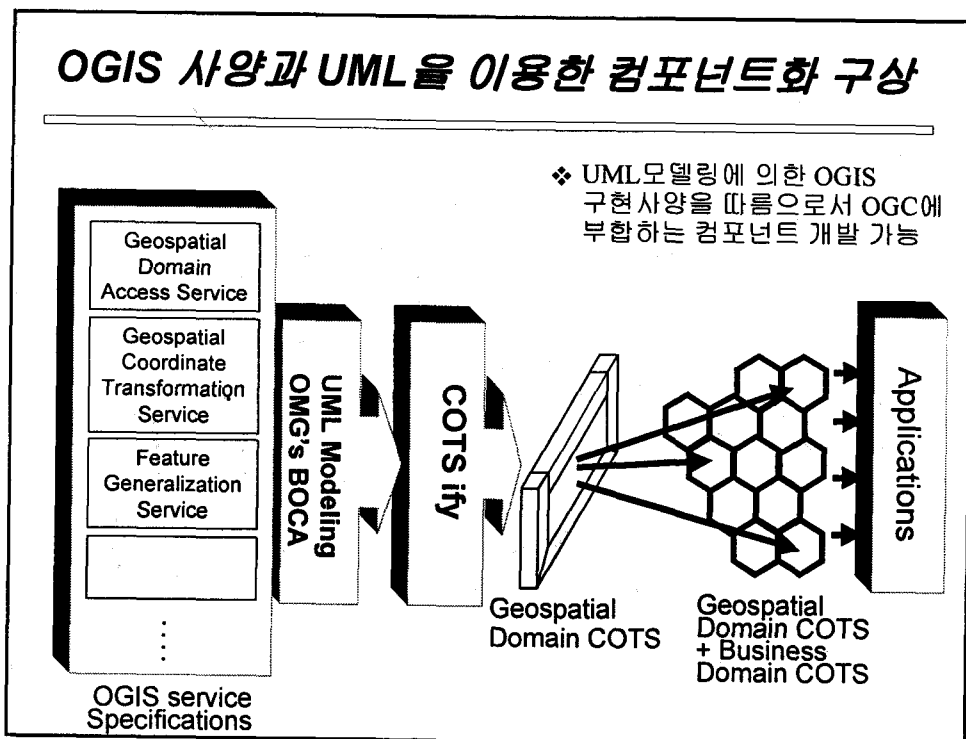
완전하게 컴포넌트화된 모습



Industry-level의 컴포넌트 표준



OGIS 사양과 UML을 이용한 컴포넌트화 구성



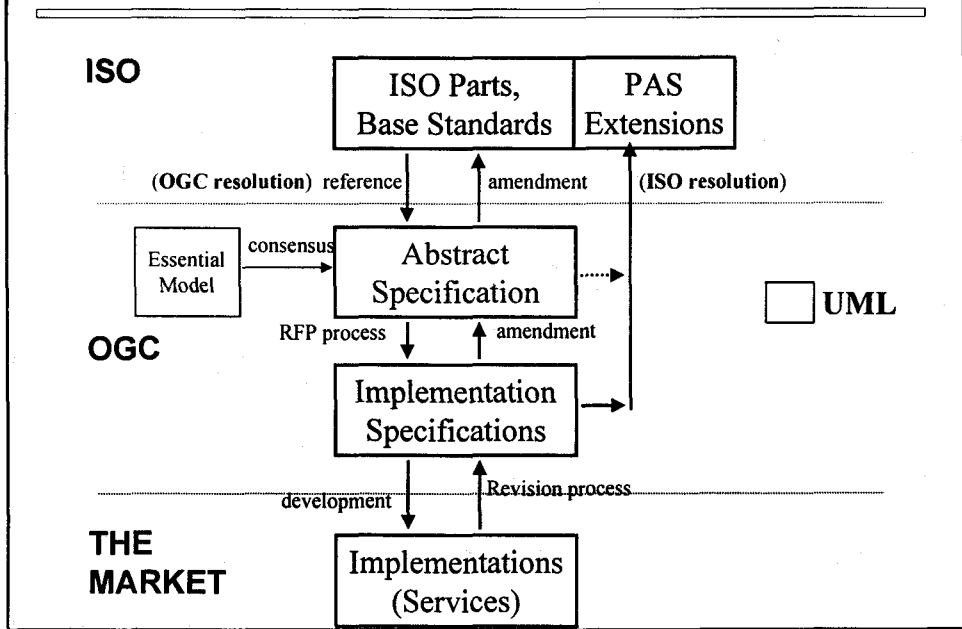
UML 소개 - 1/2

- '90년대 들어 프로그램 수준에서 소프트웨어 시스템을 객체지향 방식으로 구축하는 것만으로는 재사용이나 유지보수가 생각처럼 쉽지 않음
- 많은 개발자와 관리자들이 프로그래밍을 위한 모델링에 관심을 가지게 됨
- '80년대 말부터 등장한 객체지향 모델링
 - OMT(Object Modeling Technique) ,Booch Method
 - OOSE (Object-Oriented Software Engineering),
 - RDD(Responsibility Driven Design) , Shalaer-Mellor Method
- '97년 1월에 OMT, Booch, OOSE 세가지 방법론을 통합한 UML 1.0 발표
- 대규모의 복잡한 소프트웨어 시스템 개발에 필요한 가공물에 대한
 - 모델링을 위한 구성요소 제시
 - 이를 이용한 추상화 방법
 - 이들의 결과물 산출을 개발자들이 수비게 이해할 수 있도록 가시화 방법
 - 산출물 문서화 방법

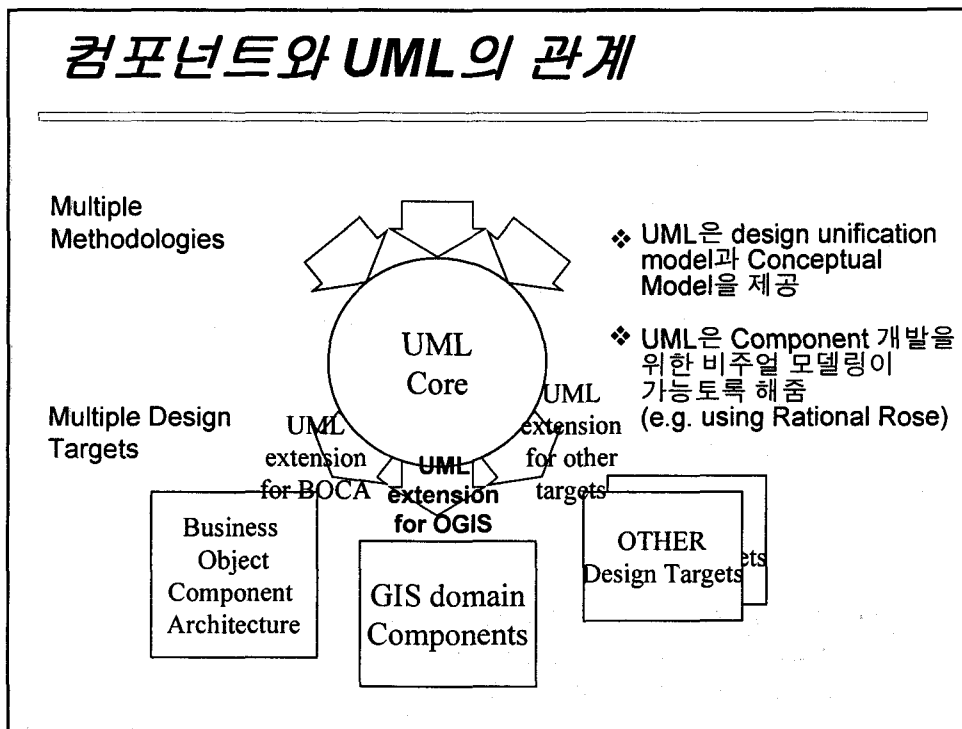
UML 소개 - 2/2

- **Use Case 모델링**
 - 사용자의 요구사항 정리
- **객체/클래스 모델링**
 - 현실세계의 복잡한 문제들을 추상화하여 풀어야 하는 문제의 범위를 개념적으로 현실의 문제보다 더 광범위하게, 현실적으로 현실의 문제보다 축소
- **개발 컴포넌트 모델링**
 - 소프트웨어의 레고 블록 식 조립 가능
- **분산/배치 모델링**
 - 개발된 후의 소프트웨어 컴포넌트를 복잡한 분산 처리나 클라이언트/서버 환경에 물리적으로 어떻게 배치할 것인지?
- **UML 1.0 의 8가지 모델링 다이어그램**
 - Use case, Sequence, Collaboration, Class, Component, Deployment, State, Activity

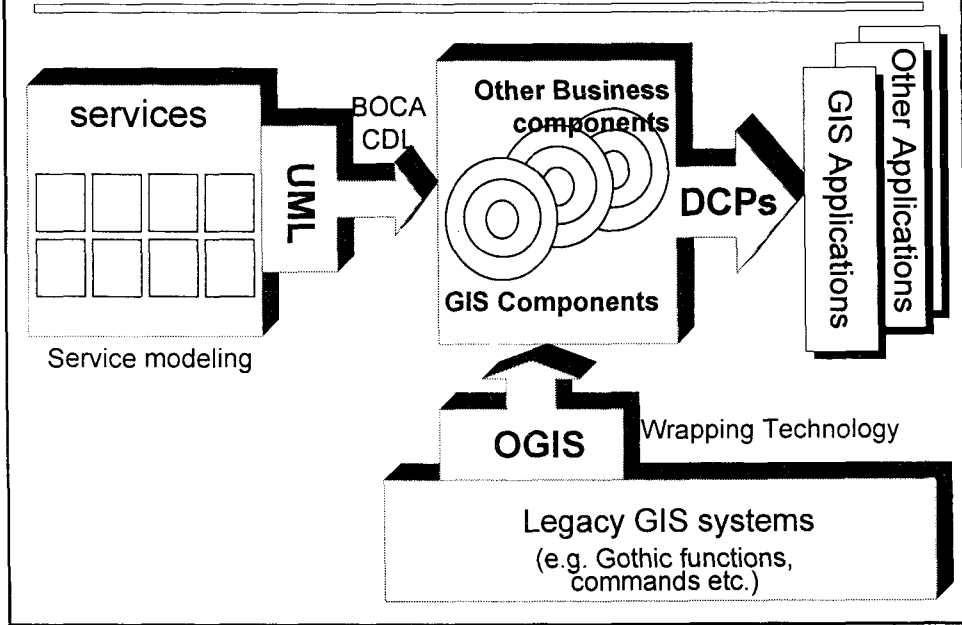
UML을 이용한 ISO와 OGC의 협력



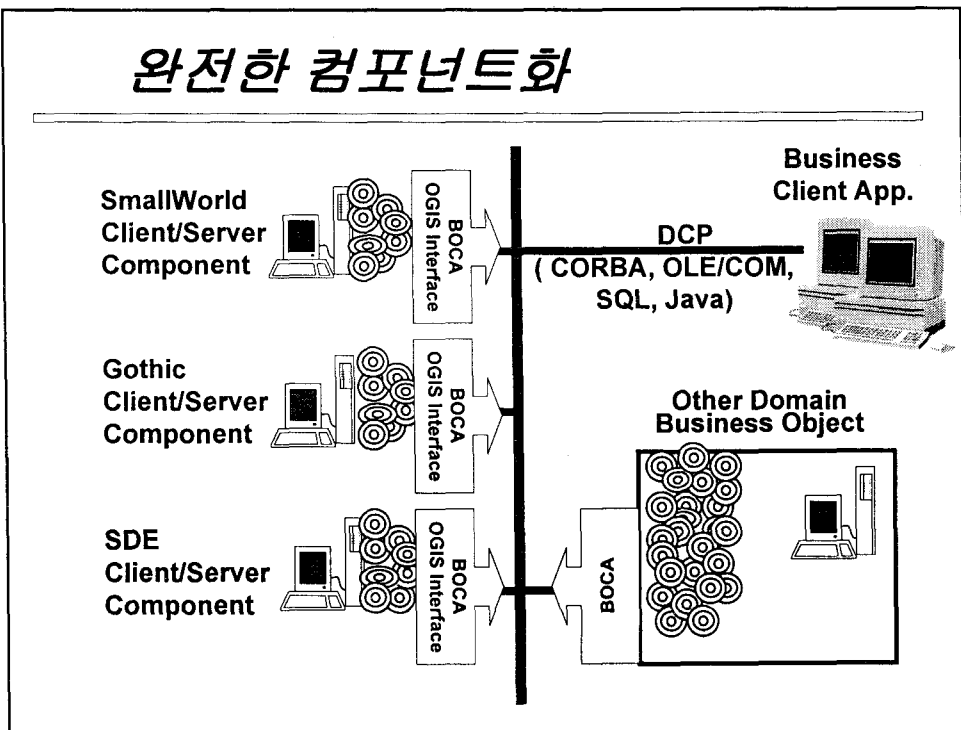
컴포넌트와 UML의 관계



OGC의 컴포넌트화 구상



완전한 컴포넌트화

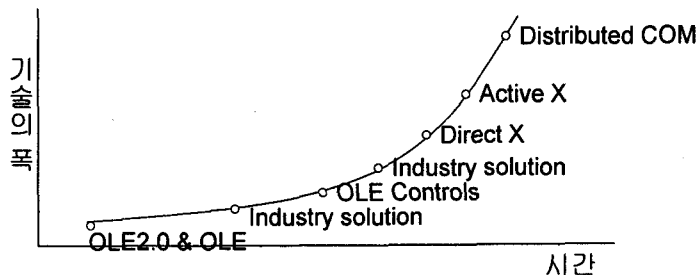


컴포넌트 분야의 기반 표준

- Microsoft: OLE/COM
- CORBA: OpenDoc/SOM

OLE/COM 소개

- 현재 뿐만 아니라 미래의 새로운 설계도 수용할 수 있도록 확장성을 지닌 마이크로소프트사의 시스템 객체기술(systems object technology)
- 시스템 객체기술(systems object technology): 현재 운용중인 운영체제에서 객체지향 원리를 구현함을 의미함. 즉, 캡슐화되고(encapsulated), 다형성을 지니며(polymorphic), 재사용이 가능한(reusable) 컴포넌트를 사용할 수 있는데 이 때, 소스 코드 수준이 아니라 binary entity로써 상호운용 가능한 것이 특징임



COM의 정의와 구조

• 정의

- Component Object Model(COM)에 기반한 일체의 기술 및 서비스를 통칭하는 마케팅 명칭을 ActiveX라고 하며, DCOM은 longer wire상의 COM이라 할 수 있음

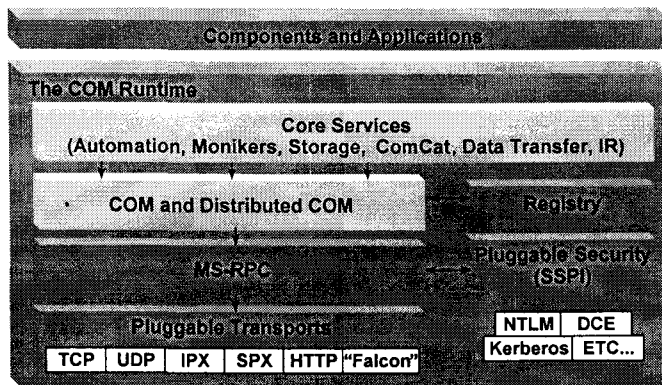
• COM의 구조

- COM의 구조는 크게 모델, ORB, 서비스의 3부분으로 나뉨
- Component Object Model: 컴포넌트, 이진 표준, wire-level 표준
- COM의 기본 원리
 - 이진 컴포넌트(binary components): in-process, cross-process, cross-machine
 - in-process 수행능력의 감소를 가져와서는 안됨
 - 가능한 단순한 모델이어야 함. 즉 확장성 및 적응성이 있어야 함

• Distributed COM (DCOM)

- 긴 통신선상에 적용하는 COM을 말한다.
- 모든 COM, 컴포넌트들이 같은 방식으로 통신이 가능하다.
- 동일 도구, 지식, 코드를 사용한다.
- 이러한 DCOM은 wire 프로토콜로서 DCE RPC 사양에 기반하고 있어 OS의 DCE RPC 구현물들과 상호운용가능(interoperable)하다.
- ORPC라고도 한다.
- 효율적이면서 scalable 함

COM"ORB"의 구조



• COM의 핵심 서비스

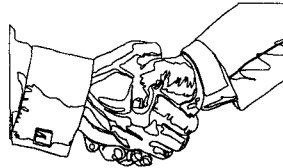
- 보안, 수명주기관리, Type Information, Monikers, 자동화 자료전송, 컴포넌트 범주화, Registry

OMG의 임무

Cooperative creation and promulgation of object-oriented standards based on *commercially available software*, guaranteeing:

- reusability of components;
- interoperability & portability.

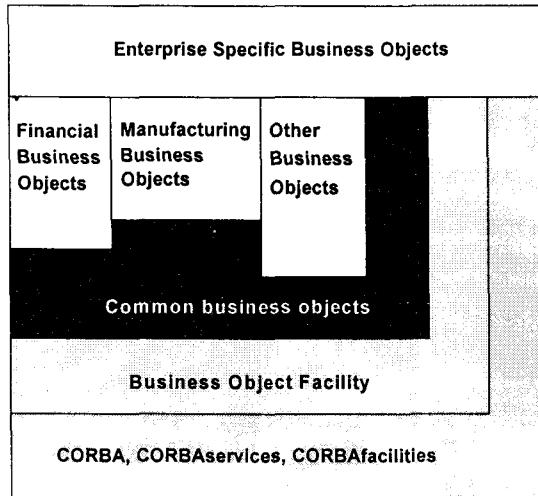
Focus on swiftly-developed, easily usable off-the-shelf standard components.



Business Objects

- Provide the infrastructure for distributed business applications
- Provide for cooperative business objects
- Empower users to construct business applications from components
- Business model and applications are the same "shape"
- Component assembly and reuse

Business Object의 위치

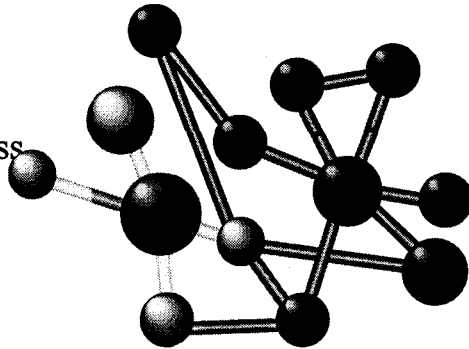


BOCA의 필요성

- BOCA(Business Object Component Architecture)
 - the fundamental unification point for business objects is the application architecture represented by the BOCA Meta-Model. The BOCA Meta-Model describes the constructs and types that are to build a business object system
- BOCA의 필요성
 - Exist a series of technical services(CORBA etc.) and a way to exchange designs between analysis and design tools(UML)
 - but lacking the application architecture, technology and specification mechanism that specialize these designs and services for distributed component business applications

협력지향적 Business Object

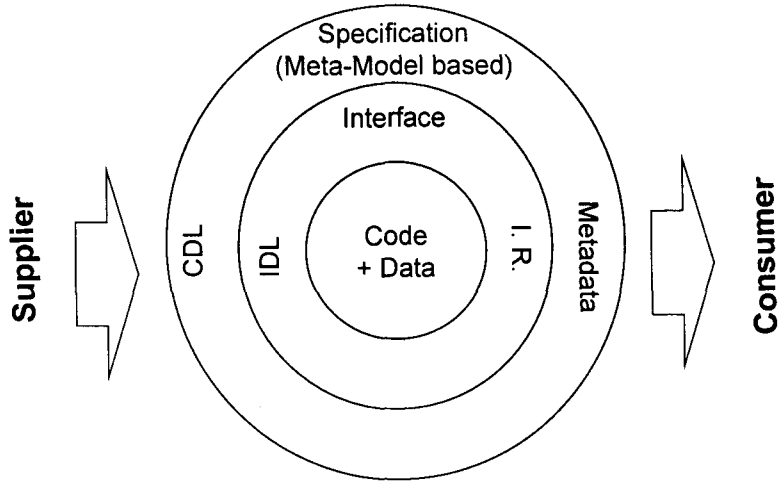
- Business object는 반드시 독립적인 컴포넌트
- 그러면서도 관련 business object와 상호운용 가능해야 함
- 이를 통해 다른 business object components와 상호연계하는 격자형 구조를 형성



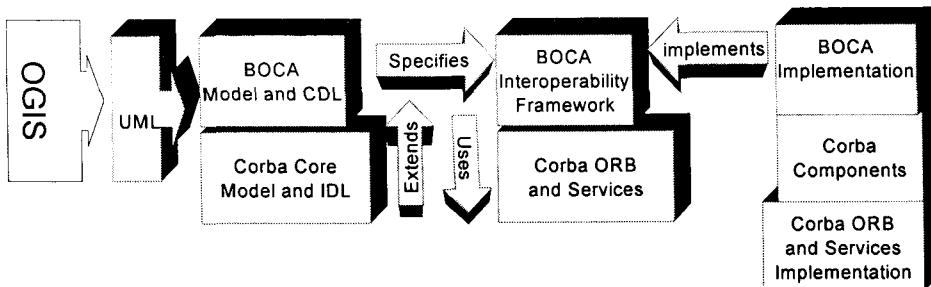
BOCA의 잇점

- High level specification including...
 - Everything in IDL plus
 - Relationships, Events, States, Rules and Roles
 - Entities, Processes, Business events, Dependents
- Integrated framework providing...
 - Persistent, Concurrent, Transactional, Secure, reflective business objects
 - Referential integrity, Type Managers (With factory), Thin-client support and Query

완전한 Component란



OMG의 컴포넌트화 구상



목 차

1. OpenGIS 태동 배경
2. OpenGIS 개요
3. OpenGIS 기술개발 절차
4. OpenGIS vs. SDTS
5. OpenGIS와 컴포넌트
6. *Case study: OpenGIS를 이용한 NGIS
기술개발 과제 통합(안)*

NGIS 기술개발 사업 개요

❖ **NGIS 구축에 필요한 공통 기반 소프트웨어 기술을 종합적이고 체계적으로 개발**

- 1999년까지 대략 184억원 투입
- 관련 정부와 참여기업 50%씩 투자

❖ **4개 중과제**

- GIS 기본 S/W 개발: Gothic기반의 generic 명령어를 이용한 DBMS independent GIS S/W 개발
- 매핑 S/W 개발: 새로운 매핑 알고리즘 및 S/W 개발
- DB Tool 개발: 공간객체관리 및 저장 시스템, OODBMS 및 RDBM 인터페이스 개발
- 시스템 통합: 이상 3개 과제의 통합 방안 및 상품화 연구

❖ **목표**

- NGIS 구축 기반기술 개발
- 국가 표준에 부합한 GIS S/W의 개발
- NGIS 구축사업의 효율적 지원

NGIS 기술개발 과제의 구조

시스템 통합 기술: 3개 중과제

기본 S/W 개발	매핑 S/W 개발	DB Tool 개발
API 개발	자료획득 및 수정 S/W	공간객체 관리시스템
분석 S/W	자료처리 및 3차원 지형분석 S/W	공간객체 저장시스템
조작 S/W	지도출력, 지도설계 자동화	OODBMS Interface
User Interface		RDBMS Interface

문제 제기 -1/2

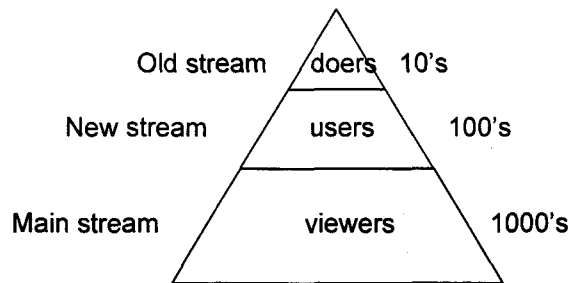
- 각 과제별 서로 다른 플랫폼
- 각 과제들은 서로 다른 GIS S/W에 기반하고 있음 (Gothic, GRASS, 또는 자체 OODB 등)
- 바람직한 System Integration에 대한 전략의 부재
- 3개의 중과제를 하나의 커다란 conventional GIS로 통합하는 것이 과연 필요한가?

?

시스템 통합 기술: 3개 중과제		
기본 S/W 개발	매핑 S/W 개발	DB Tool 개발
API 개발	자료획득 및 수정 S/W	공간객체 관리시스템
분석 S/W	자료처리 및 3차원 지형분석 S/W	공간객체 저장시스템
조작 S/W	지도출력, 지도설계 자동화	OODBMS Interface
User Interface		RDBMS Interface

문제 제기 -2/2

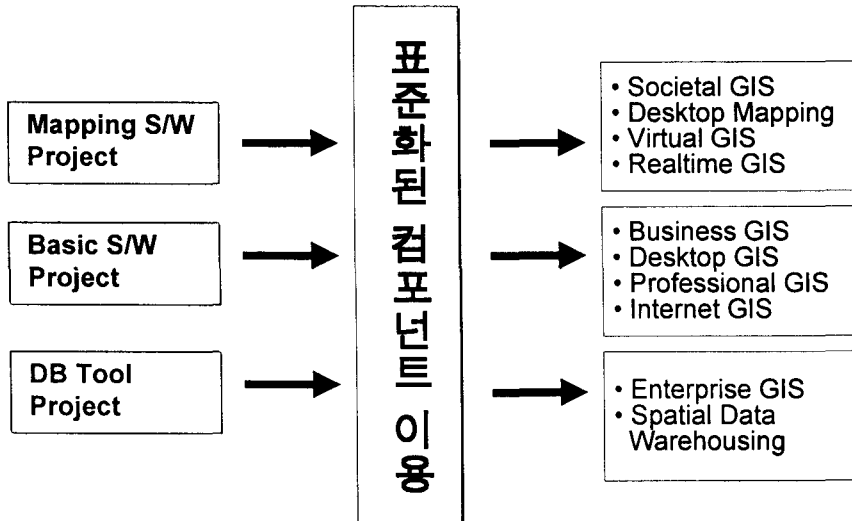
- NGIS 기술개발 과제가 계획된 당시에 비해 현재 정보기술이 급격히 변화하였음
- Viewer 시장의 증대
- Conventional GIS (Professional GIS)는 과거에 비해 GIS SW 시장에서 점차 그 중요성이 떨어지고 있음



시스템 통합 방안: OGIS 기반의 컴포넌트화

- 세분 시장에 민감하게 대처할 수 있다
- 컴포넌트의 다양한 조합을 통해 틈새시장에 대비할 수 있다
- 세계적으로 경쟁력을 갖춘 GIS 컴포넌트를 개발하는 것이 하나의 덩지 큰 traditional GIS 패키지를 개발하는 것보다 성공 가능성이 높다

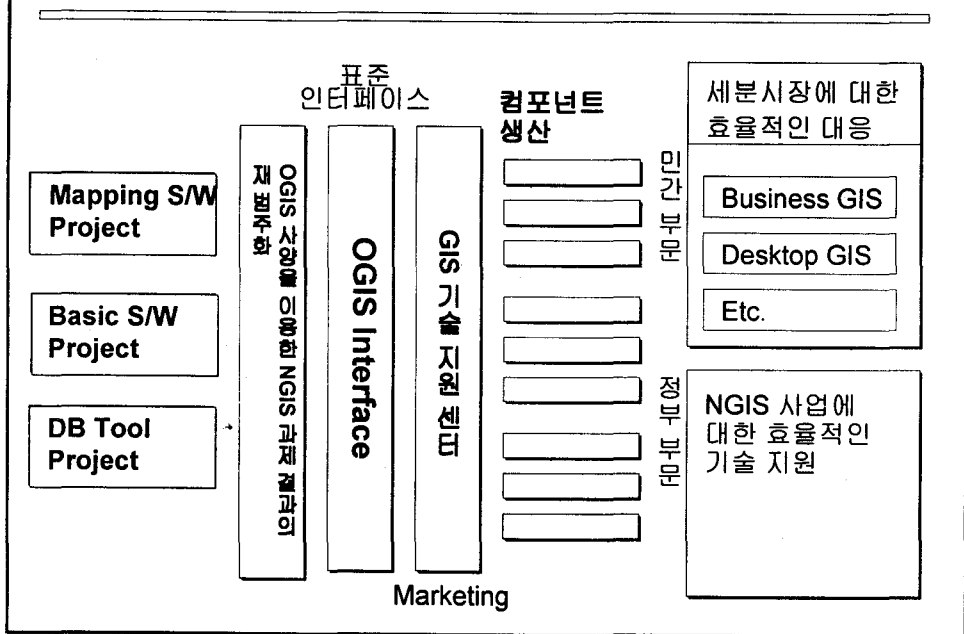
컴포넌트를 통한 시장 주도



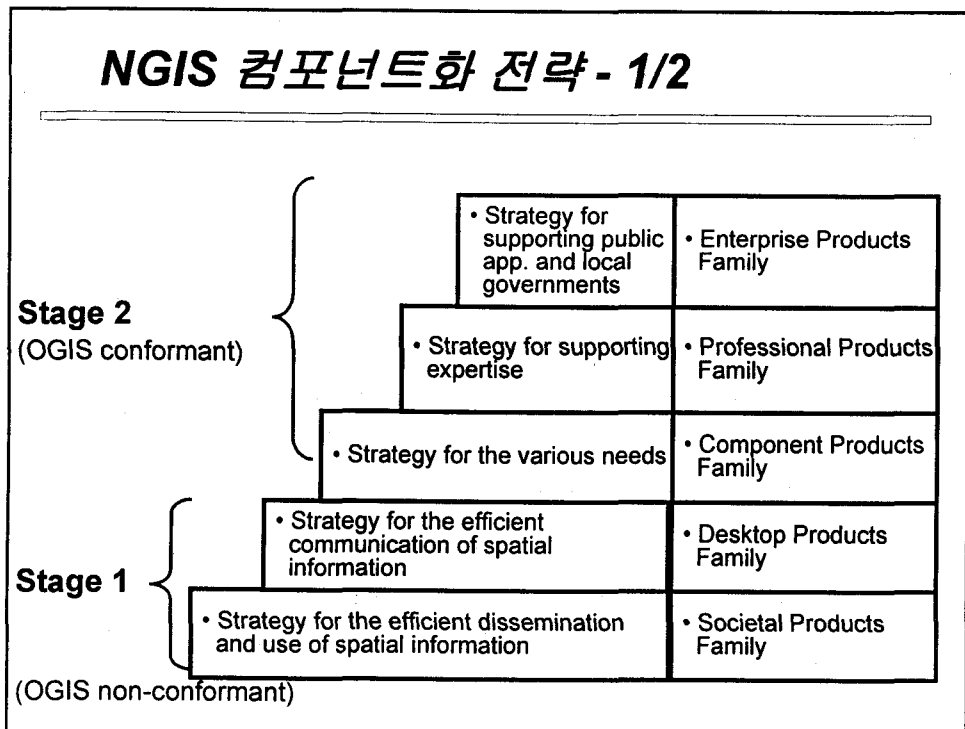
시장 세분화

- Societal 제품 계열: Societal GIS, Business GIS
- Desktop 제품 계열: Desktop Mapping, Desktop GIS
- Component 제품 계열: Component GIS, Internet GIS
- Professional 제품 계열: Professional GIS, Realtime GIS, Virtual GIS
- Enterprise 제품 계열: Enterprise GIS, Dataware House

NGIS 컴포넌트화 방향



NGIS 컴포넌트화 전략 - 1/2



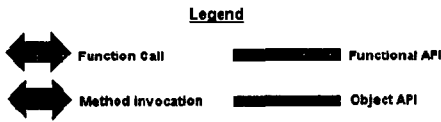
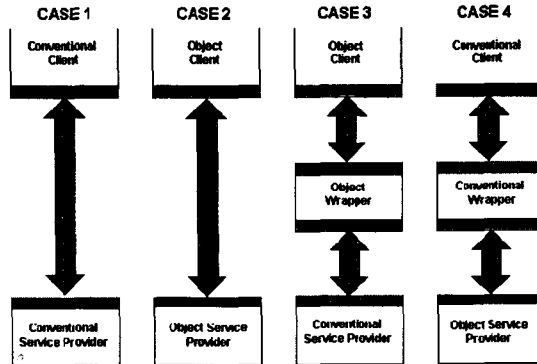
NGIS 컴포넌트화 전략 - 2/2

- **단계 1: legacy system wrapping technology 를 이용한 Conventional System Integration(OGIS non-conformant)**
 - OGIS service architecture에 따라 NGIS 과제 결과 재범주화 작업
 - 재범주화된 NGIS service를 Societal 및 Desktop 제품계열로 패키징 (OGIS non-conformant)
 - Legacy system에 대한 wrapping technology 채용
- **단계 2: 세분시장별 컴포넌트 개발 /조합 (OGIS conformant)**
 - wrapping technology와 OGIS specification을 통해 단계 1의 NGIS service들의 완전한 컴포넌트화 (OGIS conformant)
 - Professional 분야 어플리케이션을 위한 컴포넌트 상품 개발
 - 다른 business object COTS들과 상호운용할 수 있는 enterprise 어플리케이션을 위한 컴포넌트 상품 개발

Legacy System 에 대한 Wrapping Technology

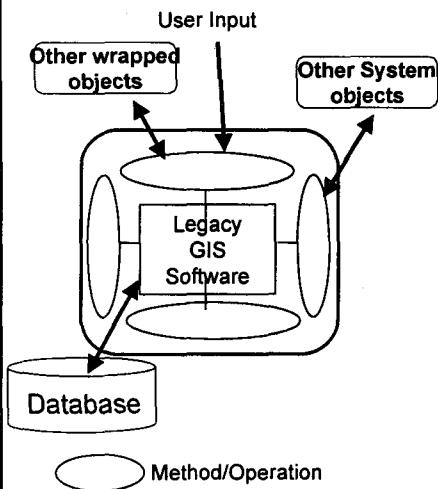
- Software wrapping is a technique in which an interface is created around an existing piece of software, providing a new view of the software to external systems, objects, or users
- Wrapping의 장점
 - establish compatibility of old code with new interoperable data description standards by supporting the translation between these standards and legacy formats at the methods interface to the wrapped software
 - facilitates rapid transition of multiple legacy systems to fewer and more interoperable systems, thereby improving interoperability and reducing maintenance and modification costs

OO와 non-OO 기술을 활용한 wrapping

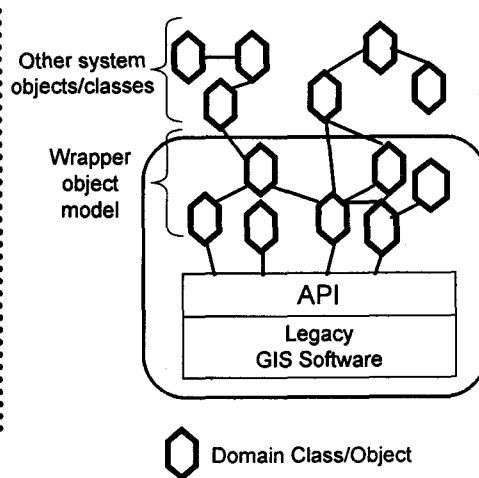


두 가지 형태의 Legacy System Wrapping Technology

Wrapped as an object, without API



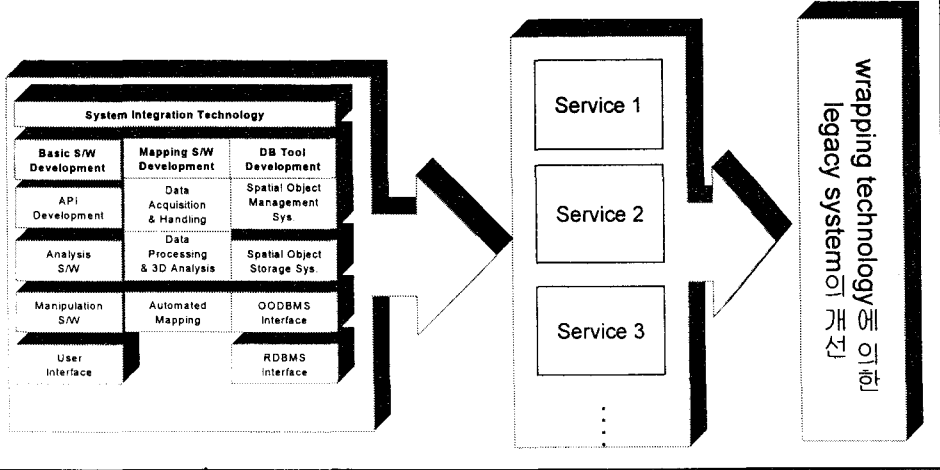
Wrapped with Object Model with API



단계 1 - 컴포넌트화 준비

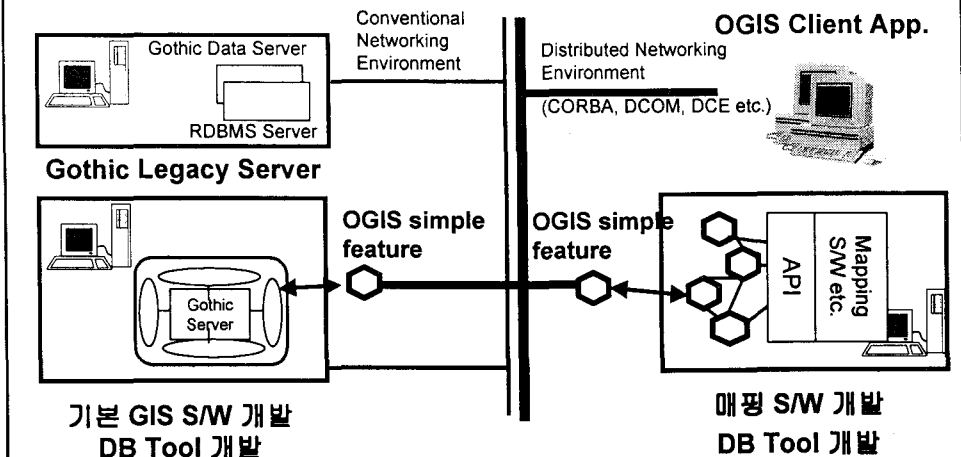
OGIS 사양을 참조하여 NGIS
과제 결과들의 재범주화

재범주화된 NGIS service
들의 패키지

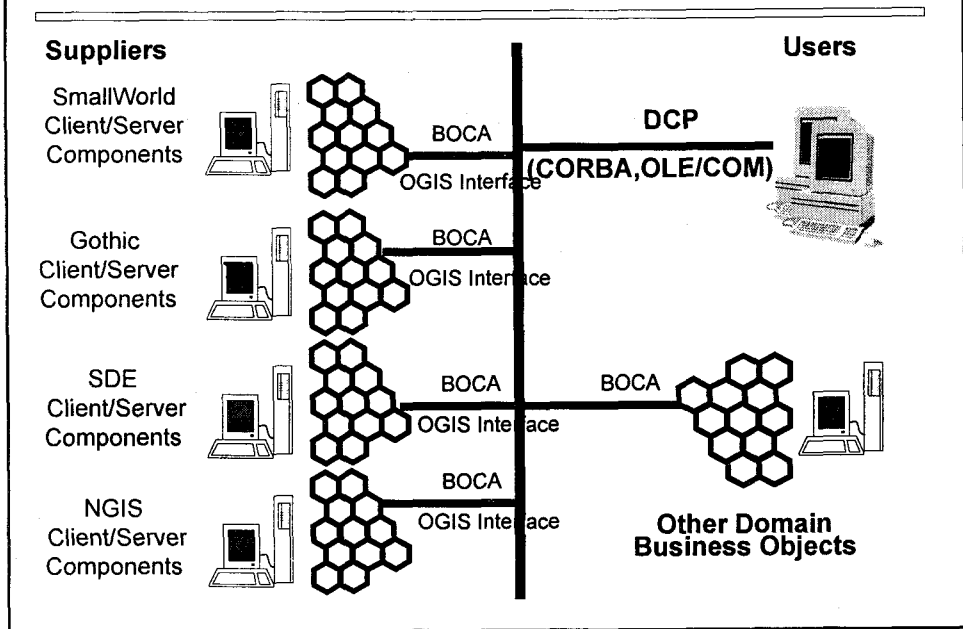


단계 1.5 - Legacy S/W Wrapping

- ❖ OGC 구현 사양의 완료시점: 2000 년경
- ❖ 완전 컴포넌트로 가는 중간 단계로서 OGIS 의 방향을 따름

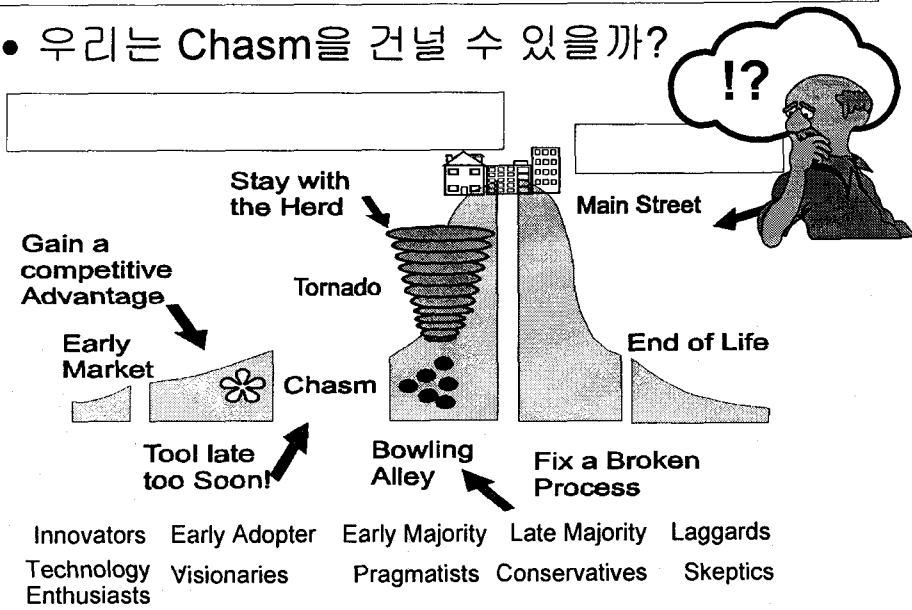


단계 2 - 완전 컴포넌트화



High-Tech 시장 개발

- 우리는 Chasm을 건널 수 있을까?



남겨진 문제들

- 창안품(Devices) vs. Whole products
- 개발 SW를 지속적으로 유지, 발전 시킬 민간부문 컨소시엄 구축
- 시장 진입전 conformance test
- 구체적인 마케팅 계획
- 컴포넌트 개발을 위한 국제적 협력

감사합니다

*<http://gis.kyungwon.ac.kr>
E-mail: eun1955@bora.dacom.co.kr*