

컴포넌트 기반 지리 정보 시스템을 위한 엔터프라이즈 서버의 설계 및 구현[†]

Design and Implementation of an Enterprise Server for Component-based Geographic Information Systems

이강준*, 홍동숙*, 박지웅*, 한기준**

Kang-Jun Lee, Dong-Suk Hong, Ji-Woong Park, Ki-Joon Han

요약 최근 정보기술(Information Technology : IT) 분야의 새로운 소프트웨어 패러다임 (paradigm)은 분산 컴퓨팅 기법이 가져온 컴포넌트 기술이다. 컴포넌트 기술은 프로그램 개발에 소요되는 시간을 상당히 단축시킬 수 있으며, 인터넷/인트라넷 환경에 적합한 분산 컴퓨팅 환경을 쉽게 구축할 수 있도록 설계되어 있다. 이러한 소프트웨어 기술의 변화는 지리 정보 시스템(GIS) 분야에도 커다란 영향을 주었으며, 또한 지리 정보 시스템의 기술적, 정책적 측면에서 컴포넌트 기술이 매우 필요하게 되었다.

본 논문에서는 분산 GIS 컴포넌트들의 효과적인 개발과 운용을 위해 선 마이크로 시스템에서 산업 표준으로 제시한 Enterprise JavaBeans Specification 1.1을 적용하여 GIS 컴포넌트를 위한 엔터프라이즈 서버를 개발한다. 사용자는 엔터프라이즈 서버를 활용하여 GIS 컴포넌트 개발을 위한 UML과 디자인 패턴 등을 관리하며, 개방형 지리 데이터 상호 운용성 사양인 OpenGIS를 준수하는 GIS 컴포넌트 컨테이너를 개발하고, 공간 XML 엔진을 이용하여 공간 메타데이터를 관리하며, 그리고 개발된 GIS 컴포넌트들의 테스트 및 검증 을 위한 TESTBED 서버를 구현할 수 있다.

ABSTRACT Recently, the new software paradigm in information technology (IT) is the component-based development (CBD) technology that has brought as a result of the distributed computing environment (DCR). The CBD has reduced a program developing time and has been designed to make a distributed computing environment easier in the Internet and Intranet environments. These changes of software technology have influence on the geographic information system (GIS) and the CBD technology is highly needed as a point of view of the technique and policy of the geographic information system.

In this paper, we designed and implemented an Enterprise Server for the efficient development and management of GIS components. Especially, the Enterprise Server was developed with the Enterprise JavaBeans Specification 1.1 that had been proposed as an industry standard by the Sun Microsystems. The Enterprise Server can allows users to manage the Unified Modeling Language (UML) and Design Patterns for the development of GIS components, develop a GIS component container which obeyed the Open Geodata Interoperability Specification (OpenGIS), manage spatial metadata through a spatial XML (eXtensible Markup Language) engine, and implement a TESTBED server for testing and verification of developed GIS components.

키워드 : Enterprise JavaBeans(EJB), 컴포넌트 GIS, 엔터프라이즈 서버, COM, CORBA, OpenGIS, XML 엔진, SDTS, UML

[†] 본 연구는 한국과학재단 '97특정기초연구(KOSEF 97-01-02-04-01-3) 과제로부터 지원받았음.

* 건국대학교 컴퓨터공학과

** 건국대학교 컴퓨터공학과 교수

{kjlee, dshong, jiwpak, kijan}@db.konkuk.ac.kr

1. 서론

최근 소프트웨어 기술의 가장 큰 이슈는 바로 컴포넌트 기술이다. 컴포넌트 기술은 소프트웨어 접근면에서 하드웨어 플랫폼에 무관하며, 구성면에서 플러그-앤-플레이 식의 소프트웨어 조립을 가능하게 한다. 또한, 관리면에서 소프트웨어 동적 재사용을 가능하게 하며, 설계면에서 컴포넌트 자체가 독립적인(standalone) 성격을 가지는 기술이다. 이러한 컴포넌트 기술은 정의된 인터페이스를 통해 접근할 수 있는 바이너리 수준의 객체로서 분산 컴퓨팅 환경에서의 원격호출(remote invoke)을 기본으로 하고 있다[8, 23]. 컴포넌트 기술에 대해서 여러 그룹이 표준 모델을 제시하였는데, 이 중 가장 두드러진 표준 모델을 제시한 그룹은 OLE/COM 사양을 제시한 마이크로소프트, CORBA 사양을 제시한 OMG(Object Management Group), 그리고 JavaBeans/Enterprise Java Beans (EJB) 사양을 제시한 SUN 등이다. 이들 각각의 표준 모델에 대해 간략히 소개하면 다음과 같다.

마이크로소프트는 COM(Compound Object Model) 이전에 소프트웨어의 기술 개발을 위한 많은 시도를 해왔다. 프로그램간의 데이터 공유를 위한 클립보드와 프로세서간 데이터 공유를 위한 DDE(Dynamic Data Exchange)를 개발하였고, 여기에 OLE 1.0 이라는 문서 객체 안에 데이터를 그대로 간직하는 삽입된 객체(embedded object) 또는 연결된 객체(linked object) 기술을 개발하였다. 그러나, 이 OLE 기술은 특수한 문서 객체만 처리한다는 문제점이 있어, 소프트웨어 기술 개발자들은 응용 프로그램에서 여러 목적으로 사용 가능한 객체를 생성하는 기술의 필요성을 느끼게 되었다. 그리하여, 이들 객체간의 표준 통신 방법의 필요성이 증대되면서 마이크로소프트사는 COM을 포함하는 OLE 2.0 기술을 소개하였는데, 이 OLE/COM 모델에서 COM 객체간의 통신 인프라스트럭처로 COM ORB가 사용된다. 또한, 현재 분산 COM간의 객체 통신을 지원하게 하는 Distributed COM (DCOM)도 제시되었다[4, 7].

OMG는 1989년 객체 기반 분산 컴퓨팅을 위한 표준 구조 개발을 목적으로 결성된 단체이며, 표준 컴포넌트 사양 개발을 전담하는 컨소시엄이다. OMG의 가장 큰 결과물은 CORBA(Common Object Request Broker Architecture)인데, CORBA는 객체간의 통신을 위한 인프라스트럭처로서 이 기종 환경하의 분산 객체 시스템을 위한 표준이다. CORBA는 객체들이 시스템 수준의 서비스에 접근하고 이용할 수 있도록

Common Object Services를 정의하고, 어플리케이션 수준에서 일반적인 함수를 제공하는 Common Facility를 정의하고 있다. 그리고, TCP/IP를 이용하여 서로 다른 ORB간에 주고 받는 표준 메시지 형식에 대한 사양 집합인 Internet Inter-ORB Protocol (IIOP)를 제공하고 있다[12].

여기에 SUN의 자바가 새로운 강력한 컴포넌트 기술의 후보자로 나서고 있다. 자바 프로그래밍 언어는 객체지향 개념에 적합하도록 설계되었고, 분산 환경 및 다중 플랫폼 환경에서 동작 가능한 바이너리 표준을 지원한다. 현재 자바는 컴포넌트 기술로서 자바빈즈(JavaBeans)를 제시하고, 서버 차원의 컴포넌트 기술을 위해 엔터프라이즈 자바빈즈(Enterprise Java Beans:EJB)를 제시하고 있다. EJB 컴포넌트 모델은 OMG를 중심으로 상호 운용성을 위해 CORBA 기술과 제휴가 활발하여 현재의 컴포넌트 모델은 크게 COM과 EJB/CORBA의 두 방향으로 양분되어 있다[20, 21, 22].

컴포넌트 기술은 현재의 인터넷과 분산/네트워크 컴퓨터에 대한 요구가 급증함에 따라 그 필요성이 더욱 대두되고 있다. 위에서 언급한 컴포넌트의 기본 개념이 가지는 이점으로 인해 현재 많은 분야에서 컴포넌트 프로그램이 개발되어 사용되고 있다. GIS 소프트웨어 분야에서도 이와 같은 필요성은 더욱 크다고 할 수 있는데, 대부분의 사람들은 현재 지리 정보 시스템이 가진 문제점에 대한 가장 명쾌한 해결책으로 컴포넌트 기술을 꼽고 있다. 그러나, 현재 컴포넌트 모델 표준의 양분화와 컴포넌트 모델만이 정립되어 있을 뿐 GIS 컴포넌트 개발을 위한 산업 표준이나 GIS 컴포넌트 개발 아키텍처가 없어, 현 시점에서 제작된 컴포넌트들은 단지 GUI나 동적 클래스 라이브러리 정도의 컴포넌트 기초 기술 수준을 벗어 나지 못하고 있는 실정이다[29].

이와 같은 기술 수준에서 제작된 국내 GIS 컴포넌트들은 재사용성을 보장받을 수 없고, 또한 인터페이스 표준이 없는 상황에서 제작되었기 때문에 다른 GIS 벤더들이 만든 컴포넌트들과 상호운용성이 어려워 효율적인 플러그-앤-플레이를 기대할 수 없게 된다. 즉, GIS 컴포넌트가 부분적으로 사용되나 GIS 산업의 생산성에는 크게 기여하지 못하게 된다. 그러므로, GIS 산업에서 GIS 컴포넌트 아키텍처의 산업 표준은 매우 중요한 위치를 차지하며, 향후 GIS 컴포넌트 시장의 우위를 차지하기 위해 반드시 필요한 것임을 우리는 예견할 수 있다. 현재 SUN에서는 컴포넌트 개발에 대한 산업 표준으로 Enterprise Java

Beans Specification 1.1을 제시하였다[20]. 이 Enterprise JavaBeans Specification 1.1을 이용하여 구현된 컴포넌트는 자유로이 컴포넌트 서버나 컨테이너를 선택할 수 있으며, 컴포넌트 개발자가 특별히 신경을 쓰지 않고 분산 컴포넌트 환경 하에서의 트랜잭션 처리가 가능하게 된다[1].

본 논문에서는 현재 컴포넌트 아키텍처 산업 표준으로 인정받고 있는 Enterprise JavaBeans Specification 1.1 사양을 준수하는 컴포넌트 기반 GIS를 위한 엔터프라이즈 서버, 즉 ESCGIS (Enterprise Server for Component-based GIS)를 설계 및 구현한다. 본 논문에서는 1장 서론을 통하여 산업 표준 컴포넌트 개발 아키텍처의 필요성을 언급하였다. 2장 관련연구에서는 컴포넌트 기술과 엔터프라이즈 기술을 분석하고, 3장 ESCGIS의 설계에서는 ESCGIS 컴포넌트 인프라스트럭처와 ESCGIS 시스템 구조를 설명한다. 4장 ESCGIS의 구현에서는 ESCGIS의 구성요소인 각 관리자별 구현에 대해 기술하며, 마지막으로 5장에서는 GIS 산업 표준을 위한 컴포넌트 아키텍처로서의 ESCGIS에 대한 결론 및 향후 연구과제에 대하여 언급한다.

2. 관련연구

본 장에서는 ESCGIS (Enterprise Server for Component-based GIS)의 설계 및 구현에 관련된 기술들인 컴포넌트 기술과 엔터프라이즈 기술에 대해 살펴본다.

2.1 컴포넌트 기술

본 절에서는 기존 컴포넌트 기술과 GIS 컴포넌트 기술, 신 컴포넌트 기술, 그리고 엔터프라이즈 기술에 대해 분석한다.

2.1.1 기존 컴포넌트 기술

소프트웨어 컴포넌트란 인터페이스를 통해서만 활용 가능한 최소단위의 소프트웨어를 말한다. 컴포넌트는 크게 분산 객체 컴퓨팅 기술 측면과 프로그램 모델 측면으로 나눌 수 있다. 분산 객체 컴퓨팅 기술 측면의 컴포넌트 모델은 COM과 CORBA로 나눌 수 있으며, 이 모델은 객체간의 통신에 대한 표준 모델을 위한 목적으로 제시되었다. 프로그램 모델 측면의 컴포넌트 기술은 객체지향 언어 프로그래밍 기술의 문제점을 상호 보완하는 새로운 프로그래밍 기법으로 제시되면서, CBD(Component-Based Development)라 불리는 개발 방법론을 가져왔다. 우리가 일반적으로 말하는

컴포넌트 기술은 이 두 가지의 요소를 같이 가지고 있다. 이 두 가지 기술 측면을 가지고 컴포넌트 기술이 최종적으로 지향하는 방향은 다른 애플리케이션, 다른 OS, 다른 하드웨어상에서 동작할 수 있는 플랫폼 무관한 바이너리 수준의 재사용성과 상호운용성이다. 즉, GIS 벤더가 개발한 GIS 컴포넌트는 소스 차원의 재컴파일이나 수정없이 서로 다른 환경의 GIS 응용프로그램을 구성할 수 있으며, 언제라도 플러그-앤-플레이어를 이용하여 교체 및 재조립을 가능하게 할 수 있다는 뜻이다. 이런 컴포넌트 기술이 가지는 개념은 GIS 산업 뿐만이 아니라 소프트웨어 산업 전반에도 매우 매력적인 요소로 작용하여 현재 컴포넌트 시장을 확대시키고 있다. 본 절에서는 현재 컴포넌트 기술의 실제 적용 사례를 들면서 기존 컴포넌트 기술의 현재 상황을 살펴본다[8, 23].

현재 컴포넌트 기술의 실제 적용은 분산 컴퓨팅을 위한 CORBA/DCOM/EJB, OLE/COM 기반의 자동화 객체 서버, ActiveX, 비주얼 컴포넌트 라이브러리(Visual Component Library)와 자바빈즈(Java Beans) 등을 말할 수 있다. 우선 CORBA/DCOM/EJB에서의 컴포넌트는 분산 환경하에서 클라이언트와 서버 형태로 분리하여 클라이언트측에서 서버측의 기능을 호출(involve) 하여 기능을 수행한다. CORBA/DCOM/EJB의 컴포넌트 기술은 컴포넌트의 분산 컴퓨팅 기술을 분산 환경에서 적용하고 있는 사례라 할 수 있으며 이 부분은 앞으로 다가올 분산 네트워크 시대를 준비하면서 지속적인 발전을 하고 있다.

OLE/COM 기반의 자동화 객체 서버는 마이크로소프트 운영 체제인 윈도우에 한하여 사용되는 기술로서 현재 ADO(Active Data Object)나 OLEDB (OLE Database) 등의 구현으로 사용되고 있다. OLE 기술은 마이크로소프트 COM을 근간으로 하고 있으며 이 기술을 통해 객체간의 통신이 가능하게 된다. 프로그래머는 자동화 객체 서버 메커니즘에 따라 객체를 개발하는데, 필요한 기능을 COM Server로 구현하고 COM Server의 인터페이스를 윈도우즈 레지스트리에 등록시킨 후 다른 사용자가 호출할 수 있도록 한다. 이렇게 등록된 COM Server는 다른 응용 프로그램에서 호출시 윈도우즈 내부에서 자동으로 실행되게 되며, 작업이 끝나면 COM Server 역시 자동으로 종료하게 된다.

ActiveX는 OLE/COM 환경에 포함되어 있는데, 현재 인터넷의 수요가 급증하면서 활용도가 가장 큰 컴포넌트 적용 사례이다. 마이크로소프트는 응용 프로그램에 플러그인 될 수 있는 모델의 제작을 위해 일찍

이 비주얼 베이직에 VBX라는 형태의 동적 적재가 가능하고, 동적으로 사용할 수 있는 모델을 제시하였다. VBX는 비주얼 베이직에서 큰 효과를 얻었으며 소프트웨어 산업에 있어 비주얼 문화를 가져오게 된 계기가 되었다. VBX를 OLE/COM 기술에 의해 재구성한 것이 바로 OCX이다. OCX는 COM 서버가 플러그인 될 수 있는 형태로 구현된 것이며, 현재 인터넷 상에서 사용 가능하도록 가변게 제작된 것이 ActiveX이다. 이러한 ActiveX 형태는 현재 컴포넌트 기술을 이용하여 가장 널리 사용되는 기술이다. 이 기술은 동적 적재가 가능한 라이브러리 형태로서 윈도우즈 OLE/COM 환경에서만 동작한다는 단점이 있으나, COM을 지원하는 모든 프로그램 언어에서는 ActiveX로 만든 컴포넌트를 사용할 수 있다. 즉, 비주얼 베이직에서 만든 ActiveX 컴포넌트를 비주얼 C++, Delphi, C++ Builder 등에서 사용 가능하다는 의미이다. 원래 C 라이브러리, PASCAL 라이브러리, 비주얼 베이직 라이브러리들은 모두가 자신들만의 중간 파일로 라이브러리를 가지고 있기 때문에 ActiveX가 있기 전에는 서로간에 작성된 프로그램 모듈을 호환할 수 있는 방법이 매우 어려웠으며, 대부분의 개발자들은 해당 언어로 재코딩하여야 되었다. 그러나, 윈도우즈 OS 환경에서 프로그램을 개발할 때 ActiveX는 아주 좋은 공용의 라이브러리 역할을 제공할 수 있다.

컴포넌트의 프로그래밍적 접근 모델면에서는 비주얼 컴포넌트 라이브러리와 자바빈즈를 들 수 있다. 객체지향 프로그래밍은 상속, 다형성 등의 고급 프로그래밍 언어 기술과 실세계를 모델링하는 방법으로 제시되어 많은 언어적 수준을 상승시켰다. 그러나, 객체들 상호간에 작용하는 표준이 없어 객체들의 공유와 재사용을 어렵게 하였으며 동적 소프트웨어 재설계에 적합한 유연한 모델을 가지지는 못하고 있다. 이런 객체지향 프로그래밍의 한계를 최근의 비주얼 프로그래밍 환경은 매우 효과적으로 극복하고 있는데 이것이 비주얼 컴포넌트 라이브러리이다. 비주얼 컴포넌트 라이브러리는 객체지향 이론에 컴포넌트 기반의 개발 방법론을 적용한다. 즉, 객체지향 프로그래밍이 자체적(stand alone)일 수 없는 단점을 이벤트라는 개념을 추가하여 극복하고 있으며, 동적인 재사용이 비주얼하게 표현될 수도 있다. 비주얼 컴포넌트 라이브러리의 좋은 예는 델파이나 비주얼 베이직의 비주얼 컴포넌트를 들 수 있는데, 비주얼 컴포넌트는 프로퍼티(properties), 메소드(method), 이벤트(event)로 구성된 확장 클래스이다. 이러한 컴포넌트 기술의 활용은 프로그래밍 모델적인 컴포넌트 개발 접근으로서 실제 비주얼 환경

하에서 프로그램 생산성을 높이는 RAD (Rapid Application Development) TOOL이라는 개념을 만들어 내었다. 이러한 비주얼 컴포넌트는 클래스 생성시 기존의 객체지향의 프로그래밍의 모습을 가지며 컴포넌트화된 형태로 사용되게 된다. 즉, 타이머 컴포넌트를 만들게 되면 타이머의 고유 속성과 메소드를 유지하는 객체지향 상속 개념과 설정된 시간마다 자동으로 이벤트가 발생하는 자체적인 객체를 설계할 수 있게 되며, 또한 이 타이머 컴포넌트는 비주얼 라이브러리 형태로 제공되어 드래그 인 드롭으로 응용 프로그램을 작성할 수 있게 된다.

비주얼 컴포넌트 기술을 자바에 적용한 형태가 자바빈즈이다. 자바빈즈는 자바의 플랫폼 무관한 언어에 비주얼 컴포넌트 라이브러리 개념을 적용한 컴포넌트 기반의 프로그래밍 접근 기술이다. 자바 컴포넌트라고 불리는 자바빈즈는 비주얼 컴포넌트 라이브러리와 마찬가지로 프로퍼티, 메소드, 이벤트를 가지고 있으며 자바 프로그래밍에서 비주얼 컴포넌트 환경의 구축을 가능하게 한다. 즉, 자바빈즈는 자바 언어가 가지고 있는 특성에 부합하여 새로운 컴포넌트 개발 환경을 제공하고 있다.

2.1.2 기존 GIS 컴포넌트 기술

GIS 컴포넌트 개발은 현재 매우 기초적인 수준에 머물러 있으나 현재 GIS 컴포넌트를 위한 움직임이 매우 활발해 지고 있는 시점으로써 개방형 GIS 표준인 OpenGIS는 Geometry와 Feature 등의 모델에 대한 표준을 제시하고 있다[15]. 상호 이질적인 지리 데이터와 지리 정보 처리 자원을 네트워크 환경에서 상호운용 가능하도록 하는 OpenGIS는 언어, 운영체제, 플랫폼 등에 종속적이지 않고, 특정 분산 환경에 국한되지 않도록 하는 14개 주제로 추상화 사양을 제시하고 있다. 또한, GIS 컴포넌트의 제작이 이미 국내외에서도 이루어지고 있는데, 국내에서는 국가 주도 하의 개방형 GIS 컴포넌트 S/W 개발을 시작하고 있는 중이다. 그러나, 기존의 GIS 컴포넌트 개발에서는 ActiveX 형태나 COM Server 형태의 윈도우즈 OS에 국한된 컴포넌트만을 개발하고 있으며, 또한 독립된 기능만을 가지는 클래스 라이브러리 형태의 컴포넌트만을 제작하고 있다. 이러한 개발은 기존의 컴포넌트가 요구하는 재사용성을 윈도우즈 기반하의 재사용성으로 제한하고 있으며 컴포넌트의 기본 지향 방향인 상호운용성과 바이너리 수준의 재사용을 얻기에는 많은 문제점을 안고 있다.

실제로 국내 ETRI에서의 GIS 컴포넌트 개발은 GUI 컴포넌트나 클래스 라이브러리 위주의 개발에 다

중 플랫폼과의 상호운용을 분산 컴퓨팅을 이용한 데이터 전송으로 해결하려는 방향으로 추진되고 있다[29]. 그러므로, 현재 개발된 GIS 컴포넌트들은 서로간의 상호운용성을 보장받을 수 없다. 우선 OLE/COM으로 개발된 GIS 컴포넌트들은 유닉스 플랫폼에서 사용할 수 없으며, 표준화된 컴포넌트 개발 아키텍처 없이 제작되었기 때문에 독립적인 기능을 수행하는 ActiveX 형태 만이 윈도우 플랫폼하의 GIS 벤더들간에 상호운용할 수 있다. 다시 말하면, 디스플레이 기능인 MAP 컴포넌트 등의 GUI ActiveX 만이 효율적으로 이용될 수 있다는 것이며, 이 같은 컴포넌트 사용 방법은 또 다른 문제점을 야기하고 있다. 즉, 실제 컴포넌트를 사용할 때 동적 클래스 라이브러리를 사용하는 형태와 같이 GIS 응용 프로그램을 제작시 컴파일 내지는 실행시 동적으로 로딩되도록 사용자가 직접 코딩으로 구현하여야 된다는 것이다. 이 말은 GIS 컴포넌트의 기술의 사용을 매우 제한시키며 응용 프로그램 제작 및 유지 보수에 대한 전반적인 효율성을 떨어뜨리게 된다.

아직까지 컴포넌트 ORB의 표준이 통합되지 않았고, 컴포넌트 개발을 위한 산업 표준 아키텍처가 제정되지 않았기 때문에 이러한 수준의 컴포넌트 기술이 국내에서 사용되고 있는 것은 매우 당연하다. 개방형 GIS 표준은 GIS 모델링과 서비스에 대한 컴포넌트 모델은 제시하고 있지만 실제 GIS 비즈니스를 위한 GIS 컴포넌트 제작 산업 표준을 제시하고 있지는 않다. GIS 컴포넌트 개발을 위해서는 개방형 GIS 표준 위에 여러 GIS 컴포넌트들을 관리하여 통합된 GIS 비즈니스 컴포넌트를 만드는 GIS 컨테이너와 같은 GIS 컴포넌트 아키텍처의 개발은 매우 중요한 사항이다.

2.1.3 신 컴포넌트 기술

기존 컴포넌트 개발 방법론에서 제기된 문제점을 살펴보면 컴포넌트들이 계속적으로 개발되고 있으나 산업 표준의 부재, 컴포넌트 인프라스트럭처를 관리하기 위한 제도적인 기술의 부재, COM/DCOM 개발시 컴포넌트의 원래 지향 방향인 상호운용성의 한계, 소프트웨어 모델링에 입각한 컴포넌트 개발 방법론의 필요성, 그리고 컴포넌트들의 운영 아키텍처에 관한 연구의 미비 등이다. 이와 같은 문제점을 극복하기 위한 신 컴포넌트 기술 동향을 살펴 보자. 급속히 발전하고 있는 컴포넌트 기술의 신 동향은 4가지로 분류될 수 있는데 컴포넌트 개발 및 운영을 위한 표준화, N-tier를 지향하는 소프트웨어 개발, 분산 컴포넌트 기술의 확대, 그리고 소프트웨어 디자인 패턴을 이용한 표준

모델링 기법이다.

먼저 컴포넌트 개발 및 운영을 위한 표준화는 선 마이크로 시스템즈의 엔터프라이즈 자바 애플리케이션 기술인 Enterprise JavaBeans 1.1 Specification [20]을 들 수 있다. EJB는 마이크로소프트사의 MTS와 유사하나 산업 표준을 위해 제시된 사양이라는 점에서 차이가 있다. 즉, MTS가 마이크로소프트 COM에 대한 분산 처리를 위한 하나의 프로그래밍에 반해 EJB는 하나의 산업 표준을 위한 서버측 컴포넌트의 제작과 운영에 대한 표준 사양이다. 현재 이 사양은 JAVA 사양과 CORBA 사양이 나와 있으며, 플랫폼 무관한 분산 컴포넌트 개발에 대한 산업 표준을 제시하고 있다. 개발자는 이 사양을 이용해서 윈도우 OS상에서 DCOM을 사용하는 EJB 환경을 구축할 수 있으며, 그렇게 되면 DCOM 컨테이너와 COM 클라이언트 컴포넌트들의 관리가 가능하게 된다. EJB 사양은 앞으로 컴포넌트 개발을 위한 산업 표준으로 자리 잡을 것이며 현재 ORACLE, SYBASE, IBM, INPRISE 등의 수 많은 컴퓨터 그룹들이 컴포넌트 개발 표준으로 EJB를 채택하고 있다.

두 번째는 N-tier를 지향하는 소프트웨어 개발이다. 현재의 소프트웨어 산업은 기존의 클라이언트-서버 환경으로 구성된 2-tier 애플리케이션이 가지는 문제점을 N-tier로 변환하는 방향에 초점을 맞추고 있다. N-tier 애플리케이션은 클라이언트측 컴포넌트와 서버측 비즈니스 컴포넌트들로 구현되어 분산 컴포넌트 통신을 하게 되며, 서버측 비즈니스 컴포넌트 개발의 중요성이 매우 증대되고 있다. 앞으로 N-tier 구조는 소프트웨어 산업 발전에 큰 영향을 미치게 될 인자가 될 것이다.

세 번째 분산 컴포넌트 기술의 확대는 CORBA와 EJB의 상호연동을 들 수 있다. CORBA는 자바의 분산 표준 기술인 EJB와 긴밀한 관계를 가지며 상호 교류하고 있다. EJB는 CORBA 관련 사양으로 Enterprise JavaBeans to CORBA MAPPING 1.1[21]을 제시하고 있으며 분산 컴포넌트 기술이 산업 표준을 기반으로 한층 진보되고 있다. 이와 동시에 자바와 CORBA Naming 서비스가 활성화되고 있으며, 분산 컴포넌트시에 반드시 고려되어야 할 분산 컴포넌트 트랜잭션에 대한 기술이 선보이고 있다. 이러한 분산 컴포넌트 기술의 확대는 현재 인터넷이나 인트라넷의 도래로 거대한 컴퓨터 네트워크로 기업이 제어 및 관리가 되고 있기 때문에 더욱 그 움직임이 가속화되고 있다.

마지막으로 최근 소프트웨어 디자인 패턴이라는 새

로운 프로그래밍 모델링 기법들이 컴포넌트 개발을 위해 제시되고 있다. 객체지향 모델링 기법에서 출발하여 OMT, Booch, OOSE의 세가지 방법론이 통합되어 개발된 UML(Unified Modeling Language)[2]은 소프트웨어 디자인 패턴을 분석하고 설계하는 방법론을 제시하고, 현재 OMG의 객체지향 분석/설계를 위한 표준으로 결정되었다. 그러므로, 표준으로 채택된 컴포넌트 개발 모델링 도구들은 인터넷 상의 복잡한 분산 컴퓨팅 환경이나 기업 규모의 시스템 구축에 있어서 컴포넌트를 사용한 전반적인 설계(설계 및 구현, 유지보수, 재사용성 등)를 가능하게 한다.

2.2 엔터프라이즈 기술

엔터프라이즈 기술은 N-Tier를 구성하는 서버 애플리케이션 제작을 위해 제시된 자바의 기술이다. 즉, 서버 애플리케이션은 N-Tier를 위해 종종 그룹화되는데 이를 엔터프라이즈 애플리케이션이라 부른다. 엔터프라이즈 애플리케이션은 현재 소프트웨어 산업에서 가장 연구가 활발한 분야로서 인터넷/인트라넷 구조에 적합한 방향으로 제작되고 있다. 엔터프라이즈 애플리케이션을 제작하기 위해 사용되는 기술인 엔터프라이즈 기술은 JDBC(Java Database Connectivity), JNDI (Java Naming and Directory Interface), Servlets, JSP(Java Server Page), RMI(Remote Method Invocation), EJB(Enterprise Java Beans), JMS(Java Messaging Service)의 7가지로 나뉘어 진다. 이들 각각을 살펴보면 다음과 같다 [1].

JDBC는 관계형 데이터베이스를 데이터 소스로 접근하기 위한 표준 자바 API이다. JDBC는 데이터베이스 벤더를 위한 표준 아키텍처를 제공하며 대부분의 자바 프로그래머는 데이터베이스 접근을 위해서 JDBC를 사용한다. JDBC는 데이터 소스에 직접 SQL 문장을 적용할 수 있으며, ANSI SQL 92 Entity Level 표준을 지원하고 있다. JNDI는 자바 객체에 대한 디렉토리 서비스를 해준다. 즉, 실행시 등록된 객체에 대해 이름에 의한 접근을 가능하게 한다. 사용자는 엔터프라이즈 빈즈와 같이 동적으로 적재되고 수행되는 환경에서 필요한 객체를 찾을 때 네이밍(Naming)과 디렉토리 서비스를 이용할 수 있다. Servlets는 자바 웹 서버에서 CGI 기능을 확장해 주게 되는데, 엔터프라이즈 환경에서 Servlets는 많은 웹 응용 프로그램에 대한 효과적인 해결책을 제시하고 있다. Servlets는 서버측 처리 기능을 가지고 있으며, Java Server Pages나 Server-Side Includes를 사

용하는 다이나믹 웹 페이지의 기능을 확장해 준다. JSP는 Java Script를 사용해서 다이나믹 웹 페이지를 만드는데 사용된다. 서버측 Script에 해당하는 JSP는 JSP를 지원하는 웹 서버(Jserver 등)에서 동작한다.

RMI와 분산 객체는 원격 메소드 호출을 가능하게 해주는 기술로서 어플리케이션과 라이브러리는 작업 연결을 형성하기 위해 RMI를 사용한다. 사용자는 RMI를 이용하여 분산 객체가 가진 메소드를 손쉽게 원격에서 제어할 수 있게 된다. EJB는 서버 상의 자바 컴포넌트와 그 컴포넌트의 동작 환경에 대한 프레임워크를 의미한다. EJB에서 작동하는 자바 컴포넌트를 특별히 엔터프라이즈 빈즈(Enterprise Beans)라고 하며, 엔터프라이즈 빈즈가 기반을 두고 있는 동작 환경을 엔터프라이즈 빈즈 컨테이너라 한다. 엔터프라이즈 빈즈는 지속적 저장 장치(Persistent Storage)와 연결되어 데이터베이스의 레코드처럼 동작하는 엔티티 빈즈(Entity Beans)와 비즈니스 로직을 담고 있는 세션 빈즈(Session Beans)로 나뉘는데, 현재 EJB 사양에서 엔티티 빈즈는 옵션 사양이다. 기존의 컴포넌트로 만들지 않으면 개발자들이 일일이 신경을 써야 했던 데이터베이스 트랜잭션, 보안, 데이터베이스 연결 풀링(pooling), 쓰레딩과 같은 기능을 EJB에서는 컨테이너 차원에서 지원하고 있다. JMS는 MOM(Message Oriented Middleware) 공급자와 SUN에 의해 설계된 자바 메시지 인터페이스 사양이다. MOM은 메시지 지향 미들웨어인데, 메시지를 지원하는 인프라스트럭처이다. 메시지 서비스는 분산 프로그래밍 구조에서 라이브러리, 서버, 혹은 다른 미들웨어 상호간에 매우 효과적인 정보 전달자 역할을 한다

3. ESCGIS 설계

본 장에서는 ESCGIS(Enterprise Server for Component-Based GIS)의 컴포넌트 인프라스트럭처, 전체 시스템 구조와 각 모듈별 설계에 대해 설명한다.

3.1 ESCGIS 컴포넌트 인프라스트럭처

ESCGIS는 컴포넌트 산업 표준인 Enterprise Java Beans Specification 1.1을 기준으로 설계되었으며 GIS 컴포넌트 운용에 적합한 EJB 서버와 EJB 컨테이너를 제공하고 있다. 이 ESCGIS를 통해 GIS 컴포넌트 기반의 지리 정보 시스템 응용프로그램 제작시 소프트웨어 모델링에 입각하여 설계할 수 있

며, GIS 컨테이너에 의한 분산 GIS 컴포넌트 조립 및 구축을 할 수 있고, 또한 개발된 GIS 컴포넌트들에 대해 실제 사용 검증을 통한 테스트를 할 수 있다. ESCGIS의 목적은 GIS 산업 표준하에서 모든 분야에 완벽히 상호운용되고 재사용 가능한 GIS 컴포넌트 개발 인프라스트럭처를 제시하는 것이다. 표 1은 ESCGIS에서 제시하는 GIS 컴포넌트의 속성별과 기능별 분류를 보여주고 있다.

표 1. GIS 컴포넌트의 속성별과 기능별 분류

대분류	속성별 분류	컴포넌트 기능별 분류
클라이언트측 GIS 컴포넌트	GIS 디스플레이 컴포넌트	ZOOM, FADE, HIDE, OVERLAY, PAN, WINDOW MANAGEMENT, LABELING
	GIS 출력 컴포넌트	PRINTING, PLOTTING, GRAPHIC FORMAT 변환
서버측 GIS 컴포넌트	GIS 접근 컴포넌트	Data Provider, Data Format Conversion, Layer, Version, Query, Configuration, Raster, Spatial Reference
	GIS 컨트롤 컴포넌트	GRID, Static, Edit, Spatial Data Mining

우선 GIS 산업 표준을 위한 ESCGIS 컴포넌트 인프라스트럭처에서는 GIS 컴포넌트를 크게 클라이언트측 GIS 컴포넌트와 서버측 GIS 컴포넌트로 구분한다. 여기서 클라이언트측 GIS 컴포넌트란 서버측 컴포넌트를 조작하는 사용자나 다른 컴포넌트와 게이트웨이 역할을 하는 컴포넌트를 말한다. 한 예로 사용자가 직접 접근하는 GUI와 같은 용도의 컴포넌트로서 GIS 데이터 디스플레이 컴포넌트와 데이터 출력 컴포넌트를 들 수 있다. GIS 데이터 디스플레이 컴포넌트는 줌(ZOOM), 페이드(FADE), 숨김(HIDE), 오버레이(OVERLAY), 팬(PAN), 윈도우 관리(WINDOW MANAGEMENT), 라벨링 (LABELING) 등이고, 데이터 출력 컴포넌트는 프린팅(Printing), 플로팅(Plotting), 그래픽 포맷 출력 등을 들 수 있다.

서버측 GIS 컴포넌트는 N-tier의 구성에 참여하는 컴포넌트로서 서버 컴포넌트일 수 있으며, 순수한 비즈니스 애플리케이션 조립에 참여하는 일반 컴포넌트일 수 있다. 여기서 서버 컴포넌트란 ORB나 RMI에 의한 서버-스텝(Server-Stub)을 가지는 컴포넌트를 의미하며, 클라이언트측 컴포넌트 부름에 항상 대기하

는 컴포넌트를 의미한다. 일반 컴포넌트란 서버측의 비즈니스를 위한 데이터 변환이나 통계 컴포넌트 같은 하나의 기능을 가진 컴포넌트를 의미한다. 서버측 GIS 컴포넌트는 크게 GIS 접근 컴포넌트와 GIS 컨트롤 컴포넌트로 나눌 수 있다. GIS 접근 컴포넌트는 GIS 데이터에 대한 접근을 관리하는 컴포넌트로서 GIS 데이터 Provider, GIS 데이터 포맷 변환, GIS Layer 관리, GIS 버전 관리, GIS 질의 관리, GIS 형상 관리, GIS 래스터 관리, 그리고 GIS 공간 좌표 변환 등을 말하며, GIS 컨트롤 컴포넌트는 GIS 접근 데이터를 가지고 서버에서 사용하는 컴포넌트로서 GRID 분석, GIS 데이터 통계, GIS 데이터 편집, GIS 데이터 측정, 그리고 공간 데이터마이닝 같은 컴포넌트들이 속한다. 이와 같은 GIS 컴포넌트를 ESCGIS의 견해에서 보면 그림 1과 같다.

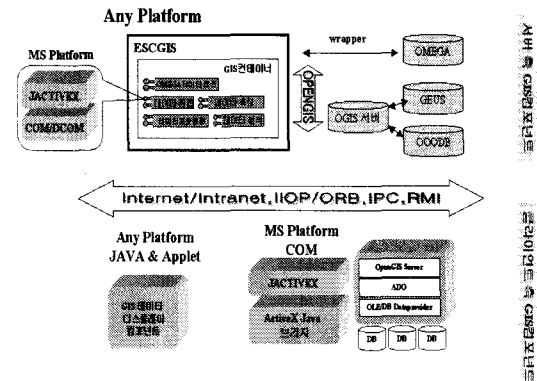


그림 1. ESCGIS 컴포넌트 인프라스트럭처

ESCGIS가 추구하는 기본적인 컴포넌트 모델은 서버측 컴포넌트를 Enterprise JavaBeans로 작성하는 것을 전제로 한다. 클라이언트측 컴포넌트는 JAVA나 COM으로 구현된 것 모두가 사용될 수 있는데, 만약에 클라이언트측 컴포넌트가 COM으로 제작되었을 경우는 COM 모델에 대한 다중 OS 지원 사양이 없기 때문에 현재는 마이크로소프트 윈도우즈 OS 환경에서만 동작할 수 있다. 그러나, 자바로 클라이언트측 컴포넌트를 만들었을 경우 컴포넌트를 이용하는 것은 어떤 플랫폼 상황하에서도 동작 가능하다.

ESCGIS는 다중 플랫폼 상황에서 동작 가능한 서버이다. ESCGIS 인프라스트럭처에서 컴포넌트 개발 효율성을 검증하기 위해 다양한 플랫폼 설정에 따르는 GIS 컴포넌트의 상호운용성과 재사용성을 살펴보면 다음과 같다.

① UNIX 플랫폼 : 서버측과 클라이언트측 모두 UNIX 플랫폼일 경우 ESCGIS가 JAVA로 GIS 컴포넌트가 엔터프라이즈 Enterprise JavaBeans로 되어 있으므로 서버측 컴포넌트는 플랫폼에 무관하다. 또한, ESCGIS는 다른 ESCGIS 서버와 분산 컴퓨팅 환경을 만들 수 있다. 클라이언트측 컴포넌트는 자바 애플리케이션이나 애플릿으로 제작 가능하고 CORBA를 이용하여 ESCGIS의 기능을 호출할 수 있다.

② 윈도우즈 플랫폼 : 서버측과 클라이언트측 모두 마이크로소프트일 경우는 기본적인 UNIX 플랫폼이 가지는 기능을 포함하면서 COM에 대한 기능을 추가적으로 가진다. 서버측의 경우 엔터프라이즈 자바빈즈를 COM을 이용하여 제작 가능하다. 즉, 이미 만들어 놓은 GIS용 ActiveX 컴포넌트들 중에서 서버측 컴포넌트의 기능에 해당하는 컴포넌트들을 JACTIVEX 기술을 사용하여 Enterprise JavaBeans로 변환을 하게 된다. 이와 같이 Enterprise JavaBeans 사양으로 만들어 놓은 COM 기반의 GIS 컴포넌트는 GIS 클라이언트 컴포넌트나 ESCGIS 인프라스트럭처하에 플랫폼 무관하게 호출되어 사용될 수 있다. 클라이언트측 컴포넌트의 경우 CORBA를 사용하여 ESCGIS 서버에 연결하는 COM 객체로 구현이 가능하다. 또한, JAVA BEANS 컴포넌트를 ActiveX로 wrapping 하는 기술을 사용하여 직접 윈도우즈 플랫폼하에서 ESCGIS 기능을 사용하는 윈도우즈 응용 프로그램을 개발할 수 있다.

이와 같이 ESCGIS 인프라스트럭처하에서 개발된 컴포넌트는 컴포넌트가 지향하는 재사용성과 상호운용성을 가지게 된다. ESCGIS는 GIS 컴포넌트 개발을 위한 산업 표준 아키텍처를 제시하며, GIS 컴포넌트 개발, 저장, 조립, 운영, 테스트 검증에 대한 방법론을 제시한다. GIS 컴포넌트를 이용하여 지리 정보 시스템을 구축하려는 GIS 개발자는 ESCGIS와 동일한 컴퓨터가 아니라 분산 환경인 Internet/Intranet 어디에 위치해도 무관하다. GIS 개발자는 ESCGIS에서 GIS 응용 프로그램 개발을 위한 다양한 모델링 및 디자인 패턴을 UML 형식으로 제공받을 수 있다. 그리고, 이미 개발된 GIS 컴포넌트들을 ESCGIS 컴포넌트 정보 관리 메커니즘에 의해 조회 및 선택할 수 있으며 이렇게 선택된 GIS 컴포넌트들을 가지고 GIS 응용 프로그램을 작성하기 위한 준비를 하게 된다. 준비된 GIS 컴포넌트들은 GIS 컨테이너에 적재되게 된다. 즉, GIS 컨테이너는 개발자가 필요로 하는 GIS 컴포넌트들을 동적으로 로딩(Loading)한다. 이때 GIS 컨테이너는 사용자 필요에 따라 개방형 GIS 포

준인 OpenGIS 컨테이너를 선택할 수 있으며, 또한 ESCGIS 에서 GIS 컴포넌트간의 메타데이터 호환을 위해 개발된 공간 메타데이터 컨테이너를 이용할 수 있다.

이들 컨테이너들은 자신에게 적재된 GIS 컴포넌트를 네이밍(Naming) 서비스에 등록하여 개발자들에게 원하는 컴포넌트 기능을 수행하도록 해준다. 이렇게 적재된 GIS 컴포넌트들에 대해 GIS 개발자는 GIS 컴포넌트 각각의 비즈니스 특성에 맞는 트랜잭션을 지정해 준다. 이에 따라 GIS 컴포넌트들의 트랜잭션은 수행되게 되며 또 다른 방법으로 개발자 자신이 직접 트랜잭션 수행에 대한 코드를 작성함으로써 트랜잭션 관리를 할 수 있다. ESCGIS는 접속된 GIS 개발자의 세션별 관리를 수행하며, GIS 개발자는 자신이 조립한 GIS 컨테이너를 동작시켜서 현재 자신이 개발한 GIS 응용 프로그램을 수행하도록 ESCGIS 테스트 관리자에게 테스트 등록을 요청한다. ESCGIS는 GIS 개발자가 힘들여 OpenGIS 서버나 공간 DBMS를 구해서 테스트해 봐야 하는 수고 없이 자신의 비즈니스 정책에 맞는 GIS 응용 프로그램을 ESCGIS 컴포넌트 인프라스트럭처에 의해 원격에서 구현 및 실행을 할 수 있으며, 직접 개발된 GIS 응용 프로그램을 평가할 수 있다.

3.2 ESCGIS 시스템 구조

ESCGIS는 컴포넌트 디자인 패턴 관리자, GIS 컴포넌트 관리자, TESTBED 서버 관리자, GIS 컨테이너 관리자, 공간 메타데이터 관리자의 5개 관리자로 나뉘어진다. 그림 2에서는 본 논문이 제시하는 GIS 컴포넌트를 위한 엔터프라이즈 서버인 ESCGIS의 전체 시스템 구조를 보여 준다.

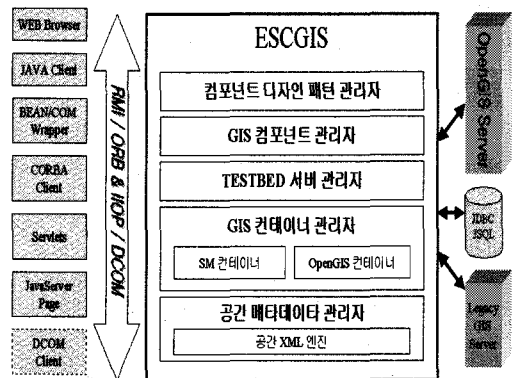


그림 2. ESCGIS 전체 시스템 구조

ESCGIS는 Enterprise JavaBeans Specification 1.1 사양에 맞추어 개발되어 관계형 데이터베이스 시스템과 JDBC로 연결되는 엔터프라이즈 빈즈 컴포넌트를 만들 수 있으며, 다양한 클라이언트측 컴포넌트에서 사용 가능하다. 다양한 클라이언트측 컴포넌트로는 순수 자바로 된 애플릿(Applet), JAVA 애플리케이션, Servlets, Java Server Page와 자바 빈즈를 ActiveX로 변환하는 BEAN/COM WAPPER, ActiveX를 자바 클래스로 변환시켜 주는 JACTIVEX, 플랫폼 무관한 ESCGIS 클라이언트 연결을 지원하는 CORBA Client 등이다. 클라이언트측 컴포넌트는 HTTP, Remote Method Invocation, Object Request Broker, Internet Inter Operate Protocol를 통해서 ESCGIS와 연결된다.

ESCGIS는 GIS 컨테이너를 통해 다양한 공간 데이터 서버와 연결되는데, 개방형 지리 정보 상호운용 사양인 OpenGIS Simple Feature Specification For CORBA(13)를 따르는 OpenGIS 컨테이너를 지원한다. OpenGIS 컨테이너에서는 OpenGIS에서 사용되는 다양한 공간 객체들의 타입을 지원하고 있으며, GIS 컴포넌트들이 GIS 응용 프로그램에 필요한 OGIS 클래스를 제공한다. OpenGIS 컨테이너를 사용하는 GIS 컴포넌트들은 OGIS 클래스를 이용하여 자신이 필요한 비즈니스 응용만을 고려하면 되고 ESCGIS가 내부적으로 OpenGIS 서버와의 초기화 및 연결에 대한 책임을 지고 있다. 또한, 여기에 Legacy GIS Server들의 연결도 가능하다. Legacy GIS Server는 기존에 존재하고 있는 다양한 GIS 서버를 지칭하는데 이들 GIS 서버들과는 GIS 서버들이 제공하는 GIS 데이터 접근 컴포넌트를 이용하여 연결을 하게 된다.

ESCGIS는 기본적으로 3-tier 구조에서 출발한다. 그리고, 클라이언트측 컴포넌트는 다시 ESCGIS의 GIS 컴포넌트일 수가 있으며, 분산 ESCGIS 서버간의 호출도 가능하다. ESCGIS에서는 GIS 컨테이너에 GIS 비즈니스를 위한 다양한 GIS 컴포넌트들이 배치되고, 여러 GIS 데이터 서버와의 연결이 가능한 N-tier 구조의 분산 지리 정보 시스템을 구축하게 해준다. ESCGIS를 구성하는 5개의 관리자들에 관한 상세한 설명은 다음과 같다.

3.2.1 GIS 컨테이너 관리자

GIS 컨테이너 관리자는 OpenGIS 컨테이너와 SM 컨테이너로 구성된다. OpenGIS 컨테이너는 OpenGIS를 사용하는 엔터프라이즈 빈즈를 위한 컨테이너이고, SM 컨테이너는 공간 메타데이터 관리자를 사용

하는 엔터프라이즈 빈즈를 위한 컨테이너이다. 엔터프라이즈 빈즈는 ESCGIS안에 적재되는 컴포넌트들을 지칭한다. 현재 ESCGIS에서의 GIS 컴포넌트들은 Enterprise JavaBeans Specification 1.1의 필수 사항인 세션빈즈(SessionBeans) 만을 지원하며 옵션 사항인 엔티티빈즈(EntityBean)는 지원 하지 않는다. GIS 컨테이너 관리자는 해당 컨테이너에 맞는 초기화를 수행하는데 각 컨테이너에 대한 설명은 다음과 같다.

OpenGIS 컨테이너는 개방형 GIS 표준인 OpenGIS Simple Feature Specification For CORBA를 따라 제작되었으며, OpenGIS 컨테이너의 OGIS 클래스는 OpenGIS의 CORBA 클라이언트측으로 임의의 공간 DBMS하의 OpenGIS 서버에 연결하여 데이터를 공유할 수 있게 해준다. 즉, OpenGIS 컨테이너는 자신에게 적재되는 GIS 컴포넌트들로 하여금 OGIS 클래스를 통하여 다양한 공간 DBMS의 공간 데이터를 이용하게 해준다. OpenGIS 컨테이너가 수행하는 역할을 보면 먼저 초기 데이터소스 바인딩시 OpenGIS 서버 접속을 위해 ORB를 초기화하며, OpenGIS Enterprise Bean 생성시 Container FeatureCollection 및 Feature Type을 바인딩한다. 그러나, GIS 컴포넌트 개발자들은 하부구조의 연결에 대한 고려 없이 OpenGIS 모델링에 의한 비즈니스 GIS 컴포넌트나 GIS 응용 프로그램을 제작하게 된다.

SM 컨테이너는 GIS 컴포넌트 개발자들에게 효과적인 GIS 데이터 소스를 제공해 준다. ESCGIS는 자체 공간 메타데이터 관리자를 가지고 있어 표준 공간 메타데이터에 대한 접근을 가능하게 하는데, ESCGIS 표준 공간 데이터 모델을 XML 형태(3)로 관리하고 있으며, 또한 자체 공간 XML을 파싱하는 공간 XML 엔진을 가지고 있다. 현재 ESCGIS의 공간 메타데이터는 SDTS(Spatial Data Transfer Standard) [11] 공간 데이터 모델을 공간 메타데이터로 사용하고 있다. SM 컨테이너는 OpenGIS 서버나 Legacy GIS가 제공하는 별도의 GIS 접근 컴포넌트 없이 GIS 컴포넌트 개발을 가능하게 한다.

3.2.2 컴포넌트 디자인 패턴 관리자

컴포넌트 디자인 패턴 관리자는 ESCGIS GIS 응용 컴포넌트 및 응용 프로그램 개발을 위한 프레임워크를 제공한다. 이 프레임워크는 하나의 프로젝트로 관리되는데, 컴포넌트 디자인 패턴 관리자는 GIS 컴포넌트 모델링 학습, GIS 컴포넌트 템플릿 제공, 그리고 GIS 디자인 패턴 DB 학습 등을 제공한다. 개발자는

맨 처음 컴포넌트 디자인 패턴 관리자에 접근하여 자신이 필요한 컴포넌트 개발 패턴 모델링을 찾아 볼 수 있고, GIS 컴포넌트 템플릿을 통해 기본적인 GIS 컴포넌트 및 응용 프로그램 개발의 프레임워크를 가지게 된다. 또한, 사용자가 개발한 ESCGIS 응용 컴포넌트들은 다른 사용자를 위한 하나의 패턴으로 데이터베이스에 저장된다.

GIS 컴포넌트 모델링 학습은 UML의 Use Case View, Component View, Logical View로 기본 GIS 시스템의 GIS 디자인 패턴을 저장하게 되며, 사용자로 하여금 자신이 개발하려는 비즈니스에 적합한 패턴을 검색하여 비교 학습을 할 수 있도록 해준다. UML의 Use Case View를 통해 GIS 개발자는 GIS 컴포넌트의 쓰임새를 확인할 수 있으며, Component View를 통해 자신이 사용하려는 GIS 컴포넌트들의 의존 관계를 확인할 수 있다. 이 의존 관계는 GIS 컨테이너 적재시 관련된 GIS 컴포넌트들이 함께 적재되어야 하는 의존관계를 나타낸다. 그리고, Logical View로서 GIS 컴포넌트의 사용방안을 확인할 수 있다. 자신이 만든 컴포넌트들을 ESCGIS 디자인 패턴 DB에 학습시키고자 하는 개발자들은 UML에 입각한 Use Case View, Component View, Logical View를 반드시 제공해야 된다. GIS 개발자들은 GIS 컴포넌트 템플릿을 이용하여 기본적인 GIS 컴포넌트 개발이나 GIS 응용 프로그램 개발에 필요한 기본 프레임워크를 얻는다. GIS 컴포넌트 템플릿은 개발자로 하여금 비즈니스 룰을 선택하게 하는데, 비즈니스 룰에는 클라이언트측 컴포넌트 개발과 서버측 컴포넌트 개발이 있으며, 각각의 GIS 사례 구축에 대한 초기 환경을 구축하게 된다. 또한, 해당 사례에 대한 기본 예제 코드를 GIS 개발자에게 생성해 준다.

3.2.3 GIS 컴포넌트 관리자

GIS 컴포넌트 관리자는 이미 등록된 컴포넌트의 인터페이스를 관리하며 클래스에 대한 도큐먼트를 제공하는데, 클래스에 대한 도큐먼트는 javadoc의 형태로 존재하게 된다. GIS 컴포넌트 관리자는 컴포넌트 바이너리를 가지고 있으며 개발자들에게 컴포넌트의 공급, 조회, 등록, 삭제를 가능하게 한다. 또한, 이와 같은 기본적인 GIS 컴포넌트 관리 외에 사용자 세션 GIS 컴포넌트 관리를 하고 있다.

사용자 세션 GIS 컴포넌트 관리는 GIS 개발자가 원격으로 자신이 개발한 GIS 컴포넌트를 테스트하고 싶을 때 개발한 GIS 컴포넌트의 동작을 위해서 반드시 필요한 클래스를 ESCGIS의 GIS 컴포넌트 관리자에 등록없이 사용 가능하게 한다. 이를 위해 GIS 컴

포넌트 관리자는 사용자가 원격에서 작성한 코드를 사용자 세션으로 관리하여 TESTBED 서버에서 실행 가능하도록 하고 있다. 이러한 세션 관리는 사용자의 해킹의 여지가 있으므로 별도의 권한과 인증된 자격을 가진 사용자만이 가능하게 해야 한다. ESCGIS에서는 사용자 세션과 같은 자유로운 컴포넌트 수행시 반드시 필요한 중요 데이터 전송이나 지속적 임시 데이터의 생성이 필요할 때 공간 메타데이터 관리자를 이용하도록 하고 있다. 즉, 직접적인 파일 생성은 불가능하며, 공간 메타데이터 관리자 함수를 이용하여 중요 데이터 교환이나 임시 보관 자료를 생성하게 된다.

3.2.4 TESTBED 서버 관리자

지리 정보 시스템은 전문화된 분야로서 일반인이 GIS DBMS를 사용하여 GIS 프로그래밍을 하기가 어려우며, 분산 컴퓨팅 환경하의 지리 정보 시스템을 설계 및 개발하는 일은 더욱 힘들다. 이러한 GIS 산업의 단된 구조에서 ESCGIS는 개방된 인프라스트럭처를 제공한다. ESCGIS는 분산 컴포넌트 GIS를 지향하는 N-tier 구조로 설계되어 있으므로 GIS 컴포넌트를 개발하려는 개발자는 공간 DBMS나 Open GIS 서버의 설치나 접근에 대해 고민할 필요가 없이 단지 자신이 원하는 GIS 비즈니스를 구현하는데 전념할 수 있게 된다.

TESTBED 서버 관리자는 ESCGIS가 가지는 N-tier 특성을 사용하여 자기 다른 GIS 응용 프로그램을 개발하려는 GIS 개발자를 위해 TESTBED 서버를 동작시켜 준다. TESTBED 서버 관리자는 GIS 컨테이너 관리자로 구성된 가벼운 ESCGIS TESTBED 서버를 관리한다. GIS 개발자는 자신이 개발 및 조립한 GIS 컴포넌트들을 테스트하는 용도로 TESTBED 서버를 사용하는데, TESTBED 서버는 클라이언트 연결 방법에 있어서 ESCGIS 서버와 일치하지만 보안 및 공간 데이터베이스 작업에 대한 제약을 갖고 있다.

3.2.5 공간 메타데이터 관리자

공간 메타데이터 관리자는 공간 데이터를 공간 XML 엔진을 이용하여 관리한다. 공간 메타데이터 관리는 공간 데이터 뿐만 아니라 GIS 컴포넌트 상호간의 중요 데이터 공유에 필요한 메타데이터를 생성할 수 있으며, GIS 컴포넌트가 동작 중에 필요한 지속적 임시 데이터를 생성하게 해준다. 공간 메타데이터 관리자는 공간 XML을 이용하여 SM 컨테이너로 하여금 공간 데이터 접근을 가능하게 해주는 데이터소스 역할을 한다. ESCGIS에서는 공간 메타데이터로서 공간 데이터 표준 교환 포맷인 SDTS를 사용하는데,

SDTS를 XML 형태로 변환한 후 공간 XML 엔진을 이용하여 공간 메타데이터 관리자의 요구에 적합한 작업을 수행한다.

공간 XML 엔진은 ESCGIS 공간 메타데이터인 SDTS XML에 대한 삽입, 수정, 삭제, 검색 등의 기능을 수행한다. 공간 XML 엔진은 SDTS 파일에 대한 수입/수출 기능을 가지고 있어 GIS 산업에 널리 유통되는 SDTS 파일을 이용하여 효과적인 GIS 컴포넌트 개발을 가능하게 한다. 또한, 공간 XML 엔진은 공간 데이터 뿐만 아니라 공간 메타데이터 관리자에 의해 요청된 새로운 메타데이터를 생성 및 조작하는 기능을 제공한다. 이 기능은 ESCGIS 각 관리자간이나 GIS 컴포넌트간의 중요 데이터 교환에 사용되며, 또한 GIS 컴포넌트 운영에 필요한 임시 데이터 조작에 사용된다.

4. ESCGIS 구현

ESCGIS는 Object Web Group(OWG)에서 제공되고 있는 Java Open Application Server (JOnAS) 위에 구현되었는데, OWG는 방대한 규모의 정보 네트워크를 위한 분산 소프트웨어 인프라스트럭처의 개발을 주 목적으로 하는 개방형 소스 단체 (Open Source Community)이다[16]. ESCGIS는 기본적으로 Enterprise JavaBeans Specification 1.1을 따르는 EJB 서버에 GIS 개념을 도입하여 GIS 컨테이너 관리, GIS 컴포넌트 디자인 패턴 관리, GIS 컴포넌트 관리, TESTBED 서버 관리, 공간 메타데이터 관리 등의 대한 전반적인 GIS 컴포넌트 개발 인프라스트럭처를 제공하고 있다.

4.1 GIS 컨테이너 관리자의 구현

ESCGIS에서는 현재 GIS 컨테이너로써 OpenGIS 컨테이너와 SM 컨테이너를 제공하고, 이를 통해 각각의 데이터 소스에 연결하여 지리 데이터를 조작할 수 있다. GIS 컨테이너는 다른 GIS 서버 특성에 맞게 새롭게 추가 및 삭제될 수 있으며, 이를 위해서는 GIS 컨테이너 관리자에게 등록 및 해제 신청을 해야 된다. ESCGIS 인프라스트럭처를 사용하여 GIS 컴포넌트를 개발하는 이들은 이러한 등록된 내용을 통해서 사용할 수 있는 GIS 컨테이너가 무엇인지를 확인할 수 있다.

4.1.1 OpenGIS 컨테이너

OpenGIS 컨테이너는 개방형 GIS 표준인

OpenGIS를 이용하여 작성된 엔터프라이즈 빈즈를 적재하고 기능을 수행하기 위해 사전 작업을 수행하는 GIS 컨테이너이다. ESCGIS에서 제공하는 OpenGIS 컨테이너는 OpenGIS Simple Feature Specification For CORBA 사양을 따르는 IDL을 이용하여 자바 OGIS 클래스를 생성하였으며, GIS 컴포넌트들은 이 OGIS 클래스를 이용하여 자유롭게 OpenGIS의 Geometry 및 Feature를 처리할 수 있다. OpenGIS 컨테이너는 데이터 소스 바인딩시 OpenGIS 접속을 위해 데이터 소스 프로퍼티를 읽어 오고, GIS 개발자들은 자신의 OpenGIS 서버에 대한 설정을 데이터 소스 프로퍼티에 세팅할 수 있다. ESCGIS는 TESTBED 서버를 위해 OpenGIS 서버를 제공하고 있는데, 현재 두개의 공간 DBMS에 대한 OpenGIS 서버를 지원하고 있다. 하나는 OpenOODB상의 OpenGIS 서버이고 또 하나는 한국 통신의 GEUS 공간 DBMS상의 OpenGIS 서버이다. ESCGIS의 OpenGIS 컨테이너를 이용하면 현재 개발되고 있는 OpenGIS 서버에 대한 검증기 역할도 가능하게 된다. ESCGIS가 동작하여 GIS 컴포넌트들을 사용하기 까지 OpenGIS 컨테이너의 내부 동작 흐름은 다음과 같다.

① ESCGIS 서버가 실행된다.

② 데이터소스 Deployment Descriptor에서 OpenGIS 컨테이너의 OpenGIS 서버의 주소 등의 속성을 읽는다.

③ OpenGIS Enterprise Beans Deployment Descriptor에서 OpenGIS 엔터프라이즈 빈즈를 OpenGIS 컨테이너에 적재한 후 생성자(Constructor)를 실행한다.

④ 적재된 OpenGIS Enterprise Beans들에 대한 Home 인터페이스들을 네이밍 서비스에 등록한다.

⑤ OpenGIS 서버 접속을 위해 ORB를 초기화한다.

⑥ ContainerFeatureCollection 및 Feature Type을 바인딩 한다.

⑦ 클라이언트로부터 네이밍 서비스 호출에 의해 Home.create() 인터페이스가 Invoke된다.

⑧ 엔터프라이즈 빈즈에서 OpenGIS를 사용한다.

4.1.2 SM 컨테이너

SM 컨테이너의 동작은 OpenGIS 컨테이너와 유사하다. SM 컨테이너는 데이터 소스를 ESCGIS 내부의 공간 메타데이터 관리자를 통해 얻을 수 있기 때문에 별도의 GIS 서버의 연결없이 동작하게 된다. GIS 컨테이너를 사용시 다폴트는 현재 OpenGIS 컨테이너

로 되어 있다. 그 이유는 OpenGIS가 개방형 표준 지리 정보 모델을 제시하고 있기 때문이다. 하지만 현재 공간 데이터를 메타데이터 표준인 XML로의 연구가 활발히 진행되고 있으며 조만간 표준 공간 메타데이터의 DTD와 DOM이 나오게 될 것이다. 현재 ESCGIS에서는 국가 교환 표준 포맷으로 채택된 SDTS 포맷 기반의 공간 메타데이터를 관리하고 있으며, 이러한 공간 메타데이터를 사용하는 GIS 컴포넌트들의 조립 및 운영을 SM 컨테이너가 관리하고 있다. SM 컨테이너의 동작 흐름은 다음과 같다.

- ① ESCGIS 서버가 실행한다.
- ② 데이터소스 Deployment Descriptor에서 SM 컨테이너의 속성을 읽는다.
- ③ OpenGIS Enterprise Bean Deployment Descriptor 에서 SM 엔터프라이즈 빈즈를 SM 컨테이너에 적재한 후 생성자를 실행한다.
- ④ 적재된 SM Enterprise Bean들에 대한 Home 인터페이스들을 네이밍 서비스에 등록한다.
- ⑤ 공간 메타데이터 관리자를 통해 공간 메타데이터를 사용하기 위한 초기 작업을 수행한다.
- ⑥ 클라이언트로부터 네이밍 서비스 호출에 의해 Home.create() 인터페이스가 Invoke된다.
- ⑦ SM Enterprise Bean을 사용한다.

4.1.3 GIS 컴포넌트 트랜잭션 정책

GIS 컨테이너는 GIS 비즈니스를 컴포넌트들의 조립으로 수행하기 위해 반드시 GIS 컴포넌트간의 트랜잭션 처리에 대해 관여해야 한다. ESCGIS에서는 GIS 컨테이너의 GIS 컴포넌트 트랜잭션 정책을 위해 Enterprise JavaBeans Specification 1.1의 Java Transaction Manager를 이용하여 GIS 컴포넌트 트랜잭션을 구현하고 있다. ESCGIS에서 GIS 컴포넌트 트랜잭션을 사용하기 위해서는 자신이 원하는 트랜잭션 스코프(scope)에 javax.transaction.UserTransaction 클래스의 begin(), commit(), rollback()을 사용하여야 된다. ESCGIS는 현재 Nested Transaction을 지원하지는 않기 때문에 사용자가 트랜잭션 안에 다시 트랜잭션을 사용할 때 기존의 트랜잭션을 중지하고 새로운 트랜잭션을 시작하든지, 아니면 기존의 트랜잭션과 같은 스코프의 트랜잭션을 그대로 사용하여야 된다. Java Transaction Manager는 ESCGIS에게 분산 컴포넌트의 내부(internal)/외부(external) 트랜잭션을 가능하게 한다. 내부 트랜잭션은 한 컨테이너내에서 컴포넌트들간의 트랜잭션을 의미하며, 외부 트랜잭션은 다른 컨테

이너나 다른 ESCGIS 서버 컴포넌트간의 트랜잭션을 의미한다.

GIS 개발자들은 자신이 만든 컴포넌트가 트랜잭션 하에서 커밋(commit)과 롤백(rollback)에 대한 관리를 직접 GIS 컴포넌트에 적용할 수 있는데, 이와 같은 트랜잭션 처리는 EJB Specification 1.1에 의한 세션빈즈 Method인 void beforeCompletion(), void after Begin(), void afterCompletion (boolean committed)의 세 개의 함수를 이용하여 수행된다. 또한, 별도의 코딩없이 GIS 컨테이너 각각 트랜잭션에 대한 처리가 가능한데, 롤백은 EJB 객체의 ejbLoad()를 호출하여 원래 생성 당시의 상황으로 컴포넌트를 초기화하고, 커밋은 각각의 컨테이너에 따라 필요한 커밋 기능을 수행한다. OpenGIS 컨테이너 커밋의 경우

insert_element_at()을 호출하며, SM 컨테이너 커밋의 경우 sm_Flush()를 이용한다.

4.2 GIS 컴포넌트 디자인 패턴 관리자의 구현

GIS 컴포넌트 디자인 패턴 관리자는 GIS 컴포넌트 개발자들에게 아주 유용한 정보를 제공한다. 이 정보는 UML로 기술되어 있는데 개발자들은 자신에게 필요한 GIS 컴포넌트 및 비즈니스 응용 프로그램에 대한 자료를 GIS 컴포넌트 디자인 패턴 관리자에서 검색하여 얻을 수 있다. UML은 컴포넌트 모델링을 위한 표준 도구로서 현재 많은 GIS 컴포넌트 개발에 사용되고 있다. UML은 많은 notation으로 표현되고 복잡한 객체지향 설계를 고려하여 제작되었지만, ESCGIS에서는 GIS 컴포넌트 제작에 가장 효율성있는 Use Case View, Component View, Logical View 만을 필요로 한다. 이는 GIS 컴포넌트 공급자가 자신의 GIS 컴포넌트에 대한 설명과 사용법을 UML로 제공해야 한다는 것을 의미한다. 그림 3은 ESCGIS에 대한 Use Case View를 UML을 이용하여 보여주고 있다.

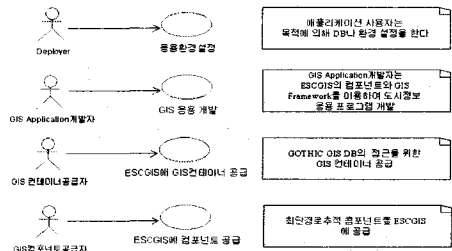


그림 3. ESCGIS의 Use Case View

그림 3을 통해서 보면 ESCGIS를 어떤 사용자들이 접근하는지를 쉽게 알 수 있는데 응용 프로그램 환경을 설정하는 Deployer, GIS 응용 프로그램을 개발하는 GIS 애플리케이션 개발자, ESCGIS에 GIS 컨테이너를 공급하는 GIS 컨테이너 공급자, ESCGIS에 GIS 컴포넌트를 공급하는 GIS 컴포넌트 공급자를 알 수 있다. 그림 4는 ESCGIS의 동작을 설명한 Logical View의 Interaction Diagram을 보여주고 있다. 이러한 Interaction Diagram을 통해 ESCGIS 사용자는 어떤 순서로 자신이 원하는 작업을 수행해야 되는지의 절차를 확인할 수 있다.

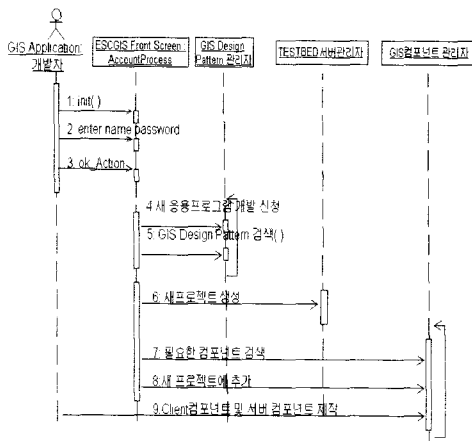


그림 4. ESCGIS의 Interaction Diagram

GIS 컴포넌트 디자인 패턴 관리자는 GIS 컴포넌트 개발자들에게 컴포넌트 개발에 대한 모델링을 강조하고 있다. 이와 같은 GIS 컴포넌트 개발 모델링은 GIS 컴포넌트 개발자들에게 반드시 필요한 컴포넌트 개발 아키텍처 중의 하나이다.

4.3 GIS 컴포넌트 관리자의 구현

GIS 컴포넌트 관리자는 ESCGIS를 위해 저장된 GIS 컴포넌트 리스트들을 관리한다. GIS 컴포넌트 관리자는 GIS 컴포넌트의 기능에 대한 도움말을 JAVA DOC의 형식으로 보여주며, 또한 컴포넌트들을 선택하여 GIS 컨테이너에 적재한다. GIS 컴포넌트 관리자에서 선택된 GIS 컴포넌트들 중 의존성 관계가 있는 컴포넌트들이 같이 선택되며, 개발자들은 선택된 GIS 컴포넌트들을 GIS 컨테이너에 적재하여 사용하게 된다. 이때 GIS 컨테이너의 종류에 따라 GIS 데이터 접근 컴포넌트들의 사용이 제한될 수 있다. GIS 데이터 접근 컴포넌트는 서버측 컴포넌트로

서 다양한 GIS 벤더들이 자신의 공간 DBMS에 접근하게 해주는 컴포넌트를 의미하는데, GIS 컨트롤 컴포넌트에게 다양한 데이터 소스에 대한 투명성을 제공한다. GIS 데이터 접근 컴포넌트는 GIS 컨테이너에 밀접한 관련을 가지고 있기 때문에 서로 다른 GIS 컨테이너에서는 다른 GIS 데이터 접근 컨테이너가 동작하지 않는다. 그러나, 이 부분은 ESCGIS가 여러개의 컨테이너를 동시에 동작할 수 있고, 이들 컨테이너간의 컴포넌트들은 서로간의 네이밍(Naming) 호출에 의해 불러질 수 있으며, 또한 같은 트랜잭션에서 동작이 가능하기 때문에 개발자는 자유롭게 필요한 비즈니스를 구축하게 된다.

4.4 TESTBED 서버 관리자의 구현

소프트웨어 개발에 있어서 TESTBED에 관한 부분이 많이 거론되고 있다. TESTBED는 사용자로 하여금 표준화 방안을 따르게 하는 아주 좋은 역할을 하는데, ESCGIS에서는 사용자가 개발한 GIS 컴포넌트를 TEST하게 해준다. ESCGIS를 통해서 개발된 컴포넌트가 TESTBED 서버에서 잘 동작하는 것은 곧 ESCGIS GIS 컴포넌트 아키텍처로 구현을 잘하였다는 것을 의미한다. TESTBED 서버는 컴포넌트 재사용에 관한 훌륭한 검증 방안을 제공하며, GIS 개발자에게 이미 개발된 ESCGIS 기반의 GIS 응용 프로그램에 대해 잘 모델링된 UML을 제공하며, 설계된 컴포넌트들이 실제 어떻게 동작하는지를 확인할 수 있고, 자신의 비즈니스에 어떻게 적용할 수 있는지를 판단할 수 있게 해 준다. 그리고, GIS 컴포넌트 개발자들은 이미 동작하는 GIS 컨테이너에서 기존의 GIS 컴포넌트를 제거하고 자신이 만든 GIS 컴포넌트를 추가한 후 자신의 GIS 컴포넌트의 성능이나 상호운용성을 검증할 수 있다. 이와 같은 작업은 컴포넌트 기술 중 플러그 앤 플레이를 통해 가능하게 된다.

TESTBED 서버 관리자는 사용자가 요청한 GIS 컨테이너 요청에 의해 TESTBED 서버를 동작시키며, 여러 정책으로 TESTBED 서버를 수행할 수 있다. 하나의 GIS 개발자에게 하나의 TESTBED 서버를 할당할 수 있으며, 여러 개발자들에게 하나의 TESTBED 서버를 할당할 수 있다. TESTBED 서버는 사용자가 조립한 GIS 컨테이너로 이루어진 가벼운 ESCGIS 서버로써 동적으로 GIS 컨테이너에 엔터프라이즈 빈즈를 적재할 수 있다. TESTBED 서버 관리자는 공동의 프로젝트를 관리하는 GIS 개발자들에게 하나의 TESTBED 서버를 동작하게 함으로써 GIS 응용 프로그래밍의 공동 작업을 가능하게 해준다.

이후 연구 방향으로는 Enterprise JavaBeans Specification 1.1의 옵션 사항인 EntityBeans를 지원하는 OpenGIS 컨테이너와 SM 컨테이너의 구현에 대한 연구가 남아있다. 현재 ESCGIS는 필수 사항인 SessionBeans만을 구현하고 있는데, EntityBeans의 구현은 본 논문에서 구현한 OpenGIS 컨테이너의 트랜잭션 기능 및 데이터 조작 기능과 긴밀한 연관이 있게 된다.

참 고 문 헌

- [1] Asbury, S. and Weiner, S. R., Developing Java Enterprise Applications, Wiley, 1999.
- [2] Bezivin, J., and Muller, P. A., "UML: The Birth and Rise of a Standard Modeling Notation," <<UML>>'98, Beyond the Notation, Vol.1618, No.1, 1999, pp.1-8.
- [3] Bray, T., Paoli, J., and Sperberg-McQueen, C. M., Extensible Markup Language (XML) 1.0, REC-xml-1 9980210, <http://www.w3.org/TR/1998/REC-xml-1 9980210>, 1998.
- [4] Chappell, D., Understanding Activex and OLE, Microsoft Press, 1996.
- [5] Gamma, E., Helm, R., Johnson, R., and Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley, 1994.
- [6] Larsen, G., "Designing Component-Based Frameworks Using Patterns in the UML," Communications of the ACM, 1999, p38-45.
- [7] Grimes, R., Professional DCOM Programming, Wrox, 1997.
- [8] Jacobson, I., Griss, M., and Jonsson, P., Software Reuse, Addison Wesley, 1997.
- [9] Larman, C., Applying UML and Patterns, Prentice Hall, 1998.
- [10] Lazar, R., The SDTS Topological Vector Profile, Cartography and Geographic Information System, Vol.19, No.5, 1992, pp. 262-299.
- [11] National Institute of Standards and Technology, The Spatial Data Transfer Standard, Federal Information Processing Standard Publication 173, 1992.
- [12] Object Management Group, CORBA, The Common Object Request Broker: Architecture and Specification, Object Management Group, Inc., 1995.
- [13] OpenGIS Consortium, OpenGIS Simple Features Specification For CORBA Revision 1.0, OpenGIS Consortium, Inc., 1998.
- [14] OpenGIS Consortium, The OpenGIS Abstract Specification Model, Open GIS Consortium, Inc., 1998.
- [15] Open GIS Consortium, The OpenGIS Guide, Open GIS Consortium, Inc., 1999.
- [16] OWG, A Distributed Software Infrastructure for Large Scale Information Networks, Open Source Community, http://www.objectweb.org/website/objectwebbpe/doc/Initiative_en/ppframe.htm, 1999.
- [17] Roman, E., Mastering Enterprise Java Beans and the Java2 Platform Enterprise Edition, WILEY, 1999.
- [18] Rumbaugh, J., Jacobson, I., and Booch, G., The Unified Modeling Language Reference Manual, Addison Wesley, 1998.
- [19] Rumbaugh, J., Jacobson, I., and Booch, G., The Unified Modeling Language: User Guide, Addison Wesley, 1998.
- [20] Sun Microsystems, Enterprise Java Beans Specification V1.1, Sun Microsystems, Inc., 1999.
- [21] Sun Microsystems, Enterprise Java Beans to CORBA Mapping V1.1, Sun Microsystems, Inc., 1999.
- [22] Sun Microsystems, Enterprise Java Beans Technology : Server Component Model for the Java Platform, Sun Microsystems, Inc., http://java.sun.com/products/ejb/white_paper.html, 1999.
- [23] 권오천, 신규상, 오영배, "컴포넌트 기반의 개발 환경 모델: 객체지향과 타 기술들의 통합적인 접근 방법," 한국정보처리학회 소프트웨어공학 연구회지, 1권3호, 1999.
- [24] 김동현 외, "UML을 이용한 자바빈즈 컴포넌트

설계," 산·학·연 소프트웨어공학기술 학술대회 논문집, 1999, p263-268.

[25] 이강준, 오병우, 한기준, "SDTS와 Informap간의 데이터 변환 시스템의 설계 및 구현," 한국지형공간정보과학회 논문지, 4권2호, 1996, pp. 109-121.

[26] 이강준, 김준중, 설영민, 한기준, "SDTS 변환 시스템," 한국개방형GIS연구회, '98 개방형 GIS 학술회의 논문집, 1권1호, 1998, pp. 183-195.

[27] 정진완, 강홍근, 김형주, 박주홍, 한기준, 허신, "국가 지리 정보 시스템을 위한 공간 객체 관리

시스템의 구조," 한국정보과학회 정보과학회지, 16권3호, 19983, pp.5-9.

[28] 한국전산원, 국가지리정보체계(NGIS) 정보유통을 위한 정보기록방식(메타데이터) 표준화 연구, 한국전산원, 1997.

[29] 한국전자통신연구원, '99개방형 GIS 컴포넌트 S/W 개발 워크샵 논문집, 한국전자통신연구원, 1999.

[30] 한국정보산업연합회, Proceedings of Software Technology Conference Korea '99, 한국정보산업연합회, 1999.



이강준
 1995년 건국대학교 컴퓨터공학과 졸업(공학사)
 1997년 건국대학교 대학원 컴퓨터공학과 졸업(공학석사)
 1997년~현재 건국대학교 대학원 컴퓨터공학과 박사과정

관심분야: 지리정보시스템, 컴포넌트 GIS, 객체 관계형 데이터베이스, 공간 데이터 마이닝, 실시간 데이터베이스



박지웅
 1992년 배재대학교 전자계산학과 졸업(이학사)
 1994년 건국대학교 대학원 컴퓨터공학과 졸업(공학석사)
 1996년~현재 건국대학교 대학원 컴퓨터공학과 박사과정

1999년~현재 동양정보넷 대표이사
 관심분야: 지리정보시스템, 디지털 라이브러리, 가상교육



홍동숙
 1999년 건국대학교 컴퓨터공학과 졸업(공학사)
 1999년~현재 건국대학교 대학원 컴퓨터공학과 석사과정

관심분야: 개방형지리정보시스템, 컴포넌트 GIS, 인터넷 데이

터베이스



한기준
 1979년 서울대학교 수학교육학과 졸업(이학사)
 1981년 한국과학기술원 전산학과 졸업(공학석사)
 1985년 한국과학기술원 전산학과 졸업(공학박사)

1990년 Stanford 대학 전산학과 visiting scholar
 1985년~현재 건국대학교 컴퓨터공학과 교수
 관심분야: 지리정보시스템, 객체 지향 데이터베이스, 공간 데이터 마이닝, 주기억-상주 데이터베이스