

UML과 RUP를 이용한 UIS 컴포넌트 모델링

(Modeling of UIS Components using UML and RUP)

조 혜 경 [†] 유 철 중 ^{**} 장 옥 배 ^{**} 김 영 섭 ^{***}
 (Hye-Gyung Cho) (Cheol-Jung Yoo) (Ok-Bae Chang) (Young-Sup Kim)

요약 국내의 각 지방자치 단체들은 그들의 업무를 전산화하기 위해 현재 다양한 UIS(Urban Information System)들을 개발하고 있다. 그러나, 그들은 기존에 개발된 다른 지방자치 단체들의 UIS들을 재사용하지 않고 있어 그들의 UIS 개발에 너무 많은 비용을 사용하고 있다. 이것은 중앙정부가 중복적인 투자를 하게하는 원인이 된다. 몇몇 지방 자치단체의 업무는 다른 지방자치 단체들의 업무와 매우 유사하다. 이 논문은 그 중복투자 문제에 대한 해결책을 제안한다. 그 문제 해결을 위해 지방자치 단체들이 Unified Modeling Language과 컴퍼넌트 기반 개발 프로세스를 이용하여 한 도시의 업무를 모델링할 것을 제시한다. 그들은 또한 기존의 모델로부터 UIS를 개발할 수 있고 다른 지방자치 단체들이 그 개발된 UIS를 재사용할 수 있다. 이 연구에서는 한 예로 P시의 상수도 업무를 사용하며 컴포넌트 기반 개발 프로세스로 Rational Unified Process를 이용한다. 본 논문은 상수도 업무에 대한 컴포넌트 기반 개발 프로세스의 적용과 재사용 가능한 업무모델/분석모델/설계모델의 구축 및 ActiveX 컨트롤 형태의 소프트웨어 컴포넌트 개발 방법을 기술한다.

Abstract Some local governments in Korea are currently developing various Urban Information Systems (UIS) to computerize their work. But they have spent too much money on their own UIS development because they have not reused the existing UIS of others. It has caused the central government to invest money repeatedly. The work of some local governments is generally very similar to those of others. The purpose of this paper is to suggest a solution to the reinvestment problem. To solve the problem, this paper suggests that they could model a city's business using Unified Modeling Language and Component-Based Development Process (CBD). They could also develop a UIS from the existing models, and then other municipalities can reuse the developed UIS. This paper illustrates the water supply business of P City in Korea and Rational Unified Process as a CBD. This paper also describes methods to apply CBD, to establish a reusable business model, analysis model, and design model, and to develop the software components of ActiveX Controls form for the water supply business.

1. 서론

GIS(Geographic Information System)는 정보화사회에서 중요한 위치를 차지하며 지리정보와 테이블형태의

데이터를 시각화시켜 사람들의 이해를 쉽게 한다[1]. 즉, GIS는 지리형태에 관한 데이터를 수집 저장하고 이를 분석 가공하여 각종 지리 관련 응용분야에 활용하도록 돕고 이미지 처리, 데이터베이스, 인공 지능 등 여러 전산학 분야를 총 망라한 포괄적인 시스템이다.

기존 GIS 분야의 대표적인 문제점은 표준화 부족과 타 회사들의 GIS 제품과의 상호운용성의 결여이다. 즉, 회사마다 자신들의 데이터 포맷을 정의해 그들의 GIS 애플리케이션에서 사용하고 있어 다른 회사 제품의 GIS 애플리케이션과 상호운용이 어렵다. 기존에 개발된 국내의 GIS관련 응용 프로그램들은 현재 독립적인 상태에서는 잘 운영되고 있으나 새로운 GIS 응용 프로그램으로

· 본 연구는 한국학술진흥재단의 "도시, 환경정보체계 구축을 위한 GIS 활용방안 연구" 과제에서 지원 받았음.

† 비회원: 전북대학교 컴퓨터과학과
hgcho@han.ac.kr

** 중신회원: 전북대학교 컴퓨터과학과 교수
cjyoo@moak.chonbuk.ac.kr
okjang@moak.chonbuk.ac.kr

*** 중신회원: 한동대학교 전산전자공학부 교수
yskim@han.ac.kr

논문접수: 1999년 12월 28일

심사완료: 2000년 4월 14일

의 확장을 위해서 단일 워크플로우 내에 각 작업단계별로 프로그램들을 통합할 경우 상호운용이 불가능하다. 이를 극복하기 위한 대안으로서 GIS 자료 교환표준인 SDTS(Standard Data Transfer Specification)[2], OpenGIS 추상과 구현 스펙을 개발하는 OGC(Open GIS Consortium) 프로젝트[3], 지리정보와 관련된 세계 표준의 제정 및 공표를 위한 ISO/TC211[4] 등의 연구가 전세계적으로 진행되고 있다.

GIS는 각 관공서의 행정업무 뿐만 아니라 군사적 목적 등 현재 그 활용분야가 다양하게 확대되고 있다. 대구 가스관 폭발사고 등과 같은 여러 인재(人災)로 관심이 집중되어온 GIS 분야에 정부는 현재 막대한 예산 투자를 진행 및 계획하고 있다. 특히, 정부는 국가 GIS(NGIS)사업의 일환으로 도시계획, 상수도, 하수도, 지적, 도로 등 각 지방자치 단체의 업무 전산화를 적극 추진 중에 있다. 이와 같은 지방자치 단체 업무를 전산화하는 GIS 응용 분야를 UIS(Urban Information System)라 부른다. UIS 개발을 위해서는 ArcInfo, ArcView, MapObjects, SDE와 같은 필수적인 GIS 기본 S/W 구입이 필요하며 또한 UIS 구축시 인건비를 포함한 시스템 개발 비용이 요구된다. 그 비용은 국내의 경우 전체 78개 도시(7대 광역시와 71개 중소도시)에 대해 대략 2,000억원 정도가 소요되고 있다[5]. 이와 같이, UIS는 우리 나라 GIS 시장에서 가장 큰 경제적인 비중을 차지한다.

현재 국내 UIS 부분의 가장 큰 문제점은 새로운 UIS 응용 시스템을 개발하려 할 때 기존에 개발된 다른 지방자치 단체의 UIS 응용 시스템을 재사용하지 못한다는 점이다. 국내 지방자치 단체들의 업무는 매우 유사한 형태를 갖는다. 예를 들어, 지적업무인 토지이용계획확인원 발급과 같은 업무는 조사결과 여러 지방자치 단체에서 거의 유사하게 행해졌다. 지방자치 단체 업무 시스템을 컴포넌트 형태로 만들어 재사용 한다면 국가 전체의 예산을 상당히 절약할 것이다

한편, 최근에 다양한 실세계의 비즈니스를 컴포넌트화하려는 시도가 상당히 이루어지고 있다[6]. GIS 분야도 마찬가지로 컴포넌트 개념을 이용한 개발법이 현재 조성되어 다양한 연구기관에서 GIS 부분 컴포넌트 개발을 프로젝트로 수행 중에 있다[7]. 컴포넌트 제작에 필수적인 조건이 되고 있는 UML(Unified Modeling Language)과 컴포넌트 기반 개발 프로세스에 대한 여러 연구들이 소프트웨어공학 분야에 발표되고 있으며, GIS 분야에서도 UML를 OGC의 OpenGIS 스펙 개발과 ISO/TC211 모든 표준제정 작업에 현재 사용하고 있다.

본 연구에서는 각 지방자치 단체의 UIS 개발에 대한 중복 투자를 막기 위해 상수도업무를 예로 들어 UIS 응용 컴포넌트의 재사용 측면을 강조한다. 이를 위해 P시를 대상으로 상수도 업무를 분석하여 RUP(Rational Unified Process) 컴포넌트 기반 개발 프로세스에 따라 UML을 사용해 상수도 시스템을 모델링하였다. 개발 결과는 3개의 ActiveX 컨트롤 형태이므로 각 지방자치 단체에서 상수도 UIS 시스템을 개발할 경우 해당 컨트롤의 재사용이 가능하고 컨트롤뿐만 아니라 모델링된 결과인 모델의 재사용도 가능할 것이다. 또한, 국내 어느 지방자치 단체에서 P시와 다른 특수한 형태의 업무가 발생하면 연구 개발된 컴포넌트를 확장하여 사용할 수 있다. 상수도 부문의 업무는 상당한 많은 세부 업무를 가지고 있기 때문에 본 논문에서는 상수도의 누수 부분에 중점을 두어 연구를 수행하고 누수업무는 공사 업무와 도면관리 기능을 반드시 요구하므로 이 두 가지를 연구에 일부 추가하였다.

본 연구에서는 컴포넌트 제작을 위한 개발 프로세스로 Rational사에서 배포한 RUP 5.0을 수용하여 산출물을 생산하였다. RUP는 여러 가지 워크플로우에서 다양한 산출물들을 제시하고 있으나 실제 소프트웨어 개발시 RUP의 전체 산출물들을 작성하는 것은 개발 프로젝트 크기에 의존적이므로 본 연구에서는 UIS 컴포넌트 개발을 위해 반드시 필요한 RUP 워크플로우 내의 산출물들을 작성하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 UIS 컴포넌트 개발을 위해 필요한 요소 기술들로 UML과 세 가지 컴포넌트 기반 개발 프로세스에 대해 설명한다. 3장에서는 UML과 RUP 적용 방법에 대해 언급하고 4장에서는 3장에서 얻어진 결과를 적용하여 직접 P시의 상수도 시스템을 모델링한다. 5장에서는 본 연구에 결과 의의 및 토의 사항에 대해 언급하고 최종적으로 6장에서는 결론 및 향후 연구 방향에 관해 기술한다.

2. 컴포넌트 모델링 방법

이 장에서는 OMG 표준 모델링 언어인 UML과 Rational사의 RUP 프로세스 그리고 RUP와 비교될 수 있는 기타 두 개의 프로세스에 대해 기술한다.

2.1 UML

UML은 1994년 Rational Software Corporation에서 Grady Booch와 James Rumbaugh가 Booch method와 OMT-2 method를 통합하여 새로운 Unified Method를 만들려는 시도에서 비롯되었다[8]. 1995년 OOSE/

Objectory의 Ivar Jacobson이 합류하여 1997년 UML Version 1.0을 발표하게 되었고 UML은 이들 세 사람 뿐만 아니라 David Harel(State Chart), Fusion, Gamma-Helm-Johnson-Vlissides 등의 연구에서 많은 새로운 개념을 도입하였다[9].

1997년 후반부에 OMG는 UML을 표준 모델링 언어로 제정하였다. UML은 단지 표기법에 대한 표준안이다. UML은 단순한 모델링 언어로서 소프트웨어 개발을 위해 RUP와 같은 특정한 분석/설계 지원 프로세스를 필요로 한다. UML은 뷰(view), 다이어그램(diagrams), 모델 요소(model elements), 일반 메커니즘(general mechanisms) 측면에서 고려될 수 있다. 뷰는 모델링될 시스템의 서로 다른 양상을 보여주고 다이어그램은 어느 한 뷰의 내용을 기술하는 그래프이며 모델 요소는 다이어그램 내에 사용되는 개념들로 클래스, 객체, 메시지, 관계성과 같은 것을 의미한다. 일반 메커니즘은 다이어그램에 노트(notes), 명세서(specifications)와 같은 부수적인 정보를 제공하는 것이다.

뷰에는 사용 사례 뷰(use-case view), 논리적 뷰(logical view), 컴포넌트 뷰(component view), 동시성 뷰(concurrency view), 배치 뷰(deployment view)가 있다. 여기서 사용 사례 뷰는 시스템의 기능성(functionality)을 보여주는 것이고 논리적 뷰는 그 기능성이 어떻게 디자인되는 가를 정적 구조(static structure)와 동적 행위(dynamic behavior)를 이용하여 보여준다. 컴포넌트 뷰는 코드 컴포넌트의 조직을 보여주며 동시성 뷰는 시스템내의 동시성을 배치 뷰는 시스템의 물리적인 구조의 배치를 보여주는 뷰이다. 각각의 뷰는 또한 다양한 다이어그램들을 포함하고 있다. 그러한 다이어그램의 종류로는 사용 사례 다이어그램, 클래스 다이어그램, 객체 다이어그램, 상태 다이어그램, 순서 다이어그램, 협동 다이어그램, 활동 다이어그램, 컴포넌트 다이어그램, 배치 다이어그램들이 있다.

사용 사례 다이어그램은 많은 외부의 액터와 그 시스템이 제공하는 사용 사례들과 연결을 보여주는 것이다. 클래스 다이어그램은 시스템 내의 클래스들의 정적 구조를 보여주며 객체 다이어그램은 클래스 다이어그램의 변형으로서 클래스 다이어그램과 동일한 표기법을 사용하고 클래스의 많은 객체 인스턴스들을 나타낸다. 상태 다이어그램은 클래스의 객체가 가지고 있는 모든 가능한 상태들을 보여주며 순서 다이어그램은 많은 객체들 간의 상호작용을 보여주고 객체들간에 보내진 메시지의 순서를 나타낸다. 협동 다이어그램은 메시지 교환을 보

여주고 객체들과 객체들간의 관계를 보여준다. 활동 다이어그램은 하나의 연산(operation)에서 수행될 활동들을 기술하기 위해 사용되며 컴포넌트 다이어그램은 코드의 물리적인 구조를 보여주고 배치 다이어그램은 그 시스템에서 하드웨어와 소프트웨어의 물리적인 아키텍처를 보여준다[9].

2.2 UML 적용 프로세스 비교

2.2.1 Craig Larman의 프로세스

Craig Larman은 "Applying UML, Patterns, and Object-Oriented Analysis and Design"에서 UML과 패턴을 이용한 새로운 프로세스를 정의하였다. 이 프로세스는 별도의 이름이 정의되지 않았으므로 본 연구에서는 'Larman의 프로세스'라 부르기로 한다. 이 프로세스는 최상위 레벨 단계(macro-level steps)로서 계획과 정교화(plan and elaborate), 구축(build), 배치(deploy)를 소개하였다[10].

계획과 정교화 단계는 요구명세서를 작성하고 프로젝트 계획을 세우고 용어 해설 작성을 시작하며 사용 사례와 사용 사례 다이어그램과 프로토타입을 구축 및 작성하며 초기의 개념적 클래스 다이어그램을 작성한다. 구축 단계는 분석과 설계에 대한 것으로서 여러 개의 반복 개발 사이클들을 가지고 있다. 하나의 반복 개발 사이클은 계획의 정제(plan refine), 동기화(synchronization), 산출물, 분석, 설계, 구축, 테스트의 세부 단계로 구성된다. 구축 단계는 핵심적이고 가장 중요한 최상위 레벨 단계이다. 이 단계에서 이루어지는 작업은 사용 사례와 사용 사례 다이어그램의 상세화, 시스템 순서 다이어그램의 작성, 연산 계약(operation contracts)과 상태 다이어그램의 정의, 협동 다이어그램과 같은 상호작용 다이어그램의 정의, 상세한 클래스 다이어그램 정의 등 중요한 기능을 포함하고 있다.

'Larman의 프로세스'는 반복 개발 사이클로서 구축 단계를 여러 번 거쳐 각 개발 사이클에 사용 사례를 할당하는 방식을 취하고 있다. 그러나, 이 프로세스는 시스템 순서 다이어그램과 연산계약과 같은 새로운 개념을 제시하여 각 객체에 책임(responsibility)을 할당하는 방식을 취하여 다른 소프트웨어 개발 방법론에 익숙한 사람들에게 적용에 대한 어려움을 가중시킨다.

2.2.2 국내 개발 프로세스

그 동안 국내에서는 UML지원 프로세스에 대한 연구가 전혀 없다가 99년 상반기에 하나의 프로세스가 제시되었다[11]. "UML을 기반으로 한 실무 중심의 객체지향 방법론"의 프로세스는 이전 'Larman 프로세스'의 축소판이라 할 수 있다. "UML을 기반으로 한 실무 중심

의 객체지향 방법론”의 프로세스는 계획단계, 분석단계, 설계단계, 구현단계, 시험단계, 보급단계로 개발 프로세스를 정의하고 있으며 이를 ‘Larman의 프로세스’와 비교할 때 계획과 정교화는 계획단계, 구축은 분석단계, 설계단계, 구현단계, 시험단계를 의미하며 배치는 보급단계로 표현되고 있다. 또한, ‘Larman의 프로세스’에서는 많은 양의 산출물을 제시하고 있으나 “UML을 기반으로 한 실무 중심의 객체지향 방법론”의 프로세스는 ‘Larman의 프로세스’ 산출물들을 상당히 제거하고 있다. 즉, 새로운 프로세스 정의는 각 단계에서 생성되어야 할 산출물들을 명확히 제시해야 하나 “UML을 기반으로 한 실무 중심의 객체지향 방법론”은 이에 관한 상세한 언급이 없다.

2.2.3 RUP

RUP는 소프트웨어공학 프로세스로서 한 개발 조직 내에 임무와 책임을 할당하는 것에 대한 방법이다. RUP은 UML을 발표한 Rational Software사가 개발하고 유지보수한다. RUP는 Rational Objectory Process를 계승한 것이며 Rational Objectory Process는 Rational Approach와 Objectory Process의 통합 결과이다. RUP는 반복적인 소프트웨어 개발, 요구명세서 관리, 컴포넌트 기반 아키텍처 사용, 소프트웨어 모델의 시각화, 소프트웨어 품질 검증, 소프트웨어 변화 제어 등과 같은 최상의 6가지 기능을 제공한다[12].

RUP는 작업자(worker), 활동(activity), 산출물(artifact), 워크플로우(workflow)의 4개 주요 모델링 요소를 가지고 있다. 작업자는 시스템 분석가, 설계자, 테스트 설계자들을 의미하며 활동은 ‘사용 사례와 액터의 발견’과 같은 것을 의미한다. 산출물은 개발 과정내의 생산물로서 모델 요소(model element), 문서, 소스 코드 등을 의미하며 워크플로우는 주요한 가치의 결과를 생산하는 활동들의 순서를 의미한다. RUP의 워크플로우는 프로젝트 관리 워크플로우(project management workflow), 업무 모델링 워크플로우(business modeling workflow), 요구분석 워크플로우(requirement workflow), 분석 및 설계 워크플로우(analysis design workflow), 구현 워크플로우(implementation workflow), 테스트 워크플로우(test workflow) 등 총 9개로 구성된다. RUP는 도입(inception), 정교화(elaboration), 구축(construction), 이행(transition)의 4단계로 구성되며 반복적이고 점차적인 개발을 지원한다[13]. 그러나, 이러한 장점에 비해 RUP 개발 프로세스는 상세한 설명과 응용을 위한 레퍼런스가 부족하여 컴포넌트 제작을 위한 실제 현장 적용에 어려움이 존재한다.

‘Larman의 프로세스’와 ‘UML을 기반으로 한 실무 중심의 객체지향 방법론의 프로세스’는 초기 요구분석을 한번만 수행하였으나 사용자의 요구 사항은 기술의 발달에 따라 조직의 변화에 따라 시간의 흐름에 따라 변화될 수 있는 여지가 크기 때문에 위의 두 가지 형태의 프로세스 정의는 바람직하지 않다. 특히, 프로젝트의 규모가 크고 프로젝트가 장기간에 걸쳐 수행되는 경우와 시시각각 변하는 많은 사용자 요구를 반영해야 할 경우 이 두 가지 프로세스는 적용성이 떨어진다. RUP는 이전의 6가지 최상의 기능들을 제시하여 이와 같은 문제점들을 해결하고 있다.

본 연구에서는 프로젝트 산출물들을 한정하지 않고 사용자의 요구사항 변화를 신속히 최대한 수용하는 개발 프로세스인 RUP를 P시 상수도업무 UIS 응용 컴포넌트 제작에 사용하며, COM 기반 ActiveX 컨트롤 형태의 UIS 응용 컴포넌트 모델링과 시스템 개발을 수행한다.

3. UML과 RUP 적용 방법

RUP의 프로세스에 따라 소프트웨어 개발을 수행하기 위해서는 RUP에 대해 좀 더 상세히 알아보고 RUP 각 프로세스 단계별로 생성되는 산출물에 초점이 맞추어져야 한다. 이 장에서는 이와 같은 내용을 기술한다.

3.1 RUP의 워크플로우

그림 1은 RUP 워크플로우에 대한 전반적인 내용을 표시한다. 그림 1에서 수평축은 프로세스의 동적인 측면으로 단계(phases), 반복(iteration), 이정표(milestones)로 표현되며 수직축은 프로세스의 정적인 측면을 나타내고 프로세스 컴포넌트(process component), 활동, 워크플로우, 산출물, 작업자로 표현된다[14].

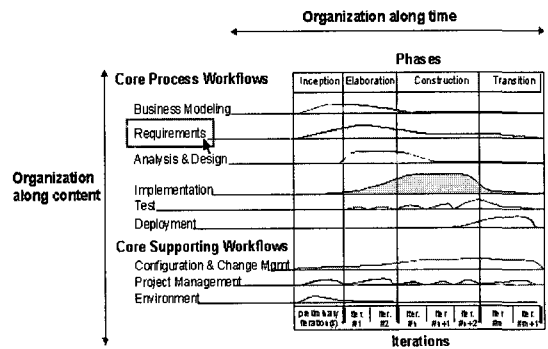


그림 1 RUP의 9가지 프로세스 워크플로우

RUP의 도입, 정교화, 구축, 이행의 네 단계 가운데 도입은 최종 프로젝트 비전과 그 프로젝트의 업무 사례(business case)를 기술하고 프로젝트 범위를 정의하며 생명주기 목적(LCO: Life-Cycle Objective) 이정표를 가진다. 정교화는 필요한 활동들과 리소스들을 계획하고 아키텍처를 설계하고 시스템의 특성(features)를 기술하며 생명주기 아키텍처(LCA: Life-Cycle Architecture) 이정표를 가진다. 구축은 제품을 만들고 제품의 비전과 아키텍처, 계획을 전개시키며 초기 연산 능력(IOC: Initial Operational Capability) 이정표를 가진다. 이행은 사용자가 만족할 때까지 제품을 제조하고 전달하고 훈련시키고 지원하며 유지보수시키고 프로젝트 릴리스 이정표를 가진다. 여기에서 이정표는 현재 단계의 해당 임무를 완성하였는가를 평가하여 다음 단계로 넘어가기 위한 기준을 의미한다. 즉, 각 단계에 따라 이정표를 설명하면 다음과 같다. LCO 이정표는 도입 단계에서 요구분석이 제대로 되고 프로젝트 범위와 비용·스케줄이 수립되었는가를 평가하며 LCA 이정표는 정교화 단계에서 안정된 아키텍처가 세워졌는가를 평가한다. IOC 이정표는 구축 단계에서 최초 제품이 발표되고 사용자 매뉴얼이 작성되었는가를 평가하며 프로젝트 릴리스 이정표는 이행 단계에서 제품에 대한 실제 사용자의 만족을 평가한다.

RUP는 사용 사례와 아키텍처의 중요성을 강조하여 사용 사례 유도 프로세스(use case driven process) 또는 아키텍처 중심 프로세스(architecture-centric process)라고도 부른다[15]. 이 두 가지 특성으로 RUP는 시스템에서 정의된 사용 사례들이 전체 개발 프로세스에 대한 기초로서 사용되고 소프트웨어 시스템의 아키텍처로서 4+1 뷰인 논리적인 뷰, 구현 뷰, 프로세스 뷰, 배치 뷰, 사용 사례 뷰를 제공한다.

그림 1의 RUP 9개 워크플로우에서 소프트웨어의 개발에 직접적으로 반드시 필요한 것은 업무 모델링, 요구 분석, 분석 및 설계, 구현, 테스트, 배치 워크플로우이다. 다양한 시스템 모델링을 위해 6개 워크플로우를 기본적으로 사용해야한다는 것이 참고문헌 [16]에 기술되어 있다. 이와 같은 이유로 6개 워크플로우들은 핵심 프로세스 워크플로우라 불리운다. 각 핵심 워크플로우에 대한 설명은 다음과 같다. 업무 모델링 워크플로우는 조직의 구조와 동적인 면을 이해하고 고객, 사용자, 개발자들이 조직에 대한 공통의 이해를 가지도록 하기 위한 목적으로 기술된다. 요구분석 워크플로우는 스테이크홀더의 요구, 시스템 특성, 소프트웨어적 요구와 같은 개발 시스템에 대한 충분한 이해를 위해 요구사항을 기술

하는 작업이 이루어진다. 분석 및 설계 워크플로우는 요구사항을 시스템 구현 방법 기술 명세서로 번역하는 과정 즉 시스템 요구사항을 시스템 설계로 변환하는 과정이며 구현 워크플로우는 서버 시스템 구현의 차원에서 소스 코드 조직을 정의하기 위해 또는 바이너리, 소스 파일과 같은 컴포넌트 차원에서 클래스와 오브젝트를 구현하는 과정이다. 테스트 워크플로우는 오브젝트와 컴포넌트간의 상호작용을 검증하기 위해, 소프트웨어 모든 컴포넌트들의 적절한 통합을 검증하기 위해, 그리고 모든 요구사항이 구현되었는가를 검증하기 위해 사용되고 배치 워크플로우는 개발된 제품을 최종 사용자에게 전달하는 것이 목적이고 베타 테스트, 소프트웨어 설치, 소프트웨어 패키징, 소프트웨어 분배 등의 활동이 일어나는 워크플로우이다[12].

3.2 RUP 워크플로우별 산출물

다음 표 1은 RUP에서 소프트웨어 개발에 주요한 워크플로우인 '업무 모델링', '요구분석', '분석 및 설계', '구현', '테스트', '배치' 워크플로우에서 생성되는 산출물과 이 산출물들의 생성에 관련된 작업자 그리고 해당 워크플로우 내의 활동들을 축약한 것이다[16].

RUP는 표 1과 같이 상당한 양의 산출물들을 제시하고 있으며 여러 워크플로우에 걸쳐 다양한 작업자들이 서로 연관되어 있다. 그러나, 모든 프로젝트 수행시 위에 나열된 산출물과 작업자들을 전부 수용한다는 것은 경제적·시간적인 이유로 인하여 현실적으로 어렵다. 이러한 RUP 산출물들의 종류의 다양성은 RUP의 장점이 되기도 하지만 시스템 개발시 RUP를 적용할 때 가장 큰 어려움과 문제점으로 작용하기도 한다. RUP 프로세스는 원래 시범(pilot) 프로젝트부터 상업용 소프트웨어 개발까지 수용할 수 있도록 만들어진 프로세스이다. 소프트웨어 개발자들은 개발할 소프트웨어의 크기나 기타 환경을 고려하여 적절한 워크플로우, 산출물, 작업자, 활동 범위를 선정해야 한다[16].

본 연구의 목적이 상업용 제품을 만드는 것이 아니므로 UIS 상수도 시스템 개발을 위해 반드시 고려되어야 할 산출물들을 UML을 중심으로 선정하였고 선정된 산출물들은 표 1에 밑줄 그어 표현하였다. 선정된 산출물과 워크플로우가 내부적으로 활동을 포함하고 있으므로 모델링 작업시 각 활동에 대한 기술은 생략한다. 또한, 본 연구에서는 소프트웨어 개발 팀의 크기에 따라 동일한 사람이 다양한 역할을 수행할 수 있으므로 각각의 작업자에게 별도의 임무를 할당하지 않았다. 다음 4장에서는 표 1에 밑줄 그어진 산출물들을 중심으로 상수도 UIS 응용 시스템의 모델링을 기술한다.

표 1 RUP내의 워크플로우, 작업자, 산출물, 활동

워크플로우	작업자	산출물	활동
업무 모델링	업무프로세스 분석가	보충적인 업무 명세 업무 사용 사례 모델 용어해설 업무 객체 모델	일반 어휘 획득 사용사례와 액터 발견 업무 사용 사례 모델 구축
	업무 설계자	업무 사용 사례 업무 사용 사례 실체화 업무 작업자 업무 개체 조직 단위	업무 사용 사례 기술 업무 작업자와 개체 발견 업무 작업자 기술 업무 개체 기술
	업무 검토자	없음	업무 사용 사례 모델 검토 업무 객체 모델 검토
요구분석	시스템 분석가	비전 스тей크홀더 요구 사용 사례 모델 보충적인 명세 용어해설 요구사항 속성	비전 개발 의존성 관리 스тей크홀더의 요구 추출 일반 어휘 획득 액터와 사용 사례 발견 사용 사례 모델 구축
	사용 사례 기술자	사용 사례 사용 사례 패키지	사용 사례 기술
	시스템 건축가	소프트웨어 아키텍처 문서	사용 사례 우선순위 부여
	사용자 인터페이스 설계자	사용 사례 스토리보드 사용자 인터페이스 프로토타입 경계 클래스	사용자 인터페이스 모델링 사용자 인터페이스 프로토타이핑
요구분석 검토자	없음	요구분석서 검토	
분석과 설계	시스템 건축가	분석 모델 설계 모델 소프트웨어 아키텍처 문서	아키텍처 분석 아키텍처 설계 동시성 기술 분배
	설계자	사용 사례 실체화 분석 클래스 설계 클래스 설계 패키지 설계 서브시스템	사용 사례 분석 서브시스템 설계 클래스 설계 사용 사례 설계
	데이터베이스설계자	데이터 모델	데이터베이스 설계
구현	구현자	컴포넌트 구현 서브시스템	서브시스템 통합 계획 클래스 구현 결점 수정 단위 테스트 수행 서브시스템 통합
	시스템 통합자	통합 구축 계획	시스템 통합 계획 시스템 통합
	시스템 건축가	구현 모델 소프트웨어 아키텍처 문서	서브시스템 조직 정의
	코드 검토자	없음	코드 검토
테스트	테스트설계자	테스트 계획 작업부하 모델 테스트 모델 테스트 사례 테스트 절차 테스트 스크립트	테스트 계획 테스트 설계 테스트 구현 테스트 평가
	시스템 테스터	결점	시스템 테스트 수행
	통합 테스터	결점	통합 테스트 수행
	성능 테스터	결점	성능 테스트 수행
	설계자	패키지와 클래스 테스트	테스트 클래스와 패키지 설계
구현자	서브시스템과 컴포넌트 테스트	테스트 컴포넌트와 서브시스템 구현	
배치	개발 관리자	배치 계획	소프트웨어 분배
	기술적 작가	릴리스 노트(사용자 매뉴얼)	사용자에 대한 도움 제공
	구현자	설치 산출물	소프트웨어 설치
	교육과정 개발자	교육 자료	사용자에 대한 도움 제공

4. P시 상수도 시스템 모델링

다음 그림 2는 상수도 업무 응용 시스템의 RUP 적용 모델링 순서[16]를 나타낸다. 4장의 내용은 전체적으로 그림 2의 순서에 따라 '업무 모델링', '요구분석', '분석 및 설계', '구현', '테스트' 워크플로우 순서로 기술된다.

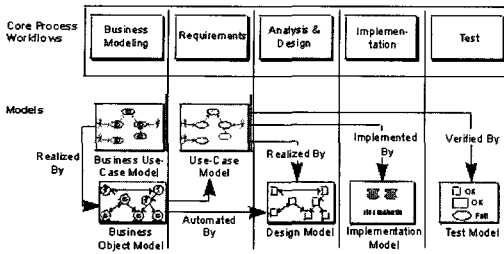


그림 2 상수도 UIS의 RUP 적용 모델링 순서

4.1 업무 모델링 워크플로우

4.1.1 보충 업무 명세

'보충 업무 명세(supplementary business specification)'는 전반적인 P시 상수도 UIS에 대한 스테이크홀더들의 이해를 위해 작성된다. 이 산출물은 P시 상수도 UIS의 전체적 배경과 본 연구에서 범위로 하는 상수도 UIS 세부 업무 정의를 목적으로 한다. '보충 업무 명세'는 시스템 개발 프로젝트와 관련된 모든 스테이크홀더들이 반드시 읽어야 할 문서이다. P시의 UIS에 관한 '보충 업무 명세' 내용은 다음과 같다.

P시 GIS 기본계획수립 연구보고[17]에서는 1998년에 P시 시청을 주변으로 1:1,000축척의 북구 9개 도엽에 대해 GIS기본계획수립연구를 수행한 바 있다. 정보통신부에서 지원하는 지하시설물도 전산화사업(정보근로화사업)의 일환으로 현재 P시에서는 GIS기본 계획을 확장하여 전체 상수도 업무의 완전 전산화를 수행 중이다. 경제적·시간적 원인으로 P시의 모든 단위 업무들을 본 연구에서 전부 개발하는 것은 어려우므로 각 단위업무에 우선 순위를 부여하여 시급한 개발이 필요한 업무를 선정하였다. 우선 순위 부여 작업은 정보통신부, 행정자치부, 건설교통부 등이 공동으로 사업을 추진하고 있는 국가GIS(NGIS)사업 결과 중 하나인 'NGIS 우선 순위 평가 기준'[18]을 사용하였다. 이 우선 순위 평가 결과에 따라 P시 지하시설물도 전산화사업의 일차 작업대상으로 평가기준에 의한 합계 점수가 47.5로 가장 높은 상수도관리 업무를 선정하였다[17].

본 연구는 P시 지하시설물도 전산화사업의 수행과정 중에 얻어진 업무분석 결과 및 경험을 최대 활용한다. 상수도 업무에는 상당한 많은 양의 대장과 도면이 소요된다. 상수도 업무로는 세부적으로 댐축조와 하천수관리, 상수원관리, 취수관리, 정수관리, 배수관리, 누수관리 등이 있다. 단, 상수도업무 중 가장 핵심이 되고 상수도 관망도와 같은 많은 도면을 사용하기 때문에 MIS(Management Information System) 보다는 GIS 성격이 강한 누수업무와 누수업무에 관련된 공사·도면 관리와 같은 기타 상수도 업무에 대해 본 연구를 적용한다.

4.1.2 업무 작업자, 업무 개체, 업무 사용 사례

'업무 작업자(business worker)'는 해당 업무 내에서 행동하는 인간의 추상화이며 '업무 개체(business entity)'는 '업무 작업자'가 접근하고 조사하고 조작하고 생산하는 객체를 의미한다. '업무 개체'는 '업무 작업자'들 사이의 공유의 기초를 제공한다. '업무 작업자'는 다른 '업무 작업자'와 상호작용하여 '사용 사례 실체화(realization)'에 참여하는 동안 '업무 개체'를 조작한다. 또한 '업무 사용 사례(business use case)'는 중요한 결과를 생산하기 위해 수행하는 행동의 순서를 나타내며 일반 '사용 사례'와 유사한 개념이다.

상수도 누수업무에서 '업무 작업자', '업무 개체', '업무 사용 사례'를 모델링하기 위해서는 먼저 현장에서 일어나는 상수도 업무를 분석해야 한다. 다음의 기술되는 내용은 실제 P시 상수도 업무 담당자들과의 인터뷰를 통하여 얻은 업무 분석 결과이다.

P시 상수도 업무 조직은 상수도사업소 내에 업무과, 시설과, 정수 1과, 정수 2과가 존재한다. 상수도사업소 업무과는 예산업무, 회계업무, 요금업무, 기업운영업무를 맡고 시설과는 상수도사업(시설물 확장·정비), 급수신청업무, 누수업무, 간이상수도업무, 갱생업무, 노후배관 교체업무를 정수 1과와 정수 2과는 정수장 운영의 역할을 담당한다[19].

또한, P시 상수도 업무는 다른 지방자치 단체와 마찬가지로 물의 흐름에 따라 업무가 이루어진다. 즉, 가정에 공급되는 물은 원수→취수→도수→정수→배수→급수의 과정을 거치게 된다. 이와 같은 물의 흐름에 따라 발생하는 상수도 업무를 알아보면 다음과 같다. 원수확보를 위한 상수시설 설치로서 댐축조와 하천수관리 업무가 있고 또한 상수원관리를 위한 상수원 보호구역 지정·확대 업무가 있다. 취수·정수에 대해서는 자연유하식, 가압식 등으로 상수원(취수원)으로부터 취수하여 도수관을 통하여 정수장에 보내어지는 취수관리 업무가 있고

며 원수를 정수장에서 정수하여 먹을 수 있는 물로 생산하여 보관하게 되는 정수관리 업무가 있다. 최종적으로 상수도 업무는 시민들의 가정 급수전까지 안정적으로 물을 보내는 배수관리, 누수관리 업무로 분류될 수 있다. 이와 같은 상수도업무 분석 결과를 ‘업무 모델링 워크플로우’에 반영하여 ‘업무 사용 사례’로 댐축조와 하천수관리, 상수원관리, 취수관리, 정수관리, 배수관리, 누수관리를 추출하였다.

P시 상수도 업무에서 ‘업무 객체’는 상수도 사업소 조직도에 있는 각 부서이고 ‘업무 개체’는 상수도 업무에 사용하는 각종 대장 및 도면 등이다. ‘업무 객체’와 ‘업무 개체’는 UML 표기법으로 클래스 형태로 나타내며 ‘분석 및 설계 워크플로우’의 일반 클래스들과 식별하기 위해 스테레오 타입 ‘«business entity»’와 ‘«business worker»’를 사용하여 표시한다. ‘업무 사용 사례’는 일반적인 UML의 ‘사용 사례’ 표기법을 사용한다.

또한, ‘업무 모델링 워크플로우’에서 ‘업무 객체 모델’과 ‘업무 사용 사례 모델’이 강조될 수 있다. ‘업무 객체 모델’은 ‘업무 사용 사례’를 실제화시키는 객체 모델로서 ‘업무 객체’와 ‘업무 개체’로 기술되며 ‘업무 사용 사례 모델’은 해당 업무가 의도하는 기능에 대한 모델을 의미한다.

표 2는 ‘업무 사용 사례’ 중 하나인 ‘누수관리 업무 사용 사례’에 관한 현장 업무 중심의 시나리오를 기술한 것이다. 단, 여기서는 선택적인 과정(alterative course)은 제외하고 일반적인 과정(normal course)을 기술한다.

표 2 누수관리 업무 사용 사례 시나리오

1. 상수도 관로 상에 누수가 발생한다.
2. 누수발생을 시민이 전화하거나 타 공사로 인한 누수인 경우 공사 담당자가 누수발생을 신고한다.
3. 상수도사업소 시설과 누수방지계 담당자가 누수접수 대장에 누수사항을 기록한다.
4. 누수방지 담당자는 누수업무 관련 유관기관 (인문기관에 단수홍보 및 경찰서에 교통통요요구)에 연락하고 대행업소에 누수처리 작업을 지시한다.
5. 누수방지 담당자는 누수 복구 작업을 위해 제수변을 조작한다.
6. 누수방지 담당자의 현장 입회 하에 대행업소가 누수 복구작업을 실시한다.
7. 누수복구작업 완료 후 대행업소는 당일 1차 전화보고를 하고 한 달에 한 번 2차 서류보고를 실시한다.
8. 시설과 누수방지 담당자는 대행업소가 제출한 공사사진, 견적서를 보고 설계하여 업무과에 공사비를 지급 통보한다.
9. 상수도사업소 업무과는 대행업소와 계약하고 공사비를 지급한다.

4.1.3 용어해설

‘용어해설(glossary)’은 프로젝트 수행 결과 생산된 산출물과 문서들을 읽을 필요가 있는 스테이크홀더들에게 정보를 제공하기 위한 것이고 해당 문제 영역에 관한 특수한 전문 용어를 정의하기 위해 사용된다. ‘용어해설’은 종종 약식의 자료사전으로서 사용되기도 하며 컴퓨터에 ‘glossary’라 부르는 파일 형태로 저장되어야 한다.

상수도 업무 관련 용어의 몇 가지 예를 들면 다음 표 3과 같다. 본 연구에서는 이와 같은 기초적인 상수도 용어를 수도법 등을 참조하여 총 120개의 단어를 정리하였다.

표 3 상수도 용어 해설

용어명칭	용어 정의
원수	식용, 공업용으로 제공하는 자연 상태의 물
취수	상수원에서 원수를 끌어들이는 것
도수	취수 시설에서 정수장까지 물을 보내는 것
정수	원수의 수질을 용도에 적합하도록 정화하는 일련의 과정
배수	정수장에서 정수 처리된 청정수를 분배하는 일련의 과정
급수	배수관에서 분기되어 각 가정의 급수전까지 물을 보내는 일련의 과정
제수변	물을 공급하는 관로에 설치되며, 물의 흐름을 제어하는 시설물
밸기밸브	관로의 높은 곳에 설치하여 공기를 자동적으로 배출시키는 밸브
누수율	누수량과 배수량의 비

4.2 요구분석 워크플로우

4.2.1 비전

‘비전(vision)’은 고수준의 요구분석과 설계 제약사항을 제공하는 것이며 개발된 시스템에 대한 관계자의 이해를 위한 것이다. ‘비전’은 애플리케이션의 목적을 설명하며 업무기회(business opportunity)와 문제 제시(problem statement), 주요한 사용자 요구, 사용자 환경 등을 기술한다. 누수 중심의 상수도 업무의 ‘비전’은 다음과 같다.

상수도 업무 중 누수관리는 현장에서 누수가 발생할 경우 누수 복구 작업을 통하여 현장사진 및 누수관련 정보를 이용해 기존 데이터베이스를 수정할 수 있어야 한다. 누수작업은 반드시 복구공사를 수반함으로 해당 누수 공사 정보는 별도로 관리되어야 하며 해당 누수지역에 대한 래스터 벡터 지도도 제공되어야 한다. 또한,

누수정보 및 누수 관련 정보를 입력할 경우 사용자의 편의성을 고려하여 쉽게 입력할 수 있도록 등록 절차를 간단히 해야 하고 편리한 사용자 인터페이스를 제공하는 윈도우 환경에서 작업이 이루어져야 한다.

4.2.2 스테이크홀더 요구와 용어해설

'스테이크홀더 요구(stakeholder needs)'는 시스템 분석가가 작성하는 것으로서 개발된 시스템에 대해 이해관계자(고객, 최종사용자, 판매자 등)가 가지는 여러 종류의 요구(requests)이다.

상수도 누수업무와 관련된 스테이크홀더로는 상수도 사업소 시설과 누수방지계, 누수공사를 실시하는 대행업소, 공사관리를 담당하는 상수도 사업소 업무과 기업회계계, 누수민원 신고자 등이 존재한다. 이들은 시스템의 고객과 최종사용자로 볼 수 있다. 상수도 사업소 시설과 누수방지계는 누수가 발생할 경우 누수관리 시스템에 누수정보를 기록 후 누수 대장과 누수 작업지시서 발급을 자동화하길 원한다. 누수공사를 실시하는 대행업소는 누수가 발생한 현장에 관한 정확한 정보를 지도와 함께 제공받아 복구 작업을 용이하게 하고 제수변 등의 제조사가 발생되지 않기를 희망한다. 공사관리를 담당하는 상수도 사업소 업무과 기업회계계는 여러 업무에서 활용하도록 설치된 시설물과 관망의 공사에 관한 최신 정보를 위해 데이터베이스의 일관성 유지와 관리를 원한다. 누수민원 신고자 입장에서는 상수도 업무 관련자의 누수정보 입력을 위해 누수 신고로 인한 번거로운 호출이 없길 희망한다.

또한, 누수관리 업무와 관련된 '용어 해설'이 새롭게 '요구분석 워크플로우'에서 추가된다. '용어해설'은 '업무 모델링 워크플로우(business modeling workflow)'부터 마지막 '테스트 워크플로우'까지 반복적으로 계속 추가된다. '요구분석 워크플로우'에서 추가된 '용어해설'의 예는 다음과 같다. 누수대장은 누수위치, 누수원인, 누수신고자, 누수 복구자, 누수 복구내용 등을 포함하고 있는 대장이며 누수현황은 현재까지 발생한 누수정보와 누수 복구정보를 나타내는 것이다. 누수 복구 작업지시서는 누수 복구자에게 복구작업을 지시할 때 전달하는 것으로서 누수위치와 지시사항, 민원내용을 포함하고 있는 서류이다.

4.2.3 사용 사례 모델

'사용 사례 모델(use case model)'은 시스템의 기능과 환경에 관한 모델이고 고객과 개발자 사이에 계약으로 사용된다. '사용 사례 모델'은 또한 '분석 및 설계', '테스트' 워크플로우 내의 활동들에 대한 입력으로서 사용된다.

4.1.2에서 파악된 '업무 사용 사례'들은 다시 이 절에서 검토된다. 사람과 같이 실세계에 존재하는 생명체여서 전산화 대상이 될 수 없거나 사람의 추상적이고 주관적인 판단을 자주 요구하여 전산화가 어려운 것들은 '요구분석 워크플로우'의 '사용 사례'에서 제거된다. 즉, '업무 모델링 워크플로우'는 실세계의 업무 관점에서 살펴본 것이고 '요구분석 워크플로우'에서 '배치 워크플로우'까지는 컴퓨터를 이용한 정보 시스템 관점에서 실세계 업무를 바라본 것이다.

본 논문에서는 상수도 시스템에 관한 '사용 사례'로서 관로관리, 시설물관리, 변류(밸브)관리, 공사관리, 누수관리, 누수관리를 식별하였고 누수관리에 대한 세부 기능으로는 누수정보 관리, 누수이미지 관리, 누수현황 관리를 파악하였다. 또한, 상수도 시스템의 모든 '사용 사례'를 지원하기 위해서는 공사관리, 도면관리 기능이 제공되어야 하므로 도면관리를 추가로 '사용 사례'로 식별하였다.

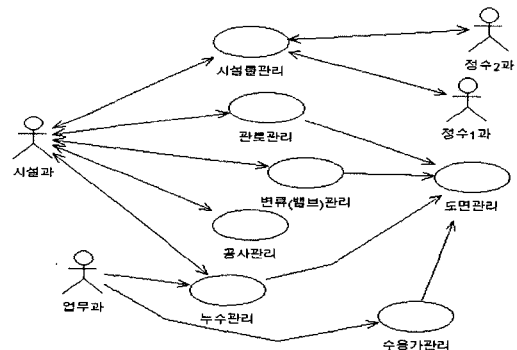


그림 3 상수도 시스템 사용 사례 모델

다음 그림 3은 상수도 시스템에 대한 '사용 사례'로 관로관리, 시설물관리, 변류(밸브)관리, 공사관리, 누수관리, 수용기관리, 도면관리를 표현한 '사용 사례 모델'을 나타낸다. 그림 3의 상수도 시스템 '사용 사례 모델'은 P시 상수도 업무 담당자로부터 승인을 받은 것이다.

위와 같이 모델링 작업자의 실제 시스템 개발 경험과 모델링 경험 등에서 '사용 사례'들이 다시 선택되고 점차 반복적으로 '사용 사례'들은 좀 더 세분화 될 수 있다. '사용 사례'는 시스템이 해야 할 대표적인 기능으로 서로 밀접히 연관된 클래스들을 사용하는 것을 하나의 '사용 사례'로 선택하는 것이 바람직하다. 그러나, 모델링 시 초기의 이러한 '사용 사례'의 선택이 올바른지 판단하기는 무척 어려운 결정이다. 이 때문에 반복적인 개

발을 통하여 '사용 사례'에 대한 검증이 계속 이루어져야 한다.

4.2.4 사용자 인터페이스 프로토타입, 사용 사례 스토리보드

'사용자 인터페이스 프로토타입(user interface prototype)'은 종이 스케치, 드로잉 도구(tool)로 제작한 그림, 또는 Visual Basic과 같은 것으로 만들어진 실행 가능한 파일 등 세 가지 형태로 작성될 수 있다. 사용자 인터페이스 생성 목적은 실제 설계와 구현을 시작하기 전에 시스템 이용가능성과 기능성의 노출과 테스트를 위한 것이다. 이것은 시스템을 개발할 때 너무 많은 시간과 자원을 소비하기 전에 올바른 시스템을 만들고 있다는 확신을 심어준다. 물론 '사용자 인터페이스 프로토타입'은 개발비용이 실제 시스템 개발비용보다 훨씬 적게 들어야 한다. 다음 그림 4는 누수관리 '사용 사례'의 '사용자 인터페이스 프로토타입'을 나타낸다.

표 4 누수관리 사용 사례 스토리보드

- a) 누수관리 사용 사례는 사용자가 최상위 메뉴에서 누수 관리를 선택할 때 시작되고 누수정보 관리 사용자 인터페이스가 디스플레이 된다.
- b) 누수관리 사용 사례 사용자는 다음 하나 이상의 단계를 수행할 수 있다.
 - b1) 누수정보를 등록/삭제/수정/조회한다.
 - b2) 해당 누수이미지를 표시한다.
 - b3) 누수이미지를 등록/삭제/수정/조회한다.
 - b4) 해당 누수공사정보를 표시한다.
 - b5) 누수공사정보를 등록/삭제/수정/조회한다.
 - b6) 해당 누수도면을 표시한다.
 - b7) 누수도면을 확대/축소/이동/조회한다.
 - b8) 누수정보/이미지/도면을 프린터로 출력한다.
 - b9) 누수대장을 프린터로 출력한다.
 - b10) 작업지시서를 프린터로 출력한다.
- c) 누수관리 사용 사례는 사용자가 최상위의 다른 메뉴를 클릭할 때 끝난다.

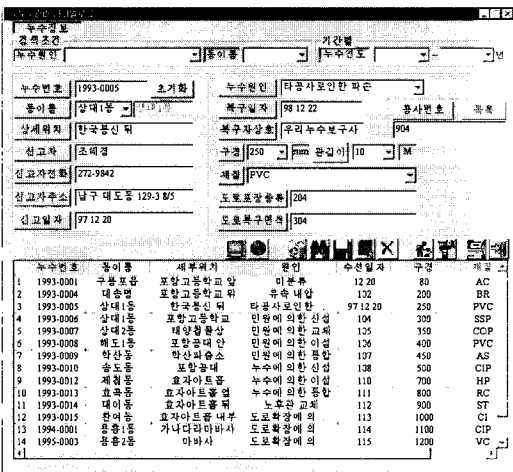


그림 4 누수관리 사용자 인터페이스 프로토타입

'사용 사례 스토리보드(use case storyboard)'는 '사용 사례'가 사용자 인터페이스에 의해 제공되는 방법에 대한 논리적이고 개념적인 기술을 의미한다. 표 4는 그림 4를 바탕으로 하여 누수관리의 '사용 사례 스토리보드'를 표현한 것이다. 표 4와 같이 '사용 사례 스토리보드'는 시나리오와 매우 유사한 형태를 지닌다.

4.3 분석 및 설계 워크플로우

4.3.1 사용 사례 실체화

'사용 사례 실체화(use-case realization)'는 설계 모델 내에 특별한 '사용 사례'를 구체화시키는 방법을 기술한 것이다. '사용 사례 실체화'는 분석과 설계 동안 '사용 사례'들에 우선 순위가 부여되자마자 작성된다.

'사용 사례 실체화'와 관련된 산출물은 '사용 사례'의 행위를 표현하는 협동 다이어그램과 순서 다이어그램이다. 상수도 시스템 '사용 사례'들 가운데 최고 우선 순위를 갖는 것은 누수관리 '사용 사례'이다.

다음 그림 5는 누수관리 '사용 사례'에 대한 일부분 순서 다이어그램이다. 그림 5는 누수관리 '사용 사례' 중 누수정보와 해당 누수이미지를 검색하며 누수현황리스트를 디스플레이하는 순서 다이어그램이다. 그림 5에서처럼 누수관리는 누수정보(leak information) 클래스가 핵심 클래스로서 이 클래스로부터 누수이미지(leak image), 누수현황리스트(leak list) 클래스들이 호출된다.

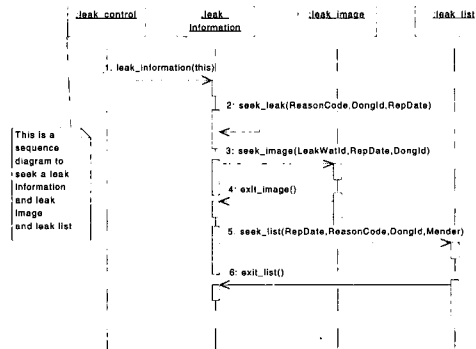


그림 5 누수관리 순서 다이어그램

4.3.2 분석 및 설계 모델

'분석 모델(analysis model)'은 '사용 사례 실체화'를

기술하는 객체 모델이고 '설계 모델(design model)'에 대한 추상화이다. '분석 모델'은 '사용 사례' 분석 결과와 '분석 클래스'의 인스턴스들을 포함한다. RUP에서는 '분석 모델'을 선택적 산출물로 정의하였다. '분석 모델'은 설계에 대한 입력으로 사용되고 설계에 대한 추상적 개요(abstract overview)를 나타내며 '설계 모델'과의 일관성이 유지되어야 한다. '분석 클래스'는 시스템 내에 존재하는 책임(responsibility)과 행위를 가지는 '설계 클래스'들에 대한 초기의 개념적 모델을 나타내므로 결국 '분석 모델'이 더욱 정교해져 '설계 모델'로 된다. 본 논문에서는 '분석 모델'의 기술은 생략하고 '설계 모델'에 대해서만 언급한다.

'설계 모델'은 '구현 모델'과 소스 코드의 추상화이며 '구현'과 '테스트' 내의 활동들에 대한 근본적인 입력이 된다. 좋은 '설계 모델'은 시스템의 '요구분석'을 만족시키고 구현 환경의 변화에 저항력이 있어야 하며 유지보수가 쉽고 구현 방법이 명백해야 한다. '설계 모델'에 관한 대표적인 다이어그램은 클래스 다이어그램이다. 그림 6은 누수관리에 대한 클래스 다이어그램을 보여주고 있다.

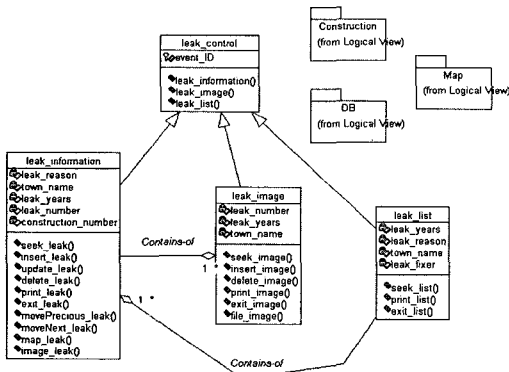


그림 6 누수관리 클래스 다이어그램

클래스 다이어그램에 표시되는 의존 관계성(dependency relationship)은 소스 코드로 매핑할 때 관련 클래스가 현재 클래스 함수의 매개변수로 사용되는 경우를 나타낸다. 의존 관계성은 클래스 다이어그램에서 가장 빈번히 발생한다. 모델링시 가장 큰 어려움은 클래스 다이어그램을 작성할 때 하나의 연산을 어느 클래스에 할당하지 또한 클래스들간의 관계성이 무엇인지를 파악하는 것이다. 일반적으로 클래스들간의 관계성에는 상속, 의존성, 집합, 기타 연관 관계가 있다.

본 연구에서는 '사용 사례 시나리오'에 따라 작성된

순서 다이어그램을 참조하여 객체에 대한 메시지의 방향을 찾아 해당 클래스에 연산을 할당하였다. 단, 협동 다이어그램은 Rational사의 ROSE와 같은 모델링 도구에서 순서 다이어그램을 작성 후 자동 생성되므로 여기에는 표시하지 않는다[16]. 또한, GRASP 패턴을 적용하여 최대한 작은 결합도와 강한 응집도를 유지하고 사용자 인터페이스 분리와 제어 클래스를 식별하여 MVC 패턴을 지원하도록 하였다[10].

4.3.3 설계 패키지과 설계 서브시스템

'설계 패키지(design package)'는 관련된 설계 모델 요소들을 함께 묶기 위해 사용된다. '설계 서브시스템(design subsystem)'과 달리 '설계 패키지'는 인터페이스를 제공하지 않고 'public'라 표시된 것을 노출한다. '설계 패키지'는 주로 관련된 것들을 함께 묶기 위해 사용하고 행위적 의미(behavioral semantics)가 요구된다면 '설계 서브시스템'을 사용해야 한다.

본 연구에서는 복잡성을 줄이고 향후 확장성을 고려하여 상수도 시스템을 누수관리, 공사관리, 도면관리의 세 가지 패키지로 분할하였다. 누수정보관리, 누수이미지관리, 누수현황관리 클래스들이 서로 밀접히 연관되고 강력한 의존성을 가지므로 이 세 가지 클래스를 하나의 누수관리 패키지로 묶었다. 또한, 공사관리는 누수공사뿐만 아니라 향후 다른 시설물의 공사도 포함하도록 별도의 패키지로 만들고 도면관리 패키지는 다른 상수도 업무에서 도면처리가 필요할 때 그리고 향후 도면의 분석·통계·편집 기능 추가를 위해 별도의 패키지로 구분하였다.

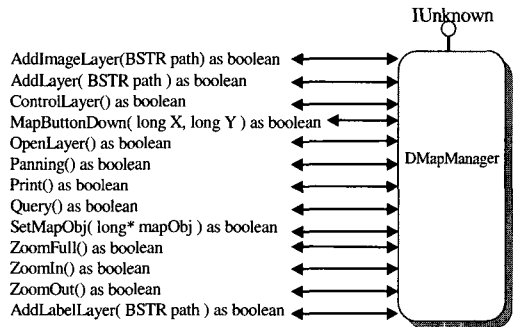


그림 7 도면관리 컨트롤의 인터페이스

누수관리, 공사관리, 도면관리는 내부에 서로 밀접히 연관된 클래스들을 가진 별도의 세 가지 패키지이면서 또한 인터페이스를 서로 노출하는 서브시스템으로 볼 수 있다. 본 논문에서는 위의 세 가지 패키지를 서브시

스텝화하여 인터페이스를 서로 노출하도록 ActiveX 컨트롤 형태로 개발하였다. 다음 그림 7은 ESRI사의 MapObjects 컨트롤[20, 21]을 사용하여 개발한 도면관리 컨트롤인 DMapManager의 인터페이스를 나타낸다.

즉, 상수도 응용 시스템은 그림 8과 같이 구성되었다. 누수관리, 도면관리, 공사관리 컨트롤들은 사용자 인터페이스와 분리되고 각 컨트롤은 외부와의 인터페이스를 통해 통신한다. 이와 같이 상수도 응용 시스템을 ActiveX 컨트롤들로 개발할 경우 어려운 점은 컨트롤들간의 인터페이스를 분리하고 인터페이스를 통해 상호작용을 발생시키는 것이다.

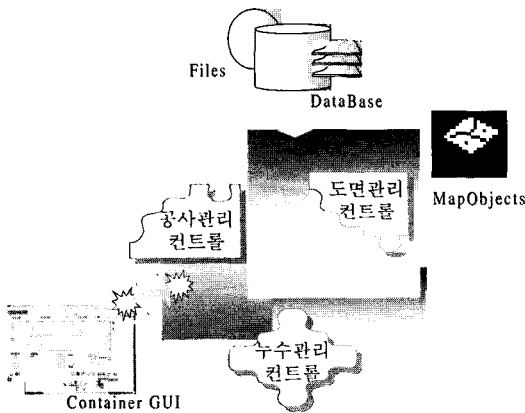


그림 8 상수도 응용 시스템 구성

4.3.4 데이터 모델

‘데이터 모델(data model)’은 ‘구현 모델’의 부분 집합이고 시스템 안에 영속 데이터(persistent data)의 논리적이고 물리적인 표현을 기술한다. ‘데이터 모델’은 시스템이 영속 데이터를 가지고 있다면 반드시 기술되어야 한다. 즉, ‘데이터 모델’은 데이터베이스 아키텍처를 결정한다.

그림 9는 누수관리 ‘사용 사례’의 ‘데이터 모델’을 보여준다. 그림 9에서는 동이름(DongName)과 동ID(DongId)를 별도의 테이블에 저장하여 누수정보 테이블(SA_Leak Water)에서 참조하도록 하였다. 또한, 누수원인과 누수관로 재질도 이와 같이 별도의 테이블에 데이터가 저장되도록 하여 누수정보 테이블에서 데이터 중복이 발생하지 않도록 하였다.

참고적으로, UML과 RUP에는 데이터 모델링 표기법이 존재하지 않으므로 그림 9에서와 같이 ‘Dong’과 ‘Reason’ 등은 테이블 이름을 나타내고 ‘DongId’와 ‘DongName’은 하나의 레코드에 존재하는 필드명으로

간주하며 테이블들의 관계성 표시는 개체 관계도(entity relationship diagram)와 클래스 다이어그램의 의미를 혼합하여 사용하였다.

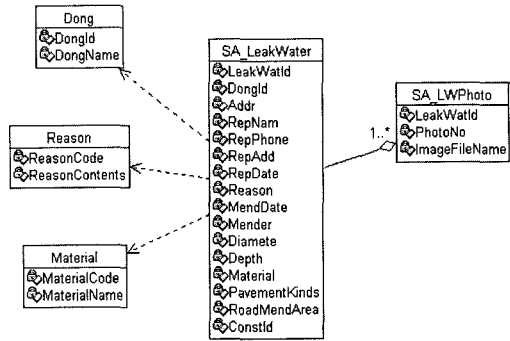


그림 9 누수관리 사용 사례 데이터 모델

4.4 구현 워크플로우

4.4.1 구현 모델과 구현 서브시스템

본 상수도 시스템은 Visual Studio 6.0에 있는 Visual C++를 이용하여 개발되었고 도면 처리를 위해 ESRI사의 MapObjects 컨트롤을 사용하였다. 데이터베이스는 MS-Office내의 Access를 사용하여 구축하였고 공간정보의 가공을 위해 ArcInfo와 같은 별도의 GIS 도면 편집 도구는 사용되지 않았다.

‘구현 모델(implementation model)’은 계속적으로 ‘설계 모델’과 일관성이 유지되어야 한다. ‘구현 모델’은 컴포넌트의 집합이고 ‘구현 서브시스템(implementation subsystem)’은 그 컴포넌트 집합을 포함하는 개념이다. 단, 여기서 언급하는 컴포넌트 의미는 일반적인 정의와 약간 차이가 있다. 즉, ‘구현 워크플로우(implementation workflow)’ 내에서 컴포넌트 의미는 실행 파일(deliverable components) 또는 실행 파일을 생산하는 소스 코드와 같은 것을 의미한다.

본 연구의 ‘구현 워크플로우’ 내에서 ‘구현 서브시스템’은 누수관리, 공사관리, 그리고 도면관리 컨트롤을 의미한다. 이 세 가지 ‘구현 서브시스템’ 내에 존재하는 ‘구현 모델’들은 각 컨트롤에 속하는 실행파일 또는 소스파일 형태의 컴포넌트 집합을 나타낸다. 그림 10은 도면관리 컨트롤 내에 들어 있는 C++의 ‘cpp’ 파일 중심의 컴포넌트 집합을 보여주는 ‘구현 모델’이다.

그림 11은 ‘구현 서브시스템’ 가운데 하나인 누수관리 컨트롤의 누수정보 이미지 디스플레이 화면이다. 그림 11의 왼쪽에는 발생한 누수정보 리스트가 보여지고 오

른쪽 위는 현재 왼쪽에서 선택된 누수정보와 관련 있는 누수이미지가 표시되며 아래 부분은 데이터베이스에 새로운 누수이미지를 등록하는 것을 보여주는 화면이다.

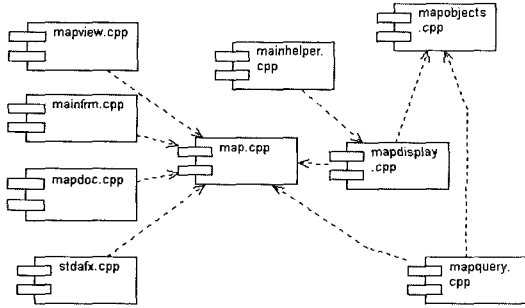


그림 10 도면관리 컨트롤 내의 구현 모델

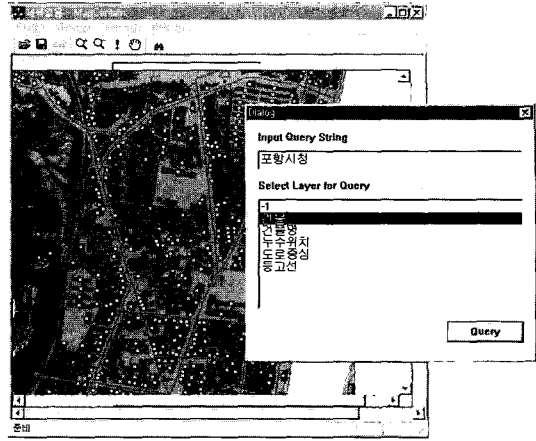


그림 12 도면관리 컨트롤을 이용한 질의

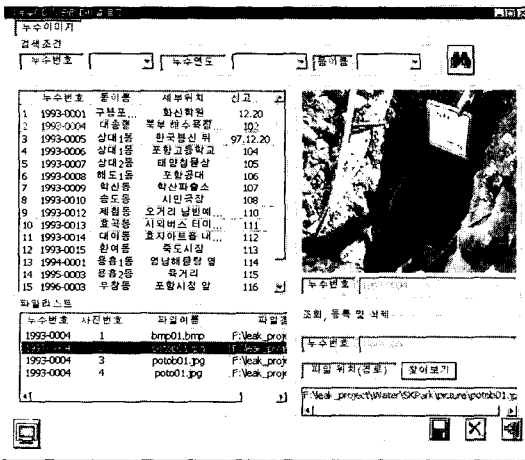


그림 11 누수정보 이미지 디스플레이 화면



그림 13 도면관리 컨트롤 질의 결과

그림 12는 도면관리 컨트롤을 '구현 서브시스템'으로 정의해 구현한 뒤 별도의 컨테이너에 삽입하여 현재 디스플레이된 건물레이어 'shape'파일에서 P시의 시청 건물 위치를 검색하는 화면이다.

그림 13은 그림 12의 질의에 대한 결과 화면이다. 그림 13의 가운데에 있는 다각형들이 P시의 시청 건물들이다. 도면관리 컨트롤은 누수관리 컨트롤에서 호출되어 사용되며 검색기능 이외에 레이어 선택과 디스플레이 레이어 우선 순위부여, 레이어 삭제 기능, 화면 확대·축소·이동 기능, 항공사진과 같은 래스터 이미지 디스플레이 기능을 지원한다.

다음 표 5는 상수도 응용 시스템 중 도면관리의 일부분 구현 함수이다. 표 5는 도면관리 컨트롤 내에서 건물 위치와 주소 등에 대한 질의를 받아 처리한 뒤 그 결과를 표시해주는 'Query()'함수를 나타낸다.

4.5 테스트 워크플로우

'테스트 워크플로우(test workflow)'에서는 '요구분석 워크플로우'에서 작성된 '사용 사례'를 기반으로 테스트 작업이 이루어진다. '테스트 워크플로우'의 대표적인 산출물로는 '테스트 사례'가 있고 이것은 '테스트 절차'와 같은 다른 테스트 작업의 기초가 된다.

'테스트 사례'의 목적은 '사용 사례'의 성공적인 구현을 검증하기 위해 필요한 조건들을 식별하는 것이다.

표 5 도면관리 컨트롤 내의 질의 처리 함수

```
// Query()
// 원하는 질의를 받아서 (예를 들어, 건물위치) 결과 값을
// 도면에 표시하여 보여주는 함수
void Query()
{
    // Query Dialog를 띄워서, 원하는 Query와 질의할
    Layer를 결정한다.
    QueryDialog.Show();

    // User가 입력한 Query를 얻는다.
    Query = QueryDialog.GetQuery();

    // User가 정한 Layer를 얻는다.
    // 이 Layer에서 원하는 질의를 실행시킬 것이다.
    Layer = QueryDialog.GetLayer();

    // 입력한 Query로 User가 정한 Layer에서 ANSI SQL
    // 문을 실행시킨다.
    // User가 입력한 Query를 ANSI SQL문으로 바꾼다.
    // 이를 위해 Layer의 Field와 Query를 연결시킨다.
    SQL_Query = Layer.Field + "like" + "%"+Query+"%";

    // SearchByExpression은 특정 Layer에서 User의 질의
    // 에 대한 Result를 찾아낸다.
    // 질의는 위에서 ANSI SQL 문장으로 바꾼 것을 사용
    // 한다.
    Result=MapObject->SearchByExpression(Layer, SQL_Query);

    // 하나 이상의 값이 나올 수 있으므로 User에게 어느
    // 것이 맞는지 물어본다.
    SelectDialog.Show(); // 선택 Dialog를 보여준다.
    SelectedResult = SelectDialog.GetSelect();

    // 원하는 곳을 확대(ZoomIn) 하고 그곳을 깜박여서 표
    // 시한다.
    MapObject->ZoomIn ( Result.position );
    // Zoom In한다.
    MapObject->Blank ( Result.Polygon );
    // 깜박여서 표시한다.
}
```

‘테스트 사례’는 테스트 조건 및 목적과 같은 상세한 설명을 나타내는 테스트 사례 기술, 테스트 입력, 테스트 결과로 구성된다. ‘테스트 사례’의 종류는 여러 가지가 있으나 가장 중요한 ‘테스트 사례’로는 소프트웨어가 서비스할 업무의 요구사항에 관한 것이다. 이 요구사항은 초기의 ‘요구분석 워크플로우’에서 ‘사용 사례’로 표현된다. 전형적으로 순서 다이어그램이 이러한 타당성 검증을 위한 기초로서 이용되고 모든 가능한 이벤트 흐름의 패스는 ‘테스트 사례’의 후보가 된다.

본 연구의 ‘테스트 워크플로우’는 20개의 ‘테스트 사례’를 중심으로 이루어졌으며, 이 ‘테스트 사례’들은 ‘사용 사례’의 시나리오와 순서 다이어그램의 모든 가능한

이벤트 흐름의 패스를 기반으로 한다. 다음 표 6은 누수 관리 ‘사용 사례’ 중 누수정보 등록에 관한 일반적인 과정의 ‘테스트 사례’ 예를 보여준다.

표 6 누수정보 등록의 테스트 사례

```
테스트 사례 기술 : 이 테스트 사례는 누수정보 등록이 올바르게 수행되는지의 테스트를 목적으로 한다.
테스트 입력 : 누수번호, 동이름, 상세위치, 신고자, 신고자 전화, 신고자 주소, 신고일자, 누수원인, 수선자상호, 수선일자 등을 입력하고 등록 버튼을 클릭한다. 누수정보 조회로 DB에 올바르게 누수정보가 입력되었는지 확인한다.
테스트 결과 : DB에 새로운 누수정보가 등록된다.
```

본 연구 결과는 클라이언트/서버 구조가 아니고 단일 환경 및 독립된 기계(컴퓨터)에서 동작하는 UIS 응용 프로그램이므로 ‘배치 워크플로우’는 연구 대상에서 제외된다. 즉, ‘배치 워크플로우’는 상업용 프로젝트 및 대규모 프로젝트 등의 목적으로 분산 환경인 경우 컴포넌트의 배치, 매뉴얼의 작성, 교육 자료, 소프트웨어 설치 파일들이 작성되는 단계이다.

5. 연구결과 의의 및 토의

표 7은 기존에 개발되었거나 현재 개발 중인 국내 상수도 응용 시스템의 현황을 나타낸다. 표 7은 국토연구원의 보고서[22]를 참고한 뒤 직접 개발업체에 전화로 문의하여 작성한 것이다.

표 7에서 알 수 있는 것처럼 국내의 상수도 응용 시스템들은 개발 참여 업체에 따라 다양한 개발 언어, GIS 기본 소프트웨어, 소프트웨어 개발 프로세스를 사용하고 있다. 국내 지방자치 단체 UIS 분야의 가장 큰 문제점은 표 7과 같이 업체마다 단일 클라이언트/서버 구조의 응용 프로그램들을 개발하고 있기 때문에 기존에 개발된 시스템에 대한 재사용과 확장성이 떨어져 시스템 개발에 중복적인 투자가 발생한다는 점이다. 또 다른 문제점으로 업체에 따라 사용하는 소프트웨어 개발 프로세스가 다양하다는 것이다. 즉, 대부분의 업체들은 기존의 구조적 기법의 소프트웨어 개발 프로세스를 이용하고 있거나 자체 개발한 프로세스들을 사용하고 있다. 이러한 프로세스들은 컴포넌트 개발에 적합하지 않다. UIS 컴포넌트의 개발을 위해서는 컴포넌트 기반 개발 프로세스가 사용되어야 한다. 위의 두 가지 문제점은 UML

표 7 국내 상수도 응용 시스템 개발 현황

지방자치단체명	개발언어	개발업체	GIS 기본 소프트웨어	소프트웨어 형태	재사용성	확장성	개발 프로세스
울산광역시	NULL, Visual C++	쌍용정보통신	Gothic, AutoCAD	단일 C/S	△	△	GUIDE
과천시	Delphi	LG EDS	SDE, MapObjects, ArcInfo	단일 C/S	△	△	SLC(EDS)
부산광역시	Delphi	LG EDS	SDE, MapObjects, ArcInfo	단일 C/S	△	△	SLC(EDS)
전주시	Visual Basic	SK씨엔씨	SDE, MapObjects, ArcInfo	단일 C/S	×	×	METHOD/1
포항시	Visual Basic	SK씨엔씨	SDE, MapObjects, ArcInfo	단일 C/S	×	×	METHOD/1
여주시	Delphi	한국항공	Mapinfo Spatial ware, Map Basic	단일 C/S	△	△	
성남시	NULL, Visual C++	쌍용정보통신	CARIS INGRES	단일 C/S	△	△	GUIDE
창원시	Delphi	한국항공	ARC/INFO (AML) Mapinfo, Map Basic	단일 C/S	△	△	
수원시	Visual Basic	SK씨엔씨	SDE, MapObjects, ArcInfo	단일 C/S	×	×	자체프로세스
P시	Visual C++	본연구팀	MapObjects, Access DB	여러 개의 컴포넌트들	○	○	RUP

과 같은 표준화된 모델링 언어를 사용하는 컴포넌트 기반 개발 프로세스를 활용하여 지방자치 단체의 UIS를 컴포넌트 형태로 개발하여 해결할 수 있다. 본 연구의 의의는 이러한 가능성을 테스트한 것이다.

한편, 국내에서는 UIS 컴포넌트의 재사용 가능성과 불가능성에 대한 두 가지 의견이 대립되고 있다. 본 시스템 개발 결과 구축되는 UIS 컴포넌트들이 해당 지방자치 단체의 데이터베이스 구조에 상당히 의존적이므로 각종 대장을 활용하는 컴포넌트들은 재사용이 어려우나 지자체들의 대장과 조서에 크게 의존하지 않는 도면관리와 같은 부분은 컴포넌트형태 개발로 재사용이 가능함이 밝혀졌다. UIS 컴포넌트 개발을 통한 완전한 재사용을 위해서는 먼저 정부 차원에서 각종 대장 및 조서와 용어의 표준화가 선행되어야 하나 표준화는 지자체업무의 광범위성과 지자체의 자립성 때문에 쉽게 이루어지지 않을 것이다. 표준화하지 않고 현재 단계에서 UIS 컴포넌트의 재사용을 이룩하는 방안은 정부에서 도면관리와 같이 컴포넌트화가 가능한 부분을 먼저 식별하여 단계별로 UIS 컴포넌트 개발과 재사용이 실현되도록 하는 것이다.

또한, 정부에서는 각 지방자치 단체들이 상수도, 하수도, 지적, 도시계획, 새주소사업 등과 같은 업무를 위해 UIS를 구축할 경우 컴포넌트 형태로 개발하도록 점차 유도하여야 한다. 정부의 자금 지원으로 개발된 컴포넌트들은 중앙의 각 해당 부처에서 관리·유지되어야 하고

다른 지방자치 단체가 유사한 시스템을 개발할 경우 반드시 먼저 등록된 컴포넌트들을 열람한 후 해당 지방자치 단체의 유형에 따라 가장 적합한 컴포넌트들을 선택하여 재사용 하도록 해야한다.

6. 결론

본 연구는 지방자치 단체들의 UIS 구축 비용을 절감하기 위한 목적으로 컴포넌트 형태의 개발 필요성을 설명하였고 컴포넌트 개발을 용이하게 하기 위해 현재 대표적인 컴포넌트 기반 개발방법론인 RUP와 표준 모델링 표기법 UML의 적용 방법을 제시하였다.

국내의 각 지방자치 단체는 현재 UIS 구축시 서로 다른 개발 방법론을 사용하고 있어 소프트웨어 재사용 뿐만 아니라 생성된 산출물들의 재사용도 어렵게 한다. 또한 GUIDE와 METHOD/1과 같은 기존의 구조적인 개발 방법론을 사용하고 있기 때문에 이 방법론들은 UIS 컴포넌트 개발에 적용하기 어렵다. 위의 문제점들을 극복하기 위해 단일화되고 컴포넌트 형태의 개발을 지원하는 컴포넌트 기반 개발방법론이 사용되어야 한다. 그러나, 컴포넌트 기반 개발 방법론에 대한 명확한 적용 방법이 없기 때문에 본 연구에서는 한 예로 RUP를 사용하여 하나의 지자체를 대상으로 적용 지침을 제시하였다.

또한, 본 연구는 국내 78개 지방자치 단체들이 동일 업무를 수행하는 UIS 시스템을 각각 별도로 구축할 경

우 중복 투자로 인한 막대한 액수의 국가재정을 절감하기 위한 방안으로 본 연구에서 직접 시스템 개발을 통해 얻어진 경험을 바탕으로 향후 국가차원에서 단계별 UIS 컴포넌트의 개발 정책 및 제공 방안을 제시하였다.

본 연구는 한 예로 P시 UIS 상수도 업무를 선정하여 모델링을 실시하였고 그 모델링 결과를 토대로 누수관리 중심의 P시 UIS 상수도 시스템을 개발하였다. 이 시스템은 윈도우 기반 하에 동작하는 3개의 ActiveX 컨트롤로 구성된다. ActiveX 애플리케이션은 Automation, ActiveX 컨트롤, OLE Document의 세 가지 형태로 개발될 수 있다[23]. 본 연구에서는 ActiveX 컨트롤만을 고려하였으며 ActiveX 컨트롤 개발 방법인 MFC와 ATL 중 구현의 용이성 때문에 MFC 방법을 사용하였다.

한편, 본 연구의 수행 결과 프로세스의 방대함으로 RUP 적용이 상당히 어렵다는 점과 RUP 프로세스를 따를 경우에 생산해야 할 산출물들이 너무 많다는 점이 입증되었다. RUP를 사용할 경우 제시된 산출물들은 개발할 시스템의 응용 분야나 개발 프로젝트의 크기·기간 등에 따라 자유롭게 선택되어야 한다. 그러나, UIS분야에서 RUP 산출물들의 다양성은 산출물의 재사용을 어렵게 하기도 하며 이는 RUP의 장점이자 단점이 되기도 한다. 지자체 UIS를 개발할 때 생성되는 산출물들의 재사용을 위해서는 현재 RUP가 지원하지 못하는 기능들을 보완하고 산출물들을 적절하게 한정된 새로운 국내 실정에 적합한 표준화된 UIS 컴포넌트 기반 개발 프로세스가 요구된다. 또한, 본 연구는 이를 위한 선행연구로서 행해졌다.

대표적인 UML과 RUP의 문제점은 다음과 같다. 현재 UML은 데이터 모델링 표기법과 데이터 흐름을 나타낼 표기법이 없으며 RUP는 프로세스가 너무 방대하여 실제 적용이 어렵고 윈도우 및 아이콘 설명 부분이 없고 용어해설에 대한 명백한 분류기준이 미흡하여 일반 변수와 전문용어의 구분을 어렵게 한다는 것이다. 향후연구과제로 이러한 문제점을 개선한 새로운 프로세스 제시가 이루어질 것이다.

참 고 문 헌

- [1] Bruce E. Davis, *GIS: A Visual Approach*, pp. 10-30, OnWord Press, 1996.
- [2] 한국통신 선로기술연구소, *GIS DB용 데이터 포맷 변환 도구 개발*, pp. 13-42, 1996.
- [3] Open GIS Consortium, Inc., *OpenGIS Abstract Specification and Implementation Specification*, <http://www.opengis.org/techno/specs.htm>, 1999.
- [4] International Organization for Standardization, *ISO/TC 211 Geographic information/Geomatics*, <http://www.statkart.no/isotc211/>, 1999.
- [5] 정보통신부 정책보고서, 국가지리정보체계(NGIS) 구축사업 활성화 방안, 1999.
- [6] Robert Orfali et. al., *The Essential Distributed Objects Survival Guide*, Wiley Computer Publishing, 1997.
- [7] Robert Hartman, *Focus on GIS Component Software*, OnWord Press, 1997.
- [8] Kendall Scott and Martin Fowler, *UML Distilled : Applying the Standard Object Modeling Language*, Addison Wesley, Inc., 1997.
- [9] Hans-Erik Eriksson and Magnus Penker, *UML Toolkit*, pp. 17-24, Addison Wesley, Inc., 1998.
- [10] Craig Larman, *Applying UML, Patterns, and Object-Oriented Analysis and Design*, pp. 17-27, Prentice Hall, 1997.
- [11] 조은숙외 2인, "UML을 기반으로 한 실무 중심의 객체지향 방법론", 정보처리논문지, 제6권, 제3호, pp 622-632, 1999.
- [12] Philippe Kruchten, *The Rational Unified Process*, pp. 5-15, Addison Wesley, Inc., 1999.
- [13] Ivar Jacobson, Grady Booch, and James Rumbaugh, *The Unified Software Development Process*, pp. 85-107, Addison Wesley, Inc., 1999.
- [14] Terry Quatrani, *Visual Modeling with Rational Rose and UML*, pp. 3-12, Addison Wesley, Inc., 1998.
- [15] Doug Rosenberg and Kendall Scott, *Use Case Driven Object Modeling With UML*, Addison Wesley, Inc., 1999.
- [16] Rational Software Corporation, *Rational Unified Process 5.0*, Manual CD, 1998.
- [17] 한동대학교 GIS연구소, 포항시 GIS 기본계획수립연구 보고서, pp. 97-139, 1998.
- [18] 국토개발연구원, 공공GIS활용체계 구축계획 수립연구, 국가GIS지원 연구사업 보고서, p. 61, 1997.
- [19] 포항시, 포항시 지하시설물도 전산화 사업 보고서, 1999.
- [20] ESRI, Inc., *Building Application with MapObjects*, 1996.
- [21] ESRI, Inc., *MapObjects Programmer's Reference*, ESRI, Inc., 1996.
- [22] 국토연구원, 지하시설물도 전산화사업의 UIS 연계전략에 관한 연구, 1999.
- [23] Don Box, *Essential COM*, Addison Wesley, Inc., 1998.



조혜경

1996년 2월 전북대학교 전산통계학과 석사 졸업. 1996년 3월 ~ 현재 전북대학교 전산통계학과 박사과정. 1996년 3월 ~ 1997년 2월 전북대학교 컴퓨터학과 조교. 1998년 1월 ~ 1999년 12월 한동대학교 GIS연구소 연구원. 2000년 3월 ~ 현재 포항1대학 전산정보처리과 겸임교수. 관심분야는 지리정보시스템, 소프트웨어공학, 전자상거래, 멀티미디어



유철중

1982년 2월 전북대학교 전산통계학과 졸업(이학사). 1985년 8월 전남대학교 대학원 계산통계학과 졸업(이학석사). 1994년 8월 전북대학교 대학원 전자계산학과 졸업(이학박사). 1982년 9월 ~ 1985년 3월 전북대학교 전자계산소 조교. 1985년 4월 ~ 1996년 12월 전주기전여자대학 전자계산과 부교수. 1997년 1월 ~ 현재 전북대학교 자연과학대학 컴퓨터학과 조교수. 관심분야는 소프트웨어공학, 에이전트공학, 컴포넌트기술, 분산객체기술, 지리정보시스템, 멀티미디어, 인지과학



장옥배

1966년 고려대학교 수학과 졸업. 1980년 조지아 주립대, 오하이오 주립대 박사과정 수료. 1987년 산타바바라대학교 졸업(Ph.D.). 1990년 ~ 1991년 영국에딘버러대학교 객원교수. 1980년 4월부터 현재 전북대학교 컴퓨터학과 교수. 관심분야는 소프트웨어공학, 전산교육, 수치해석, 인공지능 등



김영섭

1980년 연세대학교 전자공학과 학사졸업. 1983년 The University of Alabama 전기.전산기 공학과 석사 졸업. 1990년 The University of Alabama 전자.전산기 공학과 박사졸업. 1984년 ~ 1995년 Intergraph社(Alabama, USA) GIS and Imaging Division 개발부장. 1995년 ~ 현재 한동대학교 전산전자공학부 교수. 관심분야는 지리정보시스템(GIS), 영상처리, 소프트웨어공학, 인터넷 보안