

# 임베디드 시스템 개발 방법론의 요구사항에 관한 연구

김현아, 박민재, 김광훈  
경기대학교 전자계산학과 워크플로우 기술 연구실  
e-mail : [hyuna@kyonggi.ac.kr](mailto:hyuna@kyonggi.ac.kr)

## A Study the requirement of embedded system development methodology

Hyun-Ah Kim, Min-Jae Park, Kwang-Hoon Kim  
Dept. of Computer Science, Kyonggi University

### 요 약

본 논문에서는 임베디드 시스템 개발 현황과 세계적인 동향을 살펴보고, 현재 임베디드 시스템을 개발함에 있어서 방법론의 필요성 여부, 그리고 방법론에 관한 요구사항을 분석 하였다. 이에 웹을 이용한 설문을 실시하였고 임베디드 시스템의 개발 요구사항에 관한 실제 개발자 및 연구자들의 다양한 의견을 통하여 임베디드 시스템 개발 방법론에 대한 요구사항을 얻을 수 있다. 이러한 사용자 및 개발자 연구자의 요구사항의 분석 결과를 통하여 앞으로 보다 더 효율적으로 임베디드 시스템을 개발하기 위한 개발 방법론을 도출할 수 있다.

### 1. 서론

최근 급격한 발전을 보이고 있는 정보통신 및 산업 분야에서 활용되고 있는 임베디드 시스템 개발에 방법론의 필요성이 증대되고 있다. 임베디드 시스템은 범용 프로세서가 처리하기 힘든 특수한 목적의 시스템으로서 하드웨어와 소프트웨어의 특성을 고려하여 주어진 제약조건을 만족하면서 최소의 비용으로 시스템을 구현해야 한다는 특성을 가지고 있다. 전 세계에서 존재하는 컴퓨터 시스템의 90% 이상이 임베디드 시스템이라는 보고가 있을 정도로 임베디드 시스템의 활용 분야는 매우 광범위할 뿐만 아니라 부가가치가 높은 분야이기도 하다.

이러한 대세로 임베디드 시스템을 개발하는 데는 효율적으로 임베디드 시스템을 개발할 수 있도록 하는 개발 방법론이 필요하며, 임베디드 시스템 개발 방법론을 정립하기 위한 요구사항에 관한 연구 또한 필요하다.

이에 맞춰 본 논문에서는 첫째로 임베디드 시스템에 관하여 알아보고, 둘째로 세계적으로 이루어지고 있는 임베디드 시스템의 연구 동향에 대하여 알아본다. 그리고, 임베디드 시스템 개발 방법론에 관한 설문을 통하여 그에 대한 분석적으로, 본 연구의 주요한 목적이 되는 임베디드 시스템 개발 방법론의 요구사항에 관하여 분석한다

### 2. 이론적 배경

#### 2.1 임베디드 시스템

임베디드 시스템은 1950년대 통신 장비를 제어하기 위하여 컴퓨팅 시스템이 내장되면서 등장하였다. 이후 1990년대 초반까지는 군사용 기기와 산업용 기기들을 제어하기 위한 목적으로 많이 사용되어 왔다. 1990년대 후반부터 3C로 대변되는 컴퓨팅(Computing), 통신(Communication), 가전기기(Consumer Electronics) 기술들의 융합이 이루어지면서 임베디드 시스템의 영역이 크게 확대되었다. 이를 통해 임베디드 시스템은 첨단 산업으로 재등장하였으며 기존의 가전 기기와의 차별성을 강조하여 '정보 가전', 또는 기존 PC와의 차별성을 강조하여 라는 새로운 이름 'post PC'으로 불리게 되었다. 이러한 임베디드 시스템이 내장되는 기기는 표 1과 같이 다양한 영역으로 분류될 수 있다. Post PC로 대변되는 첨단 임베디드 시스템 산업은 단지 post PC 기기 시장의 확대만을 가져오는 것이 아니라 연관 산업의 전후방에서 연쇄 확대 효과가 있다는 점에서 중요성을 갖게 되었다. Post PC 산업이 확대되면 관련된 부품이나 중간 투입 재 산업이 활성화 되고 아울러 post PC를 중간재로 활용하는 새로운 산

업이 형성되기 때문이다.

임베디드 시스템이란 다른 시스템의 일부로 내장된 마이크로 프로세서 기반 디지털 시스템을 의미한다. 주로 특정 기기에 내장된 컴퓨팅 시스템을 일컫는 말이다. 임베디드 시스템은 범용 컴퓨팅 시스템과 달리 자신을 포함하고 있는 기기에 부과된 특정 목적의 컴퓨팅작업만을 수행한다. 일반적으로 임베디드 시스템은 특정 목적을 위하여 동작하는 컴퓨팅 시스템 이라고 정의할 수 있다 따라서 범용 컴퓨터를 제외하고 컴퓨팅 시스템이 내장된 모든 시스템이 임베디드 시스템이다.

표 1 임베디드 기기의 분류

임베디드 기기	예
실시간 기기	자동제어 시스템, 의료 시스템
플랫폼 기기	PDA, Tablet PC
통신 인프라 기기	게이트웨이, 액세스 포인트
가전 기기	셋탑 박스, PVR/DVR, DTV, 디지털 카메라
유비쿼터스 기기	액티브 배지, 센서

과거의 전통적인 임베디드 시스템은 올바른 결과 못지않게 제한된 시간 내에 결과를 내어야 하는 실시간성을 주요 특징으로 하였다. 차세대 임베디드 시스템이 사용되는 환경은 초고속 통신과 광역적인 이동성을 제공해야 하는 반면, 상대적으로 제한적인 하드웨어 자원과 취약한 보안성을 가진다. 이에 따라 post PC 로 대변되는 현대의 첨단 임베디드 시스템은 이동성과 연결성을 주요 특징으로 하면서 다음과 같은 요구 사항을 가진다.

- (1). 제한적인 하드웨어 자원의 효율적 관리(경량성)
- (2). 전력 소모의 최적화. (저전력)
- (3). 보안의 허점을 보완. (보안성)
- (4). 사용자의 요구 수준의 서비스 제공. (QoS)

## 2.2 임베디드 소프트웨어 연구 동향

첨단 임베디드 시스템 산업은 임베디드 기기, 즉 단말기 자체의 산업뿐만 아니라, 관련 부품과 중간 투입재의 다양한 산업으로 구성된다. 구체적으로 산업으로서 단말기, HW의 핵심 코어를 이루는 마이크로프로세서 디스플레이 무선랜 2 차 전지 등의 부품과 들을 포함하는 시스템 설계 산업이 존재한다. 한편 SW 산업으로서 SW의 핵심 코어를 이루는 임베디드 OS 와 미들웨어 등의 시스템 소프트웨어와 응용 소프트웨어 산업이 있다.

산업 전반의 동향으로는 차세대 IT 신 산업을 이끌어 나갈 주요 핵심 분야로서 임베디드 S/W 기술에 대한 중요성을 크게 인식하여 정부 및 국책 연구 기관을 중심으로 통신 정보 기술과 인프라 기술을 더욱 성장시킬 수는 신 산업 동력으로 연구 자원의 투자가 크게 증가하고 있는 추세이다. 산업체에서도 정부 연구 기관의 협력을 체결하여 홈 서버 및 모바일 제품 생산의 핵심 기술로 임베디드 기술의 상품화에 대한 노력을 시작하고 있고 학계에서는 안정적이지 못한 연구비 지원과 지속적인 연구 수행 인력의 부족으로 기존에 수행해온 분산 운영체제나 실시간 시스템의 연구를 계승하여 임베디드의 일부 부분 기술에 대한 연구를 시도하고 있다. 그러나, 미국에서와 같이 막대한 연구 기금 지원을 바탕으로 장기간에 걸친 핵심 임베디드 기술 개발이나, 유럽에서와 같이 업체 컨소시엄에 의한 실질적인 임베디드 제품 개발 지원 체계가 우리나라에서는 매우 미비한 상태임을 인식해야 한다. 미국과 유럽의 연구는 대규모의 투자를 기반으로 임베디드 소프트웨어를 개발하기 위한 다양한 연구가

진행 중에 있다. 미국은 여러 대학 및 연구 기관이 함께 협력하여 임베디드 소프트웨어를 개발하기 위한 개발 방법론과 개발 지원 도구를 연구하는데 초점을 두고 있다. 대표적인 예로 미국의 조지아 기술 연구소는 1999년부터 5년 동안 모바일 및 홈 컴퓨팅 분야의 임베디드 소프트웨어를 개발할 때 적용할 수 있는 YES(Yamacraw Embedded Software Methodology) 개발 방법론을 개발하였다. 버클리 캘리포니아 대학은 2000년도부터 다른 협력 대학 및 기업과 함께 MoBIES 프로젝트를 통한 컴포넌트 기반의 실시간 임베디드 시스템을 구축하기 위한 통합 개발 지원 도구를 개발하였다. 유럽 연합은 여러 국가의 연구 기관과 대학이 함께 임베디드 시스템 개발 체계와 프레임워크 개발에 초점을 두고 있다. MOOSE 프로젝트는 3개국(네덜란드, 스페인, 핀란드)의 13 기관이 올 해 2월 임베디드 시스템 개발을 위한 개발 체계 및 프레임워크 개발을 위한 연구를 마쳤다. 2002년부터 프랑스의 조셉 푸리에 대학과 10개국 24개 기관은 컴포넌트 기반의 임베디드 시스템을 개발하기 위한 기술, 표준 및 경험을 연구하고 산업계와 학계에 보급하기 위해 ARTIST 프로젝트를 수행하고 있다.

임베디드 S/W 는 시스템 전체를 개발하기 위한 통합적 환경이라기보다는 임베디드 S/W 를 구성하는 부분적인 기술 요구를 해결하기 위한 시도가 주류를 이루어야 하며, 현재 기존의 실시간 시스템과 운영체제 시스템 개발에서 진행해 온 세부 기술들을 임베디드 S/W 개발을 위한 확장 연구가 진행되고 있다. 국가의 정책과 기업의 연구 자금, 그리고 연구소의 기술 인프라와 학계의 핵심 기술들을 융합하여 장기간에 걸쳐 임베디드 S/W 를 위한 체계적인 생산 기술과 활용 방안, 그리고 지원 환경 등에 대한 연구 프로젝트가 절실히 필요하다.

임베디드 소프트웨어는 전 산업 분야에 걸친 산업적 파급 효과가 큰 기술로서 2002년 8월 가트너의 조사에 따르면 임베디드 소프트웨어의 국내 시장 규모는 매년 10.3%의 년 평균 성장률을 보이며 2007년에 93억 달러에 이를 것으로 전망한다.

## 3. 임베디드 시스템 개발 방법론에 관한 설문 분석

임베디드 시스템 개발 방법론에 관한 요구 사항에 대한 연구 목적으로 임베디드 시스템에 관심 있는 집단을 대상으로 임베디드 시스템의 개발에 꼭 필요한 중요 질문을 제시 총 문항 수는 20 문항, 총 응답자 수는 55 명으로 임베디드 시스템 개발 방법론에 관한 설문조사를 시행하였다.

임베디드 시스템 개발에 참여한 응답자 중 임베디드 시스템 주된 개발 분야를 살펴 보면 총 13 개의 항목 중 통신 기기 즉 휴대폰이나 단말기 쪽 분야가 가장 많았고 다음으로는 개인용 정보기기류 인 PDA 를 개발하는 것으로 나타났다. 분석 결과 통신기기 분야가 가장 활발한 개발 분야인 것으로 나타났다. 그 외 기타 의견으로는 로봇, 스케줄링, 인터넷품질측정장비, 임베디드 소프트웨어 테스트 장비, DMB 등을 의견으로 제시 하였다.

임베디드 시스템의 평균 개발기간은 6 개월 이상인 업체가 37 명으로 전체의 62%를 차지 하였고, 6 개월 미만인 개발 기간을 갖는 업체도 38%를 차지 하였다. 또한 개발 기간에 따라 개발 인원의 추이도 변화가 있지만 대다수의 회사가 5 명 이하의 인원을 투입하고 개발 기간이 길어 질수록 참여 인원수도 10 명 이상의 추이로 늘어나는 경향을 볼 수 있다.

임베디드 시스템 개발을 위해 방법론을 적용해 보았느냐는 질문에는 90%에 가까운 응답자들이 경험을 해 보지 못했다고 응답하였는데 그 이유로는 개발 일정과 잘 몰라서라는

부분이 절반 이상을 차지 하였다. 사용을 해 보았다는 11%의 응답자들은 프로토타이핑 방법론, COMET, ROOM, UML OO design pattern, XP 등을 써 보았다고 응답하였다.

또한 개발방법론의 사용 장점으로서는 체계적인 프로젝트 관리가 가장 많은 비율을 차지 하였고, 그 다음으로는 임베디드 SW/HW의 통합이 유리하는 이유를 장점으로 꼽았다.

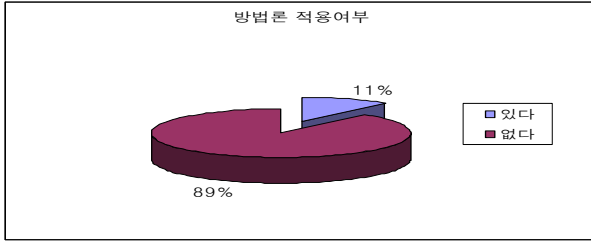


그림 1 방법론 적용 경험 여부

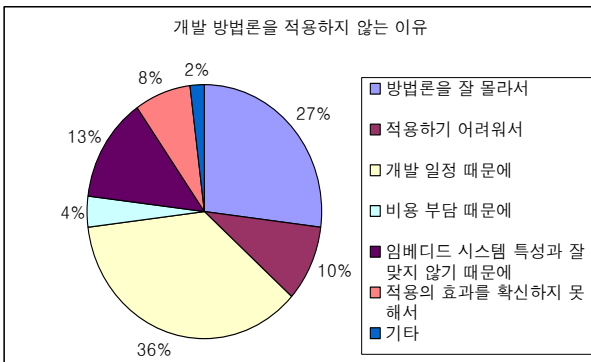


그림 2 개발 방법론을 적용하지 않는 이유

임베디드 시스템 개발 방법론이 제공해야 할 중요 기능은 표준 모델링 언어 사용(UML)이라고 응답한 사람이 38%로 가장 많았고, 프로젝트 특성에 따른 조정지침 제공이 30%이다. 이 결과로 볼 때 개발 방법론의 적용 시 변화에 적응이 용이 하도록 표준화된 모델링 언어와 조정 지침이 제공되는 것에 초점이 맞춰 질 것으로 예상 된다.

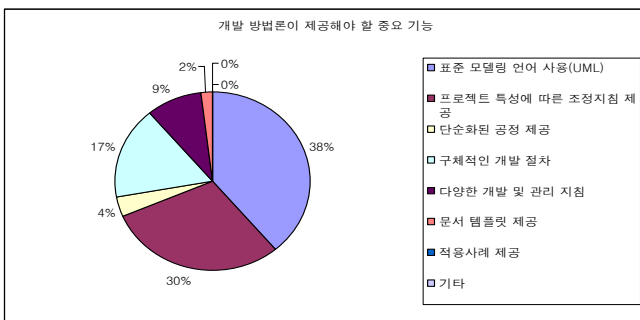


그림 3 개발 방법론이 제공해야 할 중요 기능

임베디드 시스템을 개발할 때 가장 중요한 점은 당연 품질이 최우선으로 74%의 응답 결과가 나왔고 납기나 비용 측면은 비슷한 결과 치료 중요도를 차지 하였다 그 외 기타 의견으로는 프로젝트의 특성에 따라 다소 차이는 있지만 납기가 일반적으로 우선시 되어 한다는 응답자도 있었다. 이에 반해 프로젝트 관리 측면에서는 개발 측면과는 다소 차이가 있는데, 위험 관리와 요구사항 관리가 각각 39%와 30%로 많은 비중을 차지 하였고 납기 준수 항목도 24%에 달했다.

#### 4. 임베디드 시스템 개발 방법론 요구사항

##### 4.1 임베디드 시스템 개발 방법론 요구사항 도출

실제 임베디드 시스템 개발 방법론에 관하여 발생하는 문제점 등을 살펴보면 다음과 같이 볼 수 있다.

- o 임베디드 시스템 개발 프로세스를 정의 시, 방법론 적용자가 프로세스의 각 단계에 대한 이해 불충분의 문제점
- o 알려진 임베디드 시스템 개발 방법론이 부재의 문제점
- o 개발 방법론이나 프로세스의 필요성 인식의 부족의 문제점
- o 느끼고는 있지만 어떻게 적용해야 할 지 모르는 문제점
- o 하드웨어와 소프트웨어와의 사이에서의 퍼포먼스 문제점
- o 너무나 다양한 기기를 개발해야 한다는 임베디드 시스템 특성상 고려해야 할 개발 방법의 차이점에 문제점
- o 구체적인 사례 부족의 문제점
- o 개발 방법론 적용에 어려움을 겪고 있는 중견기업 등에도 적용할 수 있어야 한다는 문제점
- o 임베디드 시스템 개발 방법론의 관심 부재의 문제점
- o 하드웨어와 소프트웨어의 통합 시에 발생 할 수 있는 여러 가지 문제점
- o 임베디드 시스템 개발에 관련된 표준의 문제점
- o 프로세스의 각 단계에 대한 인식 부족의 문제점

##### 4.2 임베디드 시스템 개발 방법론 요구사항 분석

임베디드 시스템 개발 방법론에 관하여 발생한 문제점 등을 살펴보고 분석하면 다음과 같이 볼 수 있다.

- o 임베디드 시스템 개발 프로세스를 정의 시, 방법론 적용자가 프로세스의 각 단계에 대한 이해 불충분의 문제점을 인식하여 개발 방법론의 프로세스는 임베디드 시스템의 특성에 맞는 프로세스여야 하며, 각 단계에 대한 이해가 부족하다면, 방법론에 관한 교육이 필요하다고 할 수 있다.
- o 알려진 임베디드 시스템 개발 방법론이 부재의 문제점은 임베디드 시스템 개발 방법론을 적용할 수 있도록 방법론에 대한 홍보가 필요하겠다.
- o 개발 방법론이나 프로세스의 필요성 인식의 부족의 문제점은 임베디드 시스템 개발 방법론의 장점 등을 부각 시킬 수 있는 계기를 마련하는 방법으로 문제를 해결 한다.
- o 느끼고는 있지만 어떻게 적용해야 할 지 모르는 문제점 또한 임베디드 시스템 개발 방법론에 대한 교육의 필요성을 인식하여 실시 하도록 한다.
- o 하드웨어와 소프트웨어와의 사이에서의 퍼포먼스 문제점은 하드웨어 개발 업체와 소프트웨어 개발 업체 등에서 서로간의 협의를 거친 표준 인터페이스 등을 정의 표준화에 힘써야 한다.
- o 너무나 다양한 기기를 개발해야 한다는 임베디드 시스템 특성상 고려해야 할 개발 방법의 차이점에 문제점은 각 기기 등을 총괄 개발 할 수 있는 표준 방법론의 제시가 시급하다.
- o 구체적인 사례 부족의 문제점은 실제 적용 가능하도록 구체적인 사례 등을 공개한 체험 교육 등이 필요한 실정이다.
- o 개발 방법론 적용에 어려움을 겪고 있는 중견기업 등에도 적용할 수 있어야 한다는 문제점은 적용하기 쉽고 가벼운 개발 방법론의 대안이 나와야 한다는 것이다.
- o 임베디드 시스템 개발 방법론의 관심 부재의 문제점은

임베디드 시스템 개발 방법론에 관한 홍보를 통하여 개선할 수 있는 문제점이라 보겠다.

○ 하드웨어와 소프트웨어의 통합 시에 발생 할 수 있는 여러 가지 문제점은 하드웨어와 소프트웨어 통합 시 어려움이 발생하지 않도록, 방법론 내 하드웨어와 소프트웨어를 통합하는 단계를 거쳐 문제가 발생하지 않도록 해야 한다.

○ 임베디드 시스템 개발에 관련된 표준의 문제점은 각 임베디드 시스템의 인터페이스 등의 표준화를 통하여 임베디드 개발에 빠른 성장을 이룰 수 있는 방법이라 하겠다.

○ 프로세스의 각 단계에 대한 인식 부족의 문제점은 임베디드 시스템 개발 방법론의 프로세스 각 단계에 전체적인 프로세스를 한눈에 볼 수 있고, 개발 프로젝트를 관리하기 쉽도록 개발하여야 한다.

위에서 보여지는 것과 같이 설문이나 의견 수렴을 통한 임베디드 시스템 개발 방법론에 관한 내용을 정리하여보면 다음과 같다.

(1) 현재 통일된 인터페이스가 없는 시점에서, 임베디드 시스템 개발 방법론에 관한 표준이 정립되어야 한다.

(2) 실제 업계에서 알려지지 않은 임베디드 시스템 개발 방법론에 대하여 지속적인 홍보가 필요하다.

(3) 실제 프로젝트에서 용이하게 쓰일 수 있도록 임베디드 시스템 개발 방법론은 어렵지 않아야 한다.

(4) 임베디드 시스템을 개발할 때에 개발 방법론을 적용하여, 고성능의 임베디드 시스템을 구축해 낼 수 있는 전문 인력을 교육을 통해 배출해 내어야 한다.

(5) 임베디드 시스템 개발 방법론이 중견기업에서도 어려움 없이 적용가능 하도록 할 수 있는 방법이 있어야 한다.

(6) 하드웨어, 소프트웨어 그리고 그 사이의 인터페이스에서 일어날 수 있는 모든 경우와 문제점을 모두 고려하고, 그에 대한 보상도 모두 고려하여야 한다.

(7) 현재 개발 방법론은 대부분 대규모 프로젝트에 초점이 맞춰져 있어, 소규모 프로젝트를 수행할 때에는 방법론 적용이 쉽지 않다. 따라서 소규모 프로젝트에서도 적용 가능한 방법론이 필요하다.

## 5. 결론

본 논문은 임베디드 시스템 소프트웨어를 개발하는 것과 일반적인 컴퓨터 시스템 소프트웨어를 개발하는 것과는 다르다는 측면을 고려하여, 임베디드 시스템을 개발하는데 사용되고자 하는 개발 방법론에 관한 연구로 임베디드 시스템 개발 방법론 요구사항 분석에 관한 설문을 실시하여 그 결과를 분석하였다.

현재 국내에는 임베디드 소프트웨어를 경제적으로 개발할 수 있는 개발 방법론과 개발 지원 도구가 존재하지 않다. 그리고 임베디드 소프트웨어 개발에 필요한 표준화 작업은 미비하다. 임베디드 소프트웨어 개발이 기존의 전통적인 소프트웨어 개발과 확연히 다르다는 것을 고려했을 때 전통적인 개발 방법을 그대로 적용하기에 무리가 있다. 일례로 현재 산업계 표준의 모델링 언어인 UML(Unified Modeling Language) 또한 임베디드 소프트웨어의 특성에 맞는 모델링을 지원하도록 확장되어야 할 필요가 있다. 그리고 특정 산업 분야의 제품 개발을 지원하기 위해 개발 전체 주기에 대한 프로세스를 모델링하고 필요한 기술, 지침, 산출물 등을 실제 개발 요구 사항에 기초해 정의되어야 한다. 끝으로 실

제 소프트웨어 개발에 적용하여 적용 사례를 만들고 점진적으로 개선해 나가는 것이 요구된다.

국내 기술 경쟁력을 향상 시키기 위해 임베디드 시스템 개발 방법론 및 개발 지원 도구에 대한 연구를 지속적으로 이어가야 하며 임베디드 소프트웨어 발전을 위해 국내 임베디드 시스템의 활발한 생산이 제조 업체간의 효율적인 협력과 경쟁을 통해 다양한 서비스로 생산되기를 기대한다.

## 참고문헌

- [1] J. Neighbors, "The Draco Approach to Construction Software from Reusable Components," IEEE Transactions on Software Engineering, SE-10(5), pp.564-573, September 1984.
- [2] Software Productivity Consortium, Reuse-Driven Software Processes: Guidebook Version 02.00.03, SPC-92019-CMC, Herndon, VA, Software Productivity Consortium, 1993.
- [3] M. Simos et al., Software Technology for Adaptable Reliable Systems (STARS) Organization Domain Modeling (ODM) Guidebook Version 2.0, STARS-VC-A025/001/00, Manassas, VA, Lockheed Martin Tactical Defense Systems, 1996.
- [4] K. Schmid, "Scoping Software Product Lines," Proceedings of the 1st Software Product Line Conference (SPLC), August 28-31, 2000, Denver, Colorado, USA, Patrick Donohoe (Ed.), Software Product Lines: Experience and Research Directions, pp.3-22, Kluwer Academic Publishers, 2000.
- [5] A. Jaaksi, Development Mobile Browsers in a Product Line, IEEE Software Initiating Software Product Lines, pp.73-80, July/August 2002.
- [6] S. Thiel and A. Hein, Modeling and Using Product Line Variability in Automotive Systems, IEEE Software Initiating Software Product Lines, pp.66-72, July/August 2002.