

UML을 활용한 컴포넌트 기반의 GIS 개발방법론에 관한 연구

A Study on the Component-based GIS Development Methodology using UML

박태옥(朴泰玉) · * 김계현(金桂炫)**

Taeog Park, Kye Hyun Kim

초 록 오늘날 GIS 영역을 포함하는 정보시스템 개발 환경은 소프트웨어의 복잡성 및 다양성 그리고 분산처리 및 네트워크 컴퓨팅 등의 측면에서 과거에 비해 현저하게 변화되었다. 이에 신속하게 대응하기 위하여 소프트웨어 개발 패러다임에 변화가 일어나고 있으며, 객체지향기술에 바탕을 둔 컴포넌트 기반 개발이 대세로 자리잡고 있다. GIS 개발에서도 국내외적으로 관련 표준 지침을 만들어 컴포넌트에 기반한 개발을 독려하고, 앞으로 컴포넌트 기술의 적용이 증가하는 추세이다. 이러한 추세에 부응하여 GIS를 위한 컴포넌트 기반 개발 방법론의 필요성이 대두되나 아직 연구가 충분히 이루어지지 못하는 실정이다.

본 연구는 UML을 활용한 컴포넌트 기반의 GIS 개발 방법론(ATOM: Advanced Technology Of Methodology for GIS)의 프로세스의 제안과 함께 사례연구를 통하여 이의 적용가능성을 확인하는 것이다. ATOM은 컴포넌트 개발 그 자체를 지원할 뿐만 아니라 식별된 컴포넌트와 기존 재사용 가능한 컴포넌트에 바탕을 둔 소프트웨어 개발 생명주기 전체 단계를 지원하는 전사적인 GIS 구축 방법론이다. ATOM은 생명주기 각 단계에 대한 주요 활동과 각각의 활동을 수행하기 위한 작업들을 정의하였다. 또한, 작업간 입력물과 출력물을 제시하고, 각종 문서화를 위한 표준 양식과 항목을 제시하며, 작업들의 성공적 수행을 위한 상세한 지침을 제시하여 최대한 방법론의 이해를 돕고자 하였다. 무엇보다도 ATOM의 가장 중요한 특징은 단순한 기능과 최소의 크기, 최대의 재사용을 위한 컴포넌트 추출에 목적을 두고 GIS 도메인의 여러 특징을 고려한 GIS를 위한 컴포넌트 기반의 개발방법론이라 할 수 있다.

ATOM의 사례 적용은 재사용 및 상호운용성이 뛰어난 컴포넌트의 추출과 함께 보다 체계적이고 구체적인 개발 가이드 라인을 제공하여 응용GIS 구축의 생산성 및 품질 향상에 기여할 뿐만 아니라 우리의 최종 목표인 GIS 소프트웨어 자동 생산에도 크게 기여할 것으로 사료된다.

ABSTRACT The environment to development information system including a GIS has been drastically changed in recent years in the perspectives of the complexity and diversity of the software, and the distributed processing and network computing, etc. This leads the paradigm of the software development to the CBD(Component Based Development) based object-oriented technology.

As an effort to support these movements, OGC has released the abstract and implementation standards to enable approaching to the service for heterogeneous geographic information processing. It is also common trend in domestic field to develop the GIS application based on the component technology for municipal governments. Therefore, it is imperative to adopt the component technology considering current movements, yet related research works have not been made.

This research is to propose a component-based GIS development methodology—ATOM(Advanced Technology Of Methodology)—and to verify its adoptability through the case study. ATOM can be used as a methodology to develop component itself and enterprise GIS supporting the whole procedure for the software development life cycle based on conventional reusable component. ATOM defines stepwise development process comprising activities and work units of each process. Also, it provides input and output, standardized items and specs for the documentation, detailed instructions for the easy understanding of the development methodology. The major characteristics of ATOM would be the component-based development methodology considering numerous features of the GIS domain to generate a component with a simple function, the smallest size, and the maximum reusability.

The case study to validate the adoptability of the ATOM showed that it proves to be a efficient tool for generating a component providing relatively systematic and detailed guidelines for the component development. Therefore, ATOM would lead to the promotion of the quality and the productivity for developing application GIS software and eventually contribute to the automatic production of the GIS software, the our final goal.

키워드 : 컴포넌트, 개발방법론, GIS, CBD, UML, 소프트웨어 개발, 컴포넌트 추출

* 한국정보통신대학교 정보통신교육원 팀장

topark@aiit.or.kr

** 인하대학교 지리정보공학과 부교수

kyehyun@inha.ac.kr

1. 서론

1.1 연구의 배경

지난 수십 년간 지리정보 소프트웨어에 대한 요구는 복잡 다양하게 증가하고 있으나 개발 기술은 이에 대처하지 못하고 있는 실정이며, 과거 단일 플랫폼에서 중앙집중식 형태로 개발되고 운영되던 응용프로그램들이 다양하고 복잡한 개방형 환경 어디에서나 실행 가능한 형태로의 개발을 요구하고 있다. 이것은 다중계층(multi-tier)의 개발 구조와 정보 공유 및 재사용을 요구하며, 또한 인터넷 등을 통하여 분산되어 있는 이질적인 통신 프로토콜, 하드웨어, 소프트웨어들의 통합을 요구한다. 전통적인 개발 방법으로는 이러한 요구를 만족시킬 수 없다는 인식아래 컴포넌트 기반 개발방법론의 도입을 강력히 요청하고 있다[1].

CBD에 의한 소프트웨어 개발은 CBD 하부구조를 이루고 있는 분산객체 기술 개발과 개별 소프트웨어 영역(GIS, ERP, EC 등)에 필요한 컴포넌트 개발로 구성된다고 할 수 있다[2]. 분산 객체기술 부분은 OMG의 CORBA, OSF의 DCE, 마이크로소프트의 OLE/DCOM 및 Sunsoft의 Java가 분산처리 컴퓨팅의 표준 플랫폼이 되어 다중계층의 미들웨어 역할을 수행하고 있다. 소프트웨어 영역에서 GIS 분야는 OGC에서 공간 데이터와 공간 정보 처리의 재사용성과 상호운용성에 대하여 많은 노력을 하고 있다. Open GIS Data Model의 단순 개체(simple feature)에 대한 개발 사양을 1998년에 발표한 이후 GIS 구축 분야에 많은 영향을 주고 있다[3].

컴포넌트 기반 GIS 개발 방법론은 특정 업무 영역에 대한 여러 종류의 컴포넌트를 식별 및 정의하여 전사적 GIS를 구축하는 일련의 프로세스의 집합이다. 이들 컴포넌트의 개발을 위하여 전체 시스템에 대한 명세를 작성하고, 시각적으로 표현하며 문서화 할 수 있는 표준 모델링 도구인 UML을 활용한다. 따라서 객체기반 기술을 바탕으로 UML 기반 개발 방법론이 표준으로 자리잡고 있으며, 이 방법론의 적용은 재사용이 가능한 컴포넌트의 개발과 전사적 GIS 구축이 가능하다. 그러나 현재 일관성 있고 표준적인 GIS를 위한 UML 기반 컴포넌트 개발 방법론(컴포넌트 개발이 목적) 및 컴포넌트 기반 GIS 개발 방법론(컴포넌트 개발 및 전사적 GIS 구축이 목적)이 부재하므로 새로운 개발 방법론의 수립이 필수적이라는 전제하에 본 연구에서는 특히 국내의 실정에 적합한 컴포넌트 기반의 개발방법론을 제시하였다.

1.2 연구 목표

본 연구의 목표는 UML을 활용한 컴포넌트 기반 GIS 개발 방법론을 제안하는 것이다. 제안된 방법론은 각 단계의 진행에 따라 주요 활동 및 산출물 도출의 반복적이고 점진적인 수행을 바탕으로 하고 있다. 이러한 방법론은 사용자 요구가 충분히 반영된 재사용이 가능한 컴포넌트의 식별, 분류, 명세 및 구현 방법의 과정을 기술할 것이며, 나아가 전사적 GIS 구축을 위한 지침이 될 것이다. 방법론 구축을 위한 세부 목표로는 우선적으로 방법론의 각 단계에서 UML 활용의 극대화를 통한 이해도를 높이는 것이다. 둘째, 각 단계의 산출물을 명확하게 제시하며, 이러한 산출물이 다음 단계의 입력으로 활용되는 일관성 있는 진행을 보여 주고자 한다. 셋째, 각 단계별 작업을 성공적으로 수행하기 위한 구체적인 지침과 기법들을 제시한다. 넷째, 하수도시설물관리 업무 영역에 실제 적용하여 방법론의 적용 가능성 및 GIS 구축을 위한 지침으로서의 기반을 제공하고자 한다.

1.3 연구 범위와 논문의 구성

본 연구는 CBD 관련 연구 동향, GIS를 위한 CBD 방법론의 제안, 본 연구에서 제시하는 방법론과 기존 방법론과의 비교 분석 및 방법론 적용을 위한 적용 사례 등을 주요 연구 범위로 한다. 이를 위하여 우선적으로 2장에 기술된 국외 동향으로서 OGC 활동의 파악과 함께 국내 동향으로는 개방형 GIS 컴포넌트 S/W 개발 현황에 대하여 분석하였으며, 오늘날 GIS 구축을 위하여 활용되고 있는 방법론을 살펴보았다. 한편 Catalysis 및 마르미-III를 포함하여 국내외에서 개발되어 사용되고 있는 방법론들의 장·단점 등을 분석하여 이들 분석이 본 방법론의 특징을 차별화 하는데 도움이 되도록 하였다. 이러한 동향 파악을 바탕으로 3장에서는 본 연구의 초점인 GIS를 위한 CBD 방법론이 제안되도록 하였다. 생명주기(Software Development Life Cycle)를 따라 계획부터 분석, 설계, 구현, 시험 및 배치의 각 단계에서 수행할 주요 활동(혹은 프로세스)을 기술하고, 활동의 수행 결과를 UML의 각 다이어그램을 활용하여 모델링함으로써 이해도를 높이고, 작업의 결과를 평가하는데 도움이 되도록 하였다. 특히 컴포넌트의 분류, 식별 및 정의에 본 방법론의 핵심이 있으며, 또한 컴포넌트 활용을 위한 인터페이스를 정의하며, 나아가 전사적 GIS 구축을 위한 청사진을 제시하였다. 이를 근거로 4장에서는 제안한 방법론을 하수도시설물관리 분야를 대상으로 실제 상세 적용하였다. ATOM과 타 방법론을 비교 분석함으로써 ATOM의 여러 특징을 5장에서 기술하였다. 6장에서 본 연구의 결과를 서술하였다.

2. 연구동향

2.1 GIS 컴포넌트 국외 현황

OGC는 지리공간자료 및 지리정보처리 자원을 오늘날의 분산처리 컴퓨팅 기술 환경으로의 완전한 통합을 비전으로 내세우며, 상호운용성 있는 지리정보처리를 촉진하며, 지리정보처리 기술을 산업계 정보기술 표준과 일치시키며, 상호운영 가능한 소프트웨어 인터페이스를 개발하기 위하여 지리정보 공급자와 사용자간의 협력을 도모하는 것을 목표로 설립된 단체로서 ISO/TC 211 국제 표준위원회와 협조 관계를 유지하며 1998년부터 이질적인 지리자료 및 지리정보처리 서비스에 투명한 접근을 가능하게 하는 추상사양과 구현사양을 계속 발표하고 있다. 추상사양의 목적은 구현사양을 개발하기에 충분한 개념적 모델을 만들고 문서화하는 것이다. 구현사양은 산업표준의 구현 및 소프트웨어 응용 프로그래밍 인터페이스를 위한 명확한 기술적 플랫폼 사양이다(4). OGIS는 전세계의 소프트웨어 개발자들에게 상세한 공통 인터페이스 모형을 제공하므로 소프트웨어 개발자로 하여금 다른 OGIS 호환 소프트웨어와 상호 적용할 수 있는 소프트웨어 개발을 촉진한다. 즉 컴포넌트 GIS 환경을 요구하고 있다. 그러나 GIS를 위한 컴포넌트 기반의 개발 방법론은 아직까지 제시가 안된 실정이다.

그리고 국외의 GIS 분야의 컴포넌트 개발 사례는 크게 두 가지로 분류할 수 있다. 하나는 GIS 구축 지원 도구를 중심으로 컴포넌트화가 이루어 졌고, 또 하나는 응용 시스템 구축 분야이다. 지원 도구를 중심으로 GIS 컴포넌트 생성의 사례는 ESRI(MapObject), SylvanMaps Ascent(SylvanMaps), Integrgraph (GeoMedia) 및 Mapinfo(MapX v3.5) 등의 도구 제작사들이 자사의 제품을 컴포넌트화 하여 응용시스템에서 활용하도록 한다. 응용 시스템을 위한 컴포넌트 GIS 구축사례는 외국의 많은 조직에서 살펴볼 수 있다(5).

2.2 GIS 컴포넌트 국내 현황

국내의 GIS 컴포넌트의 개발은 국가 주도로 이루어지고 있다. ETRI 주관으로 1999년도부터 2001년까지 '개방형 GIS 컴포넌트 S/W 기술 개발' 사업을 추진하였으며, 핵심 공통 컴포넌트, 데이터 제공자 컴포넌트, 공간 정보유통 컴포넌트 및 응용 컴포넌트의 4가지를 기본으로 10개 이상의 업체들이 참여하여 개발하였다. 이들 컴포넌트의 분석 및 설계는 UML을 활용하고 있다. 지방자치단체에서는 관련 컴포넌트들을 활용하면 되므로 재개발의 부담뿐만 아니라 응용시

스템 개발비용과 시간의 부담을 덜게 된다. 그리고 학계에서도 이론적 지식을 바탕으로 다만 미들웨어 도구를 컴포넌트에 기반한 설계 및 구현 정도에 대한 연구가 진행되어온 수준이다(6)(7)(8)(9). 그리고 GIS 컴포넌트 개발의 필요성을 강조한 연구가 수행된 적이 있다(10). 이렇듯 여러 연구들이 컴포넌트 기반 개발 환경에 맞추어 설계 및 구현을 하고 있으나 이를 컴포넌트 기반의 방법론에 입각하여 개발한 사례는 아직까지 나타나지 않았다. 정보통신부 등 정부기관은 앞으로 4s 응용기술 및 어플리케이션 개발에 컴포넌트 개발 및 컴포넌트 기반의 시스템 개발을 요구하고 있어 개발 환경이 CBD 환경으로 활발해질 것이다.

2.3 GIS 구축 방법론

전술한바와 같이 GIS 도메인을 위한 컴포넌트 기반의 개발방법론은 지금까지 제시된 바는 없다. GIS를 위한 방법론이 국내외적으로 제시된 적이 있으나 모두가 컴포넌트에 기반한 개발방법론은 아니다(11)(12). SLC, Method/1, Method plus, 4FRONT, IEM, Navigator, MaRMI-II, Guide 등의 기존의 방법론들을 GIS 프로젝트에 활용하고 있는 실정이다(13). 이중 SLC, IEM, MaRMI 등은 개방되어 방법론 전체를 활용할 수 있으나, Navigator, 4Front 등은 유명 회계 법인들이 자사 고객의 문제를 해결하는 차원에서 활용하는 방법론으로 상세한 내용이 개방되어 있지 않아 일반 조직에서는 활용하기가 사실상 어려운 실정이다(14).

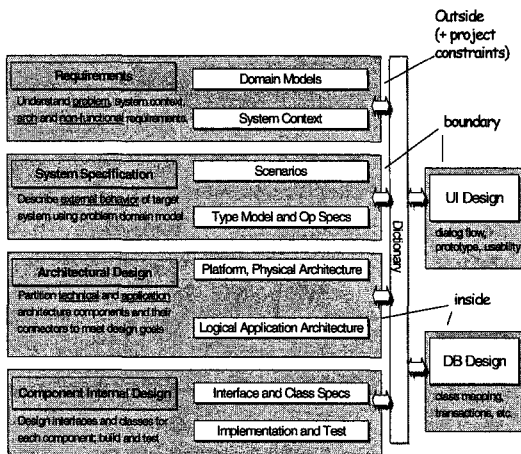
국내 GIS 분야에서는 아직도 이들을 활용한 개발이 많이 이루어지고 있고, 국외는 뉴욕 주립대학 및 기타 연구기관에서 수행된 연구로서 개발된 "The GIS Development Guide"(11)와 ESRI에서 출판된 "Managing a GIS" 프로세스가 있다(12).

이밖에 카네기멜론 대학의 SEI(Software Engineering Institute) 연구소에서 개발한 CMM (Capability Maturity Model), ISO/IEC/JTC1/SC7에서 발표한 SPICE (Software Process Improvement Capability dEtermination)(15)(16), 캐나다 Bell에서 개발한 Trillium 그리고 ESA에서 개발한 PSS-05 프로세스들은 너무나 방대하여 이해가 어렵고 많은 문서화 요구를 비롯하여 적용하는데 매우 어려움을 겪고 있어 국내에서는 대기업을 중심으로 위의 문서들을 참고로 자체적인 개발지침서를 작성하여 활용하고 있다.

2.4 컴포넌트 기반 방법론

2.4.1 국외 방법론

Catalysis 방법론은 소프트웨어 개발 프로세스를 다양한 프로젝트에 적용하도록 하나의 프로세스보다는 프로세스 패턴들을 제공한다. 이 방법론의 개발 프로세스는 분석, 설계, 구현 및 시험단계로 이루어져 있다. 나선형 모델이 이런 개발 프로세스에 적용된다. 분석, 설계, 구현 및 시험단계는 반복적으로 진행되며 전체 시스템은 점진적으로 발전한다[17]. Catalysis 방법론은 다양한 표기법과 컴포넌트와 어플리케이션 개발을 위한 지침을 제공한다. 그것은 개발 프로세스를 다양한 프로젝트에 적용하기 위해 패턴 기반의 공정을 제공한다(그림 1).



〈그림 1〉 Catalysis 방법론 개발 프로세스

이 방법론의 한계로는 작업들 사이의 흐름들이 구체적으로 정의되지 않으므로, 컴포넌트를 개발하는 동안에 작업들을 추적하고 통제하기가 어렵다. 각 작업을 위한 입력물이나 출력물들이 개략적으로 제공되기 때문에 입력물이나 출력물을 동기화 할 수 없다. 작업을 수행하기 위한 구체적인 지침 또한 제공되지 않는다. 하나의 컴포넌트 내의 클래스들 사이의 메시지 흐름이 기술되지 않는다. 그것은 각 컴포넌트에 대한 커스터마이징 부분을 제공하지 않는다[18].

CBD96 방법론은 ICON Computing's Catalysis에 바탕을 두고 있다[19]. 모델링 작업은 COOL: Spex에 의해 지원된다. 이 방법론은 요구사항 정의, 분석, 행위자명세 및 아키텍처 단계로 구성된다. 이 컴포넌트 모델링은 컴포넌트가 구현되는 동안 반복적으로 순환되며, 여기에는 도메인 모델링, 비즈니스 형태 모델링, 아키텍처 모델링, 컨텍스트 모델링, 인터페이스 모델링 및 컴포넌트 명세가 주요 작업이다. 이

방법론은 컴포넌트 모델링의 작업에 대한 정의는 제공하지만 컴포넌트 설계에 대한 구체적인 절차는 제공하지 않으며 특히 이 방법론은 커스터마이징 기법은 제공하지 않는다. 이외에 SCIPPO, Fusion 2.0, UNIFACE 및 Select Perspective 등이 있다 [20][21][22][23].

2.4.2 국내 방법론

COMO는 실제 컴포넌트 개발에 적용될 수 있는 실제적인 객체지향 컴포넌트 개발 방법과 모델링 기법을 제시한다[18]. 제시된 방법은 통합된 방법을 기반으로 두고 있고, 제시된 모델링 기법은 UML표기법에 기반을 두고 있다. 한 작업에서부터 이어지는 작업까지 매끄럽게 수행되도록 하기 위하여 개발방법의 프로세스 아키텍처를 stage, cycle, phase 및 task에 대하여 번호를 부여하는 전략을 제공한다. COMO는 컴포넌트만을 식별하고 구현하는 방법론이다.

마르미-III(Magic and Robust Methodology Integrated)는 EJB기반의 시스템 개발을 지원하며, UML 모형을 표준으로 사용하고, 쓰임새 기반 개발, 아키텍처 중심적이며, 그리고 반복적, 점진적 개발을 목표로 하고 있다. 마르미-III의 개발 공정은 컴포넌트 개발 공정과 이미 개발된 컴포넌트를 활용하여 시스템을 개발하는 공정으로 되어 있는데, 마르미-III는 컴포넌트 기반 시스템 개발 공정을 중심으로 컴포넌트 개발 공정을 함께 기술하였다. 개발 절차는 크게 계획 단계, 아키텍처 단계, 점진적 개발 단계 및 인도 단계의 4단계로 이루어져 있고, 각 단계는 논리적으로 서로 연관된 작업을 하나로 묶은 활동으로 구성되어 있다.

그리고 본 논문의 주제와 유사한 연구 결과의 방법론 CBSDP(Component-Based Software Development Process for Urban Information System)이 제시되었다[24]. CBSDP는 UIS를 GIS 도메인을 위한 것보다는 MIS 측면에 중점을 둔 내용의 전개가 이루어졌고, CBD에 기반을 둔 기술적 측면을 부각시켜 컴포넌트 도출에 중점을 두었다. 반면 기술적·관리적 측면을 고려하고, 소프트웨어 개발의 전체적인 가이드 역할을 하는 ATOM과는 구분이 된다고 볼 수 있다.

3. GIS 개발방법론의 정립(ATOM)

이 장에서는 본 논문의 주제인 컴포넌트 기반 GIS 소프트웨어 개발 방법론의 프로세스, 즉 ATOM을 제안하며, 각 프로세스에 대한 세부적 서술이 포함되었다. 본 방법론의 주요 특징을 기술함과 아울러 특징들

이 신뢰성 있게 수행되어 가는 과정을 서술하고자 한다. 특히 GIS를 위한 재사용 가능한 컴포넌트 추출을 위한 방법 및 지침이 상세하게 제시하였다.

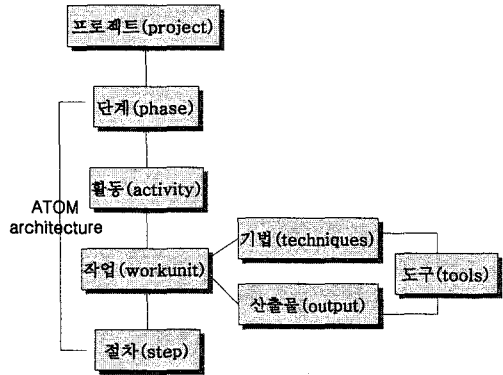
3.1 주요 특징

ATOM 방법론의 주요 특징들은 다음과 같다. 우선적으로 이 방법론의 개발 프로세스는 계획, 분석, 설계, 구현, 시험 및 배치의 소프트웨어 개발 생명주기의 패러다임을 준수한다. 이와 함께 본 방법론에서 제시된 모델링은 UML 표기법에 기반을 두고 있으며, 다양한 문서의 작성을 통하여 본 방법론의 이해도를 높일 것이다. 따라서 일관성 추적이 용이해진다. 또한 작업들 사이의 입력물과 출력물을 명확하게 정의하며, 한 작업의 출력물이 다음 작업의 입력으로 어떻게 활용되는지를 알 수 있으며, 작업을 수행하기 위한 각 단계별 구체적인 지침을 제공한다. 본 방법론은 특히 컴포넌트 및 인터페이스 정의에 또한 초점을 둔다. 이것은 컴포넌트 분류, 컴포넌트 추출 지침, 재사용 및 개발 컴포넌트의 확인, 컴포넌트 상세 명세 그리고 인터페이스의 상세 정의 등을 포함한다. 그리고 본 방법론은 아키텍처 중심적이며, 최상위 계층(layer)인 단계에서부터, 활동, 작업 및 절차의 하위 계층으로 진행하면서 작업을 상세하게 정의한다. ATOM은 개발 경로를 제시한다. CD(Component Development) 및 RAD의 경로이다. RAD는 ATOM의 전체 프로세스가 사용자 중심의 군더더기를 배제한 최소, 최적의 기능만 제시하였으므로 빠른 개발을 위한 RAD용으로 사용할 수 있고, CD는 컴포넌트 개발만을 지원하도록 프로세스를 조정하여 사용할 수 있다. 마지막으로 ATOM은 단순한 기능과 최소의 크기 및 최대의 재사용을 위한 컴포넌트 추출에 목적을 둔 GIS를 위한 컴포넌트 기반의 개발방법론이다.

3.2 ATOM의 개발 프로세스

3.2.1 메타 모형

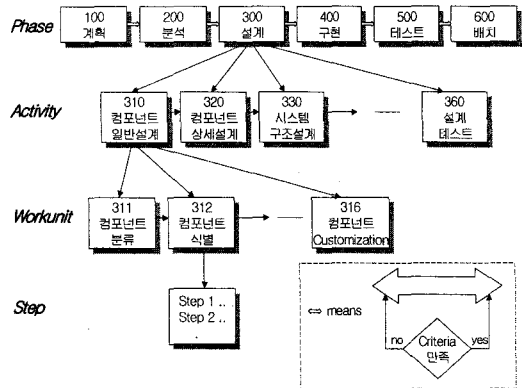
본 방법론은 최상위 계층인 단계에서부터 최하위 계층인 절차까지의 아키텍처를 가지는 <그림 2>과 같은 메타 모형을 구성한다. 이 메타 모형의 단계 -> 활동 -> 작업 -> 절차가 ATOM의 아키텍처 계층으로 작업 절차이다. 기법은 구성요소 및 적용지침을 제공하며, 산출물은 양식을 제시하며, 양식의 항목을 설명하는 것을 포함한다. 작업절차, 기법, 산출물이 ATOM을 구성하는 기본 요소이다.



<그림 2> ATOM 메타모형

3.2.2 프로세스

<그림 3>는 본 방법론의 개발 프로세스를 보여준다. 개발 프로세스는 소프트웨어 개발 생명주기인 계획(100)에서부터 배치(600)의 6단계(phase)로 구성된다. 각 단계는 산출물의 검토 회의 및 평가를 거쳐 다음 단계로의 진행을 결정한다. 다음 단계로의 진행이 승인되지 않으면 반복적으로 현 단계를 수행하고, 현 단계에서 문제점(혹은 오류)이 발견 시에도 관련 단계로 회귀하여 반복적으로 수행하는 프로세스를 가진다.



<그림 3> ATOM의 아키텍처

<그림 3>는 본 방법론의 개발 프로세스를 보여준다. 개발 프로세스는 소프트웨어 개발 생명주기인 계획(100)에서부터 배치(600)의 6단계(phase)로 구성된다. 각 단계는 산출물의 검토 회의 및 평가를 거쳐 다음 단계로의 진행을 결정한다. 다음 단계로의 진행이 승인되지 않으면 반복적으로 현 단계를 수행하고, 현

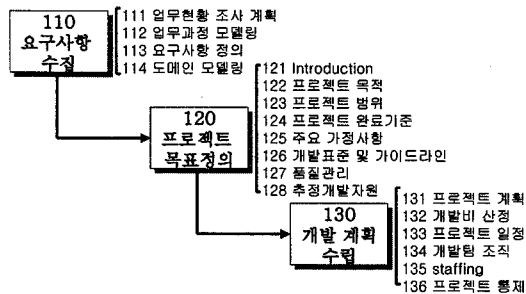
단계에서 문제점(혹은 오류)이 발견 시에도 관련 단계로 회귀하여 반복적으로 수행하는 프로세스를 가진다.

3.2.3 프로젝트 관리

그리고 소프트웨어 개발의 성공적인 수행은 기술적인 요소뿐만 아니라 관리적인 요소가 중요하므로 프로젝트 관리 생명주기에 따라서 프로젝트 관리 부분을 병행하여 설명할 것이다(25). <그림 3>의 점선 사각형 상자내의 검토회의 및 평가 등은 프로젝트 수행 중 작은 작업 단위별로 검토하여야 할 중요한 관리적인 측면이다. 또한 본 방법론에서 주장하는 점진적 및 반복적의 의미는 관리적인 측면에서도 반드시 고려되는 특성이기도 하다.

3.3 계획 단계

계획단계는 요구사항 수집, 프로젝트 목표 정의 및 개발 계획 수립의 3가지 주요 활동으로 구성되어 프로젝트를 수행하기 위한 기초 업무 조사 및 준비를 하는 단계이다. GIS 요구사항 수집은 업무 현황 조사 계획, 업무과정 모델링, 요구사항 정의 및 도메인 모델링의 작업으로 구성된다. 현업의 업무 분석을 통한 사용자 요구사항 도출과 프로젝트 범위를 정의하는 단계이다(그림 4).



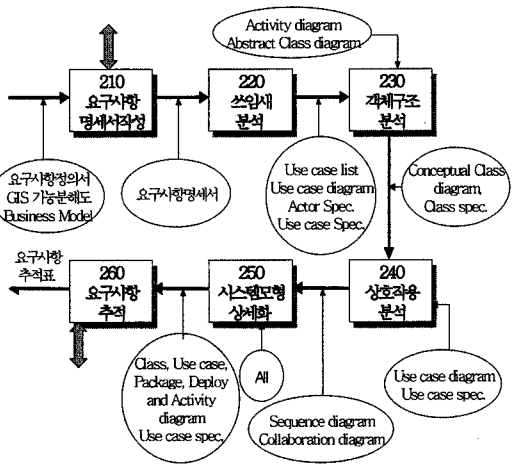
<그림 4> 계획 단계

요구사항 정의는 업무 과정 모델링에서 정의된 업무 흐름도 및 관련 자료를 바탕으로 사용자 요구사항을 정의하며, 요구사항 정의서, GIS 기능 분해도, 어휘 목록 및 클래스 목록을 작성한다. 프로젝트 목표 정의는 기술적인 측면보다는 관리적인 측면으로 프로젝트에 대한 방향과 지침, 환경을 얼마나 이해하고 설정하였는가를 파악하는 활동이다(26). 프로젝트 목표가 개략적인 프로젝트 계획이라면 개발 계획 수립은 상세 프로젝트 계획에 해당한다. 실제로 프로젝트 수행에

필요한 구체적 준비를 하는 단계이다. 계획 단계 이후 수행할 모든 작업들에 대한 목록을 작업분리구조(Work Breakdown Structure)로 정의하는 프로젝트 계획 수립 작업, 실제 소요 개발비 산정, 작업 수행 일정표 작성, 개발팀 조직도 및 인원 구성도가 명확하게 기술되어야 한다.

3.4 분석 단계

분석 단계에서는 계획 단계에서 조사된 요구사항 정의서, 기능 분해도 및 업무 과정 모델링을 분석하고, 현업의 추가 상세 조사를 통하여 목표 도메인을 이해하고, 상세 요구사항 명세서를 작성한다. 뿐만 아니라 시스템의 정적(요구사항 명세서 작성 및 객체 구조 분석) 및 동적(이용사례 분석 및 상호 작용 분석)인 측면을 번갈아 수행하면서 목표 도메인에 대한 분석이 상세히 이루어진다. 이처럼 목표 도메인에 대한 이해와 명세로부터 설계 단계에서 수행할 컴포넌트 정의 및 구현의 토대를 마련한다(그림 5). 분석 단계의 주요 목적은 시스템 도메인 내에 존재하는 시스템 기능(이용사례)과 데이터(객체)를 모두 식별하고, 그것들이 사용자요구를 만족시킬 수 있는가를 확인하는 것이다.



<그림 5> 분석 단계

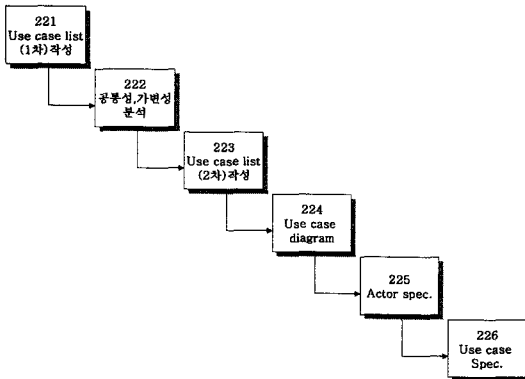
3.4.1 요구사항 명세서 작성

계획 단계에서 작성한 요구사항 정의서, GIS 기능 분해도 및 업무 과정 모델을 이용하여 요구사항 명세서를 작성한다. 그리고 조사된 요구사항을 바탕으로 도메인 내에서 요구사항을 만족시키는데 기본적으로 필요한 지도를 파악하여 제시한다.

요구사항 명세서 작성 절차는 다음과 같다. 첫째, 계획 단계에서 작성한 기능 분해도를 다시 정리한다. 특히 누락된 기능이 없는지를 요구사항 정의서 및 업무 과정 모델로부터 상세히 검토한다. 둘째, 기능 분해도의 기능 중에 최하위 기능만 뽑아내어 그 기능에 대한 상세 기능을 기술한다. 이 상세 기능은 이용사례 분석에서 사용될 1차 이용사례가 된다. 상세 기능은 하나의 트랜잭션 정도의 크기로 설정한다. 셋째, 요구되는 레이어를 정의한다. 이 레이어들은 이용사례 분석에서는 하나의 이용사례로 간주하고, 객체(클래스) 분석에서는 하나의 클래스로, 컴포넌트 식별에서는 컴포넌트로 정의한다. 이들 레이어의 내부 구조, 즉 클래스 모델은 공간 데이터베이스 설계에서 상세 작성될 것이다. 넷째, 사용자와 함께 누락된 상세 기능이 없는지를 검토한다. 요구사항정의서, GIS 기능분해도 및 요구사항명세서의 양식 및 내용은 4장 사례연구에서 확인한다.

3.4.2 이용사례 분석

이용사례는 시스템 전체나 이용사례 일부 행동을 명세화하고 순차적으로 발생하는 활동들을 기술하며, 행동 순차 모임을 설명하며, 시스템은 이러한 활동을 수행하여 행위자에게 팔목할 만한 결과를 준다. 즉 이용사례는 행위자와 시스템간의 교류를 표현한다. 이용사례도(use case diagram)는 시스템, 서브시스템, 클래스의 행동을 모형화 하는 중심 도구로서, 각각의 이용사례도는 이용사례와 행위자, 또 그들간의 관계를 나타낸다[27]. 따라서 이용사례도는 우리의 목표 GIS로부터 시스템을 활용하는 외부 행위자(사용자 포함)에게 제공하고자 하는 모든 기능(정보)을 행위자와 시스템 관점에서의 관계를 표현하는 것이다. 물론 시스템은 모든 기능에 해당하는 이용사례로 구성된다. 이용사례도를 작성하는 작업들이 <그림 6>에 나타나 있다.



<그림 6> 이용사례도 작성 프로세스

3.4.3 객체 구조 분석

객체 구조 분석 혹은 도메인 분석은 목표 도메인을 이해하는 것으로 우리의 시스템 범위 내에서 행해지고 있는 업무 과정들과 그 과정을 이루고 있는 상세한 사항들을 파악하는 것을 말한다. 이 활동의 목표는 개념적 초기 클래스도를 만드는 일이다. 이미 앞에서 작성한 업무과정 모델링의 활동도, 추상적 클래스도, 어휘 목록 및 이용사례 명세서로부터 클래스도가 작성된다. 어휘 목록에서 조사된 명사는 클래스가 되며, 또한 일부는 속성이 될 수도 있다. 동사와 동사구는 오퍼레이션 혹은 연관에 붙일 레이블을 만드는데 사용할 수 있다. 클래스도에 나타나는 각 클래스에 대한 명세서도 함께 작성한다.

3.4.4 상호 작용 분석

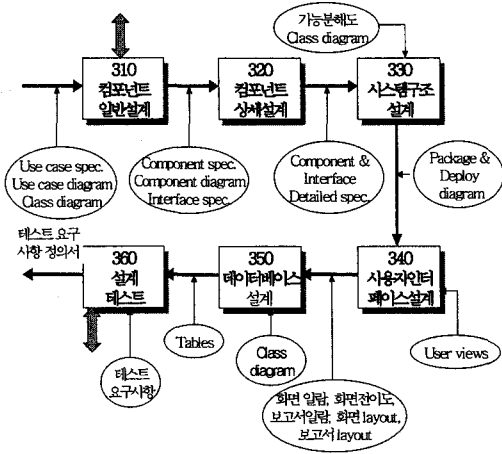
이 활동은 기 작성한 이용사례의 사건 흐름이 잘못 정의된 부분이 없는지, 클래스도에서 누락된 객체는 없는지를 파악할 뿐만 아니라 사건(메시지)의 시간 순서에 따른 객체간의 교류 관계를 보여준다. 이러한 관계를 보여주는 다이어그램이 협력도와 순차도이다. 작성 및 관점의 차이가 있으나 본 연구에서는 순차도를 중심으로 전개한다. 순차도를 작성하기 위해서는 이용사례 명세서와 클래스도의 활용을 필요로 한다. 요약하면, 상호 작용 분석이란 시스템에 존재하는 기능(이용사례)과 데이터(객체)를 상호 분석하여 누락된 이용사례 및 데이터를 파악하여 반영하는 것이 목적이다.

3.5 설계 단계

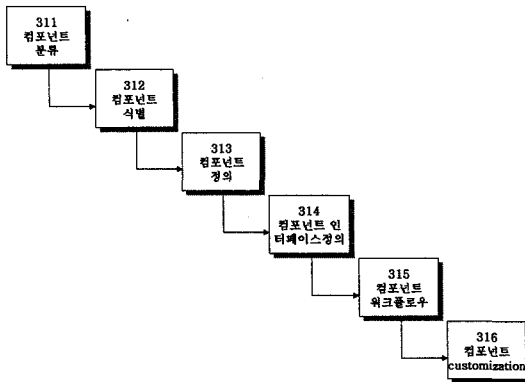
설계 단계에서는 분석 단계에서 정의한 목표 도메인 내의 사용자 요구를 해결하는 방법을 제시하는 것으로서, 본 연구의 주제인 컴포넌트를 도출하여 사용자 요구를 만족시키는 과정을 설명할 것이다. 이 절에서는 컴포넌트의 분류 및 추출 방법, 컴포넌트 인터페이스 정의, 컴포넌트 워크플로 및 커스터마이징 명세 등을 포함하는 일반설계와 컴포넌트 구현 목적의 상세 설계, 시스템과 의사소통을 위한 사용자 인터페이스 설계, 데이터베이스 설계 및 설계 내용에 대한 테스트의 활동 등을 설명한다(그림 7).

3.5.1 컴포넌트 일반 설계

이 활동은 본 연구의 핵심으로 본 연구에서의 컴포넌트의 분류, 컴포넌트의 식별 방법 및 추출된 컴포넌트의 정의 등 컴포넌트의 모든 것을 정의하는 단계이다(그림 8).



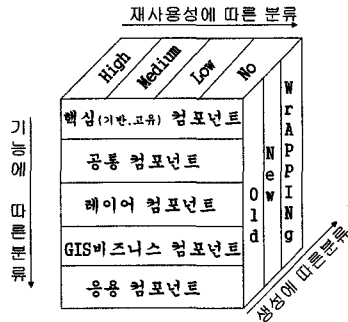
〈그림 7〉 설계 단계



〈그림 8〉 컴포넌트 일반설계 프로세스

가. 컴포넌트 분류

본 연구에서는 컴포넌트를 기능별 분류, 재사용성에 따른 분류 및 생성에 따른 분류를 통하여 컴포넌트의 특성을 이해하고 구현 여부를 파악한다. GIS 도메인에 대한 기능별 분류는 핵심(기반, 고유), 공통 및 응용 컴포넌트 외에 GIS 도메인 내의 비즈니스 프로세스 영역에 관한 single 컴포넌트인 GIS 비즈니스 컴포넌트와 레이어 컴포넌트를 추가한다. 응용 컴포넌트는 구현 단계의 응용 프로그램 목록을 정의할 때 응용 컴포넌트 후보가 될 수 있으면 응용 컴포넌트로서 다시 추가 정의한다. 재사용에 따른 분류는 재사용의 정도에 따라 No, High, Medium 및 Low로 구분한다. 또한 생성에 따른 분류는 기존의 컴포넌트를 재사용 하느냐, 새로이 개발하느냐, 혹은 기존 시스템을 Wrapping(기존 컴포넌트에서 새로운 컴포넌트 만드는 경우 포함)하느냐에 따라 Old, New 및 Wrapping으로 구분된다(그림 9)(표 1).



〈그림 9〉 GIS 컴포넌트 분류 구조

〈표 1〉 GIS 컴포넌트 설명

분류	설 명	비 고
기반	다양한 GIS 응용시스템에서 필수적으로 필요한 기능을 표준화된 인터페이스를 통하여 제공하는 컴포넌트	• 공간 데이터 액세스, 조작, 매핑 등
고유	GIS 기반 컴포넌트 외에 GIS 고유의 요소 기술을 표준화된 인터페이스를 통하여 제공하는 컴포넌트	• 네트워크 분석 • GRID 분석 • TIN 분석 기능등
공통	GIS를 포함하여 다른 정보시스템에서 공통적으로 사용 가능한 컴포넌트	• 통계, 차트, 문서작성 등
레이어	GIS만이 가지는 feature의 공간정보(위치 및 위상)를 제공하기 위한 컴포넌트	• 배수구역도 • 하천
GIS 비즈니스	GIS 도메인 내에서 업무 로직을 처리하는 최소의 기능과 크기를 가진 컴포넌트	• 통수능분석 • 우수관망해석
응용	특정 업무를 위하여 위의 컴포넌트들을 조립하여 만들어진 어플리케이션	• 도로시설물관리응용 • 상하수도관리응용

나. 컴포넌트 식별

이 작업은 목표 GIS 도메인 내에서 컴포넌트 후보를 식별하는 것으로서 이용사례도, 이용사례 명세서, 클래스도 및 클래스 명세서 등을 활용한다. 컴포넌트를 식별해내는 과정은 다음과 같다.

step 1. 클래스/이용사례 매트릭스

이것은 클래스와 이용사례들간의 관계를 나타내는 매트릭스로서 소위 CRUD(Create, Read, Update and Delete) 매트릭스로 알려져 있다. 첫째, 원시 CRUD를 작성한다(표 2). 수직으로 이용사례를, 수평으로 클래스를 나열하고, 각 칸에 우선 순위(C > D/U > R)를 지키며 하나의 기호만 기술한다. 둘째, 클러스터링 알고리즘을 이용하여 클러스터링 후의 CRUD 매트릭스를 작성한다(표 3)[28]. 셋째, 동일 클러스터에 속해있는 클래스들을 묶어서 하나의 그룹으로 만든다. 이 그룹을 추상적 컴포넌트로 정의한다. 넷째, 경험적 판단에 의하여 식별할 수 있는 GIS 비즈니스 컴포넌트를 정의한다. 독립적이고, 최소 단위의 컴포넌트 후보를 추상적 컴포넌트로부터 분리해 낸다. 예를 들면, 하수관거코드에 의한 시설물 위치 찾기(혹은 그 반대), 행정코드 번호에 의한 행정동의 명세 확인 등 코드화가 가능한 것이 사용의 독립성과 재사용을 위하여 일차적인 컴포넌트가 된다. 그리고 레이어들을 GIS 비즈니스 컴포넌트로 분류한다. 다섯째, 또한 GIS 도메인에서 경험적 판단에 의한 핵심 및 공통 컴포넌트를 정의한다. 이들은 많은 도구 제작사들 및 ETRI에 의하여 Open GIS Consortium에서 제안한 단순 개체 명세를 구현한 많은 컴포넌트들이 여기에 해당된다. 넷째, 다섯째 단계에서 분류한 이들을 경험적 컴포넌트로 정의한다. 여섯째, 추상적 컴포넌트로부터 경험적 컴포넌트를 분리하여 재 그룹핑 한다. 이 재그룹핑된 컴포넌트는 도메인의 업무 내용에 의존하는 GIS 비즈니스 컴포넌트가 된다. 일곱째, 지금까지 식별된 컴포넌트에 대하여 응용컴포넌트로 구성될 컴포넌트를 선택하여 패키지화된 형태의 응용컴포넌트를 정의한다. 여덟째, 지금까지 도출한 컴포넌트에 이름을 붙인다.

〈표 2〉 원시 CRUD 매트릭스

	C1	C2	C3	C4
U1	C	R	C	R
U2	R	R	C	C
U3	R	C	R	R
U4	R	C	R	R

〈표 3〉 클러스터링된 CRUD 매트릭스

	C1	C3	C4	C2
U1	C	C	R	R
U2	R	C	C	R
U3	R	R	R	C
U4	R	R	R	C

step 2. 기초 컴포넌트 다이어그램 작성

앞에서 도출한 컴포넌트와 클러스터링의 결과를 참조하여 컴포넌트도를 그린다. 여기에서의 컴포넌트도는 컴포넌트와 컴포넌트간의 의존 관계 및 연관 관계만 그리며 인터페이스는 생략된 아이콘 형으로 표현한 후에 개략적인 컴포넌트도를 작성한다.

step 3. 기초 컴포넌트 명세서 작성

추출된 각 컴포넌트에 대한 기초 컴포넌트 명세서를 작성한다. 여기에서는 컴포넌트 명세서의 목차를 상세히 정의하고, 기술 가능한 부분만 선택적으로 기술한다. 왜냐하면 인터페이스, 커스터마이징, 워크플로우 등 정의해야할 부분이 아직 남아있기 때문이다.

다. 컴포넌트 정의

앞 작업에서 정의한 컴포넌트들을 분류구조에 따라 정의하는 것으로 의미 있는 이름뿐만 아니라 기능별, 재사용에 따른 및 생성에 따른 분류를 통하여 각 컴포넌트들의 성격을 분명히 한다. 그리고 기존의 재사용할 수 있는 컴포넌트를 중복 개발하는 사례가 발생하지 않도록 컴포넌트의 재사용 여부의 파악과 출처를 파악하는 일도 매우 중요하다.

라. 컴포넌트 인터페이스 정의

인터페이스는 오퍼레이션의 모음으로써 클래스와 컴포넌트의 서비스를 명세화 하는데 사용하는 것으로서 컴포넌트들을 함께 묶는 수단으로써 사용된다. 순차도, 클래스도 및 이용사례도를 이용하여 인터페이스를 추출 및 명세화 한다. Open GIS Consortium에서 제시한 OpenGIS 단순 개체 명세의 인터페이스 작성 표준 지침을 따른다. 구현 단계에서도 역시 이 표준 지침을 준수할 것이다.

마. 컴포넌트 워크플로우

이 작업은 행위자와 각 인터페이스 내에 포함되어 있는 오퍼레이션별로 순차도를 작성하고, 이로부터 시스템의 기능을 파악하게 된다. 이용사례도, 컴포넌트도 및 컴포넌트 인터페이스 명세서를 활용하여 컴포넌트 인터페이스 내 각 오퍼레이션별 순차도를 작성한다. 컴포넌트가 제공하는 서비스가 인터페이스를 통해 어떻게 기술되는지를 보여주는 것이 이 작업의 목적이다. 3.4.4절의 순차도 작성 방법과 동일하다.

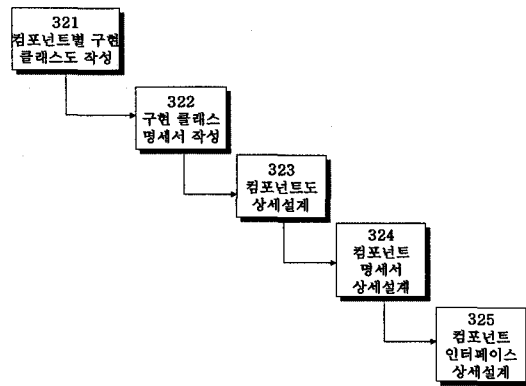
바. 컴포넌트 커스터마이징 정의

이 작업은 이미 정의한 가변성별로 커스터마이징 구조를 설계하는 것으로서 요구사항 명세서, 공통성·가변성 목록 및 순차도를 참조하여 컴포넌트 커스터마이징 명세서를 작성한다. 이 작업의 목적은 응용에 따라

서 로직의 변화하는 커스터마이징 부분을 식별하고, 컴포넌트 인터페이스에게 이들 부분을 제공하는 것이다. 응용 개발자는 컴포넌트는 내부 정보를 보여주지 않으므로 컴포넌트 인터페이스를 통하여 단지 컴포넌트를 접근할 수 있다. 그러므로 이 커스터마이징 부분이 컴포넌트 인터페이스에 제공되어야 하며, 컴포넌트 사용자는 소스코드 변경 없이 매개변수를 통하여 컴포넌트를 커스터마이징 할 수 있다.

3.5.2 컴포넌트 상세 설계

이 활동은 앞에서 작성한 이용사례도, 기초 컴포넌트도, 워크플로워, 컴포넌트 커스터마이징 및 컴포넌트 인터페이스 명세 등 모듈을 활용하여 컴포넌트에 대하여 구현을 위한 상세 설계를 한다. 이 활동의 하부 작업은 <그림 10>에 기술되어 있다. 컴포넌트 상세 설계 결과의 산출물들은 컴포넌트에 대한 모든 내용을, 즉 컴포넌트와 관련한 다이어그램, 명세서, 인터페이스, 클래스도와 클래스 명세서 등을 파악할 수 있도록 하며 컴포넌트 구현을 위한 실질적인 입력 자료가 되므로 상세 설계의 수행의 성공 여부에 따라 구현 단계의 노력에 영향을 미친다.



<그림 10> 상세(구현) 설계 프로세스

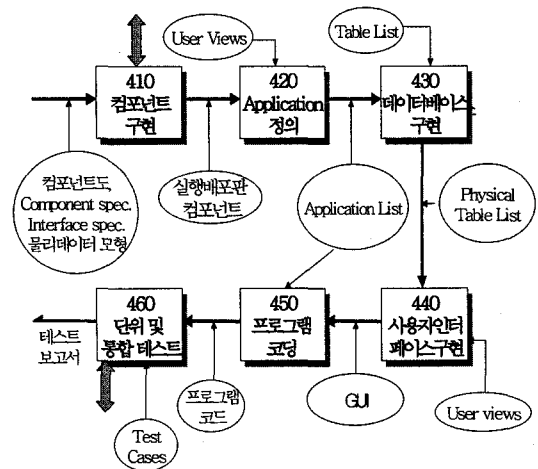
3.5.3 데이터베이스 설계

앞에서 작성한 개념적 데이터 모형인 구현 클래스도를 관계형 데이터베이스 테이블 구조로 변환하는, 즉 논리적 데이터 모형으로 변환하는 것이다. 논리적 모형은 사용하고자 하는 특정 DBMS나 구현 기술에 종속적이다. 본 논문에서는 공간정보에 해당하는 레이어(레이어)의 기하(위치)정보 및 위상정보, 그리고 속성 정보를 위하여 데이터베이스 스키마, 즉 테이블 설계를 수행한다. 데이터베이스 설계 활동은 공간정보 설계, 속성(비공간)정보 설계와 공간 및 속성정보 결합

의 3가지 세부 작업을 가진다.

3.6 구현 단계

구현 단계는 설계 단계에서 정의한 컴포넌트의 구현, 응용 프로그램의 정의 및 구현, 데이터베이스 구현, 사용자 인터페이스 구현, 단위 및 통합 테스트를 수행한다(그림 11). 그리고 작업 환경과 운영 환경을 유사하게 설치함으로써 실제 운영 시스템으로의 전환을 쉽게 할 수 있도록 한다.



<그림 11> 구현 단계

3.6.1 컴포넌트 구현

컴포넌트 구현은 컴포넌트 내 클래스들과의 상호 관계 및 인터페이스에 대한 코드를 작성하여 실행 배포 가능한 컴포넌트를 만드는 활동으로 컴포넌트도, 컴포넌트 명세서, 인터페이스 명세서 및 물리데이터 모형을 참조하여 실행배포판 컴포넌트 구현이 목적이다. 컴포넌트 모델의 환경(CORBA, COM, 및 EJB)을 설정하고, 그 환경을 지원하는 언어를 선택하여 컴포넌트를 개발한다. Visual Basic, Visual C++, Java 등의 구현 언어는 어떤 환경에서도 적용 가능한 언어로 이를 활용하여 쉽게 개발할 수 있을 것이다.

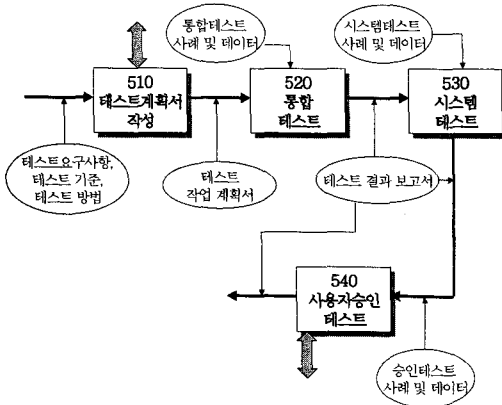
3.6.2 단위 및 통합테스트

본 연구의 단위테스트는 컴포넌트 각각에 대한 테스트를 수행하는 것으로서 크게 black-box 테스트와 white-box테스트로 나누어진다. black-box 테스트에서는 컴포넌트 인터페이스만 단지 테스트한다. 이것은 인터페이스내의 오퍼레이션들을 테스트함을 의미한다. 즉, 컴포넌트명, 컴포넌트 인터페이스명, 컴포넌트 인

터페이스내 메소드들을 테스트한다. 컴포넌트 인터페이스를 black-box 테스트를 위하여 컴포넌트 명세서 및 상세 인터페이스 명세를 사용한다. white-box 테스트의 경우는 컴포넌트내의 클래스들과 컴포넌트의 절차를 테스트한다. 클래스를 테스트하는 것은 클래스의 속성과 메소드를 테스트함을 의미한다.

3.7 시험 단계

시험 단계는 개발한 시스템에 대하여 개발자 관점과 사용자 관점에서 테스트를 수행하게 된다. 개발자의 관점에서는 시스템에 대한 모든 요구사항과 분석, 설계 명세서의 내용들이 시스템에 정확하게 반영되고 기능들이 정상적으로 수행되는 지를 검증하며, 사용자 관점에서는 사용자가 요구하는 기능 및 성능들이 시스템에 반영되어 시스템이 정상적으로 작동하는 지를 사용자 주도로 테스트하여 사용자가 원하는 시스템을 인수할 정도가 되는가를 검증하는 단계이다(그림 12).



(그림 12) 시험 단계

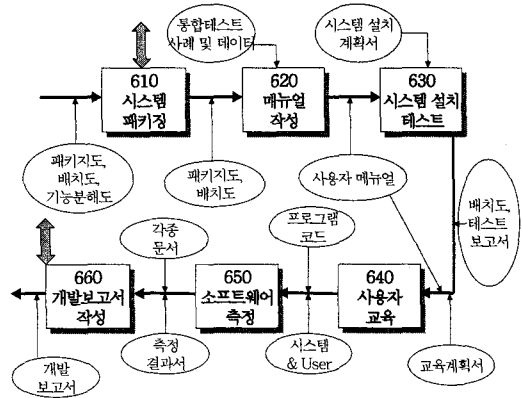
3.8 배치 단계

배치 단계는 시험이 완료된 소프트웨어를 업무 단위별 혹은 서버 시스템별로 패키징하여 하드웨어 및 운영 지향적인 시스템 구조를 보여주는 것이 목적이다. UML의 패키지도와 배치도를 사용하여 정의한다. 또한 사용자 매뉴얼 작성, 시스템 설치 테스트 및 사용자 교육 실시 등의 활동을 수행한다. 그리고 프로젝트 수행결과를 평가하기 위한 소프트웨어 측정 활동과 개발완료 보고서 작성 활동을 포함한다(그림 13).

4. ATOM의 적용 사례

이장에서는 사례 연구를 통하여 본 방법론의 적용성

과 효율성을 확인해 보고자 한다. 사례 연구의 도메인은 하수도관리 업무 분야의 하수도시설물관리 분야를 선정하였다. 하수도시설물관리는 대장관리, 도면관리, 하수관망관리 및 시설물관리의 업무들로 구성된다. 이들 중 하수관망관리, 즉 하수도 관망분석 및 우·오수 부족관거 검색 등에 초점을 맞추어 진행할 것이다.



(그림 13) 배치 단계

4.1 계획

♣♣ 113 : 요구사항 정의

본 도메인의 사용자 요구분석 결과의 요구사항 정의서(표 4), GIS 기능 분해도(표 5)를 기술하였다. 대장, 도면, 하수관망 및 시설물관리의 도출 가능한 요구사항 중 일부를 수록하였다(29).

4.2 분석

♣♣ 210 : 요구사항 명세서 작성

요구사항 명세서는 기능 분해도의 최하위 기능을 뽑아내어 상세 설명하는 것으로 (표 6)에 나타나 있다. 그리고 요구되는 레이어는 하수시설물도, 도시계획도, 우수배수구역도 및 오수배수구역도가 있다.

♣♣ 224 : 이용사례도 작성

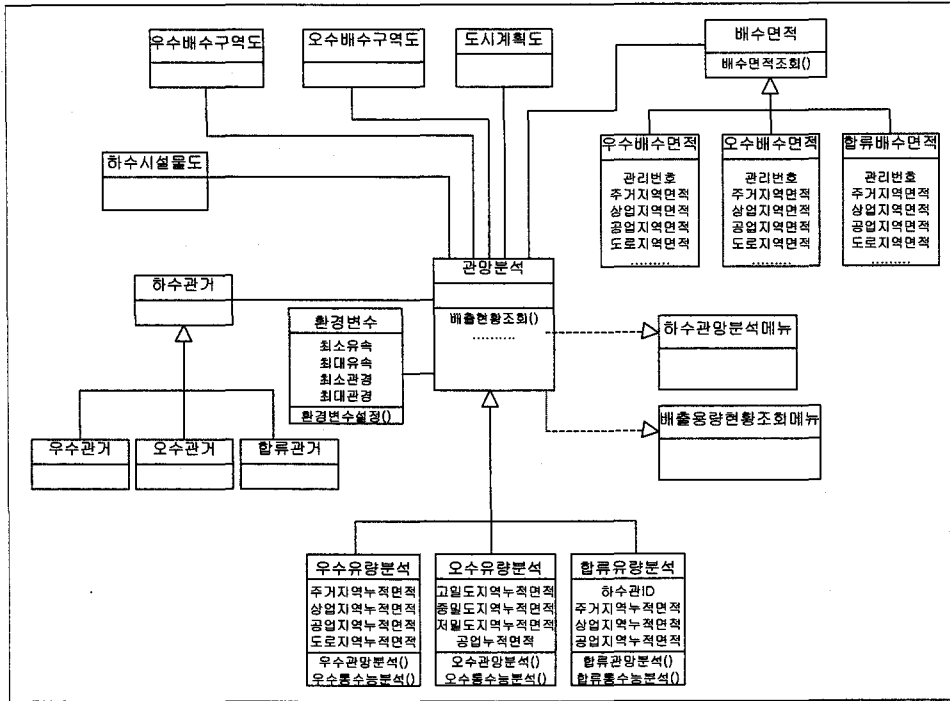
2차 이용사례 목록을 활용하여 작성한 이용사례도가 (그림 14)에 나타나 있다. 단지 이용사례도를 작성하는 가운데 이해 및 도해의 편리를 위하여 이용사례의 추가 및 삭제가 동반되었으며 이것은 단계번호 250의 시스템 모형 상세화에서 반영될 것이다.

〈표 4〉 요구사항정의서

업무구분별	요구사항	처리프로세스
1.대장관리	하수도대장장비 및 하수도대장 현황도 관리	<ul style="list-style-type: none"> 하수도시설 대장 입력 하수도시설 대장 조회
	하수도시설관리를 위한 관리번호 부여	<ul style="list-style-type: none"> 하수시설현황 관리
	하수시설의 배수구역별관리 및 현황집계요청	<ul style="list-style-type: none"> 하수시설현황 관리 하수시설별 통계처리
2.도면관리	도면갱신 및 조회 체계 마련	<ul style="list-style-type: none"> 각종도면제작 및 갱신 도면 검색
	도면의 효율적 관리	<ul style="list-style-type: none"> 도면관리
	다양한 축척 및 크기의 하수도시설현황도 제작	<ul style="list-style-type: none"> 축척별 도면 출력
3. 하수관망 관리	배수불량관에 대한 파악	<ul style="list-style-type: none"> 우수배출용량산정 및 조회 오수배출용량산정 및 조회 합류배출용량조회 및 조회
	실시간 배출용량산정기능 마련	<ul style="list-style-type: none"> 오수배출용량산정 우수배출용량산정 합류배출용량산정
	통수능 부족관거의 현황파악	<ul style="list-style-type: none"> 우·오수통수능 부족관거의 현황조회
	보수관거의 적정관경 결정방안 마련	<ul style="list-style-type: none"> 상동
	하수관 공급에 대한 민원 발생시 적절한 판단근거 제시	<ul style="list-style-type: none"> 하수관거관리
4.시설물관리	부속시설물에 대한 기본 제원 및 이력 관리	<ul style="list-style-type: none"> 시설물 제원 입력 시설물 제원 수정 및 조회 시설물 변경 관리

〈표 5〉 기능분해도(하수관망관리)

Function ID	Function name	설 명
3.	하수관망관리	하수도관망 분석관리
3.1	환경변수설정	관망분석 실행에 앞서 설정할 환경변수 정의
3.1.1	기초환경변수설정	최소, 최대유속 등 기초환경변수 정의
3.1.2	강우강도설정	관거 유형별 강우 특성 상수 정의
3.1.3	용도지역별계수설정	지역별 유출계수 및 인구밀도 등을 정의
3.2	우수배출용량산정	우수의 배출용량 등 여러 정보 검색 및 조회
3.2.1	우수배출용량산정	우수의 관거별 유량 산출
3.2.2	우수배출용량현황조회	우수의 관거별 배출용량의 현황 조회하는 기능
3.2.3	우수통수능부족관거검색	우수통수능 부족관거를 검색하는 기능
3.2.4	우수통수능부족관거 현황조회	우수통수능 부족관거를 현황 조회하는 기능
3.2.5	우수최소유속부족관거 검색	우수 최소유속 부족관거 검색하는 기능
3.2.6	우수최소유속부족관거 현황조회	우수 최소유속 부족관거 현황 조회하는 기능
3.3	오수배출용량산정	오수의 배출용량 등 여러 정보 검색 및 조회
3.3.1 ~ 3.3.6	오수배출용량산정	이하 내용 "우수"와내용 동일
3.4	합류배출용량산정	



〈그림 15〉 개념적 클래스도

♣♣ 231 : 개념적 클래스도 작성

추상적 클래스도가 업무과정 모델링 및 이용사례 분석 과정을 통하여 구체화된 개념적 클래스도가 〈그림 15〉에 기술되어 있다. 이 개념적 클래스도에는 클래스의 속성과 오퍼레이션 정의에 관심을 기울여야 하며 최종적으로 구현 의존적인 최종 클래스도가 만들어지게 된다.

♣♣ 240 : 상호작용 분석 : 순차도 작성

“하수관망분석” 이용사례에 대한 순차도가 〈그림 16〉에 기술되어 있다. 이용사례의 사건(1번에서 9번까지)을 시간 순서에 따라 왼쪽 위에서 아래로 나열되어 있고, 사건들과 관련되어 있는 객체들을 위쪽의 가로축에 정렬되어 있다. 이 다이어그램은 간단하여 UML 표기법을 지원하는 도구인 Rational Rose를 활용하여 도식하였다.

4.3 설계

♣♣ 312 : 컴포넌트 식별

이미 제시한 컴포넌트 식별 절차에 의하여 핵심(기

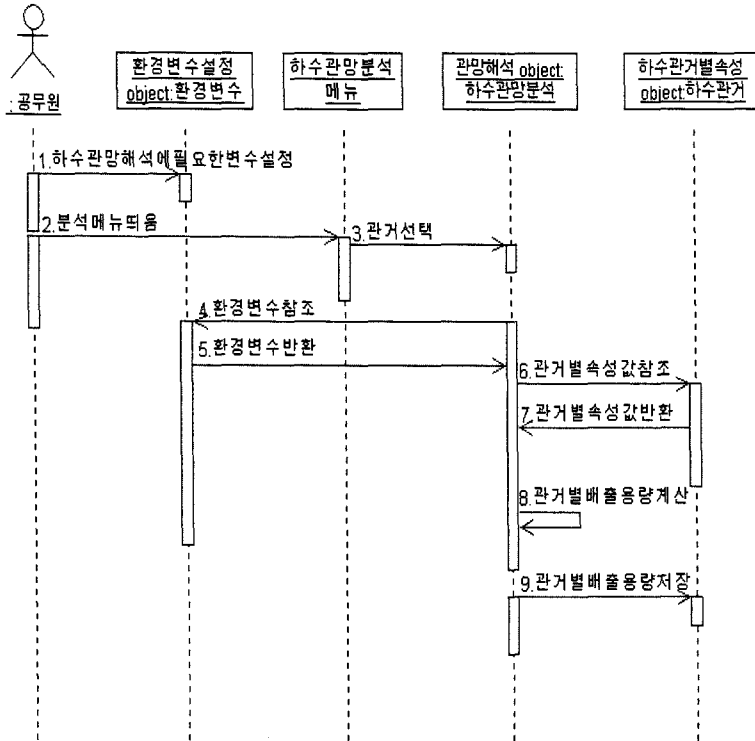
반 및 고유) 컴포넌트에 해당하는 것은 식별되며, 나머지들 재 그룹핑한 결과 각 레이어, 관망분석, 환경변수설정 컴포넌트가 식별되었다. 방법론에서 제시된 것처럼 사용의 독립성과 재사용성을 위하여 하수관거를 컴포넌트로 재식별한다. 이렇게 식별된 컴포넌트들의 관계를 나타낸 초기 컴포넌트도를 작성한다(그림 17).

♣♣ 314 : 컴포넌트 인터페이스 정의

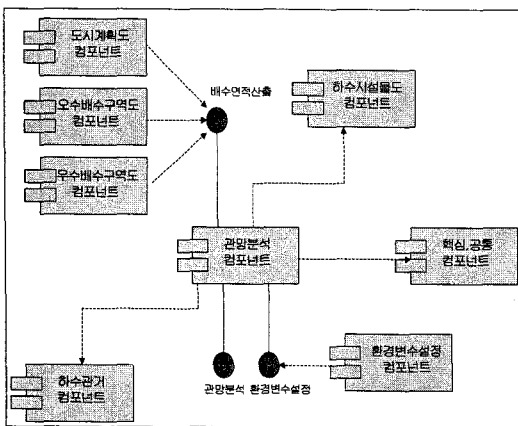
이 작업은 컴포넌트도에서 제시한 인터페이스에 포함되는 오퍼레이션을 정의하여 명세화 하는 단계로 인터페이스 “관망분석”에 대한 오퍼레이션 목록이 〈표 7〉에 나타나 있다.

♣♣ 321 : 컴포넌트별 구현 클래스도 작성

이 작업은 구현을 위한 컴포넌트도를 작성하는 것이다. 컴포넌트를 구성하는 클래스도를 구현 의존적으로 구체화한다. 전술한바와 같이 이미 8개의 컴포넌트를 우리는 추출하였다. 그중 가장 핵심적인 컴포넌트 ‘관망분석’의 구현 클래스도(그림 18)가 나타나 있다 [30].



<그림 16> 이용사례 "하수관망분석"의 순차도



<그림 17> 기초 컴포넌트도

본 사례에서 인터페이스 상세 설계는 마이크로소프트사의 ATL COM을 이용하여 작성하였다(그림 19). ISewer 인터페이스 내에 많은 오퍼레이션(혹은 메소드, 함수)이 포함되어 있다. <<propget>> 속성을 읽는 함수임을 의미하고, HRESULT는 함수 수행 결과 코드를 리턴 한다는 뜻이며, 함수 ActiveRainAnalysisDlg() 앞에 <<propget>>이 없는 것은 속성을 읽을 필요 없이 함수를 수행(우수관망해석 다이어로그 호출)하겠다는 의미를 나타낸다.

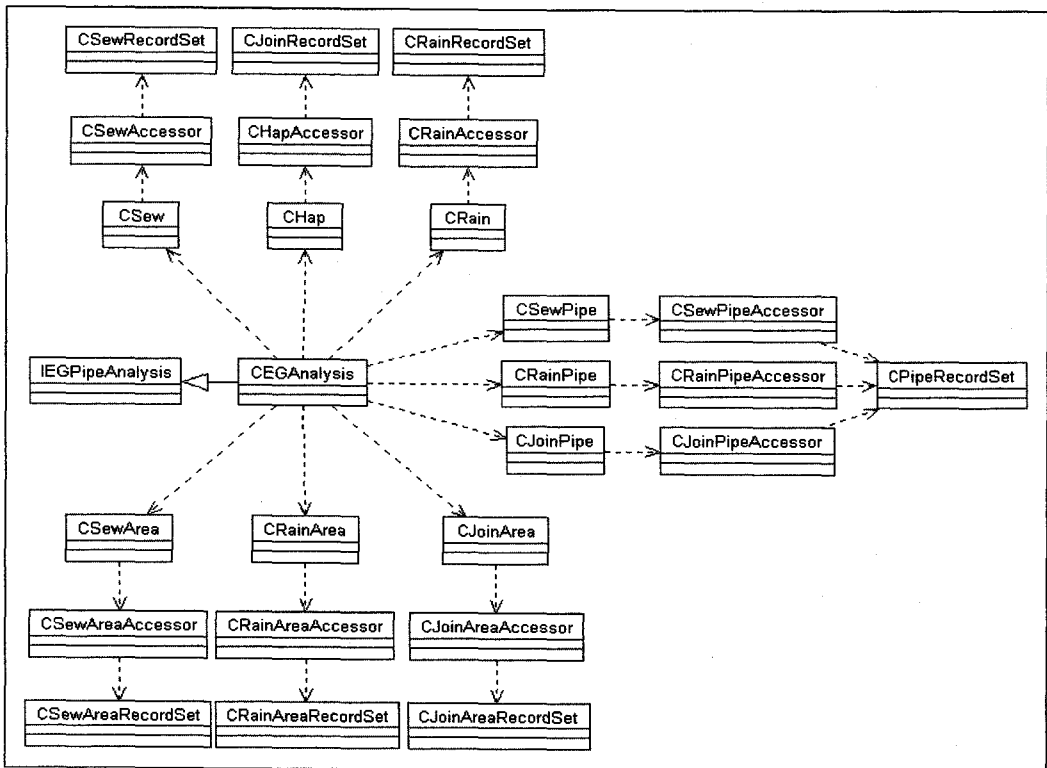
♣♣ 351 : 공간정보 설계

본 사례의 하수관망관리 시스템의 레이어에 해당하는 우수배수구역도, 우수배수구역도, 도시계획도 및 하수시설물도에 대한 도형자료(기하자료)의 추출(표 8) 및 테이블을 정의하는 것으로 관계형 데이터베이스에서는 속성 테이블 설계와 구조가 동일하다. 표에서 보듯이 자료출처가 나타나 있는데, 이들 레이어의 원 자료 소스 및 축척을 기술하였다.

♣♣ 325 : 컴포넌트 인터페이스 상세 설계

〈표 7〉 컴포넌트 인터페이스 목록

인터페이스명	오퍼레이션	설 명
관망분석	오수관망분석()	오수관망분석 다이어로그 호출 및 오수 배출용량 산정
	우수관망분석()	우수관망분석 다이어로그 호출 및 우수 배출용량 산정
	합류관망분석()	합류관망분석 다이어로그 호출 및 합류 배출용량 산정
	오수통수능분석()	오수통수능분석 다이어로그 호출 부분
	우수통수능분석()	우수통수능분석 다이어로그 호출 부분
	합류통수능분석()	합류통수능분석 다이어로그 호출 부분
	데이터베이스경로설정()	데이터베이스 경로 설정부분



〈그림 18〉 컴포넌트 "관망분석"의 구현 클래스도

〈표 8〉 추출한 레이어

Category	내용	자료출처	Coverage명	Feature 클래스
배수구역	오수	하수도계획시설 평면도(축척 1:3,000)	Area	Polygon
	우수	하수도계획시설 평면도(축척 1:3,000)		
도시계획	구역	도시계획도(축척 1:5,000)	공장	Polygon
시설물	하수관거	하수대장도(축척 1:500)	시설물	Line


```

classDiagram
    class ISewer {
        <<propget>> DataBasePath(pVal : BSTR*) : HRESULT
        <<propput>> DataBasePath(pVal : BSTR) : HRESULT
        ActiveEnvSetDlg() : HRESULT
        ActiveJoinAnalysisDlg() : HRESULT
        ActiveRainAnalysisDlg() : HRESULT
        ActiveSewAnalysisDlg() : HRESULT
        ActiveRainDischargeDlg() : HRESULT
        ActiveSewDischargeDlg() : HRESULT
        ActiveJoinDischargeDlg() : HRESULT
        <<propget>> GetFlowShortJoinID(index : long, pVal : BSTR*) : HRESULT
        <<propget>> GetFlowShortSewID(index : long, pVal : BSTR*) : HRESULT
        <<propget>> GetFlowShortRainID(index : long, pVal : BSTR*) : HRESULT
        <<propget>> GetFlowShortJoinNums(pVal : long*) : HRESULT
        <<propget>> GetFlowShortSewNums(pVal : long*) : HRESULT
        <<propget>> GetFlowShortRainNums(pVal : long*) : HRESULT
        <<propget>> GetJoinDischargeID(index : long, pVal : BSTR*) : HRESULT
        <<propget>> GetSewDischargeID(index : long, pVal : BSTR*) : HRESULT
        <<propget>> GetRainDischargeID(index : long, pVal : BSTR*) : HRESULT
        <<propget>> GetJoinDischargeNums(pVal : long*) : HRESULT
        <<propget>> GetSewDischargeNums(pVal : long*) : HRESULT
        <<propget>> GetRainDischargeNums(pVal : long*) : HRESULT
        <<propget>> GetJoinKeyFieldValue(FieldName : BSTR, Key : BSTR, pVal : BSTR*) : HRESULT
        <<propget>> GetRainIndexFieldValue(FieldName : BSTR, index : long, pVal : BSTR*) : HRESULT
        <<propget>> GetSewKeyFieldValue(FieldName : BSTR, Key : BSTR, pVal : BSTR*) : HRESULT
        <<propget>> GetJoinIndexFieldValue(FieldName : BSTR, index : long, pVal : BSTR*) : HRESULT
        <<propget>> GetSewIndexFieldValue(FieldName : BSTR, index : long, pVal : BSTR*) : HRESULT
        <<propget>> GetRainKeyFieldValue(FieldName : BSTR, Key : BSTR, pVal : BSTR*) : HRESULT
    }
    
```

〈그림 19〉 인터페이스 상세 설계

〈표 9〉 오수관망분석결과 테이블

번호	항목정의	항목명	유형	길이
01	오수관거 ID	SEW ID	N	10.4
02	고밀도지역누적면적	ACCU A	N	10.4
03	중밀도지역누적면적	ACCU B	N	10.4
04	저밀도지역누적면적	ACCU C	N	10.4
05	공업누적지역	ACCU D	N	10.4
06	상업누적지역	ACCU E	N	10.4
07	총배수면적	TOT AREA	N	10.4
08	오수발생량	SEW Q	N	5.3
09	지하수발생량	GRD Q	N	5.3
10	총발생유량	TOT Q	N	6.3
11	여유율 고려 최대처리유량	ALL_Q	N	6.3
12	최대처리유량	Q FULL	N	6.3
13	최대유속	V FULL	N	6.3
14	변경관경	CHG DIP	VC	12
15	변경유량	CHG Q	N	6.3
16	변경유속	CHG V	N	6.3

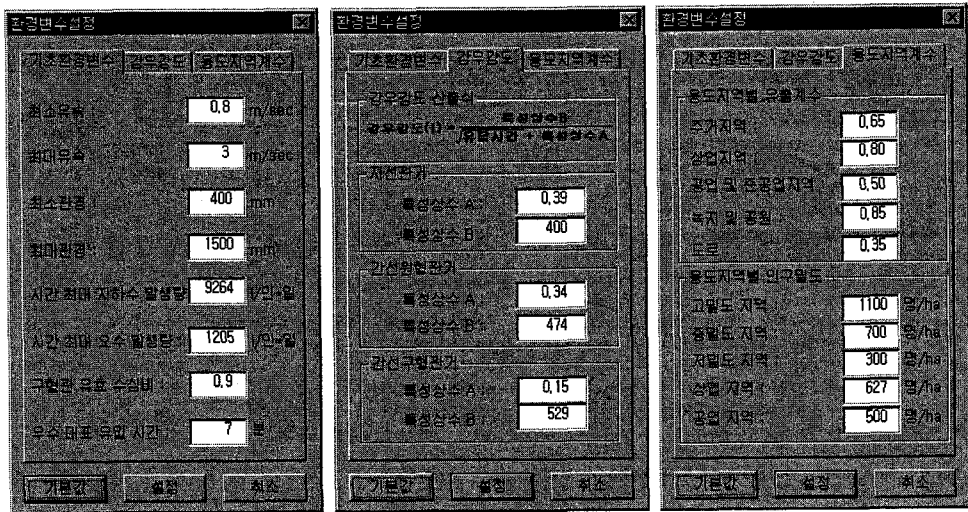
♣♣ 352 : 속성정보 설계

속성정보에 대한 테이블을 정의한다. 포함되는 속성, 속성의 크기, 타입 및 키 등에 대한 설명이 상세하게 기술되어야 한다. 환경변수, 오수·우수·합류배수 면적 테이블 및 관망분석, 오수·우수·합류관망분석결과 테이블 등이 정의되어야 한다. 오수관망분석결과(그림 18의 CSewRecordSet) 테이블이 〈표 9〉에 나타나있다. 기본키(Primary) 값은 오수관거 ID가 된다[30].

4.4 구현

♣♣ 440 : 사용자 인터페이스 구현

사용자 인터페이스 구현 및 프로그램 코딩은 Visual C++을 이용하여 개발한다. 시험 제작한 하수관거관리(하수관망분석) 시스템의 관망분석 컴포넌트인 SewerAnalysis.dll을 비롯한 4개를 이용하여 하수관망분석 시스템이 시험 개발되었다. 〈그림 20〉은 하수도 관망분석의 환경변수 설정을 위한 부분을 구현한 그림이다.



〈그림 20〉 환경변수 설정

5. 분석 및 평가

지금까지 본 연구에서 제안한 ATOM 방법론을 하수시설물관리 시스템에 사례로 적용하였다. 컴포넌트의 식별, 정의 및 구현에 이르는 과정을 살펴보았다.

ATOM은 UML을 활용하여 컴포넌트에 기반을 두고 전사적 응용GIS를 구축하는 소프트웨어 개발 생명주기 전과정을 설명하는 방법론으로서 기존의 방법론과의 비교를 통한 평가가 요망된다.

5.1 사례적용 결과 분석

사례적용 결과 도출된 컴포넌트들은 〈표 10〉과 같은 응용분야에서 재사용 가능할 것으로 파악되어 진다.

5.2 ATOM과 타 방법론과 비교

ATOM을 하나의 방법론으로서 전체를 정량적으로 검증할 수 있는 방법은 없다. 방법론을 구성하는 방법, 절차 및 기법들은 이미 검증되어 있으며, 방법론은 단지 검증된 것을 활용하여 프로세스를 전개하여 나간다. 따라서 방법론에 대한 전체적인 평가는 일반적으로 방법론에서 요구되어지는 사항을 살펴보고, 또한 다른 방법론과의 비교를 통하여 상대적 우수성을 보여주는 방법으로, 이 방법도 매우 경험적이고 전문가적 판단에 의한 것이다.

〈표 10〉 컴포넌트의 재사용 가능한 응용분야

컴포넌트명	향후 재사용 가능한 응용분야
도시계획도	- 도시설계 분야 - 적지분석 분야 - 녹지관리시스템 분야 - 하수관망분석
우수배수구역도	- 하수관망분석(우수관) - 하수관망시설 설계 분야
오수배수구역도	- 하수관망분석(오수관) - 하수관망시설 설계 분야
하수관망분석	- 하수관망시설 설계 및 유지보수 - 침수지역예측 시뮬레이션 분야 - 하수도 관리시스템 분야 - 환경정보 관리시스템 분야
하수시설물도	- 지하시설물 관리시스템 분야 - 하수도 관리시스템 분야 - 도면, 공사, 민원관리 분야 - 환경정보 관리시스템 분야
하수관거	- 지하시설물 관리시스템 분야 - 하수도 관리시스템 분야 - 도면, 공사, 민원관리 분야 - 환경정보 관리시스템 분야
환경변수설정	- 하수관망분석
핵심.공통	- 기존의 컴포넌트를 재사용 한다

5.2.1 방법론으로서의 요구사항 평가

방법론으로서 갖추어야 될 일반적인 요구사항에 대한 ATOM의 독립적인 평가가 〈표 11〉에 서술되었다. 이미 설명한 것처럼 ATOM은 사용자가 용이하게 이 방법의 프로세스만을 준수함으로써 전사적 정보시스템

〈표 11〉 ATOM의 방법론으로서 요구에 대한 평가

방법론 요구사항	반영	비고
사업전략에 따른 업무중심 접근	○	업무지향
개발생명주기 전체 단계 지원	◎	SDLC 전체 단계
다양한 경로 지원	○	RAD 및 컴포넌트 개발 경로만 지원
회사 규모와 문화에 맞도록 커스터마이징 기능	◎	자동적인 커스터마이징
작업을 효율적으로 수행하기 위한 절차, 기법, 산출물 정의	◎	
CASE 도구 지원	○	UML 및 기존 CASE 활용
이해하기 쉬운 용어 및 도형 표시	◎	다이어그램 및 표 최대 활용
ISO 준수를 통한 표준화 강화	○	OMG 및 ISO 12207 준수(SDLC 프로세스)
프로젝트 관리 지원	○	생산성 향상과 품질관리와 연계
품질 관리 활동 지원	○	생산성 향상과 품질관리와 연계
도구에 의한 방법론 조희 및 커스터마이징 지원	●	ATOM만을 지원하는 직접적인 도구 부재
도구에서 문서 템플릿 지원	●	"
개발 경로 편집 기능	○	수작업으로 가능

충분히 반영:◎, 부분 반영:○, 반영 미흡:●

을 구축하는데 역점을 두었다. 따라서, 개발생명주기 전체단계, 작업을 효율적으로 수행하기 위한 절차, 기법, 산출물의 정의, 이해하기 쉬운 용어 및 도형 표시 등의 항목이 제일 많이 반영되었고, ATOM의 전체 프로세스를 지원하는 독립적인 도구가 없음으로 도구에 의한 여러 지원 기능은 반영이 매우 미흡하다 할 수 있다. 향후 연구에서 고려해야 할 부분이다.

5.2.2 컴포넌트 중심 비교

컴포넌트 중심의 비교는 컴포넌트 개념과 컴포넌트 모델링 관점에서 컴포넌트의 식별, 정의, 구현 및 컴포넌트가 가지는 특성을 다루는 방법이다. 즉 컴포넌트 개발이 주목적인 경우에 관심을 가져야 할 부분이다. 그러나 본 ATOM은 컴포넌트에 기반하고 있고, 컴포넌트 추출만을 위한 과정(설계 단계의 상세 설계 활동까지)을 포함하고 있어 컴포넌트 중심 비교가 이루어져야 한다. 이 방법을 위한 비교는 기존의 비교 기준을 도입한다[30]. 이 기준에 따라 ATOM과 타 방법론(CBD96, Catalysis, Fusion, UNIFACE, SCIPIO)을 비교한 결과를 〈표 12〉에 기술하였다. 타 방법론과 비교하여 ATOM은 컴포넌트식별(c22), 공통성식별(c23) 및 가변성식별(c24)은 타 방법론에 비해 우세하게 지원한다. ATOM의 프로세스에서도 나타나 있듯이 기존 방법론에서 한계점으로 나타난 것을 ATOM에 반영한 결과이다.

5.2.3 방법론 중심 비교

방법론 중심 비교는 ATOM과 타 방법론과의 비교가 목적이다. 〈표 14〉의 굵은 선 안의 내용은 COMO 방법론을 제시하였던 연구[18]의 결과이며, 본 비교를 위하여 RAD 지원, GIS 도메인 적용, 문서화 및 프로젝트관리 지원 기준을 추가하였다. 〈표 13〉를 살펴보면 Catalysis와 SCIPIO는 ATOM과 마찬가지로 컴포넌트에 기반을 둔 응용시스템 개발이란 점은 같으나 컴포넌트의 식별, 정의 및 구현 과정이 명확하지가 못하다. 반면 COMO는 타 방법론과 달리 컴포넌트 기반 개발 방법론이 아닌 컴포넌트 자체의 개발에 초점을 둔다. 이런 목적으로 COMO는 컴포넌트 식별, 정의, 명세 및 구현의 과정이 명확하게 제시되고 있다. 그리고 ATOM은 컴포넌트 자체의 개발뿐만 아니라 개발된 컴포넌트를 기반으로 소프트웨어 개발 전체 생명주기를 다루는 방법론이라고 할 수 있다. 무엇보다도 가장 중요한 특징은 GIS 도메인에 적용 가능한 유일한 방법론에 있다.

6. 결론 및 향후 연구과제

본 연구에서 제안한 ATOM의 타 방법론과의 차별화 및 특징은 표 14와 같이 정리될 수 있다. 컴포넌트 개발이 목적인 경우와 전사적 GIS 개발 모두를 만족하고 있으며, 프로젝트 관리의 기능을 강화하였으며, 또한 재사용 가능한 컴포넌트 도출에 일차적인 목적을 두었다. 그리고 사용자가 보다 용이하게 사용할

〈표 12〉 ATOM과 타 방법론의 비교(컴포넌트 중심 비교)

criteria		CBD 방법론					
		CBD96	Catalysis	Fusion	UNIFACE	SCIPIO	ATOM
c1	컴포넌트	0	0	0	0	0	0
c2	하나의 클래스/컴포넌트	0	0	0	0	0	0
c3	복수의 클래스/컴포넌트	0	0	0	0	0	0
c4	하나의 인터페이스/컴포넌트	0	0	0	0	0	0
c5	복수의 인터페이스/컴포넌트	0	0	0	0	0	0
c6	제공되는 인터페이스	0	0	0	0	0	0
c7	요구하는 인터페이스	0	0	0	0	0	0
c8	컴포넌트 구현	0	0	0	0	0	0
c9	컴포넌트 상태 묘사 특성	0	0	0	0	0	0
c10	컴포넌트 행위 묘사 방법	0	0	0	0	0	0
c11	컴포넌트의 워크플로위	0	0	X	X	0	0
c12	속성 커스터마이징	0	0	0	0	0	0
c13	로직 커스터마이징	0	0	0	0	0	0
c14	작업절차 커스터마이징	X	X	X	X	X	X
c15	컴포넌트 객체	0	0	0	0	0	0
c16	복합 컴포넌트	0	0	0	0	0	0
c17	컴포넌트간 의존	0	0	0	0	0	0
c18	컴포넌트 명세	0	0	0	0	0	0
c19	컴포넌트 오퍼레이션 명세	0	0	0	0	0	0
c20	사건	X	0	X	0	X	0
c21	트랜잭션	X	X	X	X	X	X
c22	컴포넌트 식별	부분적	부분적	부분적	부분적	부분적	0
c23	공통성 식별	부분적	부분적	부분적	부분적	부분적	0
c24	가변성 식별	X	부분적	X	X	X	0

수 있도록 프로세스를 전개하고 GIS 도메인이 가지는 여러 특징들을 충분히 반영하였다고 사료된다.

향후에는 다음의 연구가 요망된다. 우선적으로 GIS의 많은 특성을 고려한 방법론으로 발전시키기 위한 연구가 필요하다. 나아가 UML 이외에 본 방법론의 기능들을 수행할 수 있는 자동화 도구의 지원을 확대하고, 아울러 OLE/COM, CORBA, EJB 등 다양한 컴포넌트 표준 모델에 의한 컴포넌트 구현 환경을 구체적으로 지원하는 방법론으로 확대 개발이 필요하다. 둘째, GIS 도메인 내의 여러 세부 도메인별로 방법론에 의한 적용사례를 발굴하여 유사 응용시스템의 개발에 활용하여 개발기간 및 개발비용을 단축하도록 할 것이다. 셋째, ATOM이 지원하는 RAD 및 컴포넌트 개발 경로(CD) 이외에 조직의 환경을 고려하여

ATOM이 기본적으로 가지고 있는 순차적 프로세스 모델을 바탕으로 프로토타입 모델, 진화적 프로세스 모델, 반복적 프로세스 모델, 점진적 프로세스 모델 등 여러 개발 경로를 구체적으로 만들어 필요한 경로에 따라 효율적으로 프로젝트가 수행되도록 지원하여야 한다. 넷째, ISO와 OMG 등의 컴포넌트 관련 표준화 기관과 여러 벤더들의 컴포넌트 기술 표준, 특히 지리정보와 관련한 ISO/TC 211 및 OMG의 OGC에서 제안하는 각종 표준을 주기적으로 반영하여 가장 최신의 표준화된 방법론이 되도록 하여야 할 것이다. 다섯째, ATOM이 제시하는 컴포넌트 분류 구조인 핵심, 공통, 레이어, GIS 비즈니스 및 응용 컴포넌트를 향후 프로젝트 및 사례연구 등을 통하여 응용분야별 도메인 분석 및 컴포넌트 분류 방법을 개발하여 컴포

〈표 13〉 ATOM과 타 방법론과 비교(방법론 중심 비교)

비교기준 \ 방법론	Catalysis	SCIPIO	COMO	ATOM
개발 프로세스 정의	가	가	가	가
작업 흐름의 정의	가	가	가	가
지원범위	컴포넌트 기반 S/W 개발	컴포넌트 기반 S/W 개발	컴포넌트 개발	컴포넌트 기반 S/W 개발
구체적인 모델링 지침의 제안	부분적	부분적	가	가
커스터마이징 부분의 제시	부분적	부	가	가
기법	부분적	부분적	가	가
적용성	어려움	어려움	용이함	용이함
성공적수행	중간	중간	높음	높음
완전성	높음	중간	중간	높음
RAD 지원	중간	중간	높음	높음
GIS 도메인의 적용	부	부	부	가
산출물 작성지침	중간	중간	중간	높음
프로젝트관리 부분 지원	중간	높음	낮음	중간

〈표 14〉 ATOM의 차별화 및 특징

번호	차별화 영역	설 명	비 고
1	개발경로	컴포넌트 기반의 응용GIS 개발(RAD)과 컴포넌트 개발(CD)의 경로를 지원한다. ATOM의 기본 목표가 전사적 응용GIS의 RAD를 추구하고 있다	전체 프로세스가 RAD를 위한 것이다
2	프로젝트 관리	개발 프로세스 중 계획 및 배치는 프로젝트 관리 부분이 가장 많이 적용되는 단계이다. ATOM은 계획 단계의 상세한 부분을 지원하며, 배치 단계의 시스템 개발 완료 후 소프트웨어 측정 등은 ATOM만이 제공하는 것으로 사료된다	생산성과 품질 측정 방법이 강조되었다
3	재사용 가능성	ATOM은 최대한으로 단순 기능과 최소의 크기를 가진 컴포넌트를 개발하여 재사용 가능성을 극대화할 수 있도록 컴포넌트 식별 지침을 제시하였다.	컴포넌트의 크기를 최소화하여 재사용 확률을 증대시켰다
4	사용자 중심의 프로세스 전개	기존의 많은 방법론들이 전략적, 홍보적, 선언적 측면을 강조하여 사용자가 실제로 사용하기에 어려움이 있어 이를 극복하는 메뉴얼적이고 교과서적으로 프로세스를 전개하였다.	사용자의 이해도를 높여 수행이 용이도록 하였다
5	GIS 도메인의 적용	GIS 도메인의 유일한 특징을 반영한 방법론이다. ① 컴포넌트 분류 구조, 컴포넌트 아키텍처, 컴포넌트 식별 및 정의 : 오직 GIS 도메인을 위한 것으로, ATOM의 컴포넌트의 분류 구조와 아키텍처, 컴포넌트 식별 방법 등은 다른 도메인에서의 활용이 불가능하다. ② 공간 데이터베이스 설계 및 구현 : 공간자료(기하 및 위상정보)의 데이터베이스 설계 및 구현. ③ 레이어 중심의 공간 모델링 : 공간 데이터 모델링을 레이어 단위별로 수행하여 공간 정보와 속성 정보를 설계 및 구현한다. ④ 요구사항 정의 : 공간 정보(위상정보, 화면 제어 정보)에 대한 요구사항이 추가된다. ⑤ 인터페이스 정의 : OGC의 구현 사양의 인터페이스를 정의하는 표준을 준수한다.	GIS 도메인만이 가지고 있는 특징 때문에 다른 도메인과 구별되어 전개되는 작업과 절차들을 포함하는 활동과 작업들이다

먼트 분류 구조와 아키텍처를 더욱 구체화시켜야 한다.

참고문헌

[1] Component Group, "Component-based Development-an Overview", White Paper, Component Group, at URL : www.componentgroup.com/whitepapers/overview.html, 2000

[2] 윤태권, 정한일, "컴포넌트 소프트웨어 산업 동향", 2000년 정보처리학회 논문집, 2000, pp 79-84.

[3] 이종훈, "개방형 컴포넌트 기술관점에서의 GIS산업 활성화 전략", 2000년 지식기반사회를 대비한 국가GIS 정책 및 기술방향에 관한 국제세미나, 2000, pp 175-190.

[4] OpenGIS Consortium, Inc., "The Open GIS Abstract Specification Model(3rd ver.)", 1998.

[5] 김은형, "GIS 선진기술모니터링 및 기술 확산", News Letter 제 15 호, 1998.

[6] 이강준, 홍동숙, 박지웅, 한기준, "컴포넌트 기반 지리정보 시스템을 위한 엔터프라이즈 서버의 설계 및 구현", 개방형GIS연구회 논문지, 1999, pp 5-20.

[7] 최상길, 이진규, 이종원, 김장수, "개방형 GIS 기반 인터넷 공간데이터 서비스 컴포넌트의 설계 및 구현", 개방형GIS 연구회 논문지, 1999, pp 21-31.

[8] 이대희, "개방형 GIS의 단순개체 사양을 이용한 공간 기준 좌표계 컴포넌트의 개발", 개방형 GIS 연구회 논문지, 2000, pp 57-62.

[9] 김계현, 이우철, 김준철, "지자체 업무지원을 위한 하수도 관망해석 컴포넌트 설계", 개방형 GIS연구회지, 제2권 제1호, 2000, pp 27-36.

[10] 김은형, "GIS 개발방법론 표준화", 2001년도 정보통신표준위원회 연구발표회, 2001, pp 83-125.

[11] New York State Local Government GIS Demonstration Project, "GIS Development Guide", New York State Archives and Records Administration Internet Document, 1993

[12] ESRI, "Managing a GIS", ESRI press, 1996.

[13] 김은형, "GIS 특성을 고려한 개발방법론 연구", 2000 개방형 지리정보시스템 학술회의 논문집", 제 3권 2호, 2000, pp 155-177.

[14] 이주현, "전략 정보 시스템 구축론", 일본전략정보시스템연구회, 이주현 번역, 1993.

[15] Carnegie Mellon University Software Engineering Institute, "The Capability Maturity Model: Guidelines for Improving the Software Process", Mark C.Paulk etc.,1995

[16] <http://www.sqi.gu.edu.au/spice/>

[17] PLATINUM technology, "Component_Based Development With Catalysis and UML", Component Technology Center.

[18] S.D. Lee, Y.J. Yang, E.S. Cho and S.D. Kim, "COMO: A UML-Based Component Development Methodology", Asia-Pacific Software Engineering Conference(ASPEC '99) Japan, DEC. 7-10, 1999.

[19] Short K., "Component Based Development and Object Modeling", Sterling Software, 1997.

[20] Veryad R., "SCIPIO:Aims, Principles and Structure," SCIPIO Consortium, April 1998.

[21] HP Company, "Engineering Process Summary: Fusion 2.0", HP Company, January 1998.

[22] Compuware Corp., UNIFACE Development Methodology V7.2, COMPUWARE Corp., 1998.

[23] Princeton Softech, "Component Based Development - A Roadmap to eBusiness Success, White Paper, Princeton Softech, Jan., 10, 2000, <http://www.princetonsofttech.com/index>

[24] 조혜경, "도시정보시스템을 위한 컴포넌트기반 소프트웨어개발 프로세서", 전북대학교 대학원 전산통계학과 박사학위 논문, 2002.

[25] 박태욱, "소프트웨어 개발 프로젝트 관리의 성공과 실패 - 비즈니스 시스템 개발 사례를 중심으로", 제6회 한국프로젝트관리기술회 정기심포지엄 논문집, 1996.

[26] 시스템공학연구소 정보기술교육센터, "프로젝트 관리", 프로젝트관리 교재, 1996.

[27] G. Booch, J. Rumbaugh and I. Jacobson,

- "The Unified Modeling Language User Guide", Addison Wesley, 1999.
- [28] 김수동, "Quick Overview of FOCUS Methodology", 발표자료, 2000.
- [29] 김계현, 김준철, "GIS 기반의 하수도 관망해석을 통한 침수피해지역 예측에 관한 연구", 한국GIS 학회지, 제9권 제3호, 2001, pp 493-507.
- [30] 이우철, "GIS를 위한 하수도 관망분석 컴포넌트 설계에 관한 연구", 인하대학교 대학원 지리정보 공학과 석사학위논문, 2001.
- [31] E.S. Cho, S.D. Kim and S.Y. Rhew, "A Comparative Study of CBD Methodologies", Workshop on Software Architecture and Components(WSAC) '99' Japan Dec. 7, 1999.



박 태 옥

1981년 부산대학교 지구과학과 졸업(이학사)
 1985년 부산대학교 지구과학과 졸업(이학석사)
 2002년 인하대학교 지리정보공학과 졸업(공학박사)
 1982년~1996년 한국전자통신연구원 선임연구원

1997년~현재 한국정보통신대학교 정보통신교육원 GIS팀장

관심분야: UIS, 개발방법론, 컴포넌트GIS, Mobile GIS, 공간데이터베이스 등



김 계 현

1982년 한양대학교 자원공학과 졸업(공학사)
 1989년 아리조나 주립대 수문학 과 졸업(공학석사)
 1993년 위스콘신주립대 토목환경 공학과 졸업(공학박사)
 1995년~인하대학교 지리정보 공학과 부교수

관심분야: 환경, 수자원, 상하수도 및 지하시설물 관리분야의 GIS활용, 주제도제작 및 GIS 표준화 등