ISSN 1598-5164

한국정보과학회

제38권 제2호

2011 가을
학술발표논문집(B)

Proceedings of
The 38th KIISE Fall Conference

한국정보과학회
KOREAN INSTITUTE OF INFORMATION SCIENTISTS AND ENGINEERS
2011년 11월 25일~26일 · 서울대학교

언어공학

컴퓨터지능

Model Driven Architecture 기반의 임베디드 테스트 프로세스에 관한 연구

김동호*, 손현승, 김우열 김영철
홍익대학교 컴퓨터정보통신공학과*
{ray*, bob}@selab.hongik.ac.kr

A Study on Embedded Test Process based on Model Driven Architecture

Dongho Kim*, Hyunseung Son, Wooyeol Kim, R. Youngchul Kim
Dept. of CIC., Hongik University, Korea*

요 약

현재 다양 이종의 임베디드 시스템이 활용과 소프트웨어가 차지하던 비중이 높아저감에 따라 소프트웨어의 검증이 중요성이 높아졌다. 그래서 다양한 이종 모바일 플랫폼에 맞는 소프트웨어 시스템의 빠르고 안정적인 개발과 체계적 시험을 위해 임베디드 시스템용 테스트 프로세스가 필요하다. 이를 해결고자, 기존에 MDA 기법을 적용해던 임베디드 개발 프로세스에, 동시에 테스트 프로세스에 추가 적용을 제안 한다. 이는 다양한 플랫폼의 개발과 더불어 체계적인 테스트도 보강이 하고자 한다. 본 논문에서는 이종 임베디드 테스팅를 위해 다중 V 모델에 MDA 메카니즘를 적용하였다. 이를 위해 개발 단계에 맞춰 동시에 테스트 프로세스의 단계, 활동, 작업, 산출물을 정의하고 이를 통해 개선된 다중 V 기반 모델 기반 테스팅에 적용하므로써, 다양한 타겟의 테스팅을 지원하고자 한다.

1. 서 론

현재 대부분의 임베디드 제품들에 있어서 소프트웨어의 중요성이 높아짐에 따라 소프트웨어를 개발하는 것뿐만 아니라 테스트의 중요성도 높아지고 있다. 그래서 매우 다양한 개발 프로세스 및 테스트 프로세스가 존재한다. 테스트 프로세스를 사용하는 중요한 이유 중의 하나는 조직의 시간과, 노력, 그리고 비용을 줄여준다. 뿐만 아니라 품질 및 소프트웨어 개발의 정규화와 같은 문제를 해결하는데 있어서 필요하다. 그러나 이런 테스트 프로세스는 개발조직의 크기나 경험, 개발 환경, 자체 개발 프로세스와의 충돌 혹은 최종 산출물의 사이즈나 품질 등에 따라 적용이 힘든 문제점들을 가지고 있다[1]. 그래서 테스트 프로세스를 각 문제에 맞춰 개발 당사자가 유동적으로 사용할 수 있는 가변성을 갖는 테스트 프로세스의 연구가 필요하다. 더군다나 최근 등장한 다양한 형태의 모바일 플랫폼상에서의 빠르고 안정적인 개발은 큰 장점이 될 수 있다. 기존에 제안한 MDA기반의 개발모델은 그림1과 같다. 기존 각 단계별로 살펴보면 크게 3부분으로 나뉜다. 각 단위는 다음과 같다. 상위 모델을 만드는 TIM(Target Independent Model) 타겟 명세적인 모델을 만드는 TSM(Target Specific Model), 타겟 의존적인 코드를 만드는 TDC(Target Dependent Code)다. 각 타겟 독립적인 모델과 명세적인 모델, 타겟 의존적인 코드는 다양한 버전의 이종 임베디드 시스템에 초점이 맞춰져 있다[3] 첫 번째 V-모델은 타겟 독립적인 모델이고 가운데 모델은 타겟 명세적인 모델이다. 그리고 마지막 모델은 타겟 의존적인 코드다. 또한 이를 지원해주기 위해서는 자동화 도구가 필요하다. 기 개발된 자동화 도구를 사용하여 모델링뿐만 TDC에서 필요한 자바, C++, C 와 같은 형태의 소스코드를 자동으로 생성한다[3].

2. 본 론

기존의 개발 프로세스[3]뿐만 아니라 테스트 프로세스도 같이 수행하여 개발과 동시에 테스트를 수행하고자 이종 시스템을 위한 임베디드 테스트 프로세스를 정의와 개선한다. 개발 시 각각의 공정마다 단계, 활동, 작업, 산출물이 있으며 테스트 또한 개발프로세스와 동시에 활동 작업, 산출물이 존재한다.

* 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업(NIPA-2011-(C1090-1131-0008))과 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임.

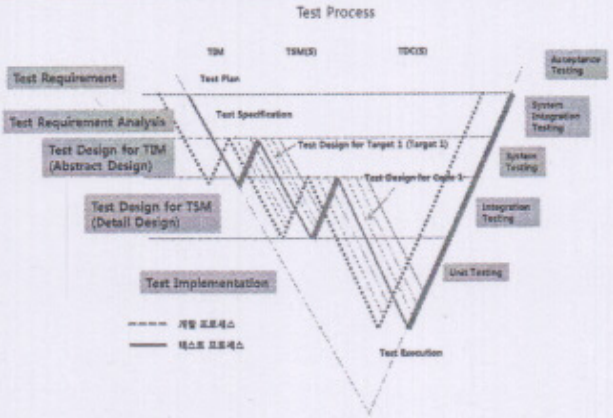

그림 1 재정의된 MDA 기반의 Multiple V-Model

2.1 Target Independent Model (TIM) 공정

타겟 독립적인 모델에서의 공정은 그림 2와 같다. 소프트웨어 개발 프로세스 활동과 테스트 프로세스 활동이 동시에 이루어지며 요구사항 단계에서 테스트 프로세스는 테스트 플랜을 작성한다. 이 단계에서는 구성된다. 테스트 분석 단계는 상위 테스트 명세활동을 하며 상위 테스트 명세 정의 및 이를 통해 상위 테스트 명세 정의문서를 작성한다. 테스트 도메인 단계에서는 상위 테스트 디자인 활동을 하며 상위 테스트 디자인 정의 작업을 하며 산출물로 상위 테스트 디자인 정의문서를 작성한다.

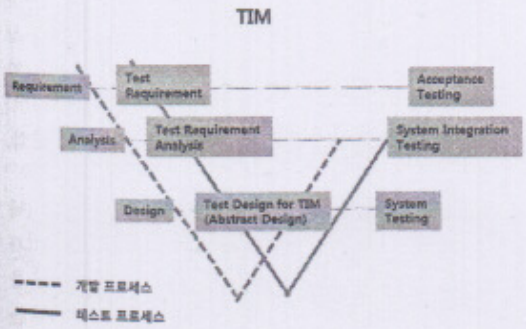

그림 2 TIM 공정 테스트 단계

실제 제품을 만드는 것이 아닌 타겟과 독립적인 상위의 공통 모델을 만들어서 향후 이를 통해 각 타겟 명세적인 모델로 변환하여 사용해야 하므로 각 타겟에서의 설계처럼 각 플랫폼에 관한 기술은 언급하지 않는다.
표 1은 TIM 공정내 테스트 단계에서 각 단계, 활동, 작업, 산출물을 보여준다. 크게 테스트 도메인정의, 분석, 설계 3단계이다. 테스트 도메인 정의는 상위의 메타 테스트 도메인을 정의하며 테스트 요구사항과 테스트 플랜으로 나뉜다. 테스트 요구사항에서는 상위의 테스트 요구사항을 기술하며 이를 통해 상위 테스트 요구사항 정의 문서를 작성한다. 테스트 플랜 활동은 테스트 플랜정의 작업과 이를 통해 작성되는 테스트 플랜 정의문서로

표 1 TIM 공정 테스트 활동과 산출물

공정	단계	활동	작업	산출물
Target Independent Model	테스트 도메인 정의	테스트 요구사항 정의	상위 테스트 요구사항 정의	상위 테스트 요구사항 정의문서
		테스트 플랜	테스트 플랜 정의	테스트 플랜 정의문서
	테스트 분석	상위 테스트 명세	상위 테스트 명세 정의	상위 테스트 명세 정의문서
	테스트 도메인 디자인	상위 테스트 디자인	상위 테스트 디자인 정의	상위 테스트 디자인 정의문서

2.2 Target Specific Model (TSM)공정

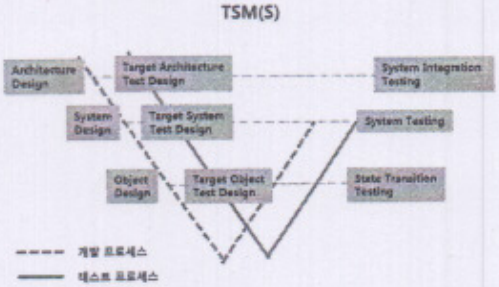

그림 3 TSM 공정 테스트 단계

그림 3은 TSM 공정을 보여준다. TSM 공정에서는 앞선 TIM공정하고 다르게 타겟위주의 모델을 보여준다. 그리고 각 분석, 디자인, 설계 전 단계에 걸쳐 각각의 타겟에 맞는 활동을 하는 공정이다. 이 공정에서 테스트 프로세스는 각 아키텍처와 시스템 오브젝트의 테스트 케이스를 목적으로 한다.

표 2 TSM 공정 테스트 활동과 산출물

공정	단계	활동	작업	산출물
Target Specific Model	타겟 아키텍처 테스트 디자인	타겟 아키텍처 테스트 디자인 정의	타겟 아키텍처 테스트 케이스 정의	타겟 아키텍처 테스트 케이스
	타겟 시스템 테스트 디자인	타겟 시스템 테스트 디자인 정의	타겟 시스템 테스트 케이스 정의	타겟 시스템 테스트 케이스
	타겟 오브젝트 테스트 디자인	타겟 오브젝트 테스트 디자인 정의	타겟 오브젝트 테스트 케이스 정의	타겟 오브젝트 테스트 케이스

2.3 TDC(Target Dependent Code) 공정

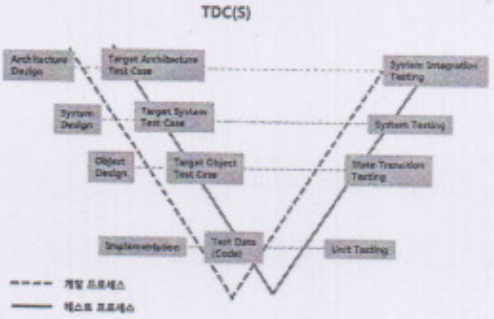

그림 4 TDC 공정 테스트 단계

그림 4는 TDC공정에서는 전체 4단계로 나뉜다. 이 공정에서는 상위의 TSM을 통해서 최종적으로 각 타겟에 맞는 코드를 발생한다. 아키텍처 테스트 디자인을 통해 발생된 테스트 케이스를 사용하여 모델 통합 테스트를 수행하며 시스템 테스트 디자인을 통해 발생된 테스트 케이스를 사용하여 시스템 테스트를 한다. 그리고 오브젝트 테스트 디자인을 통해서 발생된 테스트 케이스를 사용하여 스테이트 트랜지션 테스트를 수행하고 테스트 데이터를 통해 단위 테스트를 수행한다. 개발 단계에서 이 공정은 타겟 명세적인 모델을 통해서 타겟에 맞는 코드를 만든다. 이 때 생성된 코드를 테스트 할 때 각 타겟에 맞는 아키텍처 및 시스템, 오브젝트에 대한 테스트가 필요하며 통합 테스트 또한 필요하다. 표 3은 각 단계, 활동, 작업, 산출물을 보여준다.

표 3 TDC 공정 테스트 활동과 산출물

공정	단계	활동	작업	산출물
Target Dependent Code	타겟 아키텍처 테스트	타겟 아키텍처 테스트 코드 생성	타겟 아키텍처 테스트 실행	타겟 아키텍처 테스트 결과 문서
	타겟 시스템 테스트	타겟 시스템 테스트 코드 생성	타겟 시스템 테스트 실행	타겟 시스템 테스트 결과 문서
	타겟 오브젝트 테스트	타겟 오브젝트 테스트 코드 생성	타겟 오브젝트 테스트 실행	타겟 오브젝트 테스트 결과 문서
	타겟 통합 테스트	타겟 통합 테스트 코드 생성	타겟 통합 테스트 실행	타겟 통합 테스트 결과 문서

3. 결론

이 논문은 임베디드 시스템 상에서의 소프트웨어의 플랫폼이 빠르고 다양하게 발전됨에 따라 임베디드 소프트웨어의 개발 및 테스트가 중요해졌다 그러나 모든 개발자들이 각각의 타겟을 개발하며 테스트 하기는 쉽지 않다. 그래서 기존 소프트웨어공학에서 사용하는 다양한 플랫폼의 개발 기법인 Model Driven Architecture (MDA)를 적용한 기존 개발 프로세스의 공정, 단계, 활동, 작업, 산출물을 테스트 단계에 맞게 정의하였다.

이를 통해 다양해지는 임베디드 환경에서의 개발과 동시에 테스트를 진행하여 소프트웨어 개발 시간을 단축하고자 한다. 향후 연구로는 각 임베디드 환경에서의 각 모델들을 변환하는 도구가 매우 중요하기 때문에 이를 지원해주는 도구 또한 개선하여 발전해야 하며 테스트 각 단계별 단계, 활동, 작업 및 산출물을 사례 연구를 통해 발전시킬 것이다.

참고문헌

[1] John Wakins, "Testing IT" , Cambridge
[2]A. Kleppe, J.Warmer, W.Bast, MDA Explained: The Model Driven Architecture: Practice and Promise, Addison-Wiseley, 2003.
[3] Woo Yeol Kim, Hyun Seung Son, Y. B. Park, B. H. Park, C. R. Carlson, R. Young Chul Kim, "Automatic MDA (Model Driven Architecture) Transformations for Heterogeneous Embedded Systems" SERP'08(2008 International Conference on Software Engineering Research and Practice),LV Nevada, USA