프로덕트 라인 메타모델 정의와 변화성 모델링

김수연°, 김지영, 김행곤
대구가톨릭대학교 컴퓨터공학과
e-mail: elee--@daum.net, {kimjy,hangkon}@cataegu.ac.kr

A Method of Meta Model Definition and Variability Modeling for Product Line

Su-Youn Kim°, JI-Young Kim, Haeng-Kon Kim
Dept. of Computer Engineering, Catholic University of Daegu

요 약

프로덕트 라인은 다양하고 빠르게 변화하는 사용자 요구사항과 시스템 개발을 위한 특정 도메인 영역에서의 프로덕트 재사용에 관한 연구로 관련된 프로덕트들 사이의 공통성과 변화성에 초점을 두고 체계적인 접근을 제공할 수 있다. 본 논문에서는 체계적인 계획과 변화성 관리를 제공하는 효과적인 프로덕트 라인을 위해 프로세스와 모델링을 중심으로 전개하고, 특히 재사용 가능한 아키텍처 구성을 위해 다양한 관점의 프로덕트 라인 메타모델을 정의하고 망관리 도메인에서 프로덕트 변화성의 효율적인 생성을 지원하는 변화성의 모델링과 이들의 계속적인 변화성을 적용한 설계 패턴들을 제시하고자 한다.

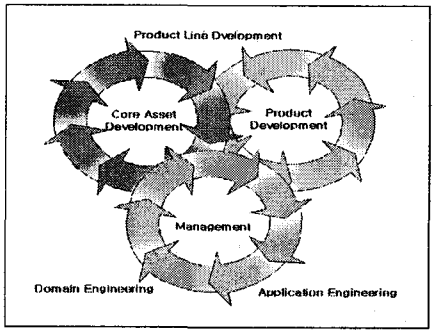
1. 서론

프로덕트 라인은 공통 Core asset으로부터 연관된 시스템 그룹의 아키텍처를 기반으로 공통성과 변화성 부분으로 구성되고 변화성 부분은 수정되거나 변경될 수 있다. 또한 프로덕트 개발의 생산성, 비용 및 품질향상을 위해 재사용 가능한 컴포넌트들을 공유함으로써 최상의 애플리케이션 개발을 위한 파라다임으로 인식되고 있다. 하지만 재사용 아키텍처 및 컴포넌트의 개발과 응용을 위한 혁신적인 개발 프로세스와 도메인 분석과 모델링시 프로덕트 라인의 다양한 관점을 통해 프로덕트 라인과 유도된 프로덕트 및 공통 asset의 정보를 표현할 수 있는 프로덕트 라인 메타 모델의 부재와 프로덕트의 변화성 식별과 모델링에 관한 문제들이 발생할 수 있다. 한편 B2B시장의 성장과 인터넷 금융, 뱅킹 기업의 확대, 컨텐츠 유료화의 바람은 네트워크 환경을 크게 변화시키며, 그 규모 또한 급격히 성장시키고 있다. 이러한 수요의 급격한 증가는 네트워크 기반의 분산 환경이 응용 개발과 활용을 위한 표준 아키텍처로 자리잡아가고 있고, 나아가 통합된 인터네트워킹 상태에서 사용자에게 서비스의 품질을 보장하고, 망의 안정과 효율을 최대화하는 망관리 시스템으로의 개발이 요구되고 있다. 이러한 망관리 시스템의 경우 이기종의 네트워크에서 다루어져야 하는 경우가 많은데, 망의 구성 요소로부터 각 종의 데이터들을 실시간으로 모니터링하고, 분석하여 관리자에게 보고함으로써 망 상태를 효율적으로 관리할 수 있다[1]. 하지만 이들 요구사항을 만족하는 망관리 시스템 개발은 기존 시스템의 복잡성 뿐 아니라 많은 변화성을 포함할 수 있다. 따라서 본 논문에서는 효율적인 프로덕트 라인을 위해 프로세스와 모델링을 중심으로 전개하고, 특히 재사용 가능한 아키텍처 구성을 위해 다양한 관점의 프로덕트 라인 메타모델을 정의하고 망관리 도메인에서 프로덕트 변화성의 효율적인 생성을 지원하는 변화성의 모델링과 이들의 계속적인 변화성을 적용한 설계 패턴들을 제시하고자 한다.

2. 관련 연구

2.1 프로덕트 라인 접근

프로덕트 라인은 기술적, 구조적으로 정의된 관리에서 아래 Core asset 개발과 프로덕트 개발을 포함한다. Core asset 개발은 도메인 공학이라고 하면 프로덕트 개발은 애플리케이션 공학이 된다. Core asset 개발과 프로덕트 개발에서 새로운 프로덕트들은 Core asset으로부터 만들어지고 Core asset들은 기존 프로덕트로부터 획득된다. (그림 1)은 미 카네기 멜론의 SEI에서 정의한 주요 프로덕트 활동 3가지를 나타낸다. 프로덕트 라인 아키텍처는 연관된 프로덕트 집합과 조직에 의해 개발된 시스템을 위한 공통 아키텍처로 향상된 생산성과 소프트웨어 품질을 제공하고, 프로덕트 라인과 관련된 연구는 다음과 같이 분류될 수 있다: 아키텍처 정의와 전개, 컴포넌트 개발, COTS 활용, 기존 Asset Mining과 소프트웨어 시스템 통합. 이들은 실제 Core asset과 프로덕트를 생성하고 전개하는 적절한 기술 적용에 대한 필요성이다[2].



(그림 1) 프로덕트 라인 행위

2.2 프로덕트 라인의 변화성

동일한 프로덕트 군에 속하는 프로덕트들은 많은 공통성을 가지지만, 또한 프로덕트 사이의 변화성도 있다. 변화성은 다양한 사용자, 다양한 설계와 구현 요구사항에 따라 제공되고, 도메인 분석시에 식별된다. 따라서 변화성은 프로덕트 라인 아키텍처 설계 단계에서 고려되어지고, 코드 레벨이 아닌 아키텍처 레벨에서 다루어져야 한다. 또한 변화성은 컴포넌트 조립시에 컴포넌트 행위가 변경될 수 있는 특정 변화점에서 가능하고, 컴포넌트 설계동안에 결정된다. 변화성은 다음과 같은 범주로 분류된다. 휘처 변화성은 특정 휘처의 정의와 구현에서 변화성, 하드웨어 플랫폼 변화성은 제어기, 메모리, 장비 등의 타입에서 변화성, 성능과 속성 변화성은 요구된 성능과 동시성 지원 및 오류관리와 같은 속성에서의 변화성을 의미한다. 이런 변화성은 5단계로 분류해서 설계될 수 있다 : 프로덕트 라인, 프로덕트, 컴포넌트, 서브 컴포넌트, 코드 단계[3].

2.3 망관리 도메인

망 관리 시스템(NMS)은 네트워크의 전반적인 상황을 체크하고, 문제점을 알려주는 기능을 수행하는 하드웨어 또는 소프트웨어를 통칭한다. 인터넷을 비롯한 비즈니스 전반의 데이터 네트워크는 다양한 사용자층, 다양한 목적과 용도로 분산되어지고 있으며, 그 보급의 확대로 인한 많은 비용의 발생과 문제점들이 도출되고 있다. 이에 따라 사용자들에겐 성능대 가격비에 적절하고 유용한 기능을 보장해 주며, 관리자에게는 망의 안정성과 유연성을 최대한 지원할 수 있는 표준화된 망관리가 필요하며 이에 대한 해답이 바로 NMS이다. NMS는 네트워크를 구성하는 각종 요소로부터 구성, 장애, 연결, 성능 등에 관한 데이터를 수집, 분석하고, 네트워크 서비스의 보장과 효율적인 제어를 지원한다[1].

3.프로덕트 라인 개발을 위한 메타모델정의

3.1 프로덕트 라인 개발 프로세스

프로덕트 라인 개발은 특정 도메인 내의 여러 애플리케이션 개발을 위해 큰 규모의 컴포넌트 레벨에서 재사용성과 프로덕트 개발의 생산성, 비용 및 품질

<표 1> 프로덕트 라인 스코핑 요소의 특성

	프로덕트 라인 기술	도메인 스코핑	Asset 스코핑
의미	비즈니스와 재사용 목적을 위한 핵심	도메인 기술서를 사용하여 표준 품질 요소 모델과 도메인을 선택	비즈니스 관점의 재사용 특성 식별
프로세스	1.기존 혹은 새로운 시스템 식별 2.프로덕트 계획 3.기능과 특성을 식별 및 기능 그룹화 4.개략적 도메인 생성 및 분석 5.프로덕트/기능분석	1.도메인 assessment 식별 2.도메인 assessment 수행 및 최적화	1.비즈니스 목적식별 2.평가를 통한 비즈니스 목적 정제 3.특성 평가 4.스코프 결정
프로덕트	프로덕트 기술서 도메인 기술서 도메인구조다이어그램	도메인 식별 기술서 도메인 평가 기술서	초기 프로덕트 맵 (프로덕트, 특성)

향상을 위해 소프트웨어 아키텍처와 컴포넌트 기반 개발의 결합으로 정의될 수 있다. 즉, 시스템 군 사이에서 아키텍처와 재사용 가능한 컴포넌트들을 공유함으로써 최상의 애플리케이션 개발을 위한 파라다임으로 인식되고 있다. 따라서 애플리케이션의 효과적인 개발에 필요한 작업의 순차적 단계인 프로세스의 정의는 기존 개발 프로세스와는 달리 두 가지 형태로 분류되어 접근되어지고, 그 전에 도메인 스코핑과 변화성을 위한 메타모델 구성을 통한 개발을 나타내고 있다(그림2).

3.1.1 프로덕트 라인의 스코핑

재사용은 프로덕트 라인의 핵심으로 그 중 컴포넌트 식별은 프로덕트 라인 스코핑 접근의 핵심이자 프로덕트 라인 개발로부터 경제적 이점을 최대화하는 행위로 간주된다. 스코핑은 프로덕트 라인 개발에서의 재사용 범위를 식별하고 평가하기 위한 것으로 프로덕트가 프로덕트 라인에 포함되고 휘처가 구현되는 프로덕트 레벨과 재사용을 위한 기초를 제공하는 도메인 레벨, 재사용 가능한 방법으로 지원되는 Asset 레벨에 따라 스코핑이 정의될 수 있다<표1>[4].

3.1.2 Core Asset 개발

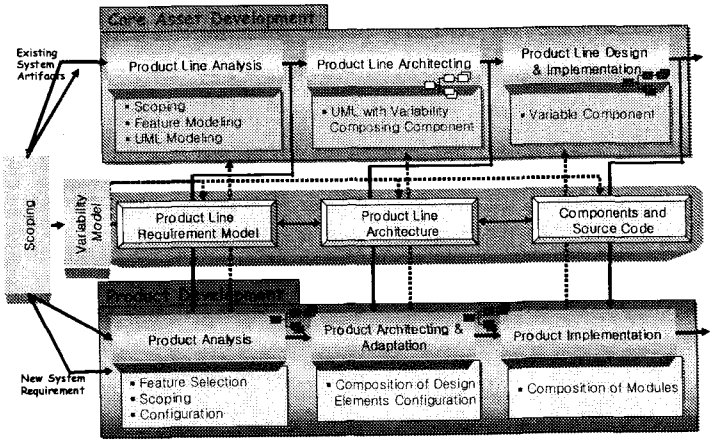
(1) 프로덕트 라인 분석 : 도메인 모델을 생성하기 위해 기존 시스템의 설계와 아키텍처, 요구사항을 분석하는 단계로 스코핑을 정의하고 휘처와 UML 모델링 작업 및 변화성 요구사항 명세 작성을 수행

(2) 프로덕트 라인 아키텍칭 : 앞 단계의 분석 모델을 통해 프로덕트 라인에서 지원되어야 하는 공통성과 변화성을 식별하고, 프로덕트 라인 아키텍처를 개발하는 단계로 변화성을 표현하기 위한 UML 확장과 하이퍼모듈 단위의 구성을 수행

(3) 프로덕트 라인 설계 및 구현 : 재사용 가능한 Asset (component, documentation, code generator 등)이 프로덕트 라인 아키텍처를 기반으로 개발되는 단계로 공통 아키텍처를 기반으로 자동화된 실체화를 통해 재사용 가능한 프레임워크를 생성

3.1.3 프로덕트 개발

(1) 프로덕트 분석 : 도메인(프로덕트 라인) 요구사항과 새로운 요구 사항을 비교하여 도메인 요구사항에 새로운 요구사항을 매핑



(그림 2) 프로덕트 라인 개발 프로세스

(2) 프로덕트 아키텍칭 : 도메인 아키텍처와 특정 시스템 요구사항에 기반한 시스템 아키텍처 개발
(3) 프로덕트 구현 : 새로운 시스템을 구축하기 위해 시스템 아키텍처를 사용하여 공통 도메인 요구사항을 위한 재사용 가능한 Asset과 특정 요구사항을 위해 개발된 컴포넌트의 조립을 통해 시스템 구축

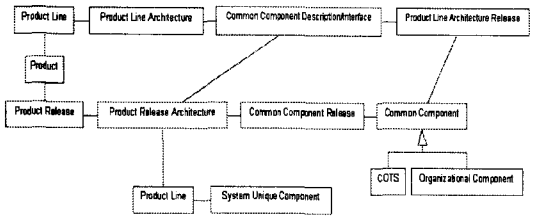
3.2 프로덕트 라인 개발 메타 모델

(그림 3)은 프로덕트 라인과 유도된 프로덕트, 공통 asset에 대한 다양한 관점을 중심으로 구성요소와 그들간의 관련성을 정의한다. 프로덕트 라인 아키텍처를 위한 메타 모델들은 다양한 아키텍처 생성 프로세스의 변화를 가져올 수 있다. <표 2>는 메타 모델과 그들을 구성하는 요소들의 개략적 명세를 나타낸다[5].

4. 프로덕트 라인 변화성 모델링

4.1 휘처 모델

휘처 모델은 프로덕트 라인 요구사항 분석의 핵심



(그림 3) 프로덕트 라인 메타 모델

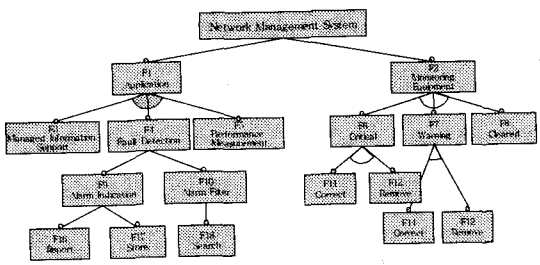
<표 2> 메타모델의 개략적 명세 요소

참조 요소	메타모델 이름
속성	메타모델의 정보를 제공하는 속성
행위	개발자들의 업무와 프로덕트 라인이 어떻게 상호작용 하는지에 대해 기술하는 이벤트 시나리오 기술
서브 요소	서브 메타 모델의 이름

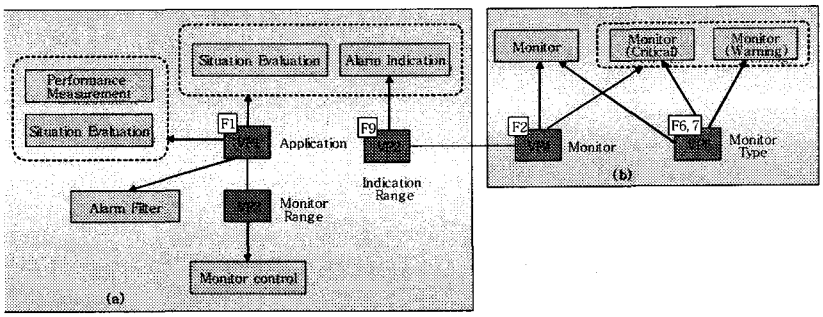
결과로 프로덕트 라인 구성의 기능적, 비기능적인 요소 뿐 아니라 공통성과 변화성을 획득하고 다양한 관점을 제공할 수 있다. 사용자는 휘처 모델을 통해 프로덕트 라인의 기능을 이해할 수 있고, 개발자들은 다양한 프로덕트 변화의 개발을 유도할 수 있다. (그림 4)는 망관리 시스템 프로덕트 라인을 위한 휘처 모델의 간략한 예를 보여준다. 망관리 시스템은 지속적인 장애 발생 여부를 감시함으로써 장애가 발생했을 경우 장애 위치와 원인, 처방 방법 등의 정보를 제공함으로서 장애를 즉시 복구 할 수 있는 기능을 제공한다. 휘처 모델은 망관리 시스템 프로덕트 라인의 기능을 트리 형태로 구조화하고 응용은 망관리 시스템의 기능을 포함하고, 감시 장비는 감시 환경을 실현하기 위해 요구된 하드웨어를 보여준다.

4.2 프로덕트 라인 아키텍처 모델

프로덕트 라인 아키텍처는 변화성의 설계 요소를 포함해야한다. 구조적인 변화성은 아키텍처 설계동안 선택적인 설계 옵션으로 종종 아키텍처 변화점의 집합으로 표현한다. 하지만 휘처 모델은 특정 설계를 함축하지 않고, 아키텍처를 구조화하기 위한 가이드라인을 제공한다. 또한 최종 아키텍처는 성능과 품질과 설계 제한 조건을 만족하는 기능적인, 비기능적인 요구사항을 고려해야 한다. (그림 5)는 망관리 프로덕트 라인 아키텍처의 논리적, 물리적 관점에서의 변화성을 보여주고 앞의 (그림 4)에서 요구



(그림 4) 망관리 프로덕트 라인을 위한 휘처 모델



(그림 5) 아키텍처에서의 변화성

사항 사이의 변화성을 만족하기 위해 변화점을 제시하고, 어떻게 언제 변화점를 적용할 것인지를 명세한다. 논리적 관점은 3개의 변화점를 포함하고, VP1은 논리적 컴포넌트와 Alarm Filter, 두 개의 컴포넌트 그룹에 영향을 끼치고, VP2, 3은 단지 개별적인 논리적 컴포넌트와 상호작용 하고, VP1과 VP3, VP2와 VP4 사이의 의존성이 존재한다. VP4와 VP5는 하드웨어 플랫폼에 대응하는 요소들과 매핑된다. 그 외 변화성은 프로세스와 전개 관점을 포함하는 다양한 아키텍처 관점에 영향을 줄 수 있다.

4.3 설계패턴에서의 변화성 모델링

변화성은 패턴을 사용해서 표현 가능하고 문제를 위한 재사용 가능한 해결책을 제공하고 구현 해결책에 대한 재사용을 지원한다. 공통성과 변화성을 지원하는 패밀리 모델을 구축함에 있어 식별과 관련된 패턴이 적용될 수 있다. 식별은 시스템을 구별하는 휘처이다. <표 3>은 3가지 타입의 식별과 패턴명, 패턴 설명 그리고 패턴 구조도를 나타낸다.

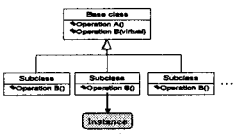
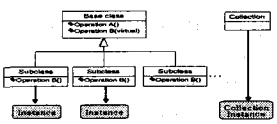
5. 결론

최근 아키텍처와 재사용 컴포넌트들의 공유를 통한 애플리케이션 개발이 최상의 파라다임으로 인식되고 있다. 컴포넌트를 기반한 재사용성과 프로덕트 개발의 생산성, 비용 및 품질 향상을 위해 본 논문에서는 효율적인 프로덕트 라인 지원 프로세스와 모델링에 대해 연구 하였다. 재사용 가능한 아키텍처 구성을 위해 다양한 관점의 프로덕트 라인 메타모델을 정의하고 망관리 도메인에서 프로덕트 변화성의 효율적인 생성을 지원하는 변화성에 대한 모델링을 하고 이들의 변화성을 적용한 설계 패턴들을 식별하고 구조화 하였다. 향후연구로는 좀더 명확한 변화성에 관한 연구를 통해 프로덕트 라인 변화성의 사용 사례와 이들을 지원하는 프레임워크 개발에 관한 연구가 필요하다.

참고문헌

[1] 김행곤외,"컴포넌트 기반 망관리 시스템 개발", KCSE'03, 5권1호, 2003.
[2] Technical Report, A Framework for Software Product Line Practice Version 3.0, Carnegie Mellon SEI, 2000.
[3] Maarit Harsu, A Survey of Product-Line Architecture, White Paper, Tampere University of Technology, 2001
[4] Klaus Schmid, "A Comprehensive Product Line Scoping Approach and Its Validation", ICSE'02, 2002
[5]Khaled Jaber, "Product Line Viewpoint and Validation Models", 2000 ACM Symposium on Applied Computing, 2000.

<표 3> 변화성 모델링을 위한 설계패턴

식별	패턴 이름	패턴 설명	패턴 구조도
Single	Single Adapter Pattern	· 일반적인 feature들이 base class에 기술되고, 특정 feature들은 subclass에 대응되는 상속 계층으로 표현 · 하나의 subclass가 하나의 시스템으로 인스턴스 · Subclass의 집합은 새로운 시스템을 위해 확장 가능 · single 인스턴스는 singleton pattern을 사용해서 구현	
Multiple	Multiple Adapter Pattern	· 하나 이상의 subclass들이 하나의 시스템으로 인스턴스 · 하나의 subclass들이 하나의 인스턴스를 가지고 인스턴스 집합은 collection으로 전개 · collection은 이름에 의한 다양한 subclass 인스턴스를 식별하고 수집하기 위해 필요	
Option	Option Pattern	· class당 두 개의 연관을 생성함으로써 표현 · 연관된 클래스는 0-1의 관련성을 가짐 · classB는 classA를 위한 옵션 · A는 B가 재사용 되는지에 상관없이 재사용 가능