

Rational Unified Process (RUP)의 추적성 적용 지침

°소동섭, 김수동

승실대학교 컴퓨터학과

dssso@selab.soongsil.ac.kr, sdkim@comp.ssu.ac.kr

Guidelines for Applying Traceability in RUP

Dong Seob So, Soo Dong Kim

Dept. of Computer Science, Soongsil University

요 약

RUP의 Unified Process Model에는 Use-Case Model, Analysis Model, Design Model, Deployment Model, Implementation Model, Test Model의 6가지의 모델이 있다. 이 모델들은 모두 일관성이 있다. 한 모델에서 나오는 요소들은 전, 후 단계에서 다른 모델들과 Trace Dependencies를 갖는다. 예를 들면, Use-Case Model에서의 한 Use Case는 Design Model에서 Use-Case Realization으로 Test Model에서는 Test Case로 Trace Dependencies를 갖는다. 소프트웨어 개발에서 추적성은 Understandability와 Change Propagation과 같은 이유로 매우 중요하다. 그러나 RUP에서는 각 모델들간의 추적성에 대한 지침이 충분히 제시되어 있지 않아, 개발 과정에서 요구사항의 변경이나 에러 수정 또는 향후 관리 시에 어려움을 겪고 있다. 본 논문에서는 이 점에 중점을 두고 RUP에서 제시하고 있는 모델들 간의 추적성 중 설계모델에서 분석모델로의 추적성 적용지침을 제시한다.

1. 서론

1.1. 연구 배경

학계나 산업계에서 분명하게 서술되고, 잘 문서화된 소프트웨어 개발 프로세스가 소프트웨어 프로젝트에서 얼마나 중요한 성공요인인지 점차 인식되어지고 있다. 산업계에서는 소프트웨어 개발을 통해서 지식과 지침들을 모아 왔다. 그러나 실질적으로 개발 프로세스가 시대흐름에 맞게 신속하게 변경되어지거나, 자세하게 정의되어 있는 프로세스는 많지 않다. 이런 현실에서 Rational Unified Process(RUP)는 성숙하고, 정확하고, 유연한 소프트웨어 공학 프로세스를 제시하고 있다[1]. RUP에서는 6개의 핵심 Workflows를 제시하고 있는데, 각 workflows의 산출물들은 다음 단계의 Workflows에서 재사용되어진다. 그러나 대부분 산출물들의 추적성을 위한 지침이 산출물의 이름을 통하여 다음 단계의 Workflows 산출물과 1:1로 맵핑되는 지침으로 한정되어 있다.

본 논문에서는 RUP의 Workflows 중 설계모델에서 분석모델로의 추적성 추가 지침을 제시하고자 한다. 본 논문의 2장에서는 관련연구로서 추적성과 분석모델과 설계모델의 비교에

대해서 설명하고 3장에서는 분석모델과 설계모델의 추적성에 대한 적용 지침을 제시한다.

2. 관련 연구

2.1. 추적성

추적성이란, 프로젝트간에 관련된 요소들을 추적하기 위한 능력을 말하는데, 여기에서 추적이란 특정한 규칙 없이 같은 개념을 갖는 어떤 요소로부터 다른 요소가 유도될 때 두개의 요소들 사이의 진행, 기록 관계를 나타내는 의존상태를 말한다 [2]. 그림 1은 RUP에서 Use Case를 중심으로 제공하는 기본 RUP 추적 전략으로 Use-Case Model에서 일관성을 가지고 다음 단계 모델로의 추적을 보인다.

2.2. 분석모델과 설계모델의 비교

분석모델과 설계모델을 9가지 점에서 비교하여 볼 수 있다 [3].

- 분석모델은 시스템을 추상화 하는 Conceptual Model인 반면 설계모델은 구현의 청사진을 보이는 Physical Model 이

다.

- 분석모델은 Design-Generic 한 반면, 설계모델은 구현에 한정되어 있다.
- 분석모델은 클래스의 Stereotype 으로 <<control>>, <<entity>>, <<boundary>> 세가지를 갖지만, 설계모델에서는 클래스에 많은 Stereotypes 를 갖을 수 있다.

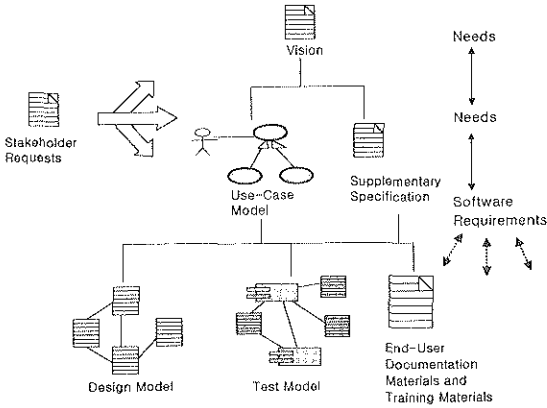


그림 1. 기본 RUP 추적 전략

- 설계모델은 분석모델에 비해 형식적이다[4].
- 설계모델은 개발하는데 분석모델에 비해 5 배 이상의 비용이 든다.
- 분석모델은 Layer 가 거의 없는 반면, 설계모델은 많은 layer 가 있다.
- 두 모델 다 동적이지만, 설계모델은 분석모델에 비해 순차에 중점을 더 두고 있다.
- 분석모델은 전체 소프트웨어 생명주기에서 관리되지 않지만, 설계모델은 전체 소프트웨어 생명주기에서 관리되어 진다.
- 분석모델은 설계모델을 생성하고, 시스템을 구체화하는 필수적인 입력 구조를 정의한다. 설계모델은 가능한 한 분석모델에서 정의된 구조를 정의하면서 시스템을 구체화한다[5].

3. 설계모델에서 분석모델로의 추적성 지침

3.1. 분석패키지와 설계서브시스템의 추적성

분석패키지(AP)는 분석모델의 산출물을 조직화하는 수단을 제공한다. 분석 패키지는 분석 클래스, Use-Case Realization, 다른 분석패키지로 구성된다. 분석 패키지는 설계 모델의 설계서

브시스템(SS)과 추적성을 갖는다. 설계서브시스템은 설계클래스와 Use-Case Realization, 인터페이스 그리고 다른 서브시스템으로 구성 된다. 그림 2 는 분석패키지와 설계서브시스템의 구성 요소들을 보여준다.

RUP 에서는 분석패키지와 설계서브시스템 사이의 추적성을 보이기 위하여 분석패키지명과 설계서브시스템의 이름을 똑같이 하여 추적성을 보이고 있다. 그러나 분석패키지내의 요소들이 설계서브시스템에서, 여러 서브시스템의 요소로 포함되어질 때, 이름만으로는 그 추적성을 나타낸다면, Naming Conventions 문제가 발생할 수 있다. 따라서, 아래 그림 3 에서와 같이 모델간의 같은 이름만으로 추적성을 나타내지 않고, 분석패키지 내의 구성 요소들이 설계서브시스템에서 어떻게 매핑 되는지를 보여주는 추적성 도표를 표 1 에서 제시한다.

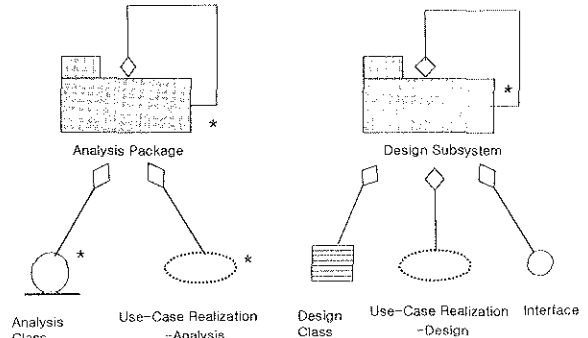


그림 2. 분석패키지와 서브시스템의 구성요소

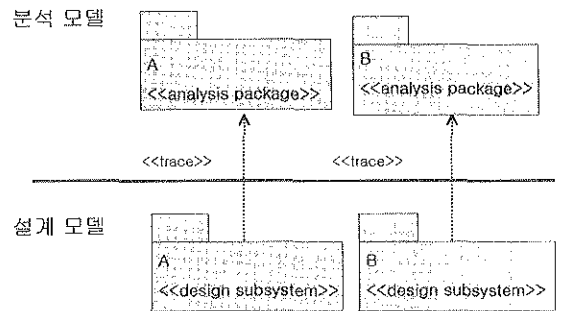


그림 3. 분석패키지에 기반한 서브시스템

표 1 에서 처음 가로줄은 분석패키지(AP)와 서브시스템(SS)을 두고, 처음 세로줄에는 분석패키지의 구성요소들을 둔다. 분석패키지의 구성요소들이 어떻게 변경되는지 매트릭스에 구성요소의 이름과 변경번호를 AP 와 SS 에 기록한다.

표 1. 분석패키지 요소와 설계서브시스템 요소의 맵핑

	AP 1	SS 1
Use-Case	AP1.WithdrawMoney	SS1.WithdrawMoney
Flow-of-events description	AP1.WithdrawMoney	SS1.WithdrawMoney
Interaction Diagrams	AP1.WithdrawMoney	SS1.WithdrawMoney
Boundary Classes	AP1.Cashier Interface	SS1.Display SS1.Key Pad SS1.Card Reader SS1.Client Manager
Control Classes	AP1.Withdrawal	SS1.Transaction Manager
Entity Classes	AP1.Account	SS1.Account SS1.Account Manager

3.2. 서비스 패키지와 서비스 서브시스템의 추적성

서비스 패키지(SP)는 응집도가 높고, 분할되지 않으며, 결합도가 낮은 특징을 가지고 있다. 따라서 여러 다른 Use-Case Realization 에서 재사용 된다. 분석모델의 서비스 패키지는 설계모델의 서비스 서브시스템과 추적성을 갖는다. 서비스 서브시스템은 대개 서비스 패키지와 1:1 로 추적되지만 다음과 같은 이유로 더 많은 산출물을 갖는다.

서비스 서브시스템(SB)은 서비스를 제공하기 위하여 필요한 인터페이스와 오퍼레이션을 갖는다. 또한 비기능적인 요구사항과 구현환경에 관련된 제약사항을 갖는다. 결과적으로 서비스 패키지보다 더욱 많은 클래스들을 포함하고, 서브시스템의 크기를 관리하기 위하여 한 서비스 패키지가 여러 서비스 서브시스템으로 분할 될 필요가 있다.

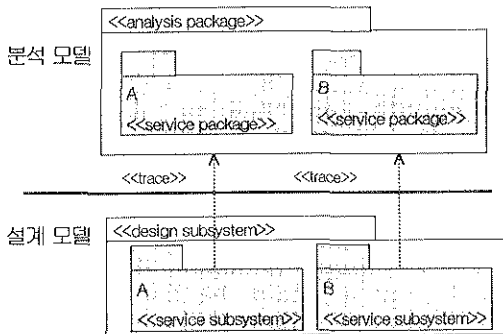


그림 4. 서비스 패키지에 기반한 서비스 서브시스템

그림 4 는 RUP 에서 제공하는 서비스 패키지와 서비스 서브시스템간의 추적을 보여주고 있다. 위와 같은 이유로 이름만으로는 추적에 어려움이 있어 서비스 서브시스템이 서비스 패키지에 맵핑되는 것을 도표를 통해 표 2 에서 제시한다.

표 2 에서 처음 가로줄은 서비스 패키지(SP)와 서브시스템(SB)을 두고, 처음 세로줄에는 서비스 패키지의 구성요소 들을 둔다. 서비스 서브시스템의 구성요소들이 어떻게 변경되는지 매트릭스에 구성요소의 이름과 변경번호를 SP 와 SS 에 기록한다.

표 2. 서비스 패키지 요소와 서비스 서브시스템 요소 맵핑

	SP 1	SB 1
인터페이스	SP1.Account transfers	SB1. Transfer
설계클래스	SP1. Invoice	SB1.Invoice

4. 결론 및 향후 연구 과제

본 논문에서는 Rational Unified Process (RUP) 에서 제시하고 있는 추적성을 보완하는 지침을 제안 하였다. 본 논문에서 제시된 방법이 RUP 의 6 가지 모델들의 추적성을 모두 제시한 건 아니지만, 적어도 추가적인 추적성의 지침 방향을 제시 함으로서, 이후 추가 내지는 수정 작업의 기반을 제시 할 수 있도록 하였다. RUP 의 모델간의 추적성에 대한 지침은 주요 연구 과제로 제안될 것이다.

참 고 문 헌

- [1] Philippe Kruchten, The Rational Unified Process An Introduction, Addison Wesley, 1999.
- [2] Murray R. Cantor, Object-Oriented Project Management with UML, John Wiley & Sons, 1998
- [3] Ivar Jacobson, Grady Booch, James Rumbaugh, The Unified Software Development Process, Addison Wesley, 1999
- [4] Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Addison Wesley, 1999
- [5] Best Practices for Software Development Teams, Rational Unified Process, <http://www.rational.com>, 1998