

제품라인모델로부터 제품모델을 추출하는 기법 및 도구의 일반화

이지원, 이관우

한성대학교 정보시스템공학과

e-mail:jiwonlee@hansung.ac.kr, kwlee@hansung.ac.kr

Generalization of methods and tools for extracting product models from product line models

Ji-Won Lee, Kwan-Woo Lee

Dept of Information Systems Engineering, Hansung University

요 약

제품 라인 공학의 핵심은 여러 제품 개발에 쉽게 재사용 될 수 있는 핵심 자산의 개발과 산출된 핵심자산을 이용하여 원하는 제품을 생산함에 있다. 그렇기 때문에 제품 라인 공학에서 원하는 제품 모델을 적기에 생산해내도록 도와주기 위하여, 제품 라인 모델의 자산으로부터 제품 모델을 추출해주는 도구를 필요로하게 된다. 사용자가 필요로하는 제품 라인 산출물의 추출을 도와주기 위해서는 제품 라인 모델로 산출될 수 있는 모든 모델을 고려할 필요가 있다. 하지만 모든 제품 라인 모델로부터 제품 모델을 추출하는 모듈을 개별적으로 구현하는 것은 비생산적이다. 따라서 본 연구에서는 사용자 맞춤형 제품 모델 추출 도구의 구현을 위해, 오픈 소스인 StarUML을 이용하여 제품 모델 추출 기법의 일반화를 제안한다.

1. 서론

제품 라인 공학[1,2,3]은 동일한 도메인에 속한 여러 제품을 쉽게 재사용이 가능하도록 핵심자산을 구축하고 만들어진 핵심자산을 통해 원하는 제품을 생산하는 과정으로 이루어져있다. 대표적인 제품 라인 공학인 FORM[1]은 휘처[1]를 기반으로 가변성을 식별한다. FORM을 바탕으로 제품 라인 모델로부터 원하는 제품을 추출하는 과정은 제품 구성에 선택된 휘처에 따라 휘처를 구현하는 산출물(컴포넌트 혹은 아키텍처)을 추출해내는 것이다.

따라서 휘처의 선택에 따라서 제품 라인 모델로부터 제품 모델을 추출해주는 도구가 필요하게 된다. 또한, 제품 라인 모델 이해관계자들에 따라 다른 제품 라인 모델 생성 프로세스를 가질 수 있기 때문에 이용자 맞춤형으로 필요한 제품 라인 모델의 추가가 자유로울 필요가 있다.

하지만 상이한 제품 라인 모델 프로세스 별로의 제품 모델 추출 모듈의 추가는 비생산적이고 비효율적이다. 그러므로 이를 해결하기위해 본 논문에서는 제품 라인 모델링 산출물들의 공통점을 분석하여 이를 일반화하고 사례에 적용시켜 효율성을 입증한다.

핵심 자산의 표현으로는 UML기반의 프로덕트라인 아키텍처 모델링[4]의 기법을 따랐으며, UML 모델링 관련 도구로 프리웨어인 StarUML[5]을 사용한다.

본 논문의 구성은 다음과 같다. 2장에서는 기반 연구로써 UML기반의 제품 라인 가변성 아키텍처 모델과 관련

연구에 대해 소개한다. 3장에서는 제품 라인 아키텍처 모델에서 제품 모델을 추출화한 기법과 이의 일반화를 설명하고, 4장에서는 3장의 결과로 나온 일반화 모듈을 구현하고 다른 프로세스 산출물에 적용시켜본 사례 연구에 대해 평가한다. 그리고 마지막 5장에서는 결론 및 향후 계획을 정리한다.

2. 기반 연구

2.1 UML기반의 제품 라인 가변성 아키텍처 모델

UML기반의 제품 라인 가변성 아키텍처 모델[6]은 UML을 이용하여 아키텍처 모델을 개념, 프로세스, 배치, 모듈의 네가지 관점에서 모델링하고 UML의 확장 메커니즘인 스테레오타입을 이용하여 가변성을 표현한 모델이다.

그림 1은 가변성을 표현한 개념 컴포넌트이다. airTempSensor컴포넌트에 airTempReading 휘처가 대응되어있음을 스테레오타입으로 표현한 것이다.



(그림 1) 개념 컴포넌트의 가변성 표현

자세한 내용은 UML기반 프로덕트 라인 아키텍처 모델

링의 가변성 표현 기법[6]을 참조하기바란다.

표 1은 네가지 관점의 아키텍처 모델을 UML로 모델링 하였을 경우, 나오는 산출물은 정리한 표이다. 본 논문에서는 해당 산출물을 통해 제품 모델 산출물을 추출하는 기법을 제안하고 그 기법을 일반화한다.

<표 1> 아키텍처 모델의 산출물

모델	산출물
개념 아키텍처모델	UMLComponentDiagram
프로세스 아키텍처모델	UMLComponentDiagram
배치 아키텍처모델	UMLDeploymentDiagram
모듈 아키텍처모델	UMLClassDiagram

2.2 관련 연구

최근에는 이클립스[7]에서 제공하는 플러그인 기법으로 휘쳐모델을 모델링하고 관리할 수 있는 휘쳐모델링 도구에 관한 연구들이 많이 진행되고 있다.

Pure::Variants[8]는 Enterprise Architect[9] UML 모델링 도구와 통합하여 제품 라인을 지원하는 상용용 이클립스 플러그인이다. Pure::Variants는 제품라인공학의 전반의 과정을 지원하는 장점을 가지고 있다. 그러나 UML모델이 아키텍처 모델을 지원하지 않기 때문에 아키텍처 모델의 표현이 불가능한 문제점을 가지고 있다.

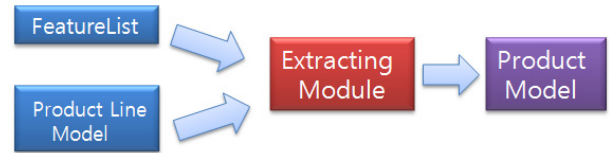
제품 라인 공학에서 컴포넌트 기반의 제품 라인 아키텍처는 가장 중요한 산출물 중의 하나이다. 그렇기 때문에 본 연구에서는 제품 라인 아키텍처로 모델을 UML로 표현하고 제품 라인 아키텍처 모델부터 제품 아키텍처 모델을 추출하는 기법을 제안한다.

또한 가변성 관리의 측면에서, Pure::Variants도구에서는 가변성을 Constraint로 표현하여준 반면, 해당 연구에서는 UML의 Stereotype을 사용하여 표현하므로써 가독성을 높인다.

다음 장에서는 스테레오타입을 이용한 가변성 표현에 따라, 제품 라인 모델에서 선택된 휘쳐에 대응되는 제품 아키텍처 모델을 추출하는 기법을 설명하고 제품 모델 추출 기법을 일반화시킨다.

3. 제품모델 추출화 기법 및 일반화

제품 라인 모델에서 제품 모델을 추출할 때, 우선 선택된 휘쳐들로부터 구성된 휘쳐리스트를 구성한다. 휘쳐리스트에 속한 휘쳐들에 대응되어있는 엘리먼트들만이 제품 모델로 생성된다. 그러므로 입력값으로 휘쳐리스트와 제품 라인 모델을 받고 이를 처리하여 제품 모델을 추출한다.

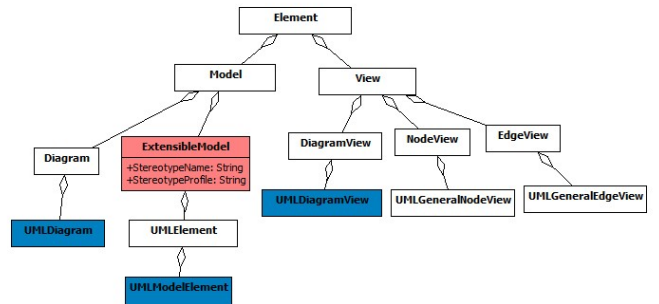


(그림 2) 제품 모델 추출 방식

본 논문에서는 가변 정보를 스테레오타입에 명시하기로 하였으므로, 엘리먼트의 스테레오타입 정보에 엘리먼트와 대응되는 휘쳐를 입력한다. 스테레오타입 정보에 있는 휘쳐들이 휘쳐리스트에 포함되지 않는 경우, 해당 엘리먼트를 제거해주면 된다.

그렇다면, 제거되어야하는 대상이 되는 엘리먼트 단위의 삭제에 의해서만 제품 모델이 생성될 수 있는 지 알아볼 필요가 있다.

그림 3은 StarUML의 메타모델[10]의 일부분이다. 스테레오타입을 기반으로 가변성을 표현하므로 StereotypeName을 애트리뷰트로 가지는 ExtensibleModel 요소의 하위에 위치한 요소들에 가변성이 있다고 판단한다.



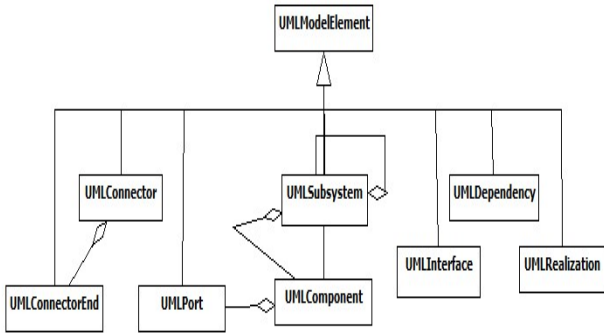
(그림 3) StarUML의 메타모델

그림 3의 UMLDiagram 요소는 모든 다이어그램을 하위 요소로 가지고 있고, UMLDiagramView는 각 다이어그램의 뷰를 가진다. 그리고 UMLModelElement는 다이어그램 안에서 표현되는 엘리먼트들을 가진다. 그림 4는 개념 아키텍처모델에서 사용되는 엘리먼트들의 관계이고 이를 UMLModelElement로 일반화시킨 것이다. 그림 4의 모든 엘리먼트들은 스테레오타입을 속성으로 가지고 있고 가변 요소를 가지고 있음을 알 수 있다.

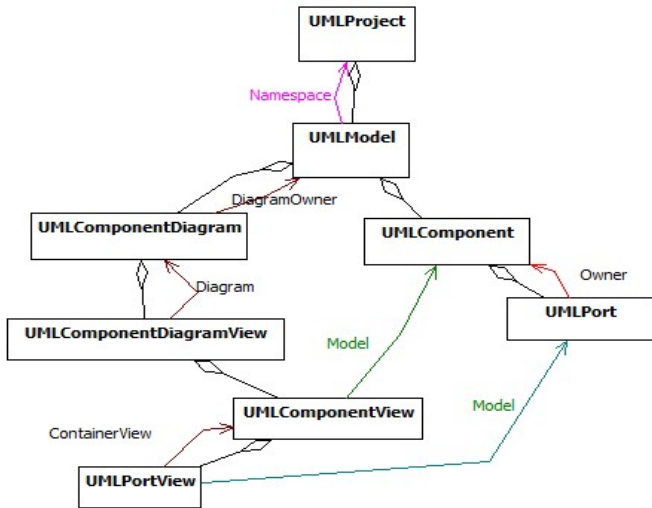
만약 UMLComponent 엘리먼트가 제거된다면, 그림 4에서 알 수 있듯이 UMLComponent 엘리먼트만 제거되는 것이 아니라, 이에 포함되어있는 UMLPort도 제거되어야한다.

그림 5는 개념 아키텍처를 표현하는 UMLComponent Diagram의 메타모델이다. 그림 5에서 Direct Association 관계로의 표현은 한 엘리먼트가 다른 엘리먼트를 참조하는 관계에 있음을 알려준다. 따라서 UMLComponent를 삭제할 경우 UMLPort뿐만아니라 UMLComponent를 Model

로써 참조하는 UMLComponentView와 이에 속해있는 UMLPortView 또한 삭제되어야한다. 이 규칙은 컴포넌트 다이어그램 뿐만 아니라 모든 UML모델에서도 적용이 될 수 있는데 표2는 엘리먼트의 일반적인 제거 규칙을 정리한 표이다.



(그림 4) 개념 아키텍처모델 엘리먼트의 일반화



(그림 5) UMLComponent Diagram 메타모델

그러나, 개념 아키텍처 모델에서 Port를 제거할 경우 Port를 참조하고 있는 ConnectorEnd가 제거되는데, ConnectorEnd는 Connector 존재의 충분조건이므로 Connector 또한 제거되어야하는 경우가 발생한다. 이는 엘리먼트와 엘리먼트의 연결관계가 한 엘리먼트가 제거될

<표 2> 엘리먼트 일반적인 제거 규칙

Condition	Action
E가 제거될 경우	E를 aggregation하는 E를 모두 제거
제거되는 E가 참조하는 E가 있는 경우	참조하는 E를 제거

으로써 사라져야하는 관점에서 고려해야할 사항이다. 표 3

(E: Element)

은 UML을 이용한 제품 라인 가변성 아키텍처 모델 추출 시 생성되는 제품 산출물에 대해서 고려해야하는 케이스를 정리한 표이다.

표 3의 고려 사항을 보면 프로세스별 산출물의 유의사항이 해당 산출물의 엘리먼트를 제거할 경우인데, 이는 모두 엘리먼트들 간의 관계에 관련되어있다. 클래스 다이어그램 산출물에서도 Class를 제거할 경우에, 제거할 Class를 참조하는 associationEnd요소를 제거해주어야한다. 하지만 associationEnd요소는 association요소의 충분조건이므로 association또한 제거되어야하는 사항을 고려해야한다.

이러한 제품 산출물 추출 기법의 일반화와 고려 사항을 바탕으로 4장에서는 사례 연구를 통해 이를 적용시킨다.

4. 구현 및 평가

표4는 엘리먼트를 제거하기 위한 과정을 표현한 Pseudo Code이다. featureList를 인자로 넘겨받아 엘리먼트의 스테레오타입에 대응된 휘처가 휘처리스트에 존재하지 않는다면 이를 삭제한다. 그리고 삭제되는 엘리먼트를 참조하는 하위 엘리먼트와 참조 엘리먼트를 모두 삭제한다. 그리고 삭제되는 엘리먼트가 다른 엘리먼트와 관계를 갖을 때, 고려해야하는 추가 제거 요소를 제거한다.

해당 프로시저는 자바로 구현을 하였다. 각 다이어그램 별로 구현을 하면, 다이어그램 별로 클래스를 만들고 개별적으로 인스턴스화하여하는 과정이 필요하게 되며, 약 1700라인의 코드를 작성해주어야한다. 반면 중복되는 코드를 일반화시켜 모듈화하였더니 약 550라인의 코드로 크기가 축소되어 효율성을 높일 수 있었다.

<표 3> 가변성 아키텍처 모델 제품 산출물 추출 시 고려 사항

UML Diagram	고려할 케이스	제거 요소와의 관계	제거 요소	추가 제거 요소
Component Dia.	port가 제거될 경우	“Role”	“UMLConnectorEnd”	“UMLConnector”
Class Dia.	class가 제거될 경우	“Participant”	“UMLAssociationEnd”	“UMLAssociation”
Deployment Dia.	node가 제거될 경우			

<표4> removeElement Pseudo Code

```

procedure removeElement(featureList):
create a queue Q
Element E incident on guidMap:
  if E has StereotypeName:
    if StereotypeName has not feature:
      E is offered into Q.
    if E has reference attribute:
      referencing E is offered into Q.
    else if referenced E is:
      referencing Infomation is offered into Q.
  else if E has relationship:
referencing E's parent is offered into Q.
  remove(Q.poll()).
    
```

또한 아키텍처 산출물 모델을 통해 일반화한 해당 프로시저의 효율성을 테스트하기 위하여 요구사항 분석단계에서 사용되어지는 유스케이스 모델과 동적 모델에 적용시켜보았다.

표 5는 유스케이스 모델과 동적 모델의 대표적인 가변점과 UML산출물을 정리한 표이다. 유스케이스모델의 대표적인 가변점인 유스케이스와 액터의 스테레وتا입에 휘차를 대응시켜 가변성을 표현하고 일반화 모듈을 적용시켜본 결과 제품 라인 유스케이스 모델에서 제품 유스케이스 모델로의 추출이 잘 수행되었다. 반면에 동적모델에서는 고려할 사항이 발견되었다. 까닭은 유스케이스 모델에서는 유스케이스가 삭제될 경우 유스케이스를 참조하는 관계가 클래스를 참조하는 관계와 같아 고려 사항에서도 해당 모듈이 적용된 반면, 동적 모델에서는 오브젝트가 제거될 때, 오브젝트를 참조하는 관계가 다른 모델과 구조는 같지만 명칭이 다르기 때문이다. 이와 같은 문제는 제거요소의 명칭을 파라미터화하여 해당 모듈에 인자값을 넘겨줌으로써 해결이 가능하다.

<표 5> 구현 결과를 테스트한 모델의 가변점 및 산출물

모델	가변점	산출물
유스케이스 모델	유스케이스	UMLUsecase
	액터	Diagram
동적 모델	오브젝트	UMLCollaboration
	스티물러스	Diagram

해당 사례연구를 통해 다이어그램 구조의 공통점을 일반화시킨 모듈이 재사용이 가능하다는 것을 확인하였다.

다음 장에서는 본 논문의 결론과 향후 계획에 대해서 설명한다.

5. 결론 및 향후 계획

해당 연구에서는 제품 라인 모델에서 제품 모델을 추출하고 사용자가 원하는 다른 모델에도 적용이 가능하도록 일반화하는 과정을 연구 및 제안하였다. 또한 일반화를 지원하는 도구의 개발을 통해 연구의 실현성을 입증하고 사용자의 편의성을 도모하였다. 향후에는 본 연구에서 다루지 않은 다른 제품 라인 모델링 모델에 대한 예외 사항을 고려하고 이에 대응 가능한 방안을 데이터화할 예정이다.

참고문헌

[1] K. C. Kang, S. Kim, J. Lee, K. Kim, E. Shin, M. Huh, "FORM: A Feature-Oriented Reuse Method with Domain Specific Reference Architectures", Ann. Soft. Eng. 5, pp. 143--168, 1998

[2] C. Atkinson et al., Component-based product line engineering with UML. Addison-Wesley, London, New York, 2002

[3] P. America, H. Obbink, J. Muller, and R. van Ommering, "COPA: A Component-Oriented Platform Architecting Method for Families of Software Intensive Electronic Products," Denver, Colorado: The First Conference on Software Product Line Engineering, 2000.

[4] 이관우, 이지원, "UML 기반의 프로덕트 라인 아키텍처 모델링", 한국정보과학회, 한국컴퓨터종합학술대회 논문집 (A), p.210~213, 2011,

[5] StarUML, <http://staruml.sourceforge.net/ko>

[6] 이관우, 이지원, "UML기반 프로덕트 라인 아키텍처 모델링의 가변성 표현 기법", 한국정보처리학회, 추계학술발표대회 논문집 제 18권 제 2호 p.1372~1376, 2011.

[7] Eclipse, <http://www.eclipse.org>

[8] 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

[9] Enterprise Architect, <http://www.sparxsystems.com>

[10]<http://staruml.sourceforge.net/docs/api-doc/index.html>