

가변성을 고려한 핵심 자산의 결정 모델 기술 및 지원도구

윤석진 이승연⁰ 신규상 양영중 박창순
 Embedded S/W 센터, 한국전자통신연구원
 { sjyoon, coral⁰, gssshin, yangyj, cpark }@etri.re.kr

Decision Model Specification of Core Assets with Variability and its Tool

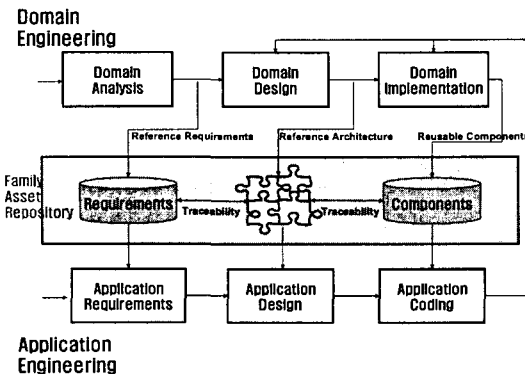
Seok-Jin Yoon Seungyun Lee⁰ Gyu-Sang Shin Young-Jong Yang Chang-Soon Park
 Embedded S/W Center, ETRI

요 약

개발한 자산의 재사용성을 증대시키기 위해서는 프로젝트 라인에서 가장 중요한 제품 간의 공통성과 가변성을 식별하고 이를 표현하는 기술이 잘 정의되어 있어야 한다. 이를 위해서는, 다양한 제품의 요구사항을 만족시켜 줄 수 있도록 자산을 모델링할 때 가변성을 제공하는 부분을 명세화하고 이를 독립적인 모델로 관리함으로써 자산의 재사용성을 향상시키는 방법이 필요하다. 본 논문은 핵심 자산 개발시 정의하는 결정 모델의 유형을 정의하고 결정요소들 간의 관계를 추적할 수 있는 지원 도구의 프로토타입을 제안한다.

1. 서 론

소프트웨어 재사용을 위한 주요 패러다임으로서 대두되고 있는 소프트웨어 프로젝트 라인(Software product lines)은 한 도메인에 속한 여러 패밀리 멤버들의 공통성과 가변성을 분석하여 재사용 가능한 자산을 개발하고 이를 특화된 제품을 개발하는데 효과적으로 이용하는 기술이다[1, 2]. 프로젝트 라인 공학의 목표는 일련의 유사한 소프트웨어 시스템의 공통성과 구별되는 특성을 이해하고 제어함으로써 시스템의 체계적인 개발을 지원 하는 것이다. 프로젝트 라인 소프트웨어 개발은 <그림 1>에서 보는 바와 같이 시각화될 수 있다[3].



<그림 1> 프로젝트 라인 소프트웨어 개발

프로젝트 라인 소프트웨어 개발은 크게 두개의 생명주기

로 나누어 볼 수 있다. 첫번째 주기는 제품 패밀리 내 제품들의 공통성과 가변성을 구체화하는 재사용 자산을 생성하는 것이고, 두번째 주기는 생성된 자산을 기반으로 특화된 어플리케이션을 개발하는 것이다. 해당 영역에 대한 제품 계열 아키텍처를 정의하고 이를 기반으로 핵심자산을 개발한다. 이때 가변성을 관리하기 위하여 결정 모델(Decision model)을 정의하여 특화된 어플리케이션을 개발할 때 자산의 재사용 범위를 결정하게 된다.

개발한 자산의 재사용성을 증대시키기 위해서는 프로젝트 라인에서 가장 중요한 제품 간의 공통성과 가변성을 식별하고 이를 표현하는 기술이 잘 정의되어 있어야 한다. 이를 위해서는, 다양한 제품의 요구사항을 만족시켜 줄 수 있도록 자산을 모델링할 때 가변성을 제공하는 부분을 명세화하고 이를 독립적인 모델로 관리함으로써 자산의 재사용성을 향상시키는 방법이 필요하다. 기존의 프로젝트 라인 개발 방법에서 정의된 결정 모델은 테이블 형태로 기술하도록 되어 있으나, 테이블 간의 연관 관계는 개발자가 직접 매핑시켜 관리하여야 하는 어려움이 있다. 핵심 자산을 개발하면서 작성한 모든 모델들에 대하여 결정 모델을 작성하기 때문에 이들간의 관계를 파악하는 것은 쉽지 않다. 또한, 가변성은 자산의 기능 뿐 아니라 기능을 수행하기 위하여 입력하는 파라미터나 수행 조건들에 의해서도 기능의 흐름이 변경될 수 있으므로 다양한 측면에서 결정 모델이 관리되어야 한다.

본 논문은 핵심 자산 개발시 다양한 가변 요소를 고려한 결정 모델 표현 방법과 결정요소들 간의 관계를 추적할 수 있는 지원 도구의 프로토타입을 제안한다. 본 논문에서 제안하는 결정 모델은 다양한 가변점과 이를 결정하는 가변치의 종류를 다양하게 정의하는 것을 목표로 하며,

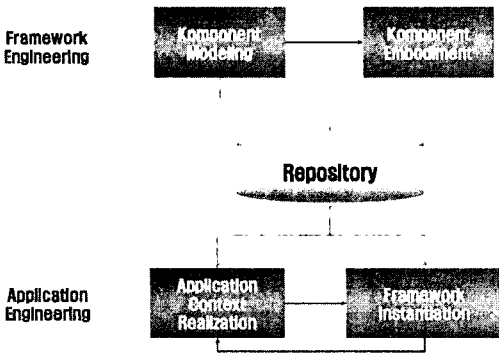
결정 모델 기술 지원 도구는 이들간의 관계를 시각적으로 정의할 수 있도록 지원한다.

본 논문의 구조는 다음과 같다. 2장에서는 본 논문의 관련 연구인 KobrA[4] 프로세스와 결정 모델에 대하여 살펴본다. 3장에서는 본 논문에서 제안하는 자산의 가변성을 관리하는 결정 모델의 모습과 기술 지원 도구의 모형을 소개하고 4장에서 향후 연구 방향을 언급한다.

2. 관련 연구

2.1 KobrA[4]

KobrA 방법론은 프로덕트 라인 패러다임을 실현하기 위한 재사용 가능한 프레임워크를 생성하고 이를 통해 실행 가능한 어플리케이션을 구축하는 작업을 지원하는 통합 방법론이다.



<그림 2> KobrA 프로세스

그림 2와 같이 KobrA 프로세스는 프레임워크 공학 프로세스와 어플리케이션 공학 프로세스로 구성되어 있다. 프레임워크 공학을 통하여 재사용 가능한 프레임워크를 생성하고, 이 프레임워크를 이용하여 어플리케이션 공학을 통하여 특정 어플리케이션을 구축한다. 즉 개별 제품들이 가지는 공통성과 가변성을 표현하는 Komponent와 그들간의 관계성을 계층적으로 명세화 한 프레임워크를 구축하고 이를 개별 제품 요구사항에 맞도록 인스턴스화한다. 범용적인 프레임워크를 개별 제품에 적용할 때는 프레임워크에 대한 결정 모델을 기반으로 자산의 가변성을 제품에 한정적으로 결정하여 자산을 인스턴스화한다.

자산에 대한 결정모델은 프로덕트 라인 멤버들간의 차별성을 규정지어주는 상위 결정 요소들과 제품들을 구성하는 가변요소들간의 관계를 정의하는 모델로서, 프로덕트 라인 멤버들의 피쳐들간의 계층구조에 따라 상위 아키텍처에서 하위 컴포넌트의 세부 모델링 요소에까지 영향을 준다. 결정모델은 아이디, 가변성, 가변성에 대한 가변치, 가변치에 대한 영향으로 구성되어 있다. KobrA에서는 자산 컴포넌트에 대하여 가변점을 기능 중심으로 기술하고 있으며 기능에 대한 가변치를 'yes', 'no'의 두 가지로 나눠 그에 대한 영향을 분석한다.

그러나, 실제로 핵심 자산을 사용하여 어플리케이션을

개발할 때에는 특정 제품에 대한 가변점이 기능에만 나타나는 것이 아니라, 공통 기능을 사용하는 입력 파라미터나 기능의 수행 조건들의 변화 때문에 자산의 기능을 그대로 사용할 수 있는 경우는 거의 없다. 가변점에 대한 가변치 또한 'yes' 나 'no'로 결정될 수 없는 경우가 빈번히 발생한다. 따라서, 결정 모델을 가변 기능을 중심으로 기술하는 테이블과 가변 속성 및 기능의 수행 조건 등을 기술하는 테이블로 나눠서 정의하고 이들간의 관계를 연결하여 관리하는 도구의 지원이 필요하다.

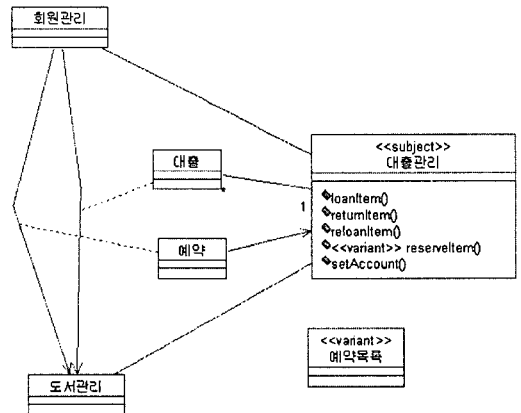
3. 결정 모델 기술 방법 및 지원 도구의 모형

본 장에서는 KobrA 결정모델의 구조를 확장하여 핵심 자산의 다양한 가변점을 고려하여 결정 모델을 기술하는 방법을 보이고 결정모델간의 추적성을 지원하는 도구의 모형을 제안한다.

3.1 결정 모델 기술 방법

결정 모델은 핵심 자산의 기능에 대한 가변성을 결정하는 부분과 핵심 자산 기능을 수행하기 위한 속성이나 조건 가변성을 결정하는 부분으로 나눌 수 있다. 전자의 경우는 가변치가 해당 메소드의 존재 여부를 묻는 것으로 정리할 수 있지만 후자의 경우는 변수의 값이나 조건문과 같은 다양한 형태로 정의되는 경우가 많다.

<그림 3>은 도서관 시스템 영역의 대출 관리 컴포넌트에 대한 다이어그램이다. <표 1>은 가변 기능인 예약 기능에 대하여 가변성을 관리하는 결정 모델을 보인다. 핵심 자산의 기능에 대한 Effect는 기능의 존재 여부에 의한 삭제 여부가 결정될 수 있다.

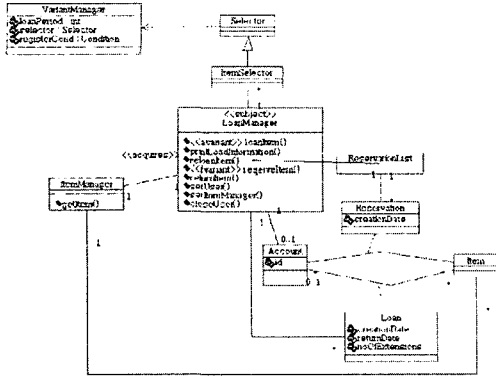


<그림 3> 대출 컴포넌트 모델링

ID	Variation	Resolution	Effect
CR1.1	예약가능하게 할 것인가?	yes	
		no	대출관리.reserveItem() 삭제

<표 1> 대출 컴포넌트의 기능 가변성을 표현한 결정 모델

핵심 자산의 가변 속성이나 수행 조건에 대한 Effect는 이를 독립적으로 관리하는 Manager 클래스의 변수를 세팅하는 방법으로 정의할 수 있다.



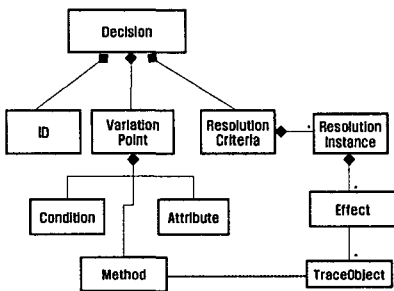
<그림 4> 대출 컴포넌트의 상세 모델링

ID	Variation	Resolution	Effect
KR1.1	대출기간	14일	VariantManager.loanPeriod=14
KR1.2	대출아이템	논문	VariantManager.selector =new PaperSelector()

<표 2> 대출 컴포넌트의 속성 가변성에 대한 결정 모델
 <그림 4>에서 보는 바와 같이 대출 기능에 대하여, 대출기간이나 대출하는 아이템은 가변 속성이 되며, 이는 실제 개발하는 어플리케이션의 요구사항에 명시된다. 이러한 속성들은 VariantManager 클래스에서 따로 관리하여 세팅할 수 있도록 모델링하면, <표 2>와 같이 결정모델의 Effect를 간단히 정의할 수 있다.

3.2 결정 모델 기술 지원 도구

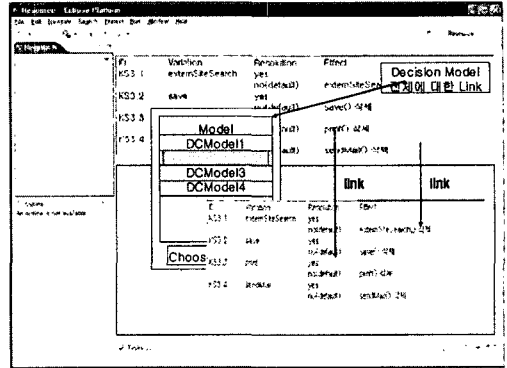
재사용성이 높은 핵심 자산을 만들어 이를 특정 어플리케이션 개발에 사용하기 위해서는 3.1에서 언급한 결정 모델을 정의하고 관리할 수 있는 도구의 지원이 필요하다. <그림 5>는 도구에서 지원하는 결정모델의 구조를 나타낸 것이다.



<그림 5> 결정 모델

하나의 결정 요소는 아이디와 가변점, 가변치를 정의할 수 있는 분기점이 있고, 정의된 가변치 인스턴스마다 영향을 받는 Effect들이 정의된다. 본 논문에서는 가변점의 종류를 가변 메소드나 클래스, 가변 속성, 가변 조건으로 나누었다. Effect를 고려할 때, 가변 속성과 가변 조건은 Manager 클래스에서 관리하기 때문에 다른 모델링 정보에 변경을 주지 않지만, 가변 메소드나 클래스는 그 가변점과 관계있는 메소드나 클래스의 삭제에도 관여할 수

있기 때문에, 가변 메소드나 클래스간의 관계를 추적할 수 있는 TraceObject를 두어 관리한다. <그림 6>은 이러한 결정 모델의 추적성을 관리할 수 있도록 지원하는 도구의 모형을 나타낸 것이다.



<그림 6> 결정 모델을 관리하는 지원 도구의 모형

제품 계열의 요구사항을 분석하여 모델링된 정보들에 대하여 작성된 결정 모델들의 리스트를 통합적으로 관리하고 가변 메소드나 가변 클래스들에 대하여 추적 관리를 지원한다.

4. 결론 및 향후 연구

소프트웨어를 개발하는데 재사용 측면에서 프로젝트 라인 공학은 소프트웨어 개발 노력에 비해 높은 이익을 제공해주는 접근 방법이다. 개발한 자산의 재사용성을 증대시키기 위해서 프로젝트 라인에서 가장 중요한 제품 간의 공통성과 가변성을 식별하고 이를 표현하는 기술이 잘 정의되어 있어야 한다. 본 논문은 다양한 제품의 요구사항을 만족시켜 줄 수 있도록 자산을 모델링할 때 가변성을 제공하는 부분을 결정 모델로 명세화하고 이들간의 연관성을 관리할 수 있는 모델 및 지원도구를 제안하였다.

향후 다양한 가변성을 표현하는 결정 모델을 관리할 수 있는 지원도구를 개발하고, 결정 모델의 Effect 정의부분을 간소화 시킬 수 있는 핵심 자산 모델링 기법에 대한 연구가 필요하다. 또한, 모델링된 핵심 자산을 구현할 때 가변포인트를 적용하여 제품 개발시 결정되는 요구사항을 핵심 자산 코드에 쉽게 반영할 수 있는 핵심 자산 코드 생성 패턴에 대한 연구가 필요하다.

5. 참고문헌

[1] J. Bosch, Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach, Addison-Wesley, Boston, 2000.
 [2] P. Clements and L. Northrop, Software Product Lines: Practices and Patterns, Boston, MA: Addison Wesley Longman, Inc., 2001.
 [3] K.C. Kang et al., " Feature-Oriented Product Line Engineering," IEEE Software, Vol. 9, No. 4, Jul./Aug. 2002, pp. 58-65.
 [4] Atkinson, C. et al, Component-Based Product Line Engineering with UML, Addison Wesley, 2002.