

RUP 기반의 Data Model 설계

최창민*, 김천식, 정정수
동명정보대학교 컴퓨터공학과

A Design On RUP based Data Model

ChangMin Choi, CheonSik Kim, ChungSoo Chung
Dept. of Computer Science, TongMyung University of Information Technology.

요 약

RUP의 Unified Process Model에는 Use Case Model, Analysis Model, Design Model, Deployment Model, Implementation model, Test Model의 6가지 모델이 있다. 이 모델들은 모두 일관성이 있다. 한 모델에서 나오는 요소들은 전, 후 단계에서 다른 모델들과 Trace Dependencies를 갖는다. 이러한 관계들은 각각의 요소들 사이의 진행, 기록 관계를 나타낸다. 그러나 일반적인 데이터간의 관계와 데이터 모델 설계는 이러한 관계없이 설계되어져 전체적인 일관성을 이루지 못 하였다. 본 논문에서는 이러한 관계를 유지 하면서 요구사항에 맞는 데이터 모델을 설계하고자 한다. 따라서 본 논문에서는 대학 종합정보시스템 구축의 일부분인 자산관리 시스템을 분석하여 데이터 모델을 제시한다.

1. 서론

학계나 산업계에서 분명하게 서술되고, 잘 문서화된 소프트웨어 개발 프로세스가 소프트웨어 프로젝트에서 얼마나 중요한 성공요인인지 점차 인식되고 있다. 산업계에서는 소프트웨어 개발을 통해서 지식과 지침들을 모아 왔다. 그러나 실질적으로 개발 프로세스가 시대흐름에 맞게 신속하게 변경되어져나, 자세하게 정의되어 있는 프로세스는 많지 않다. 이러 현실에서 RUP(Rational Unified Process)는 성숙하고, 정확하고, 유연한 소프트웨어 공학 프로세스를 제시하고 있다. RUP에서는 6개의 핵심 Workflows를 제시하고 있는데, 각 Workflows의 산출물들은 다음 단계의 Workflows에서 재사용 되어 진다. 하지만 대부분의 설계는 각 단계의 Workflows간의 관계성이 일

관성을 이루지 못 하고 있다.

본 논문에서는 각 단계의 Workflows간의 관계가 일관성 있는 데이터 모델을 설계 하기 위해 RUP방법론을 이용하여 정의한다. 또한 이를 바탕으로 종합정보시스템의 한 부분인 자산관리 시스템을 적용하여 모델링 하고자 한다. 그리고 2장에서는 CBD방법론과 유즈케이스 중심의 분석에 대하여 알아보고, 3장에서는 RUP기반의 개발 시스템 및 산출물을 알아보고, 4장에서는 사례연구를 설계 모델링하며, 5장에서는 결론을 맺는다.

2. 관련연구

2.1 CBD개발 방법론

소프트웨어 공학에서 중요한 기술적 변화로는 작업라인 기반의 소프트웨어 개발에서 조립 가능하고,

구성가능하며, 그리고 독립적으로 추출된 컴포넌트들의 조합에 의해 만들어진 시스템을 제공하는 컴포넌트 기반의 소프트웨어 개발로의 전환이라 할 수 있다. 이렇게 개발된 시스템들은 그 시스템들의 변화로 인해 나머지 다른 시스템의 부품들에서 발생할 수 있는 문제를 최소화하였다.

컴포넌트란 잘 정의된 문제 도메인의 개념상에서 명확하게 제시된 기능들로 채워진 시스템의 독립적이고 대체 가능한 부품이라 할 수 있다. 소프트웨어 공학자들은 간단하게 컴포넌트들을 선택한 시스템 내부에서 이들을 합성한다. 컴포넌트는 소프트웨어의 교체, 재사용이 용이하며 서비스는 인터페이스를 통해서만 이루어진다. 컴포넌트 개발의 각 단계는 개념적, 논리적, 물리적으로 각각 다른 형태를 취하고 있어 명확하며 이의 개발 대상은 비즈니스에서부터 어플리케이션, 통합소프트웨어에 이르기까지 다양하다. 또한 모두 연관성을 지니고 있기에 독립적인 패키지 형태로 작성이 되어도, 혹은 각각 다른 서비스를 제공하여도 이들이 모여 프레임워크를 구축하고 어플리케이션으로 발전한다.

또한 컴포넌트를 조립해 새로운 어플리케이션을 만들 수가 있어 개발기간을 단축할 수 있으며, 기존의 컴포넌트를 재사용할 수 있다면 생산성과 경제성을 높을 수 있다. 사용자는 품질 좋은 컴포넌트를 선택하여 사용할 수 있으며, 컴포넌트의 사용은 인터페이스를 통해 이루어지고 실제 구현 과정은 사용자가 필요가 없다. 이러한 컴포넌트의 장점을 이용하고 체계적으로 개발하기 위해 CBD(Component Based Development)방법론을 적용하여 컴포넌트를 만들고자 한다. CBD에 있어서도 많은 방법론들이 있는데 대부분 UML 표기법을 따르고 있으며, 다양한 방법론 중 대중적인 방법론을 정리하면 <표1>과 같다.

<표1> CBD 방법론

방법론	특징
Catalysis	<ul style="list-style-type: none"> • 흥미롭고 이론적인 접근법 • 너무 학문적이어서 사용하기가 어려움 • 실제화를 위한 프로세스 정리가 필요

CBD96	<ul style="list-style-type: none"> • Catalysis를 근간으로 보다 명확하고 간단한 방법론을 제시 • UML을 근간으로 하여 기존 객체지향 기법에 익숙한 소프트웨어 개발자는 쉽게 접근 가능 • 지원 도구들을 통하여 생산성과 품질을 최적화할 수 있는 장점 • 컴포넌트의 특징과 컴포넌트가 가져야 할 구성 요소들을 잘 정의, 새로운 방법론으로 확장이나 특정분야의 방법론으로 확장에 유용하게 사용
SELECT	<ul style="list-style-type: none"> • 이론적인 측면보다 실제적이고 개론적인 성향이 강해, 컴포넌트 기반 개발을 처음 접하는 개발자들의 컴포넌트 기반 개발 이해가 용이 • 정렬, 설계, 조립의 단계를 반복, 점진적으로 수행
RUP	<ul style="list-style-type: none"> • 객체지향 기법을 근간, 소프트웨어 개발에 필요로 하는 모든 관련 활동들을 총집합 • 대규모의 인원과 많은 시간을 필요로 하는 대형 프로젝트에 적합

2.2 RUP 방법론

RUP는 요구사항 분석 단계의 산출물로 유스케이스 모델을 산출하고 분석 단계에서 각 유스케이스의 이벤트 플로우를 보고 인터페이스 객체, 컨트롤 객체, 엔티티 객체를 도출해 나간다. 클래스 다이어그램을 도출할 경우에도 인터랙션 다이어그램을 통해서 효율적으로 도출한다. 또한 유사한 유스케이스를 그룹화 하여 시스템 컴포넌트인 패키지 다이어그램을 도출하고 패키지 다이어그램을 중심으로 클래스 다이어그램을 산출한다. 이는 시스템을 의미적으로 분할하여 클래스 다이어그램을 도출하므로 이해하기가 쉽고 분산환경에서(Deploy)할 경우에 분할이 쉬우며 요구사항 변경시 요구사항 분석단계부터 설계단계까지 각 산출물을 추적할 수 있어 요구사항 변경을 쉽게 수용할 수 있다.

2.3 Catalysis

Desmond F.D' Souza 등에 의해서 제안된 Catalysis는 1991년에 시작된 OMT의 정형화 작업에 그 뿌리를 두고 있다. Catalysis는 비즈니스 모델로부터 소스 코드에까지 이르는 전 과정에서의 추적성을 보장하고, 불명확한 모델이나 문서를 최대한 명확하도록 만드는 정확성, 컴포넌트 기반의 개발, 코드뿐 아니라, 분석이나 디자인의 산출물들까지 확대된 재사용 개념, 확장성, 유연한 프로세스를 주요 특징으로 하고 있다.

Catalysis는 각각 협력(Collaboration), 타입(Type), 정제(Refinement)의 3가지의 모델링 개념에 기초하여 구성되어 있다. 협력은 연관된 객체들간의 상호연동으로 설명할 수 있는데, UML 1.3에서 Use Case를 협력으로 실현된다. 협력은 특정 서비스를 제공해 주기 위하여 수행되는 작업들과 그 작업에 참여하는 객체들로서 이루어지는데, 협력은 큰 수준에서 점차 낮은 수준으로 세분화 될 수 있다. Catalysis는 협력 모델을 통해서 대상 시스템 혹은 도메인의 기능들을 큰 수준에서 작은 수준으로 명세한다. Catalysis 방법론에서는 협력이 후보 컴포넌트의 기준이 되는 것이다.

Catalysis는 모델 요소와 프로세스 패턴 등의 다양한 좋은 기법들을 사용하고 있지만, 그 내용들의 많은 경험을 요구하고, 특히 컴포넌트 식별 작업의 경우, 그 절차가 개념적이고, 분석자의 직관에 많이 의존된다고 할 수 있다.

2.4 CBD96

CBD96은 Computer Associates사의 개발 도구인 COOL을 이용한 컴포넌트 개발 아키텍처 모델이면서 개발 방법론이다. 컴포넌트의 식별은 시스템의 전체 도메인을 중심으로 타입 다이어그램과 비즈니스 타입 다이어그램을 추출하고 추출된 비즈니스 타입 다이어그램의 타입 중 핵심 타입(Core Type)을 중심으로 관련된 타입을 그룹화하여 비즈니스 컴포넌트를 추출한다. 이렇게 비즈니스 컴포넌트를 추출한 후 인터랙션 다이어그램(Interaction Diagram)을 통해 보다 독립적인 비즈니스 컴포넌트를 추출하도록 정

제하는 과정을 거친 후에 비즈니스 컴포넌트를 조합하여 컴포넌트 아키텍처를 정의한다.

2.5 유즈케이스 중심의 분석

유즈케이스 중심의 분석상에서 기본적인 개념은 액터(Actor)와 유즈케이스이다. 액터는 시스템의 사용자에게 의해 수행되는 특정한 역할을 말하는 것으로 시스템을 이용할 때 비슷한 행위를 하는 사용자들의 범주를 나타낸다. 유즈케이스는 시스템의 도메인에서 사용되어지는 언어들(언어)을 이용해서 자연어로 기술된다. 이러한 액터와 유즈케이스의 명세화를 통해서 유즈케이스 모델을 얻게 된다.

유즈케이스 중심의 분석은 요구사항 분석과정의 복잡성을 다루는 데에 도움을 준다. 액터의 사용적 측면에서 다른 유즈케이스들에 대해 독립적으로 발견하고 분석함으로써 하나의 유즈케이스에 초점을 둘 수 있기 때문이다. 유즈케이스 중심 분석의 개념은 간단하고 유즈케이스에 기술한 내용이 시스템의 문제 도메인에서 쉽게 발견되어 질 수 있기 때문에 고객과 시스템의 사용자가 의욕적으로 존재할 수 있는 사용자를 실제로 그들이 지닌 요구들과 행위들을 발견할 수 있다.

하지만 특정 유즈케이스는 모든 경우에 발생될 수 없다. 각 유즈케이스는 어느 특정 상황에서 시작되고 성공적으로 완수 되는 것을 기술하기 위한 것이다. 이러한 이슈는 유즈케이스를 이용한 분석에서 다루어지지 않는다. 일반적으로 유즈케이스를 이용한 분석에서는, Jacobson의 A Use Case Driven Approach에서 정의된 바와 같이, 다음과 같은 이슈들에 대해서 충분히 다루어지지 않고 있다.

- 유즈케이스는 독립적이지 않다. 그들 간에 겹쳐질 수 있고 동시에 발생될 수도 있다. 또는 서로간에 영향을 미칠 수도 있다.
- 유즈케이스는 어떤 조건이 만족되었을 때 발생된다. 상황을 발생시키고 종결시키는 상황을 가지고 있다.
- 유즈케이스의 추상화(Abstraction)의 레벨과 기술되는 문장의 길이는 임의로 정해지게 된다.

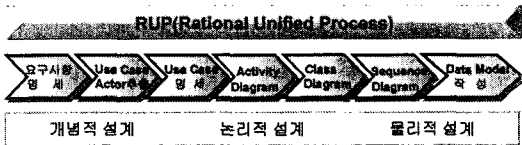
• 유스케이스는 실제로 시스템을 사용하는 시나리오의 모든 가능한 부분들을 다룬다는 보장을 하지 못한다.

계 분석에서 필요한 산출물을 각 설계 단계에서 살펴 볼 것이다. 분석단계에 따른 산출물은 <표2>과 같다.

3. RUP 기반의 개발 프로세스 및 시스템

3.1 RUP기반의 개발 프로세스

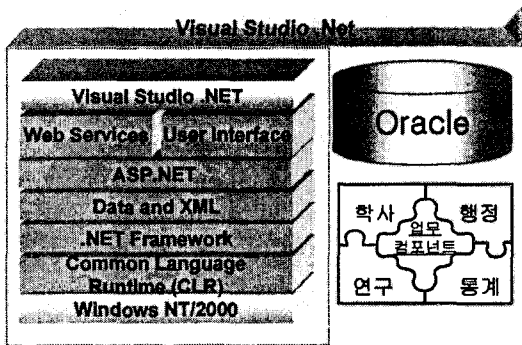
컴포넌트를 효과적으로 개발하기 위해서는 (그림 1)과 같이 실제계에서 사용자가 필요로 하는 요구사항을 통해 각각의 분석단계에서 구체 한 후, 컴포넌트 설계에서 구현하여 컴포넌트를 생성한다.



(그림 1) RUP기반의 개발 프로세스

3.2 시스템 구조

종합정보시스템은 학사, 행정, 연구, 통계를 모든 사용자가 single-sign-on을 통해서 전체의 시스템을 데이터 입력과 확인이 가능하며, 각각의 업무가 서로 연동되는 시스템을 구축한다. (그림2)는 종합정보시스템 구축의 시스템 구조이다.



(그림 2) 종합정보시스템의 시스템 구조

3.3 RUP기반의 산출물

본 논문에서 요구사항 분석 모델링과 컴포넌트 설

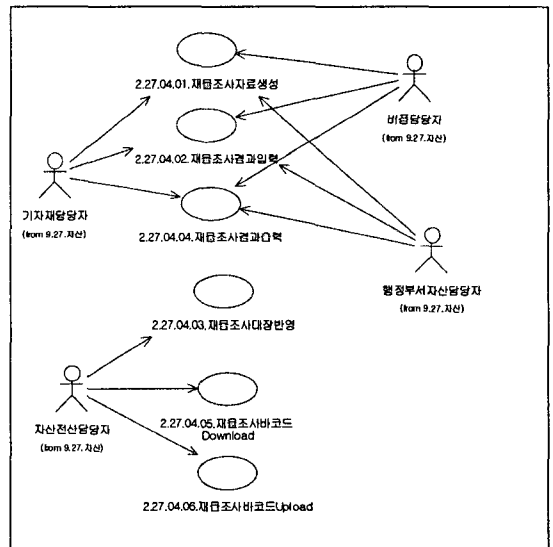
<표2>각 분석단계에 따른 산출물

구분	산출물
개념적설계	Use Case Use Case specification(명세서) Use Case Diagram Activity Diagram
논리적설계	Class Diagram Sequence Diagram
물리적설계	Package Diagram Component Diagram

4. 사례연구

4.1 자산관리시스템의 요구사항 분석

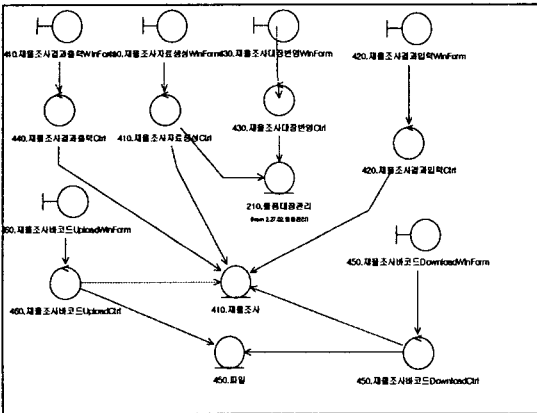
(그림 3)은 자산관리시스템의 한부분인 재물조사 요구사항을 분석한 유즈케이스 다이어그램이다. 이 유즈케이스 다이어그램은 각 부서에서 해당하는 업무간의 관련된 액터를 보여준다.



(그림 3) 자산관리시스템의 유즈케이스 다이어그램

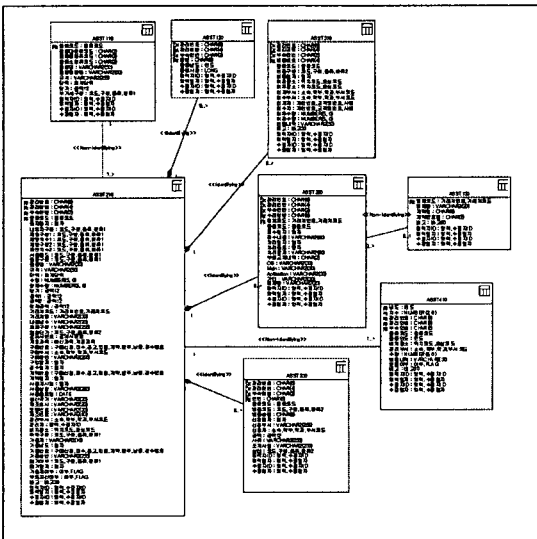
4.2 자산관리 시스템의 비즈니스 모델링

(그림 4)는 자산관리 시스템의 재물조사 비즈니스 모델을 명시한 클래스 다이어그램이다. 이 클래스 다이어그램은 각각의 업무에 대하여 화면명세에 대칭되는 boundary 클래스와 데이터를 직접 호출하는 control 클래스, 그리고 데이터 테이블에 대칭되는 entity 클래스를 추출하여 클래스간의 관계를 나타낸다.



(그림 4) 재물조사 클래스 다이어그램

4.3 자산관리시스템 데이터 모델



(그림5) 데이터 다이어그램

(그림5)는 비즈니스 모델링의 entity를 데이터 모델로 변화하여 데이터 다이어그램을 나타낸 것이다.

4.4 평가

종합정보시스템 구축의 일부분인 자산관리 시스템에서 재물조사에 대한 요구사항 분석 및 데이터 모델을 설계 하였다. 이러한 과정을 통하여 각 단계간의 관계를 유지하면서 일관성 있는 데이터 모델을 설계함으로써 소프트웨어의 개발의 품질향상 및 일관성의 이익을 얻을 수 있다.

5. 결론

RUP 기반의 개발 방법론을 유지하며 데이터 모델 설계를 진행하였다. RUP 기반의 개발 방법론으로 진행된 설계는 유즈케이스 크기에 종속적일 수 있다. 따라서 우선적으로 유즈케이스가 명확하게 도출되어야 하며, 각 모델간의 관계를 명세화 하여야 한다

그리고 이 후 다양한 영역에 적용할 수 있는 명확한 지침이나 보완에 대한 좀 더 상세한 연구가 필요하다.

[참고문헌]

- [1] 소동섭, 김수동, "Rational Unified Process(RUP)의 추적성 적용 지침" 정보과학회, 2001
- [2] Bulter Group, What is a compnent, Interact, 1998.
- [3] 시사컴퓨터, "조립식 개발방법론 CBD", 시사컴퓨터 기술보고서, 2001
- [4] Souza, Desmond, Objects, Component, and Frameworks with UML, Addison Wesley, 1999
- [5] Jacobson, I., et al, Object-Oriented Software Engineering; A Use Case Driven Approach, Addison-Wesley, 1992