

UML기반의 요구사항 추적 매트릭스 설계

A Study on the RTMD(Requirement Tracing Matrix Design) Base on UML

허계범*, 김영규*, 양동일**

Kwae-Bum Heo*, Young-Gyu Kim* and Dong-Il Yang**

요 약

오늘날 기업의 비즈니스 환경은 시장, 기술 및 사회에서 오는 여러 가지 요인들로 인하여 한층 더 복잡해지고 있다. 비즈니스 패러다임의 변화에 따른 빠른 시장 대응(Time-to Market)에의 요구 증가 및 비용 감소, 높은 생산성에 대한 요구 증가, 재사용성 증대를 통한 유연한 애플리케이션을 개발할 수 있는 절차와 매커니즘에 대한 필요성이 대두되고 있다. UML 기반의 객체지향 소프트웨어 개발 생명주기에서 요구사항 추적 매트릭스 설계(RTMD : Requirement Tracing Matrix Design) 모델을 제시하여 실제업무에 적용하였다. 이 방법은 소프트웨어 개발의 생산성 향상 및 유지보수 비용을 절감하는 효과가 있다. 그러나 관련 문서들의 완전성(completeness), 일관성(consistency), 정확성(correctness)을 검증하기 위한 절차 및 규칙이 미흡하며, 또한 이들 명세서들은 보다 체계적으로 관리할 수 있는 자동화된 시스템을 필요로 한다.

Abstract

Design traceability has been widely recognized as being an integral aspect of software development. Also many projects for software application development have failed because that the projects did not analyze user requirements sufficiently and reflected them in the projects adequately.

This thesis proposes a specification and tracing technique for object-oriented analysis and design. The subject of this study is the artifacts of UML development methodology. Therefore, to improve the quality of the software development cycle, in short, the system requirement of customers will be able to respond quickly.

Key words : RTMD, UML, CBD, object-oriented

I. 서 론

오늘날 기업의 비즈니스 환경은 시장, 기술 및 사회에서 오는 여러 가지 요인들로 인하여 한층 더 복잡

해지고 있다. 비즈니스 패러다임의 변화에 따른 빠른 시장 대응(Time-to Market)에의 요구 증가 및 비용 감소, 높은 생산성에 대한 요구 증가, 재사용성 증대를 통한 유연한 애플리케이션을 개발할 수 있는 절차와 매커니즘에 대한 필요성이 대두되고 있다

* 한림성심대학 인터넷비즈니스과

** 강원대학교 컴퓨터과학과

· 제1저자 (First Author) : 허계범

· 교신 저자 : 양동일

· 투고일자 : 2011년 5월 24일

· 심사(수정)일자 : 2011년 5월 24일 (수정일자 : 2011년 6월 17일)

· 게재일자 : 2011년 6월 30일

[1],[2],[3],[4].

소프트웨어 프로세스는 일반적으로 문제영역에 대한 사용자 요구사항을 도출하는 단계에서 시작하여 이를 바탕으로 분석 단계, 설계 단계를 거쳐 코딩, 테스트의 단계를 거치게 된다. 이러한 프로세스의 단계 중에서도 사용자 요구사항을 분석하고 설계하는 초기 단계가 무엇보다 중요하다. 즉 소프트웨어 프로젝트의 실패요인 중에서 54%가 요구사항관리 미흡에서 발생하며, 이중 22%는 요구사항 변경관리에서 비롯된다[11].

본 논문에서는 UML기반의 객체지향 소프트웨어 개발 생명주기에서 산출물, 즉 소프트웨어 개발 제안 요청서, 제안서, 사업 착수 계획서, 요구사항 정의서, 분석 모델링, 설계 모델링 단계의 매트릭스를 설계하여 보다 체계적인 요구사항들을 관리하고자 한다.

II. 관련 연구

2-1 UML

UML(Unified Modeling Language)은 객체지향 소프트웨어 시스템을 개발하는데 사용되는 모델링 언어로 시스템의 가공물을 가시화, 구축, 문서화하고, 그 시스템을 사용하는 조직의 업무를 모델링하는 방법을 제공한다. 대부분의 방법론은 모델링 언어와 프로세스로 구성되며 기본적으로 모델링 언어와 프로세스에서 일관성을 갖는다. 모델링 언어는 방법론에서 분석 및 설계사항으로 표현하기 위하여 사용하는 그래픽 표기법이며 프로세스는 설계를 수행하는 동안에 각 단계에서 취해야 할 사항에 대한 지침이라 할 수 있다. 즉 방법론을 사용하는 것은 대부분 프로세스보다는 모델링 언어를 사용한다는 것을 의미하며 의사소통의 중요한 부분이다 [1],[5],[6],[7],[8],[12],[13].

2-2 CBD 방법

CBD(Component Based Development) 방법은 시스템이나 소프트웨어를 구성하는 각각의 컴포넌트를 만들고 조립하여 또 다른 컴포넌트나 소프트웨어를 만들기 위한 작업, 절차, 기법, 산출물을 정의하여 효율적으로 수행할 수 있도록 지원한다. CBD 방법은

재사용 가능한 소프트웨어 모듈을 제작하여 기계 부품과 같이 이를 조립하여 보다 복잡한 소프트웨어를 만드는 방식을 말한다. CBD 방법은 일반적으로 반복적 개발을 원칙으로 하고 있으며, 컴포넌트를 중심으로 여러 차례의 반복 프로세스와 아키텍처를 정제함으로써 위험을 최소화시켜 프로세스에 신뢰성을 제공하여 소프트웨어의 복잡성과 생산성 문제를 해결하고자 하는 일종의 새로운 개발 패러다임을 제시하였다[1],[3],[6],[7],[8],[10].

2-3 요구사항 추적기법

요구사항 추적은 요구사항의 일관성(consistency)을 검증하고 완전성을 확인하는 V&V 모델(Verification & Validation Model)에 기반 하는 것으로 소프트웨어 개발 생명주기의 각 단계에서 요구사항의 일관성 및 완전성을 검증하는 것이다. 이와 같은 추적기법에 관한 방법에는 정보검색(information retrieval) 추적기법, 이벤트기반(event-based) 추적기법, 시나리오기반(scenario-based) 추적기법이 있다 [3],[9].

III. 요구사항 추적 매트릭스 설계

3-1 RTMD 모델

사용자 요구사항 분석 활동은 분석준비 활동에서 수집된 자료를 바탕으로 사용자의 기능적, 비기능적 요구사항을 정형화된 형태로 정의한다. 작성된 사용자 요구사항 정의내용을 중심으로 시스템을 직접 사용하게 될 사용자와 시스템과 상호작용하는 외부 시스템을 파악한다. 여기서 파악된 사용자와 외부시스템을 기반으로 시스템이 사용자에게 제공해야 하는 서비스 유형을 정의한다.

본 논문에서는 그림 1, 그림 2, 그림 3과 같은 UML 기반의 객체지향 소프트웨어 개발 생명주기에서 요구사항 추적 매트릭스를 설계(RTMD : Requirement Tracing Matrix Design)하여 소프트웨어 개발의 생산성 향상에 그 목적이 있다.

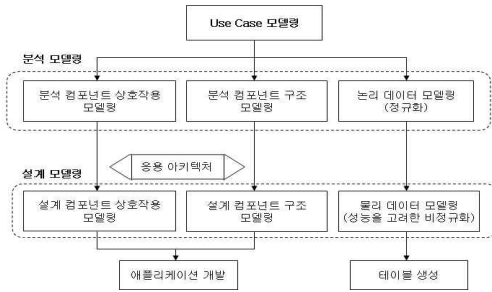


그림 1. RTMD 진행 절차
Fig. 1 Procedure of RTMD Progress

UML 2.0 다이어그램은 구성 요소들을 그래픽으로 표현한 것으로 다음과 같은 요소로 구성되며 2개의 카테고리 내의 다이어그램을 정의할 수 있다. 즉 시스템의 정적인 구조를 표현하는 구조적 다이어그램 (structural diagrams)과 동적인 행동의 다른 면을 표현하는 행위 다이어그램(behavior diagrams)이 있다. 그림 2는 UML 2.0의 다이어그램 구성 요소를 나타내고 있다[13].

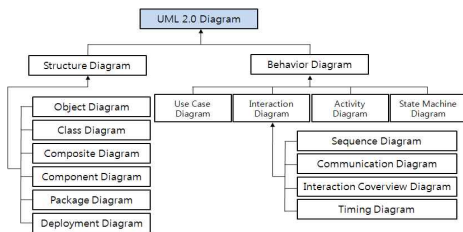


그림 2. UML 2.0의 다이어그램
Fig. 2 Diagram of UML 2.0

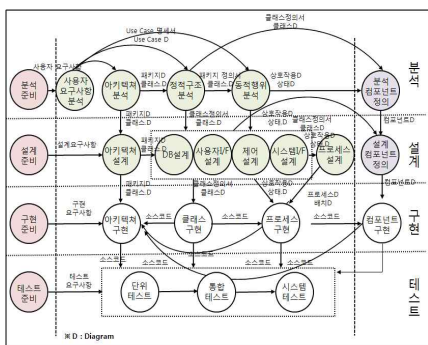


그림 3. 표준 객체지향 개발 모형의 주요 관점
Fig. 3 Viewpoints of the Standard Model for Object-Oriented System Development Activities

표준 객체지향 개발 모형의 산출물은 기본적으로 UML을 중심으로 정의하고, UML에서 정의하고 있지 않은 산출물에 대해서는 필수 항목을 포함하는 일반

적인 산출물로 정의하였다. 활동간 주요 산출물 연관성은 그림 3과 같으며, 각 버블은 표준 객체지향 개발 모형의 활동을 의미하며, 화살표는 활동간의 산출물 연관성을 나타낸다. 여기에서는 표준 객체지향 개발 모형의 산출물 중 UML기반의 주요 산출물의 활동간 연관성만을 나타내었다. 그 이유는 각 단계의 준비부분을 점선으로 구별하여 나타낸 것은 점선 오른쪽의 활동들이 시간적 수행순서를 가지지 않는 것에 반해, 각 단계의 준비활동은 시간적 순서에 의존적이기 때문이다. 또한 컴포넌트 관련 활동을 점선으로 구별하여 나타낸 것은 이 활동이 필수적인 활동은 아니며, 선택적으로 수행될 수 있다는 것을 의미한다.

3-2 Use Case 모델링

Use Case 모델링은 시스템을 사용하는 사용자 및 연계가 필요한 외부 시스템과 시스템이 제공하는 기능 사이의 상호작용을 모델링하는 것이다. Use Case 모델링 단계에서는 시스템 내부에 구축되어야 할 클래스들을 식별할 수 있는 기반을 제공하며, 사용자와 시스템간의 상호작용 형태를 분석함으로써 구체적인 입출력 데이터 및 사용자 인터페이스 정보를 제공한다.

표 1. Use Case 모델링의 산출물
Table 1. Output List of Use Case Modeling

산출물 명	내용
Use Case 다이어그램	Actor 상호간의 관계, Actor와 시스템이 제공하는 기능사이의 관계 및 기능 상호간의 관계를 표현한 다이어그램
Actor 카탈로그	Actor의 역할을 정의한 문서
Use Case 명세서	시스템이 제공하는 기능의 세부적인 처리흐름을 기술한 문서

3-3 분석 모델링

분석 모델링이란 Use Case를 통해 수집되고 정리된 요구사항을 정제하고 구조화하는 과정을 의미한다. 요구사항수집이 소프트웨어 사용자의 관점에 따라 진행하는데 반해 분석부터는 개발자의 관점에서 시스템을 설명하게 된다. 분석 모델링은 Use Case 명

세서에 나타난 이벤트의 흐름을 분석하여 이를 Sequence Diagram으로 표현하고, Sequence Diagram에 나타난 객체를 컴포넌트로 추상화하는 과정이다. 또한 데이터베이스 구축을 위한 논리 데이터 모델링 작업도 수행하게 된다.

표 2. 분석 모델링의 산출물
Table 2. Output List of Analysis Modeling

산출물 명	내 용
분석 모델 패키지	분석 모델을 관리하기 용이한 단위로 분할한 모델의 관리 구조이다.
Use Case Realization	하나의 Use Case가 실행되는 데 어떤 컴포넌트가 필요하며, 그 컴포넌트 간의 상호작용이 어떻게 이루어져 원하는 기능을 수행하는 가를 표현하는 것으로, 하나의 Use Case Realization은 하나 이상의 Component Diagram과 Interaction 다이어그램으로 구성된다.
분석 컴포넌트 상호작용 모델	분석 컴포넌트 상호작용 모델은 Use Case 명세서에 나타난 이벤트의 흐름에 따라 각각의 객체가 상호간에 어떤 메시지를 송수신하는지를 정의하는 것이다.
분석 컴포넌트 구조 모델	분석 컴포넌트 구조 모델은 하나 이상의 분석 컴포넌트 상호작용 모델에서 각각의 시나리오에 참여하는 객체들을 추상화하여 컴포넌트로 도출하여 그들 사이의 관계를 표현한 모델이다.
논리 데이터 모델	논리 데이터 모델은 관리해야 하는 데이터가 중복 없이 관리될 수 있도록 조직화하여 표현한 모델이다.

분석 모델링에서는 컴포넌트에 대한 세 가지 스테레오 타입을 정의해서 사용하며, 각 스테레오 타입에 대한 설명은 아래와 같다.

표 3. 컴포넌트 스테레오 타입에 대한 내용
Table 3. Description of Component Stereotype

컴포넌트 스테레오 타입	내 용
바운더리 컴포넌트 (Boundary Class)	시스템 외부의 Actor와 메시지를 주고받는 컴포넌트, 즉 보통 화면 컴포넌트들이나 다른 시스템 또는 장치(device)와 인터페이스하는 컴포넌트를 의미한다.
컨트롤 컴포넌트 (Control Class)	Use Case에 종속적인 Sequence를 관리하는 컴포넌트의 유형으로 하나의 Use Case에 하나의 컨트롤 컴포넌트를 생성하여 사용한다.
엔티티 컴포넌트 (Entity Class)	업무에서 필요로 하는 정보를 저장하고 있거나, 정보를 제공하는 역할을 수행하는 컴포넌트로 프로그램이 종료한 후에도 영구적으로 저장되어야 하는 정보를 관리한다.

3-4 설계 모델링

설계 모델링단계에서는 Use Case 모델링과 분석

모델링 과정을 통하여 도출해 낸 비즈니스 성격의 분석 모델을 이에 해당하는 오퍼레이션을 시스템 구현과 밀접한 형태의 모델로 변환하여 정제 하는 단계이다. 설계 컴포넌트 상호작용 모델링에서는 분석 단계에서 Boundary, Control, Entity 세 가지 스테레오 타입으로 분류하여 정의한 컴포넌트는 설계 모델에서 더 작은 단위로 분할되고 정제된다. 분석 모델에서 작성한 분석 Sequence Diagram을 설계 Component Diagram으로 변환되며, 분석 모델에서 만들어진 Boundary 컴포넌트는 설계 모델에서 JSP 파일로 변환이 된다. 이때 하나의 Boundary 컴포넌트는 하나 이상의 JSP 파일로 변환이 된다. Boundary 컴포넌트에서 호출하던 Control 컴포넌트의 오퍼레이션은 설계 모델에서 하나의 Command 컴포넌트로 분화된다. 또한 분석 모델에서 Entity 컴포넌트는 설계 모델에서 하나의 DAO로 합쳐지며 각 Entity 컴포넌트에 대한 입력, 수정, 삭제, 조회에 대한 요청은 DAO 컴포넌트의 오퍼레이션으로 변환된다.

표 4. 설계 모델링의 산출물
Table 4. Output List of Design Modeling

산출물 명	내 용
설계 모델 패키지	분석 모델을 관리하기 용이한 단위로 분할한 모델의 관리 구조이다.
Use Case Realization	하나의 Use Case가 실행되는 데 어떤 컴포넌트가 필요하며, 그 컴포넌트 간의 상호작용이 어떻게 이루어져 원하는 기능을 수행하는 가를 표현하는 것으로, 하나의 Use Case Realization은 하나 이상의 Component Diagram과 Interaction Diagram으로 구성된다.
설계 컴포넌트 상호작용 모델	설계 컴포넌트 상호작용 모델은 Use Case 명세서에 나타난 이벤트의 흐름에 따라 각각의 객체가 상호간에 어떤 메시지를 송수신하는지를 정의하는 것이다.
설계 컴포넌트 구조 모델	설계 컴포넌트 구조 모델은 하나 이상의 설계 컴포넌트 상호작용 모델에서 각각의 시나리오에 참여하는 객체들을 추상화하여 컴포넌트로 도출하여 그들 사이의 관계를 표현한 모델이다.
물리 데이터 모델	물리 데이터 모델은 관리해야 하는 데이터가 중복 없이 관리될 수 있도록 조직화하여 표현한 모델이다.

설계 Use Case Realization에서 Use Case와 Use Case Realization을 분리하는 목적 중의 하나는 이미 Baseline 화 된 Use Case에 영향 없이 설계를 수정할 수 있는 유연성과 분석 단계의 Use Case가 설계 단계에서 어떠한 형태로 변화되는지를 추적할 수 있는 Trace를 찾기 위함이다. 따라서 Use Case View에 존

재하는 Use Case와 동일한 단위로 Use Case Realization을 만들고 Dependency 관계로 Relation을 만들어 준다. 대부분의 경우 Use Case와 분석/설계 Use Case Realization은 1:1로 매핑 된다. 그러나, 분석 단계 시의 Use Case가 설계로 발전되면서 분리되거나 병합이 일어날 수 있다. 만약에 분석 단계에서 Use Case가 도출되지 않았다면 분석 단계의 Use Case를 다시 잡아 분석 단계 이후의 Process를 진행하도록 하고, 그 외에 도출은 되었으나 Realization 시 변경이 일어나면 이를 Use Case Trace Diagram을 이용하여 추적성을 관리 할 수 있도록 한다. 설계 단계의 Use Case Realization은 통합 모델러가 일괄적으로 작성하도록 한다.

3-5 UML 다이어그램의 일관성 및 완전성 검증

사용자가 작성한 모델의 정확함을 파악하기 위한 성질에는 완전성(completeness), 일관성(consistency), 정확성(correctness)등의 3가지가 있다.

완전성은 사용자 요구사항과 다이어그램의 일치성을 파악하기 위한 성질이다. 완전성은 사용자 요구사항의 분석에 따라서 당초에 목표한 기능을 제대로 수행할 수 있도록 UML 다이어그램의 설계가 되었는지, 즉 사용자의 요구사항이 제대로 반영되었는지를 파악하는 성질이다. 일관성은 다이어그램간의 일치성을 파악하기 위한 성질이다. 일관성은 하나의 사용자 요구사항으로부터 유도된 UML 다이어그램의 형태는 비록 다르더라도 그 의미하는 바가 같은가를 파악하는 성질이다. 정확성은 다이어그램과 UML 표준의 일치성을 파악하기 위한 성질이다. 정확성은 사용자 요구사항을 기반으로 작성된 UML 다이어그램이 UML의 표준에 적합하게 작성이 되었는가를 파악하는 성질이다.

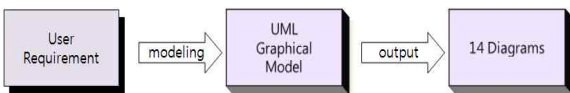


그림 4. UML의 주요 관점
Fig. 4 Viewpoints of Unified Modeling Language

UML(Unified Modeling Language)은 추상적인 요구

사항을 가시화하고, 사용자 요구사항을 보다 완전하게 명세하며, 다양한 프로그래밍 언어와 연결되어 코드 생성과 산출물들을 문서화하는데 도움을 준다. UML은 빌딩블록, 규칙(rules), 공통메커니즘과 같은 세 가지 주요 구성요소로 이루어져 있다. 빌딩블록은 다시 사물(things), 관계(relationships), 다이어그램으로 구성되어있으며 사용자가 모델링시에 사용하게 되는 각각의 구성요소를 이루게 된다.

본 본문에서는 UML 다이어그램의 정확함에 관한 성질 중 완전성과 일관성 그리고 정확성을 검증하기 위하여 메타모델기반의 검증방법을 적용한다[2],[11]. UML 다이어그램의 메타모델을 유도하고, 메타 모델로부터 일관성 검증을 위한 관계요소를 도출하며, 메타모델은 각각의 다이어그램을 구성하는 요소를 클래스로 표현하고, 구성요소 사이의 관계를 연관, 실체화, 의존, 일반화 관계로 표현한다.

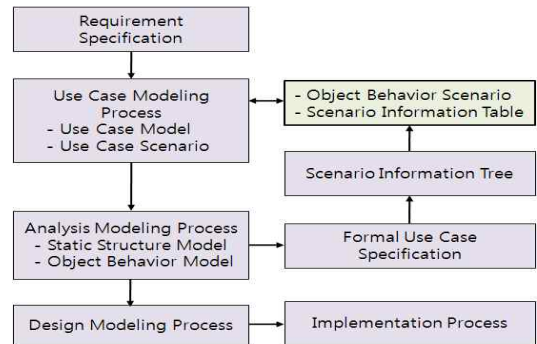


그림 5. 검증 절차
Fig. 5 Procedure for Verification

Use Case Diagram은 시스템과 시스템 사용자, 그리고 그 시스템 사용자들과 Use Case간의 관계를 나타내는 Use Case들의 집합을 다이어그램으로 나타낸다. Use Case는 시스템이 수행하는 일련의 행위, 외부 사용자의 관점에서 시스템이 어떻게 사용되는지의 과정을 서술적으로 기술한 것으로, 사용자 입장에서 시스템이 기능적으로 분할된다. Use Case와 Use Case 사이에는 일반화, 포함/확장(include/extend) 관계를 가지며, 행위자와 행위자 사이에는 일반화 관계를 가진다. Use Case는 하위의 구성요소로서 사건 흐름과 사건 예외 흐름을 가진다. Use Case는 Sequence Diagram의 시나리오와 인스턴스 관계를 가지며, 협력을 매개로 하여 Class Diagram의 오퍼레이션과 대응 관계를

가진다.

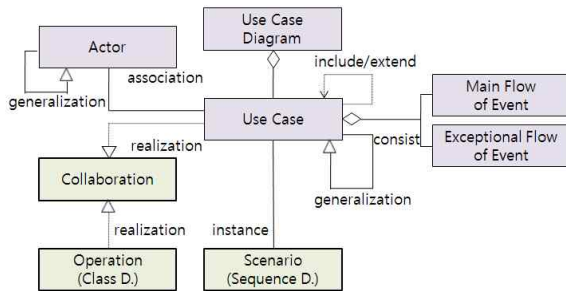


그림 6. Use Case Diagram의 메타모델
Fig. 6 Meta-Model of Use Case Diagram

Class Diagram은 시스템 객체들의 타입을 기술하고, 그들 간의 다양한 정적인 관련성을 나타낸다. 또한 클래스의 속성과 연산, 그리고 객체들의 연결 방법에 적용될 제약조건에 대해서도 기술한다. Class Diagram은 클래스, 패키지, 인터페이스, 속성, 오퍼레이션 등의 구성요소를 가지고 있으며, 일반화(generalization), 의존(dependency), 연관(association) 관계를 가진다. 하나의 클래스는 객체와 인스턴스 관계를 가지며 컴포넌트와 의존관계가 있으며 인터페이스와 실체화 관계를 가진다. 따라서 클래스는 객체, 컴포넌트등과 대응관계를 가지며 오퍼레이션은 하나의 Use Case와 대응 관계를 가진다.

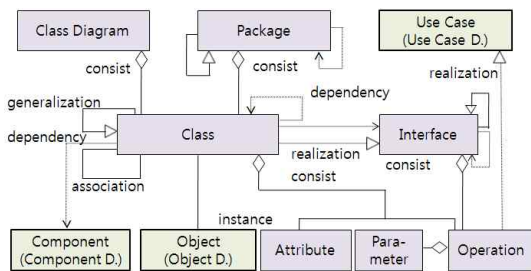


그림 7. Class Diagram의 메타모델
Fig. 7 Meta-Model of Class Diagram

Component Diagram은 Use Case Diagram, Class Diagram, Sequence Diagram등을 통해 산출되는 실행 파일, 라이브러리, 테이블, 파일과 같은 구체화되고 실체적인 것들의 관계를 표현하는 다이어그램이다. Component Diagram은 컴포넌트, 인터페이스를 중심으로 구성되며, 컴포넌트와 컴포넌트 사이에는 일반화, 의존, 연관관계를 가지며, 컴포넌트와 인터페이

스 사이에는 실체화(realization), 의존 관계를 가진다.

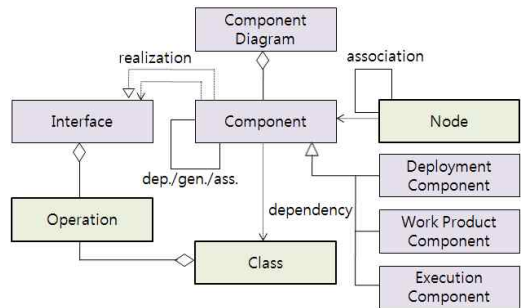


그림 8. Component Diagram의 메타모델
Fig. 8 Meta-Model of Component Diagram

Sequence Diagram에서는 시스템 사용자의 상호작용과 그들에 의해서 발생하는 연산이 표현된다. 여기서는 어떤 Use Case의 특정 시나리오를 위해서 외부 시스템 사용자가 생성하는 이벤트들과 그들이 순서, 시스템간의 이벤트들을 포함한다. Sequence Diagram은 객체, 링크, 메시지를 중심으로 구성되며, 행동에 참여하는 객체들을 우선 나열하고 객체 아래로 생명선(lifeline)을 그어 객체가 활동하는 시기를 제어초점으로 표현하고, 제어초점 간에 호출을 통해서 메시지를 주고받는다. Sequence Diagram은 하나의 시나리오가 구체화된 형태로서, Object Diagram, Class Diagram 등과 연관관계를 가진다. 따라서 Sequence Diagram에 나타나는 객체와 대응하는 객체가 Object Diagram에 나타나야 하며, 메시지에 해당하는 오퍼레이션이 Class Diagram에 나타나야 한다.

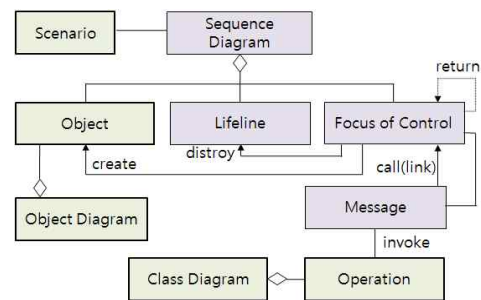


그림 9. Sequence Diagram의 메타모델
Fig. 9 Meta-Model of Sequence Diagram

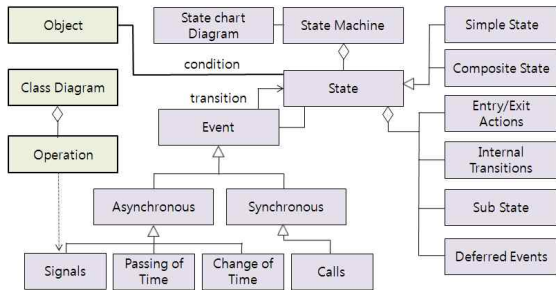


그림 10. State Chart Diagram의 메타모델
Fig. 10 Meta-Model of State Chart Diagram

State Chart Diagram은 객체의 생명주기를 나타내며, 이벤트에 의해 변화하는 객체의 상태를 도식화한 다이어그램으로 특정 객체가 취할 수 있는 모든 상태를 나타내고, 그 객체에 도달하는 이벤트들의 결과로서 객체의 상태가 어떻게 변하는지를 표현한다. 즉 State Chart Diagram은 객체의 상태머신(state machine)을 나타내며, 상태는 일반적인 형태의 단순상태와 (simple state)와 상태 내에 또 다른 상태를 포함하고 있는 복합 상태(composite state)로 나눌 수 있다. 사건은 비동기사건(asynchronous event)과 동기사건(synchronous event)으로 나눌 수 있다. 비동기사건은 임의의 시간에 발생할 수 있는 사건으로서 신호(signal), 시간경과(passing of time), 상태변화(change in state)와 같은 것이 있으며, 동기사건은 오퍼레이션을 기동하기 위한 사건으로서 호출동작(call-action)과 같은 것이 있다.

점선으로 표현된 관계는 구조적 다이어그램 또는 행위적 다이어그램 각각에 속하는 다이어그램간의 관계를 표현한 것이다.

IV. 적용 예

4-1 적용 업무

본 논문에서의 적용 업무는 행정정보 공유센터 시스템으로 응용시스템 및 데이터베이스를 재구성하여 보다 안정적인 서비스를 제공하는데 있다. 그림 12는 행정정보 공유 웹서비스 시스템을 나타내고 있다.

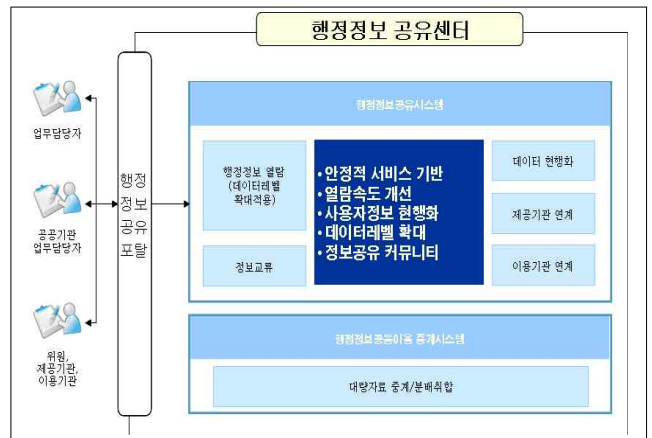


그림 12. 적용 업무
Fig. 12 Application Work

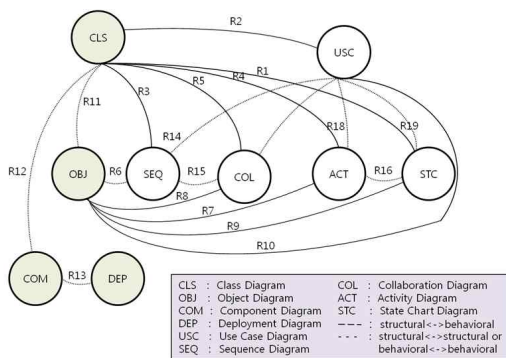


그림 11. UML 다이어그램의 관계
Fig. 11 Relationship of UML Diagram

그림 11은 UML 다이어그램들 간의 관계를 나타낸 것으로, 실선으로 표현된 관계는 구조적 다이어그램과 행위적 다이어그램간의 관계를 표현한 것이고,

4-2 적용 업무 범위

표 5. 적용 업무의 산출물

Table 5. Output List of Application Work

단계	활동	산출물	내용
사업 추진	사업 수행 준비	제안요청서	시스템개발의 대상이 되는 제안업무에 대하여 서술한 문서로서, 전체적인 개발시스템에 관련된 업무의 목적 및 배경, 목표, 제안 범위, 사업 추진 방향 및 전략 등이 포함되어야 한다.
		제안서	시스템개발의 대상이 되는 업무에 대한 제안 요청서를 바탕으로 기술한 문서로서, 전체적인 업무에 관련된 목적 및 배경, 제안 범위, 사업 추진 방향 및 전략 등이 포함되어야 한다.
		사업 착수 계획서	시스템 개발활동을 수행하기 위한 도식화한 문서로써 조직도 및 개발 일정 계획 등이 포함되어야 한다.
분석	사용자 요구사항 분석	사용자 요구사항 정의서	업무 기술서를 바탕으로 시스템에 대한 사용자의 기능적 요구사항 및 비기능적 요구사항을 기록한 문서이다.
		기능 분해도	정보화 시스템을 구축하는 가장 최저 단위의 단위 프로세스를 도출하기 위하여 전체 업무의 체계를 프로세스를 중심으로 도식화하여 업무에 대한 기능을 파악하는 것. 즉 단위 프로세스 도출과 업무 기능을 명확하게 하는 목적이 있다.
		Use Case 명세서	하나의 Use Case에 대하여 사용자와의 상호작용 내용 및 내부 처리의 내용을 기술한 문서이다.
	정적구조 분석	Class Diagram	문제영역 분석을 통해 추출된 클래스 및 그들간의 관계를 도식화한 다이어그램으로, UML의 표기법을 따른다.
		Boundary Class	시스템 외부의 Actor와 메시지를 주고받는 컴포넌트, 즉 보통 화면 컴포넌트들이나 다른 시스템 또는 장치(device)와 인터페이스하는 컴포넌트를 의미한다.
		Control Class	Use Case에 종속적인 Sequence를 관리하는 컴포넌트의 유형으로 하나의 Use Case에 하나의 컨트롤 컴포넌트를 생성하여 사용한다.
		Entity Class	업무에서 필요로 하는 정보를 저장하고 있거나, 정보를 제공하는 역할을 수행하는 컴포넌트로 프로그램이 종료한 후에도 영구적으로 저장되어야 하는 정보를 관리한다.
동적행위 분석	Sequence Diagram	클래스간의 상호작용을 시간의 흐름에 따른 이벤트의 전달로 도식화한 다이어그램으로 UML의 표기법을 따른다.	
설계	사용자 인터페이스 설계	Class Diagram	설계단계에서 새롭게 정의된 클래스 및 관계를 추가한 다이어그램으로 UML의 표기법을 따른다.
	데이터 베이스 설계	Class Diagram	설계단계에서 새롭게 정의된 클래스 및 관계를 추가한 다이어그램으로 UML의 표기법을 따른다.
		데이터 객체 구조도	엔터티 객체를 해당 DBMS에 사상하기 위한 논리적 스키마를 테이블 형태로 나타낸 문서이다. 구성필드의 이름, 타입, 길이가 명시된다.
		데이터베이스 명세서	각 지역에 따른 데이터베이스의 이름과 주요 트랜잭션이 표기된 테이블 형태의 문서이다.
	제어설계	Class Diagram	설계단계에서 새롭게 정의된 클래스 및 관계를 추가한 다이어그램으로 UML의 표기법을 따른다.
	시스템 인터페이스 설계	Class Diagram	설계단계에서 새롭게 정의된 클래스 및 관계를 추가한 다이어그램으로 UML의 표기법을 따른다.
동적행위 분석	Sequence Diagram	설계단계에서 새롭게 정의된 클래스간의 상호작용을 시간의 흐름에 따른 이벤트의 전달로 도식화한 다이어그램으로 UML의 표기법을 따른다.	

4-3 요구사항 추적 매트릭스 설계(RTMD)

표 6. 요구사항 추적 매트릭스 설계(1)

Table 6. RTMD(1)

구분	제안요청서	ID	제안서	제목	페이지	사업착수계획서		페이지
	내용		해당목차			해당목차	제목	
공 통	이용의 편리성을 감안하여 기존 시스템의 문제점을 파악하여 사용자들의 편리성 및 신속성을 위주로 구축하여야 하며, 구축과정에 UI전문가와 실제 민원처리담당 공무원을 참여시켜 사용자 환경에 대한 검증 단계를 거쳐야 한다.	RFP0301	III-1.3.6.1	개요	III-112	4.1.1.나 4.1.3.마	사용자 중심의 UI 개선	4, 16
	각종 지침, 교재, 홈페이지 등의 구성은 UI전문가가 참여하여 설계해야 하며 사용자의 검증을 거친 후 결정해야 한다.	RFP0302	III-1.3.6.2.2	구성요소 및 구축방안	III-114	4.1.1.나 4.1.3.마	사용자 중심의 UI 개선	4, 16
	모든 데이터의 자료 전송은 안전한 보안채널을 통해 이루어져야 한다.	RFP0304	III-1.3.2.3.3 III-1.3.2.3.4 III-1.3.8.4.3	시도인사행정정보시스템 연계 방안 관련 정보시스템 연계 방안 송수신 보안방안	III-72 III-74 III-168	4.1.1.나 4.1.3.카	행정정보공유 센터구축을 위한 보안 컨설팅	5, 24
	행정정보공유센터는 행정기관 및 공공기관 등 모든 기관에서 사용이 편리 형태로 개발하여야 하며, API 연계를 통한 공동이용의 경우 지원 및 테스트 등에 협조하여야 한다.	RFP0306	III-1.3.6.2.2 III-1.3.7.3.4	구성요소 및 구축방안 행정정보공동이용 열람	III-114 III-136	4.1.1.나	행정기관 및 공공기관 권한관리 강화	3
	행정정보공유센터 운영 및 적용을 위한 행정기관 및 공공기관 운영지침 및 이용기관용 지침, 교육교재 등을 제공하여야 한다.	RFP0307	III-1.3.7.3.7	사용자교육 및 홍보 기능	III-142	11	표준화계획(정보시스템 구축 운영 기술 가이드라인 적용계획)	85
	각종 프로그램, 컴포넌트, 데이터 관리 등에 필요한 관리기능과 공통지원서비스의 이용현황 등 제공 서비스에 대한 통계 기능을 제공하여야 한다.	RFP0310	III-1.3.3.3.2 III-1.3.7.3.6	사후감사 기능 통계 및 사후 감사 기능	III-84 III-140	4.1.1.3.나	공무원창구 기능 개선	3
	각종 통계는 GUI 환경으로 생성하여 화면 및 프린터로 출력 가능하여야 하고, 별도 파일로 저장 가능하여야 한다. 현황 및 통계 등을 산출하기 위하여 별도의 DB를 구성하는 등 부하를 최소화할 수 있도록 하여 공동이용 서비스에 영향이 없어야 한다.	RFP0311	III-1.3.5.3.1 III-1.3.7.3.6 III-1.3.8.5.1	현황 및 통계 조회 보장 통계 및 사후 감사 기능 통합관리 웹 프로그램 개선	III-108 III-140 III-169	6.2.1 6.3.2	객체지향웹경로 시험내역	53, 64
	새롭게 구축되는 행정정보공유센터에 접속할 수 있는 공무원창구 및 공공기관창구에 대한 사이트를 개설하여 서비스가 가능해야 한다.	RFP0317	III-1.3.6.2	행정정보공유 포탈 구성	III-113	4.1.1.나 4.1.3.가	시스템 분리 방안 법정부 행정정보공유 기반 구축	4, 9

표 7. 요구사항 추적 매트릭스 설계(2)
Table 7. RTMD(2)

솔루션정의(SD)단계			
요구사항정의서		기능분해도	
요구사항명	ID	기능명	ID
현황 및 통계조회 기능	RQ-AB-010-T-001	행정정보소재안내 통계 조회	AB050_P3.2
정보의 흐름 모니터링	RQ-AB-010-T-002		
기존 기능 충족	RQ-AB-010-T-003	행정정보소재안내 관리, 공통코드 관리, 설문조사 관리, 교육자료 관리, 서버모니터링 관리	AB050_P3, AB050_P5, AB080_P8, AB080_P6, AB080_P7
공동이용 대상정보 확대	RQ-AB-020-T-001, RQ-DB-160-T-001	공동이용대상정보별 연계시스템 구축 기능(공동이용대상정보 등록,공동이용대상정보 수정,공동이용대상정보 삭제,공동이용대상정보 조회)	AB050_P4.1, AB050_P4.2, AB050_P4.3, AB050_P4.4
공동이용 대상정보별 연계시스템 구축	RQ-AB-020-T-002, RQ-AB-020-P-001, RQ-AB-020-S-001	정보열람 관리	AB020_P1
데이터레벨 관리 기능	RQ-AB-060-T-001	데이터레벨관리 (공동이용정보별 정보항목 관리, 공동이용정보별 정보단위 관리, 공동이용정보별 열람유형 관리)	AB060_P1, AB060_P2, AB060_P3
데이터레벨 서비스 확대	RQ-AB-060-T-002	데이터레벨관리 (공동이용정보별 정보항목 관리, 공동이용정보별 정보단위 관리, 공동이용정보별 열람유형 관리)	AB060_P1, AB060_P2, AB060_P3
정보열람 시 데이터레벨 적용	RQ-AB-060-T-003	정보열람 관리, 데이터레벨관리	AB020_P1, AB060_P1, AB060_P2, AB060_P3
행정정보공유관리자 권한관리 기능	RQ-AB-070-T-009	열람권한 관리	AB070_P1.1
		단위기관 관리	AB070_P1.2
		기관총괄책임자 관리	AB070_P1.3
		제공기관 관리	AB070_P1.4
		사무 관리	AB070_P1.5
		통계 및 현황 조회	AB070_P1.6
		열람권한 관리	AB070_P8.1
		단위기관 관리	AB070_P8.2
		기관총괄책임자 관리	AB070_P8.3
		제공기관 관리	AB070_P8.4
사무 관리	AB070_P8.5		
통계 및 현황 조회	AB070_P8.6		
사후감사 기능 개선	RQ-AB-070-T-003	비정상적인 열람 탐지 기능	AB070_P15.1
		정보열람 추적관리 기능	AB070_P15.2
		민원인정보 의심항목 감시 기능	AB070_P15.3
		민원처리/열람 건수 비교 기능	AB070_P15.4
		정보열람 추이 감시 기능	AB070_P15.5
		정보주체에게 메일 발송 기능	AB070_P15.5
		사후감사 열람권한 관리 기능	AB070_P15.7
열람권한 관리 기능	RQ-AB-070-T-004	열람권한 관리	AB070_P1.1
		열람권한 승인 관리	AB070_P3.2
		열람권한 승인 관리	AB070_P4.1
		열람권한 관리	AB070_P5.1
		열람권한 관리	AB070_P8.1

표 8. 요구사항 추적 매트릭스 설계(3)

Table 8. RTMD(3)

솔루션정의(SD)단계										
기능분해도		유즈케이스 명세서		분석 Sequence Diagram		분석 Class Diagram		Boundary Class	Control Class	Entity Class
기능명	ID	유즈케이스명	ID	Diagram명	ID	Diagram명	ID	Class명	Class명	Class명
정보 열람 관리	AB020_P1	정보 열람관리	UC_AB.020_01	열람권한목록조회, 정보열람 결과문서 다운로드, 정보열람 결과문서 열람, 정보열람 신청 및 결과 처리, 준실시간 목록 조회, 준실시간 열람 처리	ASD_AB.020_01_01 ASD_AB.020_01_02 ASD_AB.020_01_03 ASD_AB.020_01_04 ASD_AB.020_01_05 ASD_AB.020_01_06 ASD_AB.020_01_07 ASD_AB.020_01_08	정보열람	ACD_AB.020_01	열람권한목록조회 Bnd 정보열람 Bnd 결과문서 다운로드 Bnd 정보열람 결과문서 열람 Bnd 정보열람 신청 및 결과 처리 Bnd 준실시간 목록 조회 Bnd 준실시간 열람 처리 Bnd	정보열람 Ctl	기관-자료연계 처리내역 Ent 공공이용-공통-송신로그 Ent 기관-공공이용 정보열람유형 Ent 공공이용-공통-수신로그
정보 흐름 모니터링 관리	AB020_P2	정보 흐름 모니터링 관리	UC_AB.020_02	정보열람 처리상태 조회, 열람권한 처리상태 조회	ASD_AB.020_02_01 ASD_AB.020_02_02	정보흐름 모니터링 관리	ACD_AB.020_02	정보열람 처리상태 Bnd 열람권한 처리상태 Bnd	정보흐름 모니터링 기능 Ctl	기관-자료처리 연계내역 Ent PMI-권한신청 Ent PMI-권한신청 공공이용정보 Ent
현황 및 통계 조회	AB020_P3	현황 및 통계 조회 관리	UC_AB.020_03	공공기관별 현황/통계 조회, 기간별 현황/통계 조회, 대상정보별 현황/통계 조회, 업무별 현황/통계 조회, 행정기관별 현황/통계 조회, 현황/통계정보 출력, 현황/통계정보 파일 저장	ASD_AB.020_03_01 ASD_AB.020_03_02 ASD_AB.020_03_03 ASD_AB.020_03_04 ASD_AB.020_03_05 ASD_AB.020_03_06 ASD_AB.020_03_07	현황 및 통계 조회	ACD_AB.020_03	공공기관별 현황/통계 조회 Bnd 기간별 현황/통계 조회 Bnd 대상정보별 현황/통계 조회 Bnd 업무별 현황/통계 조회 Bnd 행정기관별 현황/통계 조회 Bnd 현황/통계정보 출력 Bnd 현황/통계정보 파일 저장 Bnd	현황 및 통계 조회 Ctl	기관-공공이용 통계 Ent

표 9. 요구사항 추적 매트릭스 설계(4)
Table 9. RTMD(4)

진화적전달(ED)단계						
Entity(ERD)	설계 Sequence Diagram		설계 Class Diagram		테이블 목록	
Entity 명	Diagram 명	ID	Diagram 명	ID	테이블 ID	테이블 명
기관-이용 사무 Ent	열람권한 목록 조회 정보열람 신청 및 결과 처리 행정기관별 현황/통계 조회 공공기관별 현황/통계 조회 대상정보별 현황/통계 조회 업무별 현황/통계 조회 기간별 현황/통계 조회 현황/통계정보 출력	DSD_AB.020_01_01 DSD_AB.020_01_04 DSD_AB.020_03_01 DSD_AB.020_03_02 DSD_AB.020_03_03 DSD_AB.020_03_04 DSD_AB.020_03_05 DSD_AB.020_03_07	정보열람 현황 및 통계 조회	DCD_AB.020_01 DCD_AB.020_03	TB_AB227	기관-이용사무
기관-공동이용정보항목 Ent	정보열람 결과문서 열람 정보열람 신청 및 결과 처리 준실시간 열람 처리 공동이용정보별 정보항목 등록 공동이용정보별 정보항목 삭제 공동이용정보별 정보항목 조회	DSD_AB.020_01_03 DSD_AB.020_01_04 DSD_AB.020_01_06 DSD_AB.060_01_01 DSD_AB.060_01_02 DSD_AB.060_01_03	정보열람 공동이용정보별 정보항목 관리	DCD_AB.020_01 DCD_AB.060_01	TB_AB222	기관-공동이용 정보항목
기관-열람유형별정보단 위 Ent	정보열람 결과문서 열람 정보열람 신청 및 결과 처리 준실시간 열람 처리	DSD_AB.020_01_03 DSD_AB.020_01_04 DSD_AB.020_01_06	정보열람	DCD_AB.020_01	TB_AB226	기관-열람유형별 정보단위
기관-자료연계기본정보 Ent	정보열람 결과문서 열람 정보열람 신청 및 결과 처리 준실시간 목록 조회 준실시간 열람 처리 정보열람 처리상태 조회 행정기관별 현황/통계 조회 공공기관별 현황/통계 조회 대상정보별 현황/통계 조회 업무별 현황/통계 조회 기간별 현황/통계 조회 현황/통계정보 출력 공동이용대상정보 등록 공동이용대상정보 수정 공동이용대상정보 삭제 공동이용대상정보 조회 공동이용정보별 열람유형 등록 공동이용정보별 열람유형 수정 공동이용정보별 열람유형 삭제 공동이용정보별 열람유형 조회 공동이용정보별 정보단위 등록 공동이용정보별 정보단위 수정 공동이용정보별 정보단위 삭제 공동이용정보별 정보단위 조회 공동이용정보별 정보항목 등록 공동이용정보별 정보항목 삭제 공동이용정보별 정보항목 조회	DSD_AB.020_01_03 DSD_AB.020_01_04 DSD_AB.020_01_05 DSD_AB.020_01_06 DSD_AB.020_02_01 DSD_AB.020_03_01 DSD_AB.020_03_02 DSD_AB.020_03_03 DSD_AB.020_03_04 DSD_AB.020_03_05 DSD_AB.020_03_07 DSD_AB.050_08_01 DSD_AB.050_08_02 DSD_AB.050_08_03 DSD_AB.050_08_04 DSD_AB.060_03_01 DSD_AB.060_03_02 DSD_AB.060_03_03 DSD_AB.060_03_04 DSD_AB.060_02_01 DSD_AB.060_02_02 DSD_AB.060_02_03 DSD_AB.060_02_04 DSD_AB.060_01_01 DSD_AB.060_01_02 DSD_AB.060_01_03	정보열람 현황 및 통계 조회 공동이용대상정보 관리 공동이용정보별 정보항목 관리 공동이용정보별 정보단위 관리 공동이용대상정보별 열람유형 관리	DCD_AB.020_01 DCD_AB.020_03 DCD_AB.050_08 DCD_AB.060_01 DCD_AB.060_02 DCD_AB.060_03	TB_AB201	기관-자료연계 기본정보

V. 결 론

소프트웨어 개발 프로세스는 일반적으로 문제영역에 대한 사용자 요구사항을 도출하는 단계에서 시작하여 이를 바탕으로 분석, 설계, 코딩, 테스트 단계

로 이루어진다. 이러한 프로세스의 단계 중에서도 사용자 요구사항을 분석하고 설계하는 초기 단계가 매우 중요하다. 즉 초기에 발생한 오류는 추후 보정 비용 및 유지보수 비용이 많이 소요되기 때문이다.

본 논문에서는 UML 기반의 객체지향 소프트웨어 개발 생명주기에서 요구사항 추적 매트릭스 설계

(RTMD : Requirement Tracing Matrix Design) 모델을 제시하여 실제업무에 적용하였다. 이 방법은 소프트웨어 개발의 생산성 향상 및 유지보수 비용을 절감하는 효과가 있다. 그러나 관련 문서들의 완전성 (completeness), 일관성(consistency), 정확성 (correctness)을 검증하기 위한 절차 및 규칙이 미흡하며, 또한 이들 명세서들은 보다 체계적으로 관리할 수 있는 자동화된 시스템을 필요로 한다.

따라서 향후 연구과제는 실제 프로젝트에 편리하게 사용할 수 있으며, 보다 정형화되고 표준화된 생산성이 우수한 객체지향 Case Tool 개발 방법에 대한 연구가 절실히 요구된다.

참 고 문 헌

[1] 김영규, “소프트웨어 생산성 향상을 위한 프레임워크 설계 EOOM”, *강원대학교 대학원 박사학위논문*, 2008. 02.
 [2] 하일규, “정확성 및 일관성 검증규칙에 기반한 UML 다이어그램의 메타검증”, *영남대학교 대학원 박사학위논문*, 2003. 06.
 [3] 허계범, 김영규, 양동일, “객체지향 환경에서 소프트웨어 생산성 향상을 위한 프레임워크 모델 연구”, *한국항행학회, 제14권 제6호*, 2010. 12.
 [4] *한국전산원*, “CBD 방법론 기반 개발사업에 대한 감리지침 연구”, 2003. 11.
 [5] *한국전산원*, “정보시스템 객체지향 개발감리지침 연구”, 1998. 11.
 [6] LG CNS, “유스케이스모델링가이드”, GIC-221-01 ver. 1.0
 [7] LG CNS, “분석모델링가이드”, GIC-221-05 ver. 1.0
 [8] LG CNS, “설계모델링가이드”, GIC-221-06 ver.1.0
 [9] Grant Zemnt, "Recovering Towards Value Based Requirement Traceability", *DePaul University Chicago Illinois*, 2005. 03.
 [10] Mikio Aoyama, “New Age of Software Development : New Component-Based Software Engineering Changes the Way of Software Development, International Workshop on Component-Based Software“, *ICSE*, pp. 124-128, 2003.
 [11] Standish Group, *CHAOS Reports(2004)*, 2004, 01.

[12] <http://www.omg.org>; OMG, “Unified Modeling Language Specification”, Version 1.4, Object Management Group;
 [13] <http://www.objs.com/agent/index.html>; OMG Agent Platform Special Interest Group;

허 계 범 (許啓範)



1993년 8월 : 광운대학교 전자계산학과 이학석사
 1999년 2월 : 광운대학교 전자계산학과 이학박사
 2010년 현재 : 한림성심대학 인터넷 비즈니스과 교수

관심분야 : OOA, OOD, OOP, OODB, e-Biz, e-커머스, 정보시스템감리

김 영 규 (金榮奎)



1989년 8월 : 충북대학교 컴퓨터과학과 석사
 2008년 2월 : 강원대학교 컴퓨터과학과 이학박사
 2010년 현재 : 한림성심대학 인터넷 비즈니스과 교수

관심분야 : 소프트웨어공학, ERP, 정보통신

양 동 일 (梁東一)



2004년 2월 : 강원대학교 컴퓨터과학과 이학석사
 2007년 8월 : 강원대학교 컴퓨터과학과 이학박사
 2010년 현재 : 한림성심대학 인터넷 비즈니스과 교수

관심분야 : 소프트웨어공학, 유비쿼터스, 포렌식