

## MBASE를 적용한 임베디드 소프트웨어 개발 방법론

김상수<sup>o</sup>, 인 호  
{sookim, hoh\_in}@korea.ac.kr  
고려대학교 컴퓨터학과

### Embedded Software Development Methods Applying the MBASE

Sang-Soo Kim<sup>o</sup>, Hoh In  
Department of Computer Science and Engineering, Korea University

#### 요 약

제품에서의 임베디드 소프트웨어의 비중은 점점 더 증가하고 있으며, 임베디드 소프트웨어의 특성으로 인해 일반적인 소프트웨어의 개발방법론을 적용한 임베디드 소프트웨어의 개발은 시간, 생산성, 비용 등 다양한 측면에서 제품의 성공을 보장해줄 수 없는 문제점을 안고 있다. 본 논문은 임베디드 도메인에 특성화된 소프트웨어개발에 필요한 방법론을 개발하기 위하여 MBASE(Model-Based (system) Architecting and Software Engineering) 기법을 적용하였으며 임베디드 제품개발의 실패를 가져올 수 있는 모델들 간의 충돌을 찾아내고 이를 개선하기 위한 개발 기술을 중심으로 임베디드 소프트웨어 개발 방법론을 제시하였다. 임베디드 소프트웨어 개발시 적용했을 때 발생하는 일반적인 문제점을 개선할 수 있다는 결론을 도출하였다.

#### 1. 서 론

자동차, 항공기, 모바일 폰, DVD 플레이어, 의료기기, 산업용 장비 등 많은 제품들이 임베디드 소프트웨어를 포함하고 있으며 향후에는 더욱더 임베디드 소프트웨어의 비중은 증가할 것이다.[1] 이러한 제품들의 주요 성공 요소는 고품질의 임베디드 소프트웨어를 필요한 시기와 제한된 비용으로 성공적으로 개발하는 것이다. 이를 위해서는 시간과, 생산성, 비용, 품질의 최적화된 임베디드 소프트웨어를 개발하는 것이다. 많은 소프트웨어 개발방법론이 존재하지만 임베디드 소프트웨어의 특수한 요구를 만족할 수 있는 방법론이 보편화 되어있지 않다. 일반적인 소프트웨어 개발방법론을 임베디드 소프트웨어 개발에 테일러링 하여 적용하기에도 어려운 실정이며, 특히 임베디드 도메인의 경우 신뢰성(Reliability factor), 비용(Cost factor) 및 시장 출시 시간에 좌우되는 특징이 있으며 이러한 특수한 도메인을 겨냥한 개발 기술이 필요하다.

본 연구에서는 임베디드 소프트웨어의 도메인에서 적용될 수 있는 개발방법론을 제안하였다. 본 논문에서 제안된 방법론은 임베디드 시스템의 수명주기 동안 발생할 수 있는 충돌로 인한 프로젝트의 실패를 방지 하고 임베디드 소프트웨어의 분야에 맞게 특성화하기에 용이한 MBASE(Model-Based (system) Architecting and Software Engineering)[2]를 기반으로 하고 있다.

MBASE 방법론을 임베디드 소프트웨어 개발에 적용하여 임베디드 도메인의 특징인 신뢰성과, 비용 등과 관련된 모델간의 충돌을 찾아내고 이를 방지할 수 있는 개발 방법론들을 중심으로 제안하였다. 이를 위해 본 논문의 2장에서는 MBASE방법론을 소개하고, 3장에서는 임베디드 소프트웨어의 도메인을 분석후, 4장에서는 MBASE를

적용한 임베디드 소프트웨어 개발 방법론을 제시하고 결과를 도출한다. 5장에 결론과 향후 연구내용을 제시한다.

#### 2. 배경: MBASE

MBASE는 프로세스(process), 제품(product), 특성(property), 성공(success) 모델을 통합적으로 관리하는 접근법으로 1998년 USC(University of South California)의 DR. Barry Boehm의 주도로 개발되었다. 소프트웨어 시스템 개발 시 발생할 수 있는 모델들 간의 충돌을 방지하여 성공적으로 개발이 완료될 수 있도록 하는 접근법으로 소프트웨어 시스템 개발에 필요한 전 수명주기간의 프로세스를 제공한다. USC CSE(Center for Software Engineering)에서는 매년 15개 이상의 프로젝트에 MBASE 방법을 적용하고 있으며, 미 공군, 국방성 등 다양한 프로젝트에 적용되고 있다.

소프트웨어 또는 시스템 아키텍처가 만족하는지 결정하기 위하여 컴포넌트, 커넥터, 형상 및 제약사항에 대한 명세서 이외에 반드시 필요한 것이 있다. 아키텍처 하나만으로는 이것의 완전성을 평가하기에는 상당한 불합리 한 점이 있다. MBASE는 이러한 쟁점들을 해결하기 위해 제시된 시스템 개발 접근법이다. MBASE는 소프트웨어 개발 프로젝트의 제품(product), 프로세스(process), 특성(property), 성공(success) 모델의 일치성과 상호작용에 초점을 맞추고 있다.

개발자들은 소프트웨어를 개발하기 위해 수많은 개발 프로세스를 사용하고 있다. 그중 가장 대표적인 것들은 waterfall, evolutionary, incremental, spiral 모델 등이 있다. Product 모델은 운용개념, 요구사항, 아키텍처, 설계 및 코딩 등과 이들 간의 연관관계를 명세화 하는 것이다. Property 모델은 요구 및 적절한 수준과비용, 일

정, 성능, 신뢰성, 재사용성, 보안 등의 품질속성과 같은 절충 가능한 프로젝트 요소로 정의된다. Success 모델은 correctness, stakeholder win-win, business-case, IKIWISI(I'll know it when I see it) 등이 있다.[3]

소프트웨어 개발과정에서 발생할 수 있는 일반적인 모델충돌은 다음과 같다.

- Design-to-schedule 프로세스와 우선순위가 설정되지 않은 요구사항 간의 충돌
- COTS-driven product와 Waterfall process 간의 충돌
- Risk-based process와 spec-based progress payments 간의 충돌
- life-cycle architecture 없이 Evolutionary 방식으로 개발 시
- Golden rule과 stakeholder win-win 방식간의 충돌
- Spec-based Process와 IKIWISI success model

MBASE 개발 방법을 개발프로젝트에 따라 발생할 수 있는 충돌을 방지하기 위하여 다양한 방법으로 모델을 조합하여 적용할 수 있다. MBASE 개발방법을 구성하는 근간은 Stakeholder win-win(success model), Win-Win Spiral Model(Process model), LCO 및 LCA 산출물 명세서(Product model), 주요지정 별 Milestones(Property model)이다. 특히, MBASE에서는 도메인의 특성에 맞는 접근방법과, 프로젝트의 특정 도메인에 적합하도록 첫번째 단계의 모델 레이어로 부터 출발하는 일관적이고 통합된 도메인 묘사 방식을 적용하고, 소프트웨어 수명주기 프로세스 간에 유연성을 제공, 참조 모델을 활용하고 모델기반으로 개발하는데 적합한 접근 방법 등 일반적인 다른 객체지향 접근법과 차별화 된 객체지향 개발방법을 적용하고 있다.

### 3. MBSSE를 적용한 임베디드 도메인 분석

임베디드 시스템은 일반적인 시스템과 다른 다양한 특징을 때문에 임베디드 시스템의 개발은 일반적인 시스템 개발과는 많은 차이점을 가지고 있다. 임베디드 시스템 개발 기술은 hard timing 제약, 메모리 및 전원사용의 한계, 규정된 하드웨어 플랫폼 기술, 시스템 비용과 같은 특수한 제약사항에 역점을 두어야 한다.

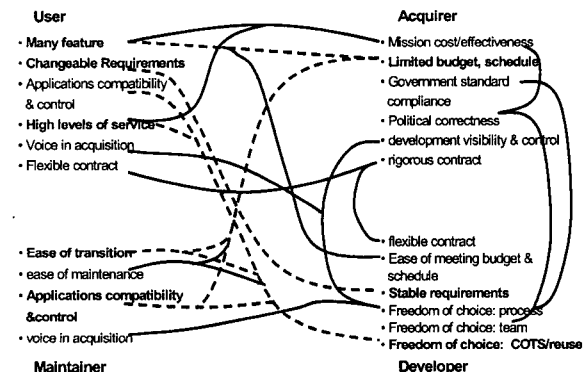


그림 1. 임베디드 소프트웨어 개발 시 모델간의 충돌

그림 1은 MBSAE에서 제시하는 방법에 따라 임베디드 시스템 개발 과정에서 발생할 수 있는 모델간의 충돌을 찾아낸 것이다. 임베디드 시스템의 주요한 특징인 비용, 신뢰성을 포함한 높은 품질, 플랫폼 기술 등 제약사항, COTS 및 legacy 시스템의 재사용 등과 관련된 충돌을 찾아냈다.

표 1. 모델간의 충돌과 관련된 요소

Clash	Stakeholder	Phase
Property-Product	User, Developer	Requirements
Product-Property	User, Acquirer	Architecture
Product-Property	User, Developer, Maintainer	Implement
Product-Success	User, Developer, Maintainer	Implement
Process-Property	User, Developer	Integration

표 1은 임베디드 소프트웨어 개발시 발생할 수 있는 주요 충돌 내용을 충돌형태, 충돌과 관련된 이해당사자 그리고 주로 발생하는 시점별로 구분하여 나타낸 것이다. 임베디드 소프트웨어 개발시 발생하는 충돌들은 일반적인 소프트웨어 개발시 나타나는 충돌들을 거의 대부분 포함하고 있었으며 임베디드 소프트웨어의 특성 때문에 갖는 충돌들도 찾아 낼 수 있었다.

### 4. MBASE를 적용한 임베디드 소프트웨어 개발 방법론

#### 4.1 Product-Property-Success 모델충돌 해결방안

획득부서는 제한된 개발비용과 일정을 고려하지만 사용자는 다양한 기능을 원하고 개발자는 안정된 요구사항을 근거로 개발을 하기를 원하기 때문에 이들 간에 충돌이 일어날 수 있다.

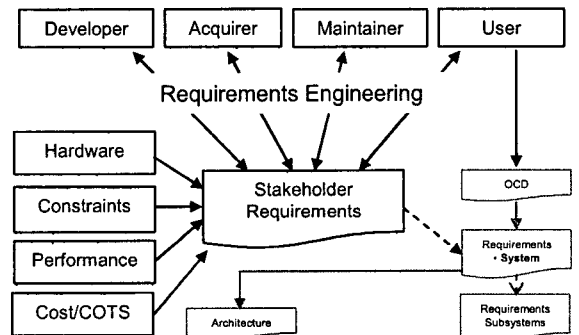


그림 2. 요구사항 도출 프로세스

대부분의 소프트웨어 개발에서처럼 임베디드 시스템 개발에는 다양한 이해당사자(stakeholder)들이 존재한다. 이것은 시스템엔지니어링 프로세스를 수행하는 동안 확연하게 드러난다. 사용자는 요구사항의 잦은 변경을 원하고 개발자는 요구사항이 안정된 상태에서의 개발을 원한다. 그림 2에 제시한 바와 같이 이해당사자와 주변 환경을 종합적으로 고려한 요구공학 절차를 적용하여 단순한 사용자 요구사항이 아닌 이해당사자 요구사항(stakeholder requirements)을 개발해야 한다.

MBASE에서의 Success 모델은 이해당사자를 모두가 win 할 수 있는 접근법을 제공하고 있다. 개발 프로젝트가 성공 요건은 어느 하나의 이해당사자만의 만족해서는 안된다. 예를 들어 사용자 측면에서는 성능과 일정 면에서 제품에 만족하더라도 획득기관과, 개발자가 비용적인 측면에서 만족하지 않는다면 프로젝트가 성공했다고 볼 수 없는 것이다.[4] 모든 이해당사자들이 win 할 수 있도록 하기 위해서는 개발의 전 과정에서 요구사항을 이해당사자 요구사항으로 개발하고 관리해야 한다.

4.2 Product-Success 모델 충돌 해결방안

임베디드 소프트웨어 개발자가 COTS를 선택하여 개발할 경우 사용자 및 시스템관리자가 원하는 적용성과 적합성과의 충돌이 발생한다.

임베디드 소프트웨어의 경우 하드웨어 플랫폼 기반으로 개발되어야 하기 때문에 특정 제품 또는 시스템에 맞게 소프트웨어를 개발해야 하는 제약사항을 갖는다. 이러한 경우 개발 프로세스의 적용에 따라 달라질 수 있다. COTS를 사용하는 경우 Waterfall 프로세스를 사용할 경우에는 충돌을 발생 시킬 수 있다. MBASE 개발 방법론에서는 COTS 및 legacy 시스템을 활용하기 적합한 접근법으로 WinWin Spiral 모델 등을 제공하고 있다. 요구사항의 정의 정도와 변화에 따라 대응할 수 있는 방법론을 적용해야만 임베디드 소프트웨어의 특성에서 발생할 수 있는 문제점을 해결할 수 있다.

4.3 Process-Property 모델 충돌 해결방안

임베디드 소프트웨어가 독립적인 개체로서 인증될 수 없다는 것은 중요한 사실이다. 임베디드 소프트웨어의 개발 프로세스를 결정하는 가장 중요한 요소로 작용하기 때문이다. 하드웨어 시스템의 개발과 병행하여 개발되며 상호간에 정보를 교환하는 방식으로 개발이 진행되어야 한다. 그림 5에서와 같이 시스템 레벨에서의 각 절차간의 절충 과정은 개발 수명주기 간 지속적으로 이루어져야 한다. Co-design 접근법은 일반적인 개발방법에서 설계단계부터 하드웨어와 소프트웨어를 분리하여 설계하는 방식이 아니라 상호 제약사항과 요구사항을 보완해가면서 동일한 프로세스에서 개발하는 방법으로 임베디드 시스템 개발 시에 적절한 접근법이다.

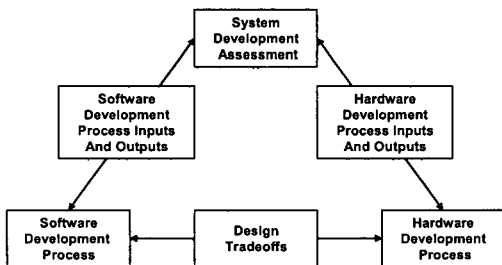


그림 5. 하드웨어와 소프트웨어 개발 프로세스 간의 관계

이상에서 살펴본바와 같이 MBASE를 기반으로 한 임베디드 개발 방법론은 기존의 개발방법을 적용했을 때

발생할 수 있는 요구도출, 아키텍처, 설계 및 구현단계에서의 문제점을 개선할 수 있는 접근법이 될 수 있다는 것을 확인할 수 있었다. 표 2에서 나타난 몇 가지 장점들은 개발과정에서 발생할 수 있는 모델들 간의 충돌을 방지하기 위해서 MBASE 방법론에서 제시하고 있는 방안들을 적용한 결과이다.

표 2. MBASE 적용결과

구분	기존 문제점	MBASE 적용결과
Product-Property-Success	다양한 이해관계자들의 요구사항을 고려하지 않은 소프트웨어 개발	요구사항 도출과정에서 모든 이해 당사자들이 절충을 통하여 합의된 요구사항을 도출함으로써 모든 이해당사자가 win 가능
Product-Property	비용과 개발 효과측면의 고려 미흡	비용, 성능 관련 다양한 토크를 제공함으로써 최적의 개발 효과를 달성 가능
Product-Success	하드웨어 플랫폼 및 COTS 사용 시 적용성과 적합성 문제 발생	COTS, legacy 등의 활용에 적합한 객체모델기반 개발방법론의 적용으로 문제 해결 가능
Process-Property	하드웨어와 독립적인 설계 구현 과정을 거친 후 통합시험 시 문제점 발견	하드웨어와 소프트웨어 Codesign 개발방법을 적용 절충을 통한 개발을 통해 하드웨어에 적합한 소프트웨어 개발가능

5. 결 론

본 논문에서는 MBSE 방법을 적용하여 임베디드 소프트웨어의 분야에 특성화된 개발 방법론을 제시하였다. 임베디드 시스템 개발 시 발생할 수 있는 모델들 간의 충돌을 찾아내고 이를 방지하기 위하여 요구사항 단계를 포함한 각 단계에서의 개발기술을 MBASE 접근방법을 기반으로 제시하였다. 연구결과 MASE 접근법은 임베디드 시스템에 일반적인 방법론 적용시 발생 가능한 몇 가지 문제점을 해결할 수 있다는 결론을 얻었다.

본 논문에서 제시한 개발방법을 임베디드 도메인에 일반화할 수 있는지를 검증하기 위하여 15개의 강의 프로젝트에 적용하였지만 평가나 검증 결과로 제시하기에는 미흡한 점이 있었다. 향후에 MBASE를 임베디드 도메인에서 일반적으로 적용할 수 있도록 하기 위하여 임베디드 버전 개발을 위한 연구를 수행할 계획이다.

참고문헌

- [1] Bas Graaf, Marco Lomans, Hans Toetenel, "Embedded Software Engineering: The State of the Pratrice", IEEE Software, November/December, 2003
- [2] Boehm, B. and Port D., "Escaping the Software Tar Pit: Model Clashes and How to Avoid Them," Software Engineering Notes, Association for Computing Machinery, 1999.
- [3] Boehm, B., Port, D., Alsaid, M., "Avoiding the Software Model-Clash Spiderweb", IEEE Software, November, 2000.
- [4] Boehm, B., H. In, "Identifying Quality-Requirement Conflicts". IEEE Software, March 1996.