

제19권 제1호

ISSN 2005-0011





제37회 춘계학술발표대회 논문집(하)

일자: 2012년 4월 26일(목)~28일(토)

장소: 순천대학교 70주년 기념관

주최: 사단법인 한국정보처리학회

후원:  지식경제부  2012 여수세계박람회
Ministry of Knowledge Economy EXPO 2012 YEOSU KOREA

협찬: 롯데정보통신, 삼성SDS, SK C&C, 정보통신산업진흥원,
한국게임과학고등학교, 이투스앤자루, 커미트, 한국생산성본부,
한솔인티큐브, 나라인포테크, 마크애니, 동하테크, LG엔시스 (무순)

春

 한국정보처리학회
Korean Information Processing Society

347. 효과적인 SCL 엔지니어링 틀 설계를 위한 관련 S/W 틀 분석 KIPS_C2012A_0312
 채창훈*, 정남준, 최효열, 안용호(한국전력공사) • 1215
348. LTL Synthesis를 이용한 다중 로봇 시뮬레이터 개발 KIPS_C2012A_0313
 김성희*, 권령구, 권기현(경기대학교) • 1219
349. 앱개발 도구 : HTML5, App Inventor, M-BizMaker 어느 것을 선택할 것인가 KIPS_C2012A_0318
 김시우*(숭의여자대학교), 전정훈(동덕여자대학교) • 1223
350. 실시간 GPS 좌표추적을 이용한 성범죄자 추적 및 알림 어플리케이션 KIPS_C2012A_0319
 이동성, 김정윤, 황선명*(대전대학교) • 1226
351. 달빅 DEX 파일 브라우저의 설계 및 구현 KIPS_C2012A_0334
 소경영*, 정택희(전북대학교), 박종필, 고헌만(상지대학교) • 1228
352. 안드로이드 애플리케이션 GUI 테스트 도구 적용 및 사례연구 KIPS_C2012A_0342
 김태균*, 권기현(경기대학교) • 1231
353. 결합 위치 추적을 위한 테스트 케이스 자동 생성 기법 KIPS_C2012A_0346
 박창용*, 김준희, 류성태, 윤현상, 이은석(성균관대학교) • 1235
354. 이종 임베디드 테스트를 위한 MDA (Model Driven Architecture)기반의 테스트 프로세스 개선 및
 확장에 관한 연구 KIPS_C2012A_0358
 김동호*, 손현승, 김우열, 김영철(홍익대학교) • 1239
355. 테스트 프로세스 개선 모델(TPI next)을 통한 테스트 성숙도 모델 확장에 관한 연구
 KIPS_C2012A_0361
 김기두*(한국정보통신기술협회), 김영철(홍익대학교) • 1243
356. 사용자 니즈를 통한 사용자 선호도 요구사항 추출 및 우선순위화 KIPS_C2012A_0364
 박보경*, 김영철(홍익대학교) • 1247
357. 원인-결과 다이어그램과 접목을 위한 메시지-순차적 다이어그램 확장 연구 KIPS_C2012A_0365
 우수정*, 손현승, 김영철(홍익대학교) • 1251
358. 클라우드 컴퓨팅에서 BPEL분석 및 검증을 위한 Onion언어로의 변환 KIPS_C2012A_0376
 최재홍*, 온진호, 이문근(전북대학교) • 1255
359. 비즈니스 프로세스 프레임워크상에서의 비즈니스 프로세스 모델, 서비스와 컴포넌트기반
 개발의 매핑을 통한 소프트웨어 재사용 패러다임 KIPS_C2012A_0377
 서채연*, 문소영, 김영철(홍익대학교) • 1259
360. 정보시스템감리와 회계감사의 적정성 비교 KIPS_C2012A_0431
 권호열*(강원대학교) • 1262
361. 공공부문 정보화사업 PMO 도입의 과제 KIPS_C2012A_0432
 권호열*(강원대학교) • 1264

정보저리용용(IT교육 등)

362. 항만건설공사 전자설계·준공도서 서비스 시스템 개발 KIPS_C2012A_0008
 정성윤*, 김남곤(한국건설기술연구원) • 1269
363. 도로현황조사시스템 구축방안 연구 KIPS_C2012A_0011
 김영진*, 김병곤, 임재규(한국건설기술연구원) • 1271
364. 정량적 기고서 분석을 통한 MPEG 표준화 과정 연구 KIPS_C2012A_0016
 이광훈*, 김현규, 장의선(한양대학교) • 1275
365. 매쉬업을 위한 Open API 유사성 탐색 방법 KIPS_C2012A_0019
 이용주*(경북대학교) • 1279

이종 임베디드 테스트를 위한 MDA (Model Driven Architecture)기반의 테스트 프로세스 개선 및 확장에 관한 연구

김동호, 손현승, 김우열, 김영철
홍익대학교 컴퓨터정보통신공학과
e-mail : ray@selab.hongik.ac.kr

Test Process Improvement and Extension Based On Model Driven Architecture(MDA) For Heterogeneous Embedded Testing

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

요 약

현재 소비자의 요구에 따라 다양한 타겟 상에서 임베디드 소프트웨어 개발이 폭주되고 있다. 같은 서비스를 제공하는 어플리케이션을 다양한 플랫폼에 맞게 개발하려면 많은 시간과 비용이 소모된다. 또한 이를 위한 테스트 비용도 증가하게 된다. 이는 테스트 비용의 지출이 전체 개발비용에 막대한 영향을 미친다. 그래서 다양한 플랫폼 상에서의 테스트 비용을 감소하기 위해 기존 소프트웨어공학 기법 중 하나인 Model Driven Architecture (MDA)를 적용한 기존 임베디드 개발기법에 테스트 프로세스를 개선 및 강화할 것을 제안한다[1][2]. 또한 다양한 타겟에 맞는 이종 테스트케이스 개발에 밑거름이 될 것이다.

1. 서론

최근 임베디드 소프트웨어 시장은 다양한 환경과 기기상에서 제품이 출시되고 있다. 이에 따라 개발사들은 각 플랫폼 별로 제품을 출시해야 하는 문제가 있다. 개발뿐만 아니라 테스트 또한 각 플랫폼 별로 해야 하기 때문에 많은 비용과 시간을 소모하게 된다. 그래서 본 그래서 공통의 상위 모델을 만들고 이를 기반으로 다양한 플랫폼을 지원할 수 있는 MDA에 관한 필요가 높아지고 있다. 더군다나 최근 등장한 다양한 형태의 모바일 플랫폼상에서의 빠르고 안정적인 개발은 큰 장점이 될 수 있다. 기존에 제안한 MDA기반의 개발모델은 크게 3부분으로 나뉜다. 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 또한 이를

지원해주기 위해서는 자동화 도구가 필요하다. 기 개발된 자동화 도구를 사용하여 모델링뿐만 TDC에서 필요한 자바, C++, C 와 같은 형태의 소스코드를 자동으로 생성한다. 이를 통해 다양한 환경과 플랫폼상에서의 임베디드 소프트웨어 개발에 유리하다 그러나 개발중심의 방법론이기 때문에 테스트 부분은 취약하다. 그래서 테스트 프로세스를 보강한 방법을 기존에 제안했다. 그러나 기존 모델 기반의 개발 방법과 테스트 프로세스의 접목에서 수많은 가변성을 플랫폼으로 재 정의하고 이를 기반으로 하는 이종 임베디드 테스트를 위한 MDA관점의 테스트 및 각 단계별 활동, 작업, 산출물 등을 재정의하였다[2][3].

2. 관련연구

2.1 MDA 기반의 개발 방법

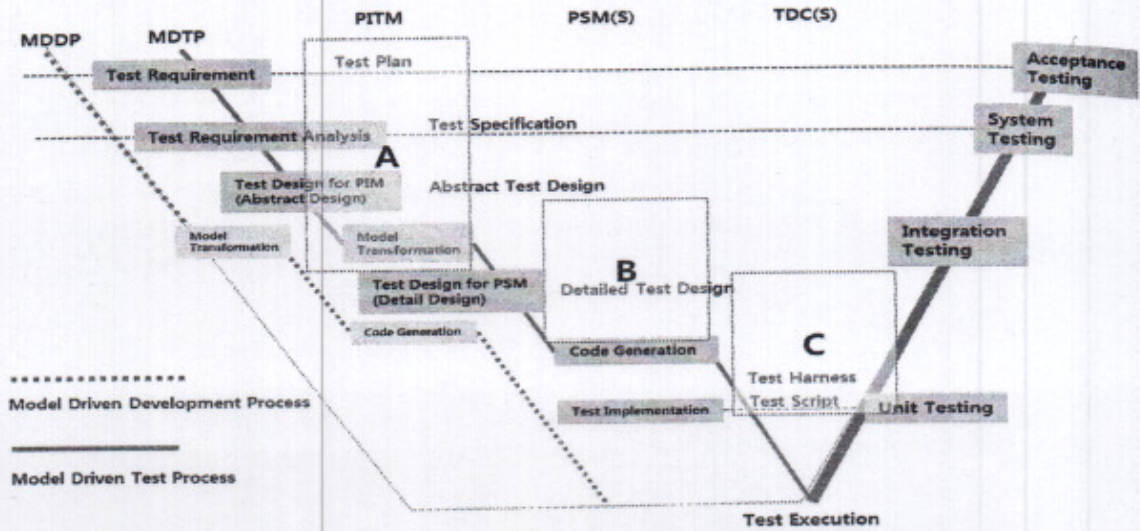
기존 임베디드 소프트웨어 개발 프로세스는 모델, 프로토타입, 프로덕트 순으로 개발한다[4]. 이 방법은 반복적이고 점진적이다. 이렇게 하여 하드웨어에

* 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업(NIPA-2012-(H0301-12-3004))과 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2011-0004203)으로 수행된 연구결과임.

소프트웨어를 적용하면서 개발할 수 있고 새로운 시스템을 개발할 때 좋은 방법이다. 그러나 이 방법은 단일시스템에만 적용할 수 있는 단점이 있다. 왜냐하면 모델을 만들고 여기에 대한 프로토타입을 만들 때

이미 소프트웨어는 하드웨어에 종속성을 갖기 때문이다. 기존 방법은 이종의 시스템의 개발과 소프트웨어의 재사용이 어렵다. 결국 MDA 기법은 이런 문제가 해결이 가능하다[2][3].

Model Driven Development and Test Process



(그림 1) 모델 기반 개발 및 테스트 프로세스

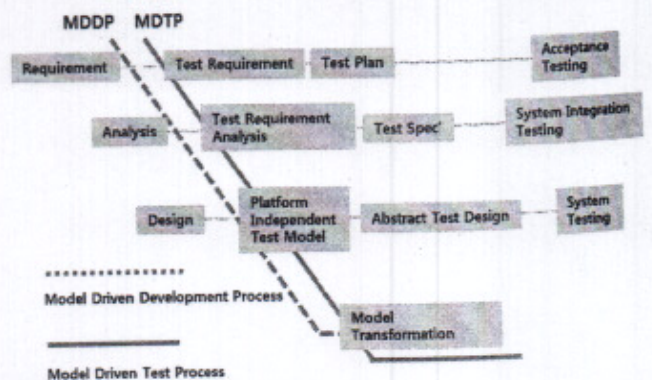
그래서 기존 개발 프로세스에 이종의 시스템 개발을 위해서 MDD 를 접목하였다. 이 방법은 하드웨어에 의존되는 코드를 독립모델과 종속모델로 분리하여 독립모델을 재사용 가능하도록 하고 종속모델을 자동 생성하여 줌으로 복잡한 이종시스템 개발을 쉽고 빠르게 수행해 준다. 이 개발방법은 이종 임베디드 시스템 개발이 가능하고 하드웨어에 종속적이지 않는 소프트웨어 재사용이 가능하다[1][2][5]

그림 1 은 모델 기반 개발 및 테스트 프로세스의 전체적 흐름을 보여준다. 이종의 임베디드 환경에서의 개발과 테스트를 위해 기존 소프트웨어공학에서 다양한 플랫폼상에서의 개발을 도와주는 Model Driven Architecture (MDA)의 사상을 임베디드에 접목하였다. 각 각의 임베디드 개발 환경뿐만 아니라 테스트 환경은 매우 다르다. 플랫폼 독립적인 상위의 모델을 정의하고 이를 기반으로 자동화된 도구를 통하여 모델 변환을 하여 플랫폼 명세적인 모델을 만들고 이 모델을 기반으로 다시 각 임베디드 타겟에 맞는 코드를 생성하는 것이 MDA 기반의 개발 프로세스였다. 그러나 테스트에 대한 정의 및 프로세스가 취약하여 각 개발 프로세스에 대응되는 테스트 프로세스를 정의하였고 자세한 각 프로세스는 다음과 같다[6].

2.2 임베디드 테스트 프로세스

기존 임베디드 테스트에서는 V-모델을 기반에 둔 다중 V-모델을 사용하고 모델이 올바르다고 판단되면 모델로부터 코드를 생성 프로토타입에 임베디드 된다. 프로토타입의 실험용 하드웨어 시스템이 사용되고 최종 산출물이 나올 때까지 실제 하드웨어 제품으로 대체된다. 프로토타입을 사용하여 요구사항 변경을 적용하는 것이 최종 제품을 변경하는 것보다 더 값싸고 빠르기 때문이다. 그리고 테스트에 대한 언급도 있으나 대부분 단일 플랫폼 위주이다. 그래서 다양한 플랫폼상에서 다양한 제품 군을 지원하기에는 부족한 감이 있기 때문에 이런 문제를 해결하고자 이종 임베디드 환경에서의 테스트 프로세스가 필요하다. [4]

(A) Platform Independent Process



(그림 2) 플랫폼 독립적 프로세스

3. 개선된 이종 임베디드 프로세스

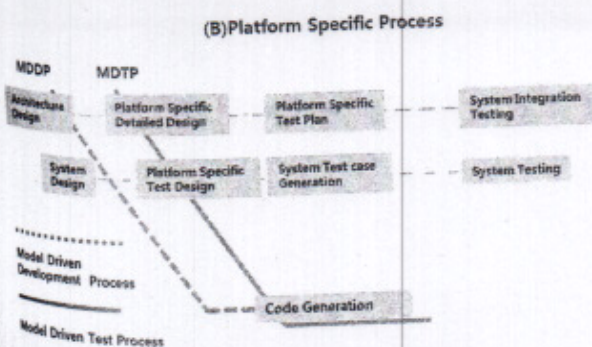
기존 임베디드 소프트웨어 개발 프로세스에 병렬적으로 테스트 프로세스를 접목을 제안 하였다. 이는 개발과 동시에 이에 대한 테스트 단계를 적용했다.

그림 2 은 플랫폼 독립적인 공정을 나타낸다 개발 프로세스와 테스트 프로세스는 병행하여 같이 수행하며 각 단계별 매핑되는 부분을 설명하면 다음과 같다. 개발단계에서 요구사항은 테스트 요구사항으로 매핑되며 이 데이터를 기반으로 테스트 플랜을 작성한다. 테스트 플랜에서는 플랫폼 독립적인 부분에 관하여 테스트 플랜을 정의하고 문서를 작성한다. 분석단계는 테스트 요구사항 분석단계로 매핑되며 테스트 스펙을 작성하고 문서화 한다. 디자인 단계는 플랫폼 독립적인 테스트 모델을 정의하는 단계로 플랫폼에 독립적이고 공통적인 부분을 추출하여 이 데이터를 기반으로 플랫폼 독립적인 테스트 디자인을 하며 이를 문서화 한다.

<표 1> 플랫폼 독립적 테스트 모델

공정	단계	활동	작업	산출물
플랫폼 독립적 테스트 모델 (FITM)	테스트 도메인 정의 (FD)	테스트 요구사항 정의 (TRD)	테스트 플랜 정의 (TFD)	테스트 플랜 정의서 (TFDD)
	테스트 도메인 분석 (A)	테스트 요구사항 분석 (TRA)	테스트 명세 (TS)	테스트 명세 정의서 (TSDD)
	테스트 도메인 설계 (D)	플랫폼 독립적인 추상화 테스트 모델 (FIATM)	플랫폼 독립적인 추상화 테스트 디자인 (FIATD)	플랫폼 독립적인 모델의 테스트 설계서 (FITD)

표 1 은 플랫폼 독립단계에서의 활동과 작업 산출물을 정의한다. 크게 3 부분으로 테스트 도메인정의, 테스트 도메인 분석, 테스트 도메인 설계로 나뉜다. 테스트 도메인정의는 테스트 요구사항을 정의하고 이를 기반으로 테스트 플랜을 정의한다. 이를 바탕으로 테스트 플랜을 정의한다. 테스트 도메인 분석은 테스트 요구사항을 분석하고 이를 통해 테스트 명세를 만들고 문서화한다. 테스트 도메인 설계는 플랫폼 독립적인 테스트 모델을 정의하고 디자인하고 이를 문서화하는데 테스트 도메인 설계부분이 이중의 임베디드 개발 환경의 중요한 부분이 된다. 기존 MDD 개발방법이 플랫폼의 독립적인 상위의 모델을 기반으로 모델 변환을 거쳐 플랫폼 명세적인 모델로 변환하는데 이때 플랫폼 독립적인 모델을 기반으로 각 플랫폼으로 변환하기 때문에 최대한 플랫폼에 명세적이지 않으면서 공통적인 부분을 정의하고 디자인하여야 한다.



(그림 1) 플랫폼 명세적 프로세스

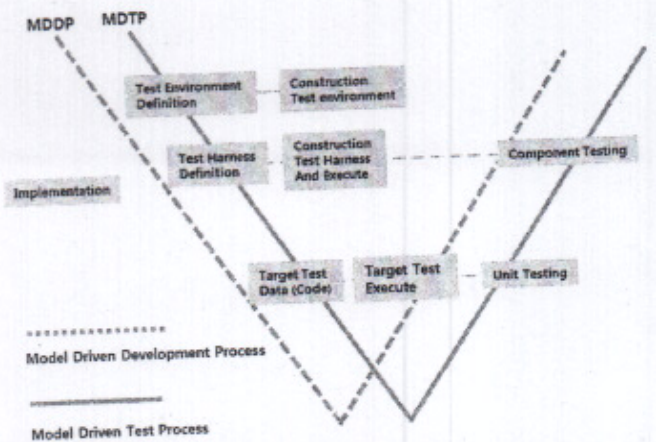
그림 3 은 플랫폼 명세 프로세스를 나타낸다. 이때 상위의 플랫폼 독립적인 부분에서 하위의 각 플랫폼에 맞는 부분을 모델링하고 정의하고 디자인하는 단계로 상위의 공통적인 모델에서 모델변환을 거쳐 플랫폼 명세적인 부분으로 바뀌게 되는데 이때 각 플랫폼 별로 다른 특징을 갖기 때문에 특징에 맞게 변환하는 것이 중요하다. OMG 에서는 이를 위해 가급적 자동화된 틀을 권장한다[2]. 각 단계를 살펴보면 각 플랫폼 별로 아키텍처 및 시스템과 오브젝트들이 다르기 때문에 이에 맞춰서 모델이 변경되고 테스트 또한 이 구조를 기반으로 정의 되어야 한다. 그래서 각 플랫폼에 맞는 아키텍처를 기반으로 테스트 디자인을 하며 이를 통해 플랫폼 명세적인 테스트 플랜을 정의한다. 또한 각 시스템 및 오브젝트에 맞는 테스트케이스도 생성하고 이를 문서화한다.

<표 2> 플랫폼 명세적 테스트 모델

공정	단계	활동	작업	산출물
플랫폼 명세적 테스트 모델 (FSIM)	도메인 명세적 상세 분석 (A)	플랫폼 명세적 상세 분석 (ASB)	플랫폼 명세적 시스템 상세 디자인 (PSSDD) 및 테스트 케이스 생성 (PSSTG)	플랫폼 상세 시스템 테스트 케이스 (PDST) 및 정의서 (PDSTDD)
	도메인 명세적 설계 (D)	플랫폼 상세 테스트 디자인 (FDTD)	플랫폼 상세 테스트 케이스 생성 (PTG)	플랫폼 상세 테스트 케이스 (PST) 및 문서 (PSTDD)

표 2 에서 자세히 살펴보면 플랫폼 도메인에 종속적이며 명세적인 부분을 정의하기 위해 플랫폼 명세적 테스트 요구사항을 만들고 이를 플랫폼 명세적 테스트 플랜을 작성한다. 이 데이터를 기반으로 각 플랫폼 도메인 종속적인 분석을 하며 이때 각 플랫폼에 맞는 테스트 디자인을 통해 플랫폼 시스템에 맞는 테스트케이스를 생성하고 이를 문서화 한다. 그리고 각 오브젝트에 맞는 테스트 디자인을 하고 테스트케이스를 생성하며 이를 문서화 한다[7].

(C) Platform Dependent Process



(그림 2) 플랫폼 의존적 프로세스

그림 4 는 플랫폼 의존적이고 실제 코드레벨 단위의 테스트 프로세스를 정의하고 이에 맞는 테스트케이스를 생성 및 실행하는 단계이다.

상위의 플랫폼 명세적인 데이터와 문서를 기반으로 각 플랫폼에 맞는 코드를 자동화된 도구를 통하여 생성하고 이 때 각 단계별 테스트에 맞는 디자인이 필요하다. 이를 개발 단계에 맞게 테스트 프로세스를 재정의하였다. 크게 4 부분으로 플랫폼 아키텍처 테스트디자인, 플랫폼 시스템 테스트 디자인, 플랫폼 타겟 오브젝트 테스트 디자인, 테스트 디자인으로 나뉜다. 플랫폼 아키텍처 테스트 디자인은 플랫폼 별로 아키텍처가 다르고 상위의 플랫폼 명세적인 모델이 나오지만 코드 단위에서 또 추가가 되거나 수정이 되는 부분이 있기 때문에 이에 맞는 테스트케이스를 생성한다. 플랫폼 시스템 테스트 디자인은 개발되어 적용되는 임베디드 디바이스에 맞는 테스트 케이스를 생성하는 단계이다. 테스트 오브젝트 디자인 부분은 오브젝트 레벨 단계에서의 테스트 케이스를 생성한다. 테스트 데이터는 구현된 코드를 기반으로 테스트 케이스를 추출하고 실행하는 단계이다. 표 3 은 이와 관련된 단계, 활동, 작업, 산출물을 정의하였다. 상위의 플랫폼 명세적인 모델을 기반으로 각 아키텍처와 타겟에 맞는 코드를 생성하고 테스트 케이스를 생성하여 테스트를 실행한다. 이때 각 별로 아키텍처와 시스템, 오브젝트가 다르기 때문에 각 타겟에 맞게 디자인하고 테스트케이스를 생성하는 것이 중요하다[7].

각 단계별 활동 및 작업 산출물은 다음과 같다. 대상정의 단계에서는 각 대상 별 아키텍처에 맞는 테스트 디자인을 통해 최종적으로 테스트 케이스 및 문서를 산출하고 이를 시스템 통합 테스트에 적용한다. 대상분석 단계는 플랫폼 시스템 테스트를 디자인하고 테스트케이스를 생성 및 문서를 산출하고 이를 시스템 테스트에 적용한다. 대상 설계 단계에서는 타겟 오브젝트 테스트 디자인을 하고 각 타겟의 오브젝트 테스트 케이스를 생성하고 문서를 산출하고 이를 기반으로 스테이트 트랜지션 테스트를 실행한다. 대상 구현 단계에서는 테스트 데이터를 기반으로 테스트를 실행하며 유닛 테스트가 이 단계에서 주로 실행된다.

<표 3> 타겟 의존적 공정

용경	단계	활동	작업	산출물
타겟 의존 코드	테스트 구현 (I)	테스트 환경 정의 (TED)	테스트 환경 구축 (TEC)	테스트 환경구축 보고서 (TECD)
		테스트 하네스 정의 (THD)	테스트 하네스 구축 및 실행 (THE)	테스트 하네스 구축 및 수행 결과 보고서 (THCED)
		테스트 스크립트 정의 (TSD)	테스트 스크립트 실행 (TSE)	테스트 수행 결과 보고서 (TED)
		타겟 의존적 테스트 데이터 정의 (TDTD)	타겟 테스트 실행 (TTDE)	타겟 테스트 수행 결과 보고서 (TTED)

4. 결론 및 향후 연구

현재 소비자의 요구에 따라 다양한 타겟 상에서 임베디드 소프트웨어 개발이 폭주되고 있다. 같은 서비스를 제공하는 어플리케이션을 다양한 플랫폼에 맞게 개발하려면 많은 시간과 비용이 소모된다. 또한 이를 위한 테스트 비용도 증가하게 된다. 이런 문제를 해결하고자 본 논문에서는 이중의 임베디드시스템에서의 테스트를 위해 기존의 Model Driven Architecture(MDA)를 사상으로 하는 모델기반의 테스트 프로세스를 재정의 하였고 기존 논문에서 부족했던 부분인 각 공정 별 단계, 활동, 작업, 산출물을 정의하였다. 이를 기반으로 모델변환 도구에서 필요로 하는 각각의 활동, 작업, 산출물을 정의함으로써 모델변환 도구가 좀 더 높은 수준의 테스트케이스 및 산출물을 만들어 낼 수 있으리라 본다. 그러나 현재 기존 개발된 모델변환 도구와의 매핑을 통한 사례연구가 부족한 상황이다. 이를 위해 향후 연구과제로는 이를 기존 모델변환 도구에 적용하고 개선점을 찾아서 기존 모델변환 도구의 성능과 품질을 높이고 이중 임베디드 환경에서의 테스트 케이스를 생성하는 도구를 개선하고자 한다. 이를 바탕으로 다양한 이중의 임베디드 플랫폼과 개발환경에 맞는 맞춤형 테스트 프로세스를 정의하고 관련된 도구를 만들고자 한다.

참고문헌

- [1] A. Kleppe, J. Warmer, W. Bast, MDA Explained: The Model Driven Architecture: Practice and Promise, Addison-Wisley, 2003.
- [2] 김동호, 손현승, 김우열, 김영철. 임베디드 시스템 테스트를 위한 Model Driven Architecture (MDA) 기반의 테스트 프로세스에 관한 연구, 정보과학회, 2011
- [3] John Wakis, "Testing IT", Cambridge
- [4] B. Brockman, E. Notenboom, Embedded Software Testing, Addison Wesley, 2003
- [5] 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
- [6] 김우열, 이중 모바일 임베디드 플랫폼을 위한 모델 변환 프레임워크, 홍익대학교 2011
- [7] Dong ho Kim, Woo Yeol Kim, Hyun Seung Son, Hyung-Mook Lee, HeaKyung Seong, R. Young Chul Kim, "Test Case Extraction based on Use Case Approach for Heterogeneous Embedded System", The 1st Yellow Sea International Conference on ubiquitous Computing, Shandong Univ, China, 2011