

## PL 아키텍처를 위한 도구 설계

송문섭<sup>o</sup> 황길승  
한국전자통신연구원  
{sirius<sup>o</sup>, kshwang}@etri.re.kr

### Design Tool for Product Line Architecture

Moon-Sub Song<sup>o</sup> Kil-Seung Hwang  
Electronics and Telecommunications Research Institute

#### 요 약

제품계열(Product-Line) 개발 방법은 특정 영역에 대해 핵심 자산을 구축한 후 제품 특성에 맞게 자산을 변경하여 신속하게 제품을 생산하는 방법이다. 제품계열의 이용한 제품 생산이 생산성과 효율성을 높이기 위해서는 자산 구축이 제품 패밀리로부터 정확히 추출, 생성되어져 있어야 한다. 특히 핵심 자산 중에서 가장 중요한 자산인 제품계열 아키텍처가 중요하다. 본 논문에서는 도메인 전문가가 제품 영역에 대한 분석을 끝낸 후 제품계열 아키텍처를 생성, 편집을 용이하도록 도와 줄 수 있는 기능과 제품계열 아키텍처에서 제품 아키텍처로 쉽게 유도할 수 있도록 도와주는 기능을 고려하여 설계하고자 한다. 또한 설계된 내용을 기반으로 아키텍처 다이어그램 편집기 프로토타입을 구현하였다.

#### 1. 서 론

소프트웨어공학에서 자산의 재사용을 통한 생산성 향상은 끊임없는 주제였다. 재사용될 자산의 종류나 형태에 따라 문서의 재사용, 코드의 재사용, 또는 모듈의 재사용, 객체나 컴포넌트의 재사용, 프레임워크의 재사용 등으로 구분해 볼 수 있다.[1] 특히 최근에는 유럽과 미국에서 제품계열 개발 방법을 적용한 제품 생산의 사례가 증가하고 있으며, 임베디드 시스템에서 그 효과가 높은 것으로 나타나고 있다.[2] 그러나 현재 제품계열 개발 방법론을 지원해 주는 제품계열 지원 도구는 활발히 개발되어지고 있지 않다. 자산의 형태가 컴포넌트일 경우에는 컴포넌트 개발 도구의 기능을 어느 정도 활용할 수 있었지만, 제품계열 개발과 컴포넌트 기반 개발 방법의 차이로 인해 부족한 기능들이 있다. 특히 제품계열 개발 방법에서 핵심적인 부분인 제품계열 아키텍처의 생성과 관리는 기존 도구에서는 지원되어지고 있지 않다.[3][4] 본 논문에서는 도구 사용자 중에서 도메인 전문가가 해당 도메인에 맞는 제품계열 아키텍처를 쉽게 생성할 수 있도록 지원해 주고 제품 개발자가 제품계열 아키텍처로부터 제품을 빠르게 생산할 수 있도록 지원해 주는 도구를 설계하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 제품계열 개발 방법과 소프트웨어 아키텍처에 관해 살펴보고 3장에서는 제품계열 아키텍처 생성 및 관리 지원 도구를 설계하겠다. 4장에서는 설계한 지원도구의 프로토타입을 이용하여 활용 예를 보이고 5장에서는 결론 및 향후 연구과제 설명하겠다.

#### 2. 관련 연구

##### 2.1 제품계열 개발 방법론

S/W 생산 기술로 각광받고 있는 제품계열 개발 방법론은 기존의 다른 방법론과 몇 가지 다른 특징이 있다. 가장 큰 특징은 아키텍처 기반 개발 방법이라는 것이다.[3][4] 특정 영역에 대한 도메인 분석 결과를 바탕으로 공통성과 가변성을 적용한 공통 아키텍처를 생성한 후 이를 변경하여 유사 제품들을 개발하는 방식으로 임베디드 시스템의 S/W 분야에 많이 활용되고 있다.[5] 또 다른 특징으로는 단순한 S/W 개발에 관한 방법뿐만 아니라 제품계열과 회사의 규모에 따라 조직의 구성과 제품 개발 과정에 대한 방법까지 제시해 준다는 것이다.[6]

현재 제품계열 공학은 미국의 CMU SEI와 독일의 IESE 연구소를 중심으로 한 유럽에서 많이 연구되어지고 있으며, 상업적으로 제품계열 방법을 잘 사용하고 있는 회사와 제품으로는 핀란드의 노키아(Nokia)사의 모바일 브라우저와 미국의 보잉사(Boeing)사의 항공 제어 모듈이 대표적이다.

##### 2.2 소프트웨어 아키텍처

소프트웨어 아키텍처는 표현하고자 하는 시스템 또는 프로그램에 따라 다양한 뷰(View)로 표현되어져 왔으면 표현 언어(Architecture Description Language)도 시스템의 특성에 따라 달라져왔다. S/W 개발에서 소프트웨어 아키텍처의 역할은 제품 개발 단계에서 개발 관계자들 사이의 의사소통을 원활하게 해 주며 프로젝트 초기 단계에서 결정해야 할 사항들을 명확하게 해 줄 수 있다. 또한 소프트웨어 아키텍처 패턴 등을 이용하여 유사 시스템 개발에 재사용할 수 있다.[6][7]

하나의 아키텍처 뷰로 시스템 전부를 표현할 수 없으므로 표현하고자 하는 내용에 따라 다양한 뷰로 표현되며 일반적으로 4+1 뷰가 많이 사용되어지고 있다. 아키텍

텍처 표현 언어로는 비표준 언어인 개별 ADL과 OMG(Object Management Group)의 UML 2.0의 확장 메커니즘인 프로파일이다.

3. PL 아키텍처 지원 도구 설계

제품계열 개발 방법에서 핵심 자산인 아키텍처는 두 가지 형태로 사용된다. 첫 번째는 핵심 자산 구축 시 제품계열에 포함된 제품들의 특성들을 모두 포함하는 공통 아키텍처이고 두 번째는 개별 제품에 맞는 제품 아키텍처이다. 제품 아키텍처는 제품계열 아키텍처의 가변성을 결정하는 유도 과정을 거쳐서 생성된다. 따라서 도구에서도 이러한 흐름을 지원해 주어야 한다.

3.1 PL 아키텍처 지원 도구 구조

PL 아키텍처 지원 도구는 제품계열의 핵심 자산인 공통 아키텍처를 생성, 편집할 수 있는 인터페이스를 아키텍트에게 제공해주며, 공통 아키텍처로부터 개별 제품 아키텍처를 생성할 수 있도록 지원해 준다. 지원 도구의 구조는 그림 1과 같이 공통 아키텍처를 생성하는 모듈, 제품 아키텍처를 생성하는 모듈, 두 모듈의 수행을 위해 필요한 정보를 관리하는 DB, 사용자와 인터랙션을 담당하는 시각화 모듈로 구성되어 있다.

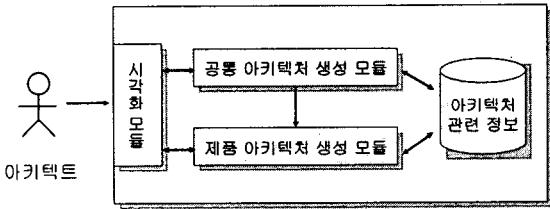


그림 1 PL 아키텍처 지원 도구 구조도

3.2 PL 아키텍처 지원 도구 기능

PL 아키텍처 지원 도구의 기능으로는 아키텍트가 도메인 지식을 이용하여 제품계열의 핵심 자산인 공통 아키텍처를 쉽게 생성할 수 있도록 지원해주는 기능, 공통 아키텍처로부터 개별 제품 아키텍처를 유도하는 기능, 사용자와 지원도구 간의 정보 교환과 아키텍처를 시각적으로 보여주는 시각화 기능 등이 있다. 각각의 모듈 별 구조 및 기능은 다음과 같다.

3.2.1 공통 아키텍처 생성 모듈

정의된 아키텍처 구성 요소들을 이용하여 공통성과 가변성을 포함한 제품계열 공통 아키텍처를 생성하고 관리하는 모듈로 그림 2와 같은 구조이다. 구성요소 관리의 아키텍처를 구성하고 있는 요소들을 정의하거나 요소들의 속성 값의 변경 등을 처리한다. 가변성 관리는 제품계열에서 가장 중요한 기능으로 공통 아키텍처에 포함된 가변성에 관련된 정보를 관리하며, 제품 아키텍처를 유도하기 위한 필수 기능으로 사용된다. 공통 아키텍처 제어는 공통 아키텍처 생성을 위한 모듈 내부 관리와 외부 모듈과의 인터페이스 역할을 담당한다.

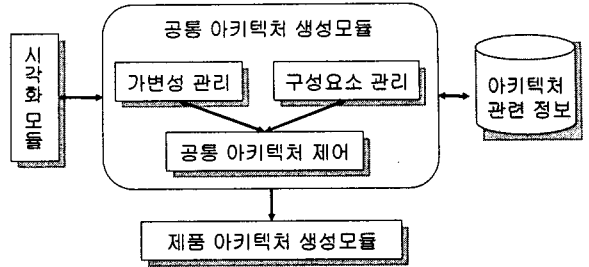


그림 2 공통 아키텍처 생성 모듈 구조도

3.2.2 제품 아키텍처 생성 모듈

제품 아키텍처는 제품 별로 생성하는 것이 아니라 제품계열 공통 아키텍처로부터 제품 특성에 맞게 유도된다. 제품 아키텍처 생성 모듈의 구조는 그림 3과 같다. 가변 결정 모듈은 공통 아키텍처에 표현된 제품계열의 가변성을 제품 특성에 맞게 결정하는 모듈로 결정에 따라 정의된 처리를 수행하게 된다. 지원 도구에서는 가변성 관리 단계에서 결정에 따른 영향을 미리 정의하도록 지원해줘야 하며 가변 결정 단계에서는 결정에 따른 영향을 자동으로 수행해줘야 한다. 추가 요소 생성은 개별 제품에 특화되어서 필요한 요소들을 추가로 제품 아키텍처에 반영할 수 있도록 지원해 주는 기능이며 제품 아키텍처 제어는 제품 아키텍처를 생성하기 위한 관리 및 외부 모듈과의 인터페이스 역할을 담당한다.

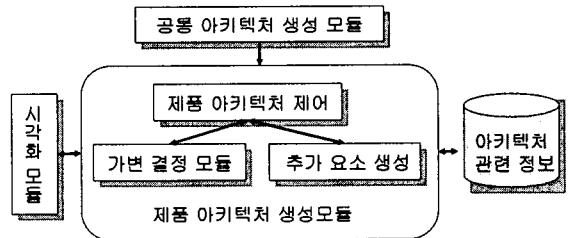


그림 3 공통 아키텍처 생성 모듈 구조도

3.2.3 시각화 모듈

시각화 모듈은 공통 아키텍처 생성 모듈과 제품 아키텍처 생성 모듈에서 사용자가 입력해야 할 정보를 쉽게 입력할 수 있도록 도와주며, 산출되는 모델 정보와 데이터를 사용자에게 시각화해서 제공해 준다. 그림 4는 시각화 모듈에서 제공해 주는 세부 기능들을 나타낸다. 가변 테이블은 공통 아키텍처의 가변성을 관리하기 위한 인터페이스를 제공해 주며, 결정 트리는 가변 테이블에 정의된 가변성을 제품 아키텍처 생성 시에 결정하는데 도움을 주는 기능이다. 팔레트 창은 공통 아키텍처 생성 및 추가되는 제품 아키텍처 구성 요소를 입력하기 위한 아키텍처 구성요소 모음이다. 미니 맵은 생성된 아키텍처가 한 화면을 넘을 경우 전체적으로 보거나 화면 이동을 쉽게 해 주는 기능이다. 속성 창은 아키텍

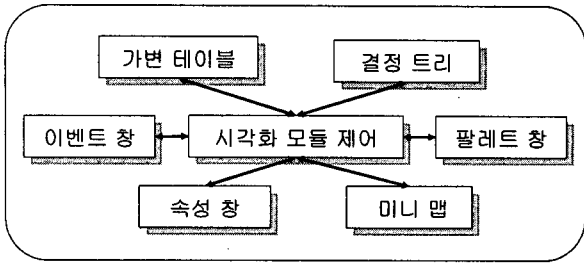


그림 4 시각화 모듈 구조도

텍처 구성요소 별로 갖고 있는 속성을 설정하거나 표시해 주는 기능이며, 이벤트 창은 아키텍처 지원도구에서 수행되는 결과들을 사용자에게 제공해 주는 창이다.

4. 프로토타입

설계한 PL 아키텍처 지원 도구의 프로토타입은 그림 5와 같다. Eclipse 3.0의 플러그인 패키지로 개발하였으며, 팔레트와 미니맵 등은 Eclipse에서 제공해 주는 기능을 이용하였다.

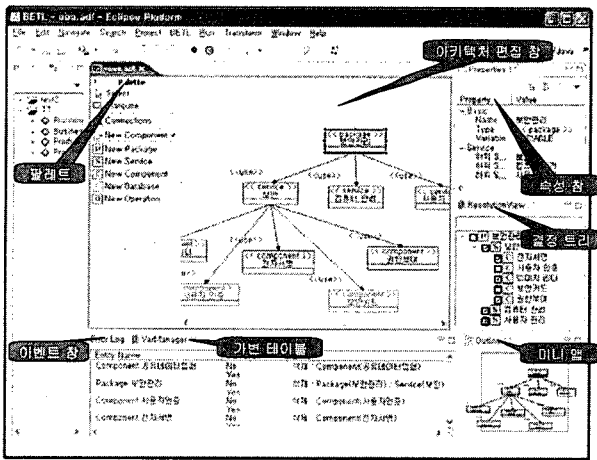


그림 5 PL 아키텍처 지원 도구 프로토타입

4.2 아키텍처 구성 요소

PL 아키텍처의 구성 요소들로는 엔티티, 릴레이션, 포트, 커넥터 등이 있을 수 있다. 이러한 구성 요소들은 지원도구의 팔레트 기능을 통해 쉽게 사용되어질 수 있다. 스테레오 타입을 이용하여 좀 더 구체적으로 구성요소들을 표현할 수 있도록 지원해 준다.

5. 결론 및 향후 연구과제

본 논문에서는 제품계열 개발 방법론에서 가장 중요한 핵심 자산인 제품계열 아키텍처의 생성, 편집 등을 지원해 주는 지원 도구를 설계하고 설계한 도구의 프로토타입을 이용하여 지원 도구의 갖추어야 할 기능에 대해 살펴보았다. 설계한 도구에서는 아키텍처가 제품계열 공통 아키텍처를 쉽게 생성할 수 있도록 지원해 주며, 생성된 공통 아키텍처를 이용하여 유사한 제품들의 아키텍처를 빠르게 생성할 수 있도록 가변성 관리 및 영향 관리 등의 기능을 도구에서 지원해 주었다.

향후 연구과제로는 PL 아키텍처의 구성 요소를 UML 2.0 프로파일을 이용하여 정의한 후 도구에서 사용할 수 있도록 지원해 주는 것과 생성된 제품 아키텍처를 기반으로 소스 코드를 생성할 수 있도록 지원해 주는 기능들이 추가되어야 한다. 또한 가변성 관리를 위한 규칙 정의 언어에 대한 연구도 진행되어야 한다.

참고문헌

- [1] Krueger, C.W., "Software product line reuse in practice", *Application-Specific Systems and Software Engineering Technology, Proceedings, 3rd IEEE Symposium*, pp.117-118, 2000
- [2] P. Clements and L. Northrop, "A Framework for Software Product Line Practice, SEI Report, 2002.
- [3] Boehm, B., Wolf, S., "An open architecture for software process asset reuse", *Software Process Workshop, Process Support of Software Product Lines, Proceedings of the 10th International*, pp.2-4, 1996.
- [4] Popp, T. J. "Software architecture development for product line software", *Digital Avionics System conference, Proceedings. 18th Volume2*, 24-29. 1999.
- [5] David M. Weiss, Chi Tau Robert Lai, *Software Product-Line Engineering : A Family-Based Software Development Process*, Addison-Wesley, 1999.
- [6] J. Bosch, *Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach*, Addison-Wesley, Boston, 2000.
- [7] Kang, k.c., Jaejoon Lee, Donohoe, p, "Feature-oriented product line engineering", *Software, IEEE* vol 19, pp.58-65, 2002.

4.1 PL 아키텍처 메타 모형

PL 아키텍처의 메타 모형은 그림 6과 같다. 아키텍처는 하위에 여러 서비스들로 구성되며 서비스는 공통성과 가변성의 성질을 갖고 있다. 또한 하나의 서비스는 여러 개의 개념 컴포넌트들로 구성되며 개념 컴포넌트들도 공통성과 가변성을 갖고 있다. 개념 컴포넌트는 추가로 객체들로 세분화하여 표현될 수도 있다.

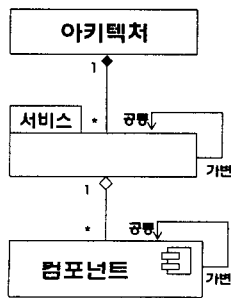


그림 6 PL 아키텍처 메타 모형