

UML기반 프로덕트 라인 아키텍처 모델링의 가변성 표현 기법

이관우, 이지원
한성대학교 정보시스템공학과
e-mail: kwlee@hansung.ac.kr, jiwonlee@hansung.ac.kr¹⁾

A Technique for variability Modeling of UML based Product Line Architecture

Kwan-Woo Lee, Ji-Won Lee,
Dept of Information Systems Engineering, Hansung University

요 약

프로덕트 라인 공학에서 프로덕트 라인 아키텍처는 가장 중요한 산출물 중에 하나이다. UML (Unified Modeling Language) 2.x 부터는 아키텍처를 모델링하기 위한 유용한 모델링 요소를 제공하고 있다. 이러한 UML을 이용하여 프로덕트 라인 아키텍처를 모델링하기 위해서는 가변성의 표현이 명시적으로 이루어져야 하지만, UML 자체에는 가변성을 표현하기 위한 방법 및 기법을 명시적으로 기술하고 있지 않다. 본 논문에서는 UML에서 제공하는 확장 메커니즘을 이용하여 가변성을 표현하는 방법을 제안한다. 즉, 모델링요소에 태그값(Tagged Value) 및 스테레오타입을 넣어주어 다양한 관점에서의 가변성을 표기하는 방법을 제안한다.

1. 서론

소프트웨어 프로덕트 라인 공학 (software product line engineering)에서 재사용의 대상이 되는 산출물을 핵심자산이라고 부르고, 핵심자산 중에 가장 중요한 자산은 프로덕트 라인 아키텍처이다. 프로덕트 라인 아키텍처는 만들어질 때 가변성과 재사용성을 고려하여 구성되어야 하므로, 아키텍처 모델 상에서도 프로덕트 라인 범위 내의 시스템간의 가변요소를 표현해주고 관리해줘야 한다.

그동안 아키텍처를 모델링하기 위한 많은 아키텍처 기술 언어[1]가 개발되어 왔으나, 이들 아키텍처 기술 언어는 각자 고유한 표기법을 정의하고 있으므로, 서로 다른 아키텍처 기술 언어로 표기된 모델 간에 호환성이 떨어지는 문제가 발생한다.

최근 들어 산업계 표준으로 자리잡고 있는 모델링 언어인 UML (Unified Modeling Language) [2]은 아키텍처 개념을 표현하기 위한 표기법을 확장해 오고 있다. 하지만, 현재 UML 2.3의 표기법에는 커넥터 개념 및 다중 관점을

표기하는 데는 한계가 있다 [4]. 따라서, 이러한 문제를 해결하기 위해 UML을 기반으로 아키텍처를 모델링하기 위한 연구 [3, 4]가 있어 왔다.

하지만, 이러한 연구 [2, 3]는 프로덕트 라인 내의 시스템 간의 가변성을 표현하지는 못하므로, 본 논문에서는 기존의 연구 [4]를 확장하여 아키텍처 모델의 가변성을 표현하는 기법을 제안한다. 즉, UML의 확장 메커니즘 (즉, 태그 값과 스테레오타입)을 이용하여 하나의 프로덕트 라인 아키텍처 모델의 가변요소뿐만 아니라 다양한 관점의 프로덕트 라인 아키텍처 모델의 가변성을 효과적으로 적용할 수 있는 방법을 제안한다.

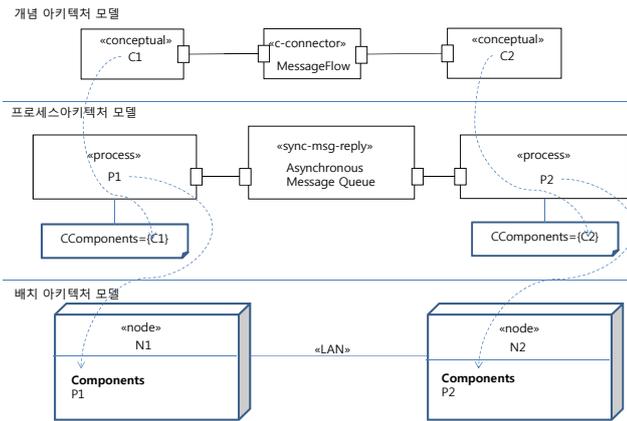
본 논문의 구성은 다음과 같다. 2장에서는 기반연구에 대해 소개한다. 3장에서는 가변성 모델링 기법을 제안하고 4장에서는 적용 사례를 설명한다. 마지막으로 5장에서는 토의 및 결론에 대해 기술한다.

2. 기반 연구

본 장에서는 본 논문의 기반 연구인 [4]에서 정의한 UML 기반의 아키텍처 모델링에 대해 간단히 설명하고, 아키텍처의 가변성 표현에 대한 기존 연구에 대해서 조사

1) 본 논문은 정보통신산업진흥원의 SW공학 요소기술 연구 개발 사업에 의해 지원되었음을 밝힙니다.

한다.



(그림 1) UML 기반 다중 관점의 아키텍처 모델

2.1 UML기반의 아키텍처 모델링

2.1.1 개념 아키텍처 모델링

개념 아키텍처는 시스템 혹은 시스템의 기능 단위를 개념 컴포넌트 집합, 그들 간의 논리적인 상호작용을 개념 커넥터 집합, 그리고 개념 아키텍처와 개념 커넥터의 연결로서 정의된다. 개념 컴포넌트는 UML 2.x의 UMLComponent로 표현되며, <<conceptual>> 혹은 <<functional>> 스테레오타입을 사용하여 다른 관점의 아키텍처 컴포넌트와 구분한다. 개념 커넥터는 UML 2.x에서 제공되는 커넥터를 이용하여 표현될 수도 있지만, 복잡한 상호작용 행위를 지닌 커넥터를 표현하기 위해서 개념 커넥터도 UMLComponent를 이용하여 모델링하고, 개념 컴포넌트와 구분하기 위해 <<c-connector>> 스테레오타입을 사용하여 표현한다. (그림 1 참조)

2.1.2 프로세스 아키텍처 모델링

프로세스 아키텍처는 시스템 혹은 시스템들의 동시성 구조를 나타내는 것으로서, 프로세스 컴포넌트, 프로세스 통신 커넥터, 그리고 이들 간의 연결 관계를 통해 구성한다. 프로세스 아키텍처 또한 UML 컴포넌트 다이어그램으로 표현이 가능하나 개념 아키텍처와 구분하기 위해 스테레오타입으로 프로세스 컴포넌트는 <<process>>로 표시하고, 프로세스 통신 커넥터는 통신 방법에 따라 다양한 서브타입으로 구분하여 표시한다. 가령, 그림 1에서 보는 바와 같이, <<syn-msg-replay>>는 응답이 있는 동기화된 통신 커넥터를 의미한다.

한편 프로세스 컴포넌트에는 하나 이상의 개념 컴포넌트가 대응되므로, 프로세스 컴포넌트의 태그값을 이용하여 할당된 개념 컴포넌트를 모델링한다. (그림 1 참조)

2.1.3 배치 아키텍처 모델링

배치 아키텍처는 소프트웨어가 물리적인 하드웨어 노드에 어떻게 할당될 것인가를 나타내는 것이므로, 노드 컴포넌트, 링크 커넥터, 그리고 이들 간의 연결 관계로 구성된다. 배치 아키텍처는 UML의 배치 다이어그램을 통해 표현이 가능하다.

배치 아키텍처에서는 프로세스 아키텍처의 프로세스 컴포넌트가 배치 아키텍처의 노드 컴포넌트에 할당되므로, 이를 명시적으로 표현하게 된다. (그림 1 참조)

2.2 관련연구

프로덕트 라인 아키텍처의 가변성을 표현하기 위한 효과적인 방법은 가변점 (variation Point)와 가변치(variant)를 이용하는 것이다. 가변점은 아키텍처 요소 중에서 변화 요소가 있는 지점을 의미하고, 가변치는 해당 지점에 결합될 수 있는 가변요소를 의미한다. 기존 연구에서는 프로덕트 라인 아키텍처의 가변성을 모델링하기 위해 가변점과 가변치 관점에 초점을 두었다. 가변성을 표현하기 위해 UML의 확장 메커니즘을 이용하였다.

Clauss [5]는 클래스 다이어그램 레벨에서 UML의 스테레오타입을 이용하여 가변점은 <<variation point>>로, 가변치는 <<variant>>라고 표현해 주었다.

UML 2.0기반의 가변요소 추출 및 표현 방법 [6]에서는 선택기능에 대한 가변성 표현으로 컴포넌트 추상 레벨에서 컴포넌트에는 <<optional component>>를, 커넥터에는 <<optional>> 스테레오타입을 정의하여 표현하였다.

그러나 이 연구들은 하나의 아키텍처 모델의 가변성의 표현에 초점을 두었고, 다중 관점에서의 아키텍처에 대한 가변성을 제안하고 있지 않는다. 따라서 본 논문에서는 다중 관점의 UML 아키텍처 모델을 기반으로 프로덕트 라인의 가변성을 표현하는 기법을 제안한다.

3. 가변성 모델링기법

본 연구에서는 각 관점 (개념, 프로세스, 배치)의 아키텍처가 가지는 가변요소에 대한 모델링 기법을 제안할 뿐만 아니라, 서로 다른 관점의 아키텍처 모델 간의 대응에 있어서, 가변성을 표현하는 기법도 제안한다. 이를 위해서 UML에서 제공하는 확장 메커니즘인 스테레오타입과 태그값을 이용하여 아키텍처의 가변요소를 표현한다.

3.1 각 관점 아키텍처 모델의 가변성 표현

아키텍처 모델은 각 관점마다 아키텍처 모델요소 (예, 컴포넌트, 포트, 커넥터 등) 자체는 다소 상이할 수 있으나, 가변성요소를 표현하는 개념은 동일하다. 즉, 아키텍처 모델의 가변성을 표현하기 위해서는 아키텍처 모델 내의 가변점을 찾고, 각 가변점의 가변성 종류를 명시하게 된다.

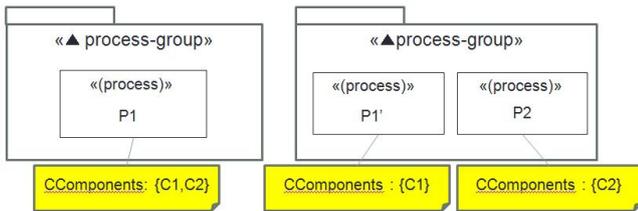
먼저, 아키텍처 모델의 가변점은 크게 블랙박스 가변점과 화이트박스 가변점으로 구분된다. 블랙박스 가변점은 모델요소 자체가 가변점인 경우를, 화이트박스 가변점은

모델요소 내부에 가변요소가 있음을 나타낸다. 둘째, 가변점이 가지는 가변성의 종류는 크게 선택적 혹은 대안적 가변요소로 구분된다. 선택적 가변요소는 시스템에 따라 선택될 수 있는 가변성을 나타내고, 대안적 가변요소들은 그 중에 오로지 하나만 하나의 시스템을 위해 선택될 수 있는 가변성을 나타낸다.

본 논문에서는 아키텍처 모델요소가 가지는 가변점의 종류 및 가변성을 UML의 스테레오타입을 이용하여 표현한다. 즉, 특정 컴포넌트 자체에 선택적 가변성이 있으면 <<●>>으로, 컴포넌트 자체의 대안적 가변성은 <<▲>>으로 표기하고, 컴포넌트 내부의 선택적 가변성은 <<○>>으로, 컴포넌트 내부의 대안적 가변성은 <<△>>으로써 표현한다. 그리고 컴포넌트 내부에 여러 종류의 가변요소가 존재할 경우에는 <<□>>로 표현해 준다.

3.2 다중 관점의 아키텍처 모델간의 가변성 표현

2.1.1에서 언급하였듯이, 개념 컴포넌트는 프로세스 컴포넌트에 대응된다. 이를 표현해주기 위해 UML의 확장 메커니즘을 이용하여 모델링 요소에 태그값(Tagged Value)을 넣어주는 기법을 사용하였다.



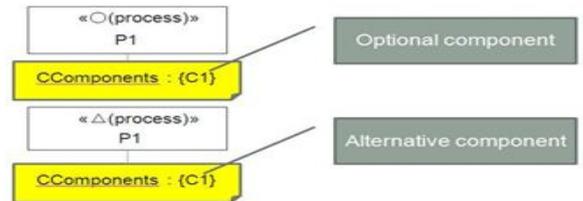
(그림 2) 개념-프로세스 컴포넌트 간의 대응 가변성

이러한 대응관계 모델링함에 있어서 가변성이 존재한다. 가령 예를 들면 개념 컴포넌트 C1, C2가 어떠한 시스템에서는 하나의 프로세스 컴포넌트 P1에 모두 대응되는 경우가 있고, 다른 시스템에서는 각기 다른 프로세스 컴포넌트 P1', P2에 대응되는 경우가 존재할 수 있다.

따라서, 그림 2에서 보는 바와 같이, 개념 컴포넌트 C1, C2가 프로세스모델에 할당될 때의 나타나는 두 가지 경우의 가변성을 표현하기 위해서 <<process-group>>의 스테레오타입으로 구분된 두 개의 패키지를 대안적-블랙박스의 가변성을 갖도록 표현하였다. 즉, 두 개의 패키지 중에서 하나의 패키지만 특정한 시스템을 위해서 선택될 수 있음을 나타내므로, 개념 컴포넌트 C1, C2가 할당된 하나의 P1 프로세스 컴포넌트가 선택되거나, C1, C2가 각각 대응된 P1', P2 프로세스 컴포넌트가 특정한 시스템을 위해서 선택되는 가변성을 표현한다.

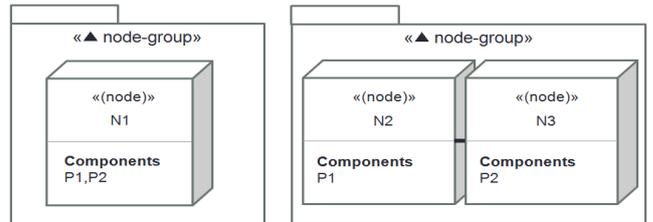
한편, 프로세스 아키텍처 내부에 할당되어 있는 개념 컴포넌트의 가변성 표현을 위해서 내부에 가변성이 존재함을 나타내도록 그림 3와 같이 선택적-화이트박스 가변점(<<○>>) 혹은 대안적-화이트박스(<<△>>)로 표현

한다. 가령, 프로세스 컴포넌트 P1에 할당된 개념컴포넌트 C1이 선택적 가변성을 갖는 컴포넌트라면, 이러한 내부의 가변성이 프로세스 컴포넌트 상에 표시되도록 하기 위해서 P1의 스테레오타입을 <<○>>으로 표시한다. 또한, P1에 할당된 개념컴포넌트 C1이 대안적 가변성을 갖는 컴포넌트라면, P1의 스테레오타입을 <<△>>으로 표시함으로써, 내부에 할당된 개념 컴포넌트가 대안적 가변성을 갖는다는 것을 의미한다.



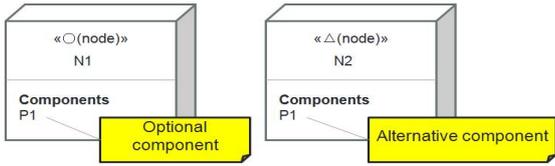
(그림 3) 프로세스 컴포넌트에 할당된 개념 컴포넌트의 가변성 표현

배치 아키텍처도 프로세스 아키텍처와 마찬가지로 컴포넌트 간의 할당 가변성이 존재한다. 가령, 프로세스 컴포넌트 P1, P2가 어떠한 시스템에서는 노드 컴포넌트 N1에 할당되지만, 다른 시스템에서는 각기 다른 노드 컴포넌트 N2, N3에 할당된다고 가정하자. 이러한 가변성을 표현하기 위해서 그림 4에서 보는 바와 같이, UML의 확장 메커니즘을 이용하여 모델링 요소에 <<node-group>>의 스테레오타입으로 구분된 두 개의 패키지를 대안적-블랙박스의 가변성을 갖도록 표현한다.



(그림 4) 배치 아키텍처의 프로세스 컴포넌트의 할당 가변성

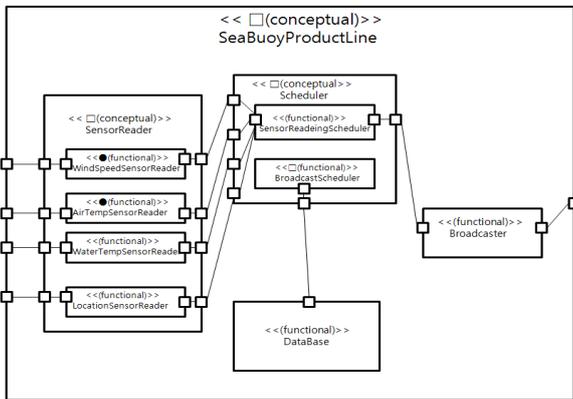
또한, 배치 아키텍처의 노드 컴포넌트 내부에 할당되어 있는 프로세스 컴포넌트의 가변성 표현을 위해서 그림 5과 같이 선택적-화이트박스 가변점(<<○>>) 혹은 대안적-화이트박스(<<△>>)로 표현하여, 노드 컴포넌트에 할당된 프로세스 컴포넌트가 선택적 가변성을 갖는지, 대안적 가변성을 갖는지 표현한다.



(그림 5) 노드-프로세스 컴포넌트의 할당 가변성

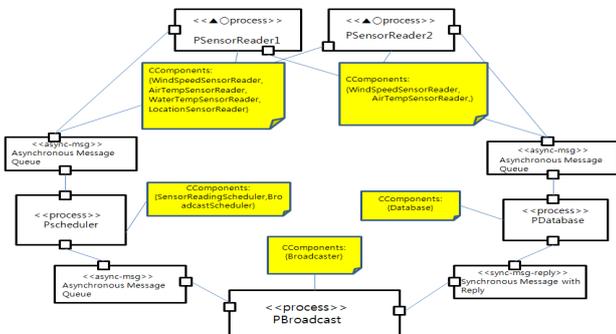
4. 적용 사례

본 논문에서 제안된 기법의 적용가능성을 평가하기 위해 바다 부표 프로덕트 라인(Sea Buoy Product Line)에 사례연구를 수행하였다. 그림 6은 개념 아키텍처의 가변성을 표현한 것으로서, SeaBuoyProductLine이라는 컴포넌트 안에 여러 서브컴포넌트가 위치하고 있고, 그 중 SensorReader 컴포넌트는 자체의 가변성은 없지만 내부에 두개의 선택적 가변성이 존재하므로, <<□>> 스테레오타입으로 표현하였다.



(그림 6) 개념 아키텍처의 가변성 표현

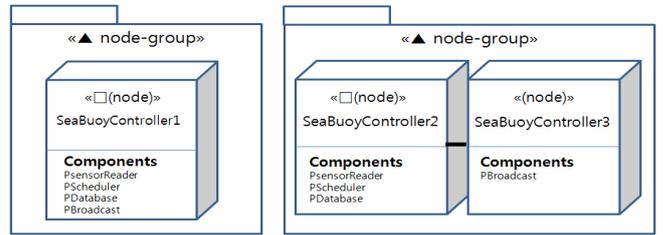
그림 7은 프로세스 아키텍처모델에 대한 예시로서, PSensorReader1컴포넌트와 PSensorReader2컴포넌트는 자체의 대안적 가변성이 존재하고 내부에 할당된 개념컴포넌트의 선택적 가변성이 존재한다. 그러므로 스테레오타입으로 <<▲○>>를 주어 표현해주었다.



(그림 7) 프로세스 아키텍처의 가변성 표현

배치 아키텍처에도 그림 8과 같이 어떠한 시스템에서는 SeaBuoyController1 노드에 네 개의 프로세스 컴포넌

트가 할당되지만 다른 시스템에서는 서로 다른 노드에 할당됨을 각각 group으로 묶어 대안적 가변요소를 표현하였다. 그리고 노드 내부의 가변성이 존재함을 <<□>> 스테레오타입으로 표현하였다.



(그림 8) 배치 아키텍처의 가변성 표현

5. 토의 및 결론

본 논문에서는 개념, 프로세스, 배치 아키텍처의 세 가지 관점으로 정의된 프로덕트 라인 아키텍처에 대해서, 각 관점의 아키텍처가 가지는 가변성에 대해 연구하고, 표준화된 UML의 확장 메커니즘을 이용하여 표현하는 방법을 제안하였다.

현재 UML모델에서의 태그값은 시각적으로 표시가 되지 않는다. 따라서 가변성 정보를 효과적으로 표현해주기 위해서는 UML모델 외에 다른 시각화 기법이 필요한데, 현재, 공개소프트웨어인 StarUML 도구를 확장하여 UML 모델에서 시각화하기 어려운 태그값을 효과적으로 시각화하기 위한 플러그-인을 개발 중에 있다.

향후에는 제안된 기법의 효율성 및 적용가능성 평가를 위해서 다양한 사례연구에 적용하여 이를 입증할 계획이다.

참고문헌

[1] Medvidovic, N., Taylor, R. N., "A classification and comparison framework for software architecture description languages", IEEE Transactions on Software Engineering, Vol. 26, Issue 1, 2001.
 [2] OMG Unified Modeling Language™ (OMG UML), Superstructure, Version 2.3, <http://www.omg.org/spec/UML/2.3/Superstructure>, May 2010
 [3] P. Clements, et. al., "Documenting Software Architectures: Views and Beyond", Addison Wesley, 2002
 [4] 이관우, 이지원, "UML 기반의 프로덕트 라인 아키텍처 모델링", 한국정보과학회, 한국컴퓨터종합학술대회 논문집 (A), p.210~213, 2011,
 [5] Clauss, M., "Modeling variability with UML", GCSE2001 - Young Researchers Workshop, September 2001.
 [6] 최유희, 하수정, 차정은, 박창순, UML2.0 기반의 가변요소 추출 및 표현 방법, 한국정보과학회. 가을 학술발표논문집, Vol.34, No2(B), pp.66~71, 2007.