

임베디드 시스템 개발의 제품 계열적 적용

이지현* 조진희 합동한 김진삼
한국전자통신연구원 임베디드 S/W 기술센터 S/W 공학연구팀

e-mail : {jihyun, chojh, dhham, jinsam}@etri.re.kr

Methodology for Embedded System Development based on Product Line

Jihyun Lee* Jin-Hee Cho Dong-Han Ham Jin-Sam Kim
S/W Engineering Research Team, Embedded S/W Technology Center, ETRI

요 약

임베디드 시스템의 개발이 시장 적시성과 개발 생산성을 만족시키도록 개발 방법을 향상시키기 위해 제품 계열(Product Line) 기반의 개발 방법을 제시한다. 기존에 생성된 소프트웨어 컴포넌트를 재사용할 수 있는 컴포넌트 기반 시스템 개발 방법론을 제시하고 종합적이고 체계적인 형태의 재사용 프로세스를 구성하며 각 단계별로 필요한 작업과 제품 생산 원칙을 살펴보도록 한다.

1. 서론

임베디드 시스템은 휴대폰, 디지털 TV, 게임기와 같은 다양한 분야에서 특정 기능을 처리하기 위해 시스템의 일부로 내장되어 시스템의 내/외부에 대한 센싱, 모니터링, 제어 기능을 수행하는 마이크로프로세서 기반의 디지털 시스템이다. 임베디드 시스템은 일반적인 범용 시스템과 달리 부과된 특정 목적의 기능만을 수행한다. 따라서 내장된 소프트웨어를 갖고 특정 작업을 수행하는 모든 시스템을 임베디드 시스템이라 할 수 있다.

임베디드 시스템은 실시간적 기능 처리, 고신뢰 서비스, 파워 제약성, 하드웨어를 통한 소프트웨어의 연계와 같은 특성을 가지며 본질적으로 최종 제품의 부가가치를 창출하는데 핵심적인 기능을 한다. 예를 들어 사람 한 명이 하루 동안 접하는 임베디드 시스템은 보통 20 개가 넘는다는 보고가 있으며 이러한 시스템의 사용은 통신, 가전, 의료, 금융을 포함한 전 산업으로 확대될 것으로 예상되고 있다. 향후 10 년 안에 임베디드 소프트웨어를 내장한 임베디드 시스템은 급격히 증가될 것으로 보고되고 있다. 대기업의 모바일 단말기의 출시에서 볼 수 있듯 임베디드 시스템의 생산성, 시장 적시성, 품질은 매우 중요하게 부각되고 있는 실정이다[1].

시장 적시성을 만족하는 소프트웨어와 시스템을 개발하기 위해서는 시스템을 개발하는 개발 프로세스가 매우 중요한 요소가 된다. 이러한 프로세스에는 개발에 필요한 단계 별 업무, 역할, 결과물, 기법뿐만 아니라 단위 개발 업무, 결과물, 인력, 스케줄의 진행 및 조절과 같은 관리가 함께 포함된다. 이와 같이 개발과 관리에 대한 방법 체계와 해당 조직에서의 해당 방법 체계의 숙달은 개발하고 있는 제품의 품질 및 생산성과 직접적으로 관련 맺게 된다.

본 논문에서는 임베디드 시스템 개발에 필요한 요구 사항을 기반으로 소프트웨어 공학의 제품 계열 기반 접근법을 적용하여 생성되는 기대 효과 및 응용하고자 하는 제품 공학의 요소 기술 및 관점에 대해 논의 하고자 한다. 본 논문은 제 2 장에서 임베디드 시스템 개발을 위한 요구 사항에 대해 제시하고, 제 3 장에서 제품 계열 기반 기술에 대해 살펴본다. 제 4 장에서 임베디드 시스템 개발에 필요한 프로세스를 정의하고, 5 장에서 제품 계열 기반 기술 적용에 따른 효과를 예측해 본다. 마지막으로 제 6 장에서 향후 연구를 제시하고 결론을 맺는다.

2. 임베디드 시스템 개발 요구 사항

임베디드 시스템을 범용 시스템에 포함되는 대표적

인 예인 데스크 탑 시스템이나 서버 시스템과 임베디드 시스템을 비교할 때 임베디드 시스템이 갖는 독특한 특성들은 다음과 같다.

첫째, 임베디드 시스템은 제한된 하드웨어와 저전력을 사용하면서도 실시간 처리를 지원한다. 둘째, 하드웨어의 개발이 먼저 이뤄진 후에 소프트웨어를 개발하는 절차가 보편적으로 이뤄지고 있다. 이에 반해 제품의 시장 출시 사이클을 빠르게 돌아 가고 있다. 이런 특성은 임베디드 시스템을 개발할 때 보다 용이한 개발과 효과적인 유지 보수, 타당한 개발 비용을 위해 임베디드 개발 요구 사항에 맞는 개발 방법을 필요로 한다.

임베디드 시스템의 개발을 위한 개발 요구 사항은 임베디드 시스템을 개발하는 업체들의 공통된 의견과 논문을 통해 발표된 학계의 기술 보고서에 기반하여 표 1 과 같이 요약된다.

표 1. 임베디드 시스템 개발 요구 사항

항목	요구 사항 기술
1	제한된 하드웨어 자원 사용
2	저전력 사용
3	대부분 실시간적 기능 지원
4	하드웨어를 통한 소프트웨어의 연계
5	초기 아키텍처의 중요
6	하드웨어 중심적 설계 선호
7	결함 포용
8	사용자의 요구에 맞춘 서비스 제공
9	고가의 설치(installation) 비용
10	고가의 갱신(update) 비용

임베디드라는 특정 속성에 맞는 임베디드 제품을 개발하기 위해 전체 개발 과정은 개발 첫 단계에서부터 개발 요구 사항을 만족해야 한다. 참고로 현재 임베디드 시스템에서의 기능 지원은 하드웨어에서 점차 소프트웨어로 비중이 높아지고 있는 상태이다. 일례로 전투기에 내장되는 소프트웨어가 1960년에는 8%에 그쳤으나 2000년에 와서는 80%를 차지한다[2]. 이는 임베디드 시스템 전반의 기능, 성능, 개발 기간을 결정하는 핵심이 소프트웨어에 종속적이고 소프트웨어를 수행하는 하드웨어 플랫폼은 빠르게 개발되고 있기 때문이다.

그 동안의 임베디드 시스템은 주먹구구식의 경험적 방법에 의해 이루어져 왔고, 지금도 많은 곳에서 이러한 형태로 개발되고 있다. 그리고 보편적인 시스템에 적용되던 전통적인 개발 방법은 임베디드 시스템의 개발에 적합하지 않는 부분이 많다. 그러므로 재사용 자산 기반의 체계화된 개발 방법론이 마련된다면 임베디드 시스템을 개발할 때의 제품 개발 비용을 낮추고 개발 효율을 높여 각 산업별 시장 적시성과 품질을 만족하는 제품으로 생산할 수 있다.

임베디드 시스템의 개발에 적용하기 위한 제품 계열 기술을 다음 장에서 살펴본다.

3. 제품 계열

일반적으로 시스템 개발은 하나의 제품을 개발하는데 국한되었다. 90년대 소개된 컴포넌트 기반 개발 방법은 시스템을 구성하는 기본 단위의 컴포넌트의 재사용을 언급하고 있다. 하지만 재사용되는 개념이 컴포넌트에 기반해 제한되어 있었고, 제품 계열로 넘어오면서 재사용의 개념은 보다 확장되었다. 제품 계열 개발 방법은 제품군을 개발할 때 범용적으로 사용할 수 있는 핵심 자산(예: 컴포넌트, 아키텍처)을 생산한 후 각 제품의 개발 요구 사항에 맞춰 가변적인 부분을 제품 구성 요소로 개발하고 재사용하는 핵심 자산과 함께 결합시키는 형태로 이뤄진다.

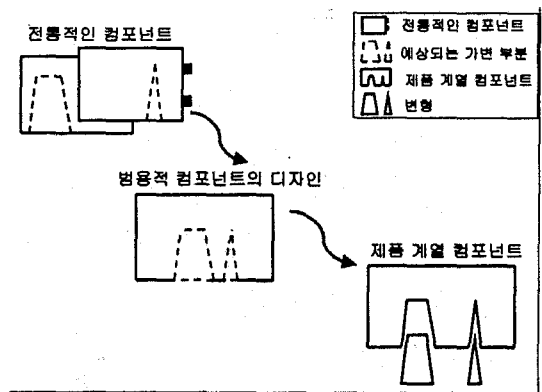


그림 1. 제품 계열 컴포넌트의 개념도

그림 1은 제품 계열 개발 자산 중 컴포넌트가 전통적인 컴포넌트와 어떻게 다른 지를 표현하는 개념도이다. 전통적인 컴포넌트는 개발된 후 같거나 비슷한 기능의 컴포넌트가 필요할 때 그대로 또는 개조를 통해 기능을 변경한 후 재사용되었다. 이에 반해 제품 계열 컴포넌트는 제품군(product family)이 갖는 공통성을 각 제품을 생산할 때 재사용할 수 있도록 최대한 범용적으로 만들고, 제품 별로 예상되는 가변적인 기능을 포함할 수 있도록 가변점(variation point)을 초기 설계 단계에서부터 고려한다. 컴포넌트에서의 가변성(variability)은 제품 계열 컴포넌트의 가변점이 특정 가변 특성을 구현한 “변형”을 받아들일 수 있는 약속이 되어 컴포넌트가 범용적으로 사용될 수 있게 한다. 그림 1에서 가변점은 “범용적 컴포넌트의 디자인” 부분에서 점선으로 표현된 곳을 의미한다. 공통된 부분들로 구성된 제품 계열 컴포넌트는 “변형”의 결합으로 제품 개발 수준의 재사용 자산이 된다.

이와 같이 제품 계열을 구축하기 위해서는 제품군에 존재한 공통성(commonality)과 가변성(variability)을 도출하고, 제품 수준의 재사용 컴포넌트를 개발하는 것에 초점을 둔다. 제품 개발 수준의 핵심 자산은 제품 개발에 필요한 구성 요소로 사용될 수 있도록 지원한다. 제품 계열 수준의 자산은 가변성에 대한 구현, 확장, 변경을 수행하고 제품군 개발에 이용될 수 있도록 저장하고 향후 재사용을 위해 관리한다.

제 4 장은 제품 계열 기술을 적용하여 임베디드 시스템을 개발할 때 수행되어야 하는 활동들을 설명하고 이들 활동 간의 관계 및 개발 프로세스에 대해 기술한다.

4. 임베디드 시스템 개발의 제품 계열 기반 기술 적용

임베디드 시스템은 하드웨어 구성 요소와 소프트웨어 구성 요소의 결합으로 이루어지며 이 두 구성 요소의 연관성으로 시스템의 구성이 복잡해진다. 따라서 시스템 전반에 대한 아키텍처의 설계 및 구현은 매우 중요하다.

시스템 구성의 기본적인 관점은 하드웨어 관점과 소프트웨어 관점으로 나뉘나 아키텍처를 바라보는 다양한 관점에 의해 세분화될 수 있다. 관점 별 아키텍처는 아키텍처 설계 산출물로 표현되고, 이러한 다른 관점의 시스템 세분화는 범용적 사용이 가능한 자산 생산을 위해 바람직하다.

본 논문에서는 임베디드 시스템의 복잡한 개발 환경을 지원하고 임베디드 시스템 제품군을 개발할 수 있도록 체계화된 개발 지원 프로세스를 정의한다. 임베디드 시스템 개발 프로세스는 소프트웨어 중심적 (software-centric) 방향으로 프로젝트 전체에서 소프트웨어의 모듈화, 개발 핵심 자산의 재사용, 재사용 자산의 프레임워크화를 지원하도록 구성된다. 개발 지원 프로세스는 제품 계열 기술에 따라 “제품 계열 개발”과 “제품 개발” 프로세스로 나뉜다.

4.1. 제품 계열 개발 프로세스

제품 계열 개발은 제품 개발에 필요한 범용적으로 사용될 수 있는 재사용 가능한 모델을 설계, 구현, 검증하는 작업으로 구성된다.

본 논문에서 정의하고자 하는 제품 계열 개발 프로세스는 그림 2 와 같다.

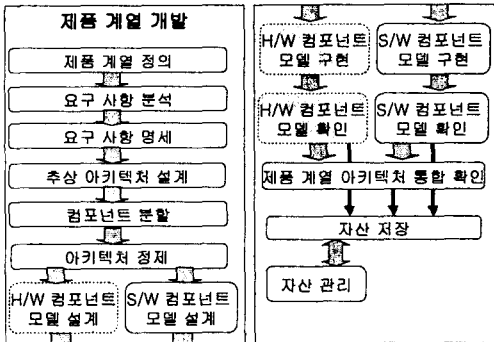


그림 2. 제품 계열 개발 프로세스

그림 2 의 제품 계열 개발 프로세스는 제품 계열에 대한 정의와 요구 사항 분석을 통해 상위 레벨의 추상화된 형태로 시스템의 구조를 아키텍처로 표현한다. 아키텍처는 기능 관점에서 컴포넌트로 분할되고, 각

컴포넌트는 그 기능이 하드웨어(H/W)가 담당하는 지 또는 소프트웨어(S/W)가 담당하는 지에 따라 H/W 컴포넌트와 S/W 컴포넌트로 나뉘어 설계되고 구현된다.

제품 계열 개발에서 구현된 컴포넌트는 제품 개발 프로세스에서 시스템을 구축하기 위해 코드 레벨의 컴포넌트로 실체화(realization)된다. 그림 2 와 그림 3 에서 사선으로 보이는 작업은 본 논문의 범위에서 배제되는 부분을 나타낸다.

제품 개발 프로세스를 살펴 보기 전에 본 논문에서 추구하고자 하는 개발 원칙은 다음과 같다.

표 2. 임베디드 시스템 개발 원칙

항목	개발 원칙	해결책
1	재사용 강화	제품 계열 자산 기반의 제품 개발
		컴포넌트 기반 소프트웨어 구성
2	시장 적시성 준수	제품 계열 구축
		제품 생산을 위한 제품 계열 관리 및 재사용
3	품질 강화	비기능적 요구 사항 분석과 설계
		초기 단계 검증
		최악 응답 시간 발생 상황 예측과 수정
		제품 계열 자산의 조직 내 반복적인 사용

임베디드 시스템 개발의 중요 개발 원칙은 첫째 제품 계열과 컴포넌트를 이용해 재사용성을 향상시키는 것이다. 둘째 제품 생산에 필요한 프레임워크를 사전에 생산하여 제품을 생산할 때마다 재사용하는 것이다. 이는 시장 적시성에 대한 사전 준비 체계를 마련한다. 마지막으로 시스템의 성능을 결정하는 요소인 비 기능 요구 사항에 대한 고려를 제품 계열 설계 단계에서부터 수행하여 제품이 개발되었을 때 제품과 설계간의 추적성을 마련한다는 점이다. 이는 임베디드 시스템 속성상 비기능적 요구 사항의 영향을 검증할 수 있는 수단이 된다.

4.2. 제품 개발 프로세스

제품 개발은 제품 계열 수준의 자산을 사용하여 제품을 구성하는 요소들을 설계하고 구현하는 작업에 초점을 둔다. 사전에 범용성을 염두에 두고 생산된 자산들은 제품 개발에 필요한 프레임워크로 제공된다.

제품 개발은 개별 제품에 대한 요구 사항 분석과 요구사항 명세 후 제품에 기반이 되는 아키텍처의 설계와 아키텍처의 구성 요소인 제품 컴포넌트의 개발 단계로 구성된다.

제품 아키텍처와 제품 컴포넌트의 생산은 제품 계열에서 개발된 아키텍처와 컴포넌트의 실제 코드가 구현되도록 개발 흐름을 진행한다.

제품 개발에서 테스팅은 제품 계열 개발 프로세스

초기 단계부터 제품 계열 수준의 자산을 확실히 하기 위해 검증한다. 제품 개발 단계에서는 설계에서 고려된 제품 요구 사항이 올바르게 반영되고 원하는 시스템으로 실재화되었는가를 확인하는 단계이므로 설치 비용이 높은 임베디드 시스템 분야에서 필수적이고 강화되어야 할 부분이다.

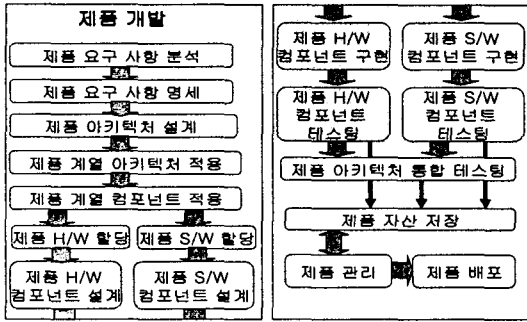


그림 3. 제품 개발 프로세스

임베디드 시스템은 응용 분야 별 제품 군이 나뉘는 다. 하나의 제품군에 대한 제품 계열은 다른 제품군에 대한 제품 계열과 별도로 관리될 필요가 있다. 제품군은 표 2에서와 같이 임베디드 시스템의 응용 분야 별로 분류했다.

표 2. 임베디드 시스템 응용 분야 분류

임베디드 시스템의 응용 분야	제품 예
전통적인 실시간 응용	자동제어 시스템, 의료 시스템
플랫폼 응용	PDA, 스마트폰
통신 인프라	라우터, 게이트웨이, 스위치, 액세스 포인트
가전기기	셋탑 박스, DTV, 디지털 카메라
유비쿼터스 컴퓨팅	센서, 단말기, RF 통신 모듈

5. 기대 효과

임베디드 시스템은 개별적인 제품의 확대 이외에 관련된 연관 산업에 대한 파급 효과로써 개발 방법론의 지원은 중요하다. 또한 방법론의 사용은 업계에서의 요구 사항이기도 하다.

제품 계열은 반복적인 사용에 따라 점차 재사용 가능 부분으로 제품 개발 지원 범위를 확장한다. 제품 계열의 초기 개발 시간과 비용은 단일 시스템의 개발에 비해 증가될 것으로 예상되나 제품군에 포함된 각 제품을 개발하면서 시간과 비용이 감소될 것으로 예측된다. 이러한 결과는 휴대단말 업체인 노키아를 포함한 업계에서의 실제 사례들이 발표되었고[3], 전체적으로 제품 계열 기술이 시간과 예산 계획을 향상시키고 제품 생산성을 높여 왔다고 보고하고 있다. 제품 계열의 장점을 제품 생산/관리/테스팅 비용 감소로 평

가하고 있다[4].

현재까지 국내 임베디드 시스템의 개발은 개발 방법의 체계화가 개발 전 범위에 있어 이루어지지 못하고, 시장 적시성과 개발 생산성 또한 극대화되지 못하였다. 국내 기술의 현황은 해외 기술 의존도가 매우 높고 수익 창출이 미비했으며 임베디드 시스템 제조에 대한 경쟁력을 빠르게 향상시켜 오지 못한 것도 사실이다.

임베디드 시스템 분야의 해외 기술 의존도 높은 현재 상황과 임베디드 소프트웨어를 내장한 제품이 기하급수적으로 늘고 있다는 점을 고려했을 때 임베디드 시스템 개발에 대한 핵심 기술과 표준화와 관련된 연구 및 보급이 요구된다.

6. 결론 및 향후 연구

본 논문은 제품 계열에 기반한 시스템을 개발하는데 필요한 개발 프로세스를 정의했고, 각 프로세스에서 필요로 하는 작업에 대해 제시했다. 임베디드 시스템의 개발 프로세스, 원리, 기법, 도구의 개발 체계는 방법론을 구성하는데 본 논문에서는 이 중 프로세스와 원리의 범위 안에서 방법론을 제시했다.

본 연구는 국내에서 임베디드 시스템의 개발을 위해 사용하는 방법론이 거의 없는 상황에 적용 가능성이 높고 개발자의 요구 사항에 맞게 적용 가능한 방법론으로 체계화하는데 중점을 두고 진행 중이다.

부가적으로 제품 계열과 제품의 자산을 관리하기 위한 프로세스를 고려할 것이며 본 논문에서 제시된 프로세스의 확장을 작업 중에 있다. 방법론 확장에는 세부 공정, 기법, 산출물의 정의와 양식을 마련하는 작업이 포함된다. 본 방법론은 시장 적시성, 재사용성, 품질 모두를 만족할 것으로 예상되며 실례로의 적용 사례 개발을 계획 중에 있다.

참고문헌

- [1] L.D.J. Eggemont, ed., "Embedded Systems Roadmap 2002," STW Technology Foundation/PROGRESS, Utrecht, The Netherlands, March 2002.
- [2] Jack Ferguson, "Crouching Dragon, Hidden Software: Software in DoD Weapon Systems," IEEE Software, July 2001, pp. 105-107.
- [3] Ari Jaaksi, "Developing Mobile Browsers in a Product Line," IEEE Software, July-Aug., 2002, pp. 73-80.
- [4] P. Clements and L. Northrop, "A Framework for Software Product Line Practice," SEI Interactive, Sep. 1999, pp. 1- 19.