

이클립스를 위한 UML 기반 특성모델링 플러그인의 설계

임용섭⁰ 김지홍
경원대학교 전자계산학과
{redbean⁰, wiskjh}@ku.kyungwon.ac.kr

Design and Implementation of Feature Modeling Plug-in for Eclipse

Yongsub Lim⁰ Jihong Kim
Dept of Computer Science, Kyungwon University

요 약

소프트웨어 프로젝트 라인 공학은 공통성과 가변성 식별을 통해 소프트웨어 재사용을 극대화 시키는 접근방법으로 이를 지원하는 도구에 대한 연구가 다수 진행되고 있다. 하지만 기존의 도구는 특성 다이어그램만을 지원하고 있고, 특성구성 정보를 이용한 프로젝트의 조립을 수행하지 못하고 있다. 따라서 본 연구에서는 소프트웨어 개발자에게 공통적 언어인 UML을 이용하여 특성모델을 표현하고 특성구성에 따라 조립을 수행하는 특성모델링 지원도구를 제안한다. 이러한 UML 기반의 특성모델링 지원도구를 통하여 보다 쉽게 특성모델링과 프로젝트 조립을 수행할 수 있다.

1. 서 론

최근 단시간에 좋은 품질의 소프트웨어 개발하기 위한 방법으로 소프트웨어 프로젝트 라인에 대한 연구가 활발히 진행되고 있다. 소프트웨어 프로젝트 라인은 특정 시장이나 목표의 요구를 만족시키기 위하여 공통된 핵심 소프트웨어 자산을 공유하는 소프트웨어 시스템들의 집합이다[1].

소프트웨어 프로젝트 라인은 각 프로젝트들의 공통성과 가변성을 식별, 명세, 관리하는데, 이를 위한 방법으로 특성모델이 많이 이용되고 있다. 따라서 이러한 특성모델링을 지원하는 도구에 대한 연구가 다수 진행되고 있다[2,3,4]. 또한 재사용가능한 플러그인 기법으로 보다 편리한 도구 개발환경을 제공하고 통합개발환경에 모델링기능을 통합할 수 있는 장점을 가진 이클립스[5]를 기반으로 개발되는 도구도 늘어나고 있는 추세이다[3,4].

현재 연구 개발되고 있는 특성모델링 지원 도구들은 FODA[6]에서 소개된 특성 다이어그램만을 지원하고 있다. 이러한 FODA의 특성 다이어그램의 표현은 간단명료한 특징을 가지고 있지만, 일반 소프트웨어 개발자가 접근하기에 어려운 점이 있다.

한편, 일반적인 UML 모델러는 단일 시스템을 위한 객체지향적 모델링만을 지원될 뿐 복수 시스템의 공통성과 가변성의 표현이 조립과 같은 적극적인 재사용으로 연결되지 않아 소프트웨어 프로젝트 라인 공학을 실질적으로 지원하기에는 부족한 점이 많다.

따라서 본 연구에서는 UML을 이용한 특성모델링을 지원하고 특성구성에 따른 프로젝트 조립을 수행하는 도구를 제안한다.

2. 관련연구

2.1 소프트웨어 프로젝트 라인

소프트웨어 프로젝트 라인은 특정 소프트웨어 시스템에서 사용되는 핵심자산을 공유함으로써 재사용을 극대화하여 생산성을 얻고자하는 접근방법이다. 소프트웨어 프로젝트 라인은 공통성과 가변성을 갖는데, 여기서 가변성이란 시스템을 고객화하거나 변경하기 위한 능력을 말한다. 이러한 가변성은 프로젝트 라인이 가지는 가변적인 부분을 요구사항의 관점에서 표현되는데, 일반적으로 특성모델이 많이 이용된다.

모든 프로젝트에 존재하는 공통성은 단순하게 컴포넌트와 같은 기반구조로 연결되고 재사용될 수 있으므로 쉽게 다룰 수 있지만, 특정 프로젝트의 부분으로 존재하는 가변성은 기반구조가 가변성으로 인해 영향을 받는 부분을 정의해야 하므로 다루기 어렵다[7].

2.2 이클립스와 특성모델링 플러그인

이클립스 플랫폼은 이클립스 프로젝트에서 개발한 확장가능한 개발 플랫폼으로, 런타임 커널을 제외한 모든 것을 확장하는 플러그인 방식으로 보다 광범위한 기능의 확장을 지원함으로써 기존의 플러그인을 다른 플러그인이 쉽게 재사용할 수 있도록 설계되었다.

이클립스 플러그인으로 작성된 Pure:Variants[3]는 트리-뷰를 사용하여 특성모델링과 특성구성을 지원하는 상업용 도구이다. FeaturePlugin[4]은 Pure:Variants와 비슷하게 트리-뷰를 사용하여 특성모델링과 특성구성을 지원하는 연구용 도구로 카디널리티를 지원한다는 점이 특징이다. 특성모델은 트리구조를 가지고 있기 때문에 트리-뷰로 매핑될 수 있는데, 트리-뷰는 트리구조의 노드를 옆방향으로 확장하기 때문에 많은 수의 특성을 다룰 수 있지만, 다이어그램으로 표현하지 않기 때문에 직관적으로 이해하고 다루기에 불편한 점이 있다.

3. UML 기반 특성모델링 도구의 설계

본 연구에서 제안하는 도구는 UML을 이용하여 특성모델을 작성하고, 특정한 프로덕트 멤버를 구성하는 특성을 결정하면 이에 따른 실제 프로덕트를 생성한다.

UML 기반 특성모델링 지원도구는 [그림1]과 같이 3가지로 구성된다. 지원도구는 UML로 특성모델을 작성할 수 있도록 하는 특성모델러, 특정 프로덕트 멤버의 특성구성을 지원하는 특성구성기, 특성 구성정보와 특성-요소매핑정보를 기반으로 특정 프로덕트를 구성해주는 프로덕트 조립기로 구성되어 있다.

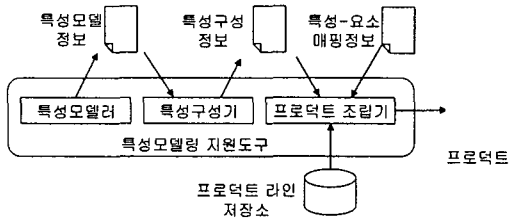


그림1. 특성모델링 지원도구 구조도

3.1 특성모델러

특성모델러는 UML로 특성모델을 표현할 수 있도록 돕는 역할을 수행한다. 특성모델러는 UML을 그대로 이용하므로 일반적인 UML모델러의 기능과 유사하지만, 사용자가 작성한 특성모델을 특성구성기에서 인식할 수 있는 특성모델정보로 도출할 수 있어야 한다. 본 연구에서는 [그림2]와 같이 특성모델정보를 다루기 위한 메타모델을 정의하였으며, UML을 이용한 특성모델의 표현은 PLUS[8]에서 소개된 방법을 이용하였다.

3.2 특성구성기

특성구성기는 특성모델러로 작성된 특성모델정보를 기반으로 특정 프로덕트 멤버가 어떠한 특성으로 구성되는지를 지정할 수 있도록 돕는 역할을 수행하고, 특성구성정보를 생성한다. 예를 들어, 도서관 프로덕트 라인에서 예약이나 대출연장 특성이 프로덕트에 적용되는지를 입력받아 특성구성정보를 생성한다. 특성구성기의 사용자 인터페이스는 특성모델러와 연결되어, 특성모델 다이어그램을 이용하여 특성구성을 할 수 있도록 지원한다.

특성구성기는 사용자가 특성을 선택할 때, 특성의 가진 구성들에 따라 다른 특성의 선택에 영향을 미친다. 예를 들어, A특성이 B특성을 필요로 한다면 A특성이 선택될때, 자동적으로 B특성이 선택되어야한다. 이러한 구성들을 반영할 수 있는 모델로 [그림2]와 같은 모델을 사용하였다.

3.3 프로덕트 조립기

프로덕트 조립기는 특성구성기로 통해 생성된 특성구성정보, 각 특성과 관련된 컴포넌트 또는 클래스들이 존재하는 프로덕트 라인 저장소, 각 특성과 어떠한 컴포넌트, 클래스, 메소드 요소들이 연결되는 지에 대한 특성-

요소매핑정보를 기반으로 실제 프로덕트를 구성한다.

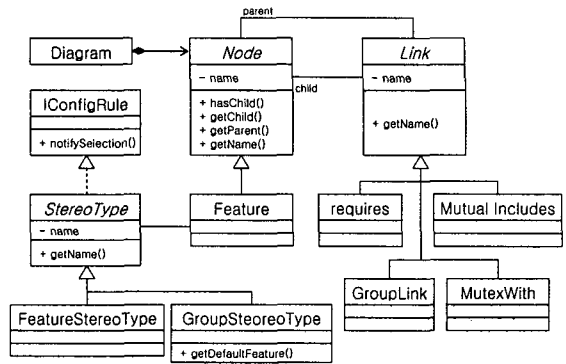


그림2. 특성모델러와 구성기를 위한 메타모델

4. 프로토타입 및 실행

4.1 프로토타입

이클립스 플랫폼은 프레임워크의 형태로 메뉴, 툴바, 편집기와 같은 기본적인 도구를 제공한다. 따라서 본 연구에서는 이클립스 플랫폼의 편집기를 그래픽 편집기로 확장함으로써, UML을 표현하고 다룰 수 있는 특성모델링 지원도구의 프로토타입을 개발하였다.

프로토타입은 [그림3]과 같은 아키텍처로 개발되었다. 특성모델러와 특성구성기는 다이어그램을 지원하므로 각 UML요소를 표현하는 뷰어와 사용자 이벤트에 따른 다양한 기능을 제공하는 컨트롤러가 필요하다. 또한, 특성모델정보와 특성구성정보를 생성하기 위해서는 이러한 정보를 가진 메타특성모델과 특성구성모델이 필요하며, 최종적으로 프로덕트를 조립하는 기능을 제공하기위해 프로덕트 조립기가 포함된다. 본 연구는 이클립스 3.1버전을 기반으로 하여 뷰어와 컨트롤러에 Graphical Editor Framework[9]를 이용하고, 모델에 Eclipse Modeling Framework[10]를 이용하여 개발의 노력을 최소화 하였으며, HyperJ[11]를 이용하여 프로덕트 조립기를 개발하였다.

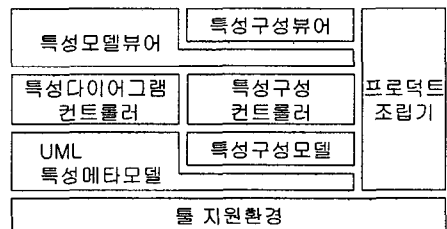


그림3. 특성모델링 지원도구 아키텍처

4.2 실행

[그림4]는 UML을 기반으로 특성모델링을 수행할 수 있는 특성모델러를 보여준다. 사용자는 특성이아이콘과 관계아이콘을 선택하여 다이어그램에 특성을 표현하고 특성간의 관계를 표현할 수 있다.

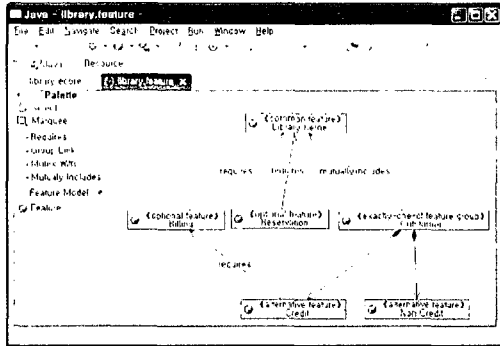


그림4. UML기반 특성모델러의 실행화면

[그림5]는 특정 프로덕트의 특성들을 구성할 수 있는 특성구성기의 사용자 인터페이스를 보여준다. 특성 구성기는 특성표현의 왼쪽상단의 체크박스를 통하여 특성이 프로덕트 조립에 사용되는지를 결정할 수 있다.

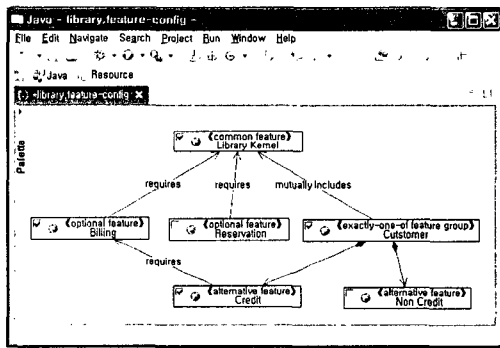


그림5. 특성구성기의 실행화면

[그림6]은 도서관 프로덕트 라인에서 예약특성이 선택된 프로덕트를 조립하는 프로덕트 조립기 실행의 한 가지 예를 보여준다. 프로덕트 조립기는 특성구성정보로 다음과 같은 HyperJ의 프로덕트 조립 명세서를 생성하여, HyperJ로 하여금 조립을 수행하도록 함으로써 실제 프로덕트를 손쉽게 구성해 주었다[12].

```

hypermodule LibrarySystemAspect
hyperslices:
  Feature.Core,
  Feature.Reservation;
relationships:
  mergeByName;
end hypermodule;
    
```

그림6. 프로덕트의 조립지시를 위한 HyperJ의 명세서

5. 결론 및 향후연구

기존의 특성모델링 지원도구들은 FODA의 특성 다이어그램만을 이용하여 특성모델을 표현하고 특성구성을 지원하였다. 특성 다이어그램의 표현방법은 간단명료하여 유용한 점을 가지고 있지만, 일반 개발자가 접근하기

에는 어려운 점이 있다. 또한 기존의 도구에서는 특성모델정보가 조립과 같은 적극적인 재사용으로 연결되지 못하였다.

따라서 본 연구는 UML을 이용하여 특성모델을 표현하고, 특성구성에 따라 조립을 실제 프로덕트의 조립을 수행하는 특성모델링 지원도구 프로토타입을 개발하여 보다 쉽게 특성모델을 이용하고, 손쉽게 실제 프로덕트를 조립할 수 있었다. 본 연구에서 제안한 도구의 향후연구방향으로는 프로덕트 라인 아키텍처의 지원과 Aspect Oriented Programming의 지원 등이 있을 수 있다.

참고 문헌

- [1] P. Clements and Linda Northrop, "Software Product Lines: Practices and Patterns", Addison Wesley, 2002
- [2] M. Selbig, AmiEddi, 2000-2004. Tool available at <http://www.generative-programming.org>
- [3] Olaf Spinczyk, Danilo Beuche, "Modeling and Building Software Product Lines with Eclipse", Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages and applications, pp.18-19, 2004
- [4] M. Antkiewicz, K. Czarnecki, "FeaturePlugin: Feature Modeling Plug-In for Eclipse", OOPSLA'04 Eclipse Technology eXchange (ETX) Workshop, 2004
- [5] Eclipse.org, available at <http://www.eclipse.org>
- [6] K. Kang, et. al, "Feature-Oriented domain analysis feasibility study. Technical Report CMU/SEI-90TR-21", Software Engineering Institute, Carnegie Melon University, Pittsburgh, 1990.
- [7] Lars Geyer, Marin Becker, "On the Influence of Variabilities on the Application-Engineering Process", In proceedings of the Software Product Lines Conference, Springer-Verlag LNCS 2379, pp.1-14, 2002
- [8] Hassan Gomma, "Designing Software Product Lines with UML", Addison Wesley, 2004
- [9] Graphical Editing Framework Project, available at <http://www.eclipse.org/gef>
- [10] Eclipse Modeling Framework Project, available at <http://www.eclipse.org/emf>
- [11] Harold Ossher, Peri Tarr, "Multi-Dimensional Separation of Concerns and The Hyperspace Approach", Proceedings of the Symposium on Software Architectures and Component Technology, 2000
- [12] 임용섭, 김지홍, "Aspect를 이용한 소프트웨어 프로덕트 라인 아키텍처의 재구성", 인터넷정보학회 춘계학술발표논문집 제6권 제2호, pp.401-404, 2005