



한국정보처리학회

12권 제1호  
Vol. 12 No. 1



# 2010

## 한국 소프트웨어공학 학술대회 논문집

Proceedings of 2010 Korea Conference on  
Software Engineering

- 일시 : 2010년 2월 8일(월)~10일(수)
- 장소 : 한화 휘닉스 파크(강원도 평창)

주최 : 한국정보과학회, 한국정보처리학회

주관 : 한국정보과학회 소프트웨어공학 소사이어티  
한국정보처리학회 소프트웨어공학 연구회  
정보통신산업진흥원 부설 소프트웨어공학센터  
한국 전자 통신 연구 원 ( E T R I )

후원 : KAIST 프로세스 개선센터(SPIC), 고려대학교 고신뢰 융합  
소프트웨어공학센터(CCEEDS), 단국대학교 금융IT를 위한  
소프트웨어공학연구센터(SERC-FIT), (주)모아소프트, 서강  
대학교 SW 요구 및 검증공학 센터(ReVeT), 포항공과대학교  
융합소프트웨어개발 연구센터(COSDEC), 숭실대학교 모바일  
서비스 소프트웨어공학센터(MSSEC)

2. SKF-학습 기반의 잠수함 추적 및 위치추정 알고리즘 연구 김민규, 김제영, 윤희병(국방대) .....	450
3. 자가 치유 시스템을 위한 치유 전략 수립 자동화 이상도, 유길종, 이은석(성균관대) .....	457
4. 자율주행 로봇의 경로설정을 위한 온톨로지 황선명(대전대), 김행곤(대구카톨릭대), 김상수(대전대) .....	465
5. 모니터링을 통한 EPC Network 장애 감시 기법 연구 손민영, 박성진, 윤상현, 박준석, 염근혁(부산대) .....	471
6. 모바일 어플리케이션 GUI 자동 생성 기법 류성태, 이은석(성균관대) .....	478
7. Ontology와 Clustering을 이용한 IPTV의 개인화된 프로그램 추천 시스템 김종우, 김주완, 윤병대, 이슬기, 강상길(인하대) .....	486
8. 광역지형 렌더링을 위한 방법에 관한 연구 정지환, 이재영(픽소니어), 황선명(대전대) 김성호, 채희만(국방과학연구소) .....	493

## 임베디드 SW

1. 임베디드 소프트웨어에서 코드 리팩토링과 CBMC 모델검증 도구의 적용 김성희, 권기현(경기대), 한혁수(상명대), 이상은, 이혁재, 남일규(NIPA) 한주동(엠디플렉스) .....	499
2. 실시간 복합 이벤트 모니터링을 이용한 로봇 프로그램의 모델 주도 개발 이기성, 이찬근(중앙대) .....	506
3. 이종 임베디드 소프트웨어를 위한 테스트 프로세스 개선 연구 김우열, 손현승, 김영철(홍익대), 김재승(트라이콘) .....	513
4. 모바일 인터넷 디바이스(MID)용 Cloud 서비스 개발을 위한 아키텍처 패턴 김지원, 이정우, 김수동(숭실대) .....	516



## 이종 임베디드 소프트웨어를 위한 테스트 프로세스 개선 연구

김우열\*, 손현승\*, 김재승\*\*, 김영철\*

홍익대학교 일반대학원 소프트웨어공학 연구실\*

트라이콘\*\*

{john, son, bob}@selab.hongik.ac.kr\*

ceo@tricon.co.kr\*\*

**요약:** 이종의 임베디드 소프트웨어 개발 프로세스에서는 하드웨어 플랫폼 별로 설계 모델을 생성한다. 여기에 기존의 테스트 프로세스를 적용하면 설계 모델마다 테스트 케이스를 생성해야 되므로 불필요하게 많은 수의 테스트 케이스가 발생되고 테스트 프로세스 절차가 중복적으로 사용된다. 이러한 문제들은 테스트 비용과 시간을 증가시켜 테스트를 어렵게 한다. 본 논문에서는 기존의 문제를 해결하기 위해, 이종의 임베디드 소프트웨어를 위한 테스트 프로세스를 제안한다. 제안한 프로세스는 이종의 개발 프로세스에 테스트 프로세스를 접목하여 효율적인 테스트를 수행할 수 있게 한다.

**핵심어:** 이종 임베디드 시스템, 테스트 프로세스, 테스트 케이스 레벨화, MDD(Model Driven Development)

### 1. 서론

최근의 임베디드 시스템은 정보 가전, 정보 단말 등은 물론이고 산업제어기기, 로봇, 사무자동화, 빌딩 자동화, 산업 자동화, 군사, 통신, 물류/금융, 자동차/운송장비, 의료, 게임, 항공 관제 등 아주 폭넓고 다양하게 사용된다[1]. 이러한 임베디드 시스템의 환경은 네트워크의 발달과 하드웨어가 다행해져 단일 시스템에서 이종 시스템으로 변화해 가고 있다[2].

이종 임베디드 시스템의 소프트웨어는 재사용을 하기 어렵다[3]. 그래서 이종 임베디드 시스템에 MDA(Model Driven Architecture)를 적용한 개발 프로세스를 제안하였다[4,5,6]. 이 개발 프로세스는 개발위주의 방법으로 모델부터 코드까지 자동으로 발생시켜준다. 하지만 생성된 코드에 대한 품질을 보장하지는 않는다.

여기에 기존의 테스트 프로세스를 적용하면 설계 모델마다 테스트 케이스를 생성해야 되므로 불필요하게 많은 수의 테스트 케이스가 발생되고 테스트

프로세스 절차가 중복적으로 사용된다. 이러한 문제들은 테스트 비용과 시간을 증가시켜 테스트를 어렵게 한다.

본 논문에서는 앞에서 언급한 문제를 해결하기 위해서는 개선한 테스트 프로세스를 제안한다. 제안한 프로세스는 다음과 같은 특징을 갖는다. 첫 번째로 개발 프로세스에서 사용된 모델을 테스트 프로세스에서 사용하여 별도의 테스트 설계를 사용하지 않도록 한다. 두 번째로 개발프로세스와 테스트 프로세스를 동시에 수행할 수 있도록 한다. 세 번째로 테스트 케이스 레벨화를 통해서 테스트 케이스를 통합할 수 있다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련연구로서 이종 임베디드 시스템 개발 프로세스와 기존의 테스트 프로세스 프로세스에 대해서 기술한다. 3 장에서는 제안한 테스트 프로세스를 설명한다. 마지막으로 4 장에서는 결론 및 향후 연구에 대해서 언급한다.

### 2. 관련 연구

#### 2.1 이종 임베디드 시스템 개발 프로세스

기존 임베디드 S/W 개발 프로세스는 Model, Prototype, Product 순으로 개발한다. 이 방법은 반복적이고 점진적이다. 이렇게 하여 하드웨어에 소프트웨어를 적용하면서 개발할 수 있고 새로운 시스템을 개발할 때 좋은 방법이다. 그러나 이 방법은 단일 시스템에만 적용할 수 있는 단점이 있다. 왜냐하면 Model 을 만들고 여기에 대한 Prototype 을 만들 때 이미 소프트웨어는 하드웨어의 종속성을 갖기 때문이다. 결국 이 방법은 이종의 시스템을 개발과 소프트웨어의 재사용이 어렵다.

그래서 기존 개발 프로세스에 이종의 시스템 개발을 위해서 MDD 를 접목하였다. 이 방법은 하드웨어

\* 본 연구는 교육과학기술부와 한국산업기술진흥원의 지역혁신인력양성사업으로 수행된 연구결과임.



의존코드를 독립 모델과 종속 모델로 분리하여 독립 모델을 재사용 가능하도록 하고 종속모델을 자동 생성하여 줌으로 복잡한 이중시스템 개발을 쉽고 빠르게 수행해 준다. 이 개발방법은 이중 임베디드 시스템 개발이 가능하고 하드웨어에 관계없이 소프트웨어를 재사용할 수 있다.

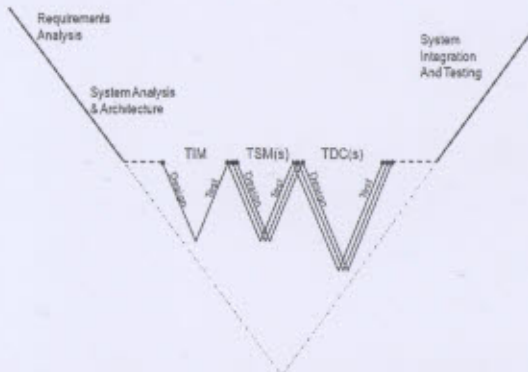


그림 1 이중 임베디드 시스템을 위한 개발 프로세스

그림 1은 이중 임베디드 시스템을 위한 개발 프로세스이다. 개발 프로세스는 요구사항 분석, 시스템 분석, TIM(Target Independent Model), TSM(Target Specific Model), TDC(Target Dependent Code), 시스템 테스트 순으로 수행된다. 하나의 TIM은 여러 개의 TSM을 생성할 수 있고 TSM 또한 여러 개의 TDC를 생성할 수 있습니다. 또한 이 방법은 자동화 도구를 통해서 TIM을 TSM으로 TSM을 TDC로 만들어준다.

기존의 이중 임베디드 개발 프로세스는 개발위주의 방법으로 모델부터 코드까지 자동으로 발생시켜 주지만 생성된 코드에 대한 품질을 보장하지는 않는다. 또한 기존의 테스트 프로세스는 V 모델에 적용되어 있어 개발 프로세스와의 테스트 프로세스를 동시에 수행하기 어렵다.

## 2.2 기존의 테스트 프로세스

기존의 일반적인 테스트 프로세스는 테스트 분석과 설계 단계, 테스트 구현과 실행, 테스트 완료 평가 및 리포트, 테스트 마감활동 5 단계로 구성되어 있다. 테스트 분석과 설계 단계는 테스트 베이스 검토와 상황/요구사항/데이터 식별, 기법의 할당, 용이성 평가, 환경 구축들을 수행한다[7]. 테스트 구현과 실행 단계에서는 테스트 케이스의 명세화와 우선 순위 선정, 테스트 데이터 생성, 프로시저 작성을 하고 테스트 실행과 기대 결과를 비교한다. 테스트 완료 평가 및 리포트는 테스트 구현과 실행에서 생성된 내용을 바탕으로 테스트 보고서를 작성한다. 테스트 마감활동은 테스트 프로세스의 평가를 한다. 이러한 테스트 프로세스는 개발 프로세스와는 독립적으로 수행한다.

V-모델에 적용된 테스트는 크게 나누면 개발 프로세스, 테스트 프로세스, 테스트 레벨로 나눌 수 있다. 개발 프로세스는 요구사항 분석, 설계, 구현 순으로 수행된다. 테스트 레벨은 인수테스트, 시스템테스트, 통합 테스트, 단위 테스트로 구성되어 있다. 여기서 테스트 프로세스는 테스트 레벨에 따라 각각 따로 수행된다. 그렇기 때문에 기존의 테스트 프로세스는 테스트 레벨마다 테스트 분석과 설계 단계, 테스트 구현과 실행, 테스트 완료 평가 및 리포트, 테스트 마감활동을 계속 수행해야 한다. 그림 2를 살펴보면 점선의 사각형처럼 각 테스트 레벨마다 각각 테스트 프로세스가 별도로 수행되는 것을 볼 수 있다.

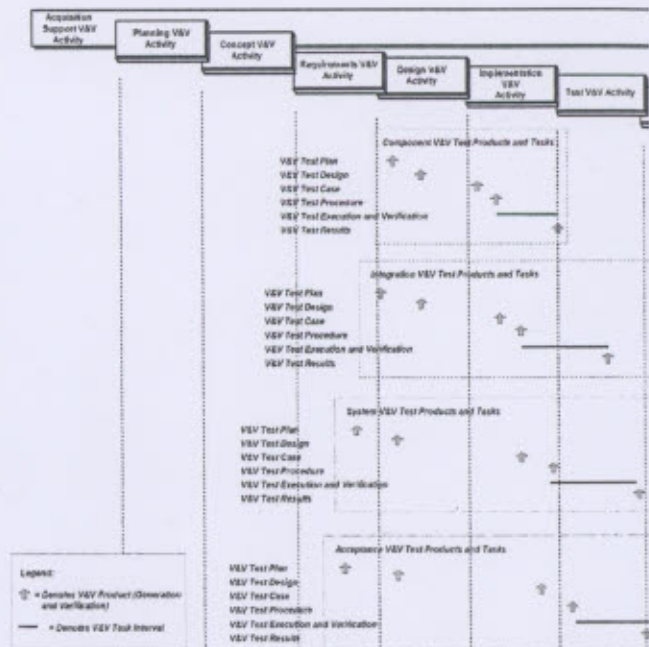


그림 2 V&V의 테스트 프로세스 (IEEE 1012-2004)[8]

기존의 테스트 프로세스의 문제점을 종합해 보면 개발 프로세스와 테스트 프로세스가 따로 수행되기 때문에 개발 산출물을 통해서 테스트 설계를 다시 해야 한다. 또한 테스트 레벨에 따라서 테스트 프로세스가 반복적으로 수행되어 테스트 프로세스의 불필요한 중복이 생긴다.

## 3. 제안한 테스트 프로세스

개발 프로세스에서 생성되는 산출물은 UML을 이용하여 표현한다. 이 때 생성되는 UML은 시스템의 레벨에 따라서 계층화 될 수 있고 각 다이어그램들 간에 수평적인 관계를 형성하게 된다. 테스트 케이스는 이러한 UML의 모델 특징을 유기적으로 관계를 가질 수 있도록 테스트 프로세스가 제안되어야 한다.

제안한 프로세스는 개발프로세스가 수행됨과 동시에 제안한 테스트 프로세스가 동시에 수행한다. 각



단계는 테스트 계획(Test Plan), 테스트 명세(Test Specification), 테스트 설계(Test Design), 테스트 소스(Test Source), 테스트 레벨화 및 우선순위화(Test Levelization & Prioritization), 테스트 케이스 수행(Test Case Execution) 순으로 수행된다.

각 세부 절차에 대해서 살펴보면 다음과 같다.

#### 테스트 계획(Test plan)

테스트 활동에 있어서 포함되는 테스트 범위, 방법, 자원, 스케줄들을 문서로 기술한다. 테스트 계획은 테스트 계획 식별, 테스트 아이템, 테스트 해야 할 사항, 테스트 하지 말아야 할 사항, 방법, 테스트 실패 기준, 테스트 임무, 테스트 요구 환경, 의무, 요구되는 팀원과 교육, 스케줄, 위험성과 유연성에 대해서술한다.

#### 테스트 명세(Test Specification)

테스트 명세로부터 테스트 케이스를 끌어내는 표준화된 방법이다.

#### 테스트 설계(Test Design)

소프트웨어의 단일 특징이나 복합된 특징 그리고 관련된 테스트를 식별하는 것을 상세히 기술하여 문서화한다. 간단히 설계단계에서 이루어진 문서들을 검증하고 테스트 활동에 대해 구체적인 활동을 문서화 하는 단계이다. TIM 레벨의 테스트 설계를 생성하고 TSM 레벨의 테스트 설계를 한다. TIM 은 하드웨어에 독립된 모델로 추상화 단계에서 테스트 케이스를 생성한다. TSM 은 하드웨어에 종속적인 모델로 구체화된 테스트 케이스를 생성한다.

#### 테스트 소스(Test Source)

개발 프로세스에서 실제 코딩이 이루어지는 단계로 이 단계에서는 테스트에 필요한 데이터와 코드, 혹은 테스트 스텝 등이 실제 생성되는 단계이다.

#### 테스트 레벨화 및 우선순위화(Test Levelization & Prioritization)

각 개발 프로세스 단계에서 생성한 산출물을 통해서 시스템 레벨에 맞도록 테스트 케이스를 분류하는 작업을 수행하는 단계이다. 또한 Goal, Value, Risk 등을 적용하여 테스트 케이스를 우선 순위화 한다.

#### 테스트 케이스 수행(Test Case Execution)

레벨화된 테스트 케이스를 수행하는 단계이다. 수행방법은 시스템 레벨을 먼저 수행하고 문제가 있는 서브-시스템들의 하부 레벨인 컴포넌트, 클래스, 오브젝트의 테스트 케이스를 수행한다. 이렇게 하여 레벨 별 테스트 케이스(시스템, 컴포넌트, 클래스, 오브젝트)를 레벨 별로 수행한다.

#### 4. 결론

본 논문에서는 기존의 문제를 해결하기 위해 이중의 임베디드 소프트웨어를 위한 테스트 프로세스를 제안하였다. 제안 테스트 프로세스는 개발 프로세스와 테스트 프로세스가 동시에 병렬적으로 수행되도록 개발 프로세스와 테스트 프로세스를 접목 하였다. 또한 개발 프로세스에서 생성된 모델을 테스트 프로세스에서 재사용하여 테스트 설계 단계의 노력을 줄였다. 그리고 테스트 케이스 레벨화를 하여 시스템 레벨간에 테스트를 통합하여 테스트 케이스 수행시에 시간 및 비용 절감할 수 있도록 하였다.

향후 연구로는 UML 모델에서 테스트 케이스가 자동으로 생성될 수 있는 방법을 연구 중이다. 또한 Goal, Risk, Value 를 이용한 테스트 케이스 우선순위화로 효율적으로 테스트 케이스를 생성할 수 있도록 연구 중이다.

#### 참고문헌

- [1] 김홍남, 박승민, 김두현, "임베디드 소프트웨어 최근 기술 동향", 정보과학회지, 제 24 권, 제 8 호, pp. 5-11, 2006. 8.
- [2] 김현, 이강우, 이주행, 강태근, 문애경, 서영호, 조준면, "URC 에서의 소프트웨어 로봇 기술", 한국통신학회지(정보통신), 제 21 권, 10 호, pp. 36-43, 2004. 10.
- [3] Axel jantsch, *Modeling Embedded System and SOCs*, Mogan Kaufmann, 2004.
- [4] 손현승, 김우열, 김영철, "이중 임베디드 시스템의 멀티태스킹을 위한 MDA(Model Driven Architecture) 기반의 설계", 한국정보처리학회논문지, Vol. 15-D, No. 3, pp. 355-360, 2008. 06.
- [5] Woo Yeol Kim, Hyun Seung Son, Young B. Park, Byung H. Park, C. R. Carlson, R. Young Chul Kim, "The Automatic MDA(Model Driven Architecture) Transformations for Heterogeneous Embedded Systems " SERP'08, Vol. 2, pp. 409-414, 2008. 07. 14.
- [6] Woo Yeol Kim, Hyun Seung Son, R. Young Chul Kim, C. R. Carlson, "MDD based CASE Tool for Modeling Heterogeneous Multi-Jointed Robots", CSIE 2009, Vol. 7, pp. 775-779, 2009. 04. 01.
- [7] 권원일, 박용영, 이현주, 조현길, *개발자도 알아야할 소프트웨어 테스팅 실무*, STA, 2008.
- [8] IEEE Std 1012-2004, IEEE Standard for Software Verification and Validation (Revision of IEEE Std 1012-1998), 2004.