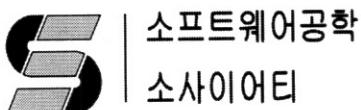


Korea Conference on Software Engineering 2012
(KCSE-2012)

2012

한국 소프트웨어공학 학술대회 논문집

- 일시: 2012년 2월 8일(수) ~ 10일(금)
- 장소: 강원도 보광 휘닉스파크 内 한화리조트



제 14 권 제 1 호
Vol. 14 No. 1



주최: 한국정보과학회, 한국정보처리학회
주관: 한국정보과학회 소프트웨어공학 소사이어티
 한국정보처리학회 소프트웨어공학 연구회
 한국전자통신연구원
후원: 비트컴퓨터,
 고려대학교 고신뢰 융합소프트웨어 연구센터,
 단국대학교 금융 IT 를 위한 소프트웨어공학연구센터,
 서강대학교 SW 요구 및 검증공학기술 연구센터,
 포항공대 융합소프트웨어개발 연구센터,

KCSE 2012 프로시딩 목차

SW 개발방법론

Agile 방법론에 대한 동양 문화 관점에서의 고찰 김민호, 조성범, 서승현, 이창호, 민상윤 (KAIST)	1
XML 기반 신뢰성있는 관점 지향 기법 (우수논문) 김은선, 이병정, 이재호 (서울시립대학교)	12
파라매트릭 소프트웨어 매트릭스 기반의 LOC 정의 (short) 신원, 장천현 (건국대학교)	14
글로벌 소프트웨어 개발 방법 적용 사례 연구: A-SQUARE 프로젝트 (산업체논문) 박태현, 강성원 (KAIST)	18

시멘틱웹

모바일 상황 인식 서비스 환경에서 향상된 추론 결과를 획득하기 위한 온톨로지 모델 선택 기법 (우수발표논문) 권오현, 박준석, 염근혁 (부산대학교)	42
온톨로지 재사용을 위한 온톨로지 저장소와 SWRL 구성 도구의 구현 및 적용 이인환, 남태우, 류쟈치, 염근혁 (부산대학교)	44
OWL Ontologies 를 이용한 XML Schema 파일 관리 (short) 박은해, 김정선 (한양대학교 컴퓨터공학과)	50
감정 분석을 이용한 동적 기능 제공 프레임워크: 모바일 기기 기반 (short) 황정우, 이윤규, 인호 (고려대학교)	54

요구 공학

Goal 지향 유스케이스 기반의 요구사항 추출에 관한 연구 박보경, 문소영, 김동호, 서채연, 김영철 (홍익대학교)	58
A Study on Research Trends in Requirement Engineering (short) 최수진, 박수용 (서강대학교)	65
UML 기반 결함 트리 자동생성과 런타임 결함식별 김준희, 강신욱, 이은석 (성균관대학교)	73

SW 개발 환경에서의 암묵적 전달/측정을 위한 지식관리 프로세스에 대한 고찰	<u>148</u>
이성연, 권용찬, 고민석, 정로미, 민상윤 (KAIST)	
TPI Next 와 CMMI 의 접목을 통한 효율적 테스트 프로세스 개선 접근 방법론	<u>156</u>
김명호, 배승진, 이충희, 이준형, 민상윤 (KAIST)	
TMN 관리기능을 이용한 USN 기반 데이터 통합운용관리시스템의 품질평가모델 개발	<u>164</u>
김성회, 권기현 (경기대학교), 한주연 (한국정보통신기술협회 시험인증연구소)	

테스팅 1

소프트웨어 역공학을 통한 HeliScope OFP 의 테스트 케이스 생성	<u>172</u>
이종훈, 이동아, 유준범, 송승화, 김두현 (건국대학교)	
Pre-Testing 를 위한 M&S 기반 테스트 케이스 추출 연구 (short)	<u>181</u>
우수정, 손현승, 김우열, 김재승, 김영철 (홍익대학교)	
지능형로봇 테스트를 위한 상태기반 모델링	<u>184</u>
유지동, 김대곤, 장영원, 노혜민, 유철중 (전북대학교)	

SW 프로세스 2

모바일 도메인의 피드백 환경 구축을 위한 상호작용 모델링 방법	<u>191</u>
신수혜, 박준석, 염근혁 (부산대학교)	
원자력 해체시설 특성관리 시스템을 위한 CBD 프로세스의 적용 방안 (short) 우수논문)	<u>198</u>
조운형, 박승국, 최윤동, 문제권 (한국원자력연구원)	
프로젝트의 성공적 수행을 위한 상주감리의 위상 정립 및 향후 발전방향에 관한 연구 (short)	<u>200</u>
전재영, 조희준, 권호열 (강원대학교)	
CMMI 기반의 품질보증 프로세스 개선 지침 모델 개발 (short)	<u>204</u>
김정민, 도성룡, 한혁수 (상명대)	

테스팅 2

스마트 폰 앱 호환성 테스트를 위한 기존 GUI 테스트 오라클의 한계	<u>208</u>
이정욱, 김태연, 채홍석 (부산대학교)	

Pre-Testing를 위한 M&S 기반 테스트 케이스 추출 연구

우수정, 손현승, 김우열, 김재승, 김영철

홍익대학교 컴퓨터정보통신공학과

소프트웨어공학전공

{woo,son,john,jskim,bob}@selab.hongik.ac.kr

요약: Pre-Testing 은 사용자의 요구사항을 가상환경에서 가상 객체로 구축하고, 테스트 케이스로 가상 객체를 테스트 하는 방법이다. 이 Pre-Testing 은 시스템 개발하기 전에 요구사항을 검증하는 용도로 사용한다. 본 논문은 Pre-Testing 을 위한 테스트케이스 추출방법에 관한 것이다. 제안한 방법은 개선된 Message-Sequence Diagram 을 Cause-Effect Diagram 으로 변환하여 테스트케이스를 생성한다. 기존의 Cause-Effect Diagram 을 이용한 Test-Case 발생 방법에 개선된 Message-Sequence Diagram 을 접목하여 보다 많은 요구사항을 테스트 케이스에 반영 가능하다.

핵심어: M&S, Message Sequence Diagram, Cause-Effect Diagram, Decision Table, Test Case, Pre-Testing

1. 서 론

스마트 폰 사용자는 2009년 47만명에서 2011년 2000만명을 돌파했다. 스마트 폰 사용자에 따라 재미를 위한 어플리케이션, 정보나 지식을 얻을 수 있는 어플리케이션, 건강을 위한 어플리케이션 등 많은 수의 어플리케이션이 개발되고 있다. 그 중에서 실생활에서 필요한 어플리케이션이 많이 개발되고 있다. 하지만 많아진 어플리케이션이 사용자의 요구사항에 적절히 반영하지 못해 품질이 떨어지는 경우가 많다. 예를 들어 무료 문자 서비스 어플리케이션 같은 경우 그 수가 많지만 사용자는 문자 서비스만 원하는 것이 아니라 음성 서비스도 같이 요구하기 때문에 이를 지원하는 어플리케이션을 더 선호한다.

이처럼 소프트웨어 개발에서 최근 중요한 이슈는 사용자 요구사항을 충족시키는 것이다. 요구사항에 맞지 않을 경우 소프트웨어 품질은 물론, 개발 비용과 개발 완료 시간에 부정적인 영향을 줄 수 있다.

이러 수정하는 비용은 개발의 후반부로 갈수록 증가하여 마지막 단계인 유지, 보수 단계에서의 에러수정 비용은 개발 초기인 요구사항 단계 보다 100~200 배에 이른다.[1] 그러므로 요구사항을 시스템 개발에 충실히 반영해야 한다.

UML 은 수학이나 철학 등에서 적절한 원료를 가지고 사용자가 쉽게 소프트웨어 모델을 구축 할 수 있도록 구문과 의미를 잘 정의해 놓은 언어이다.[2] UML 은 각 요소를 보여주고 그것들 사이의 일반화 관계 및 연관 관계 그리고 속성 등을 잘 표현해 준다. 그래서 개발자와 사용자가 모두 UML 을 통해서 쉽게 의사소통이 가능하다. 또한 UML 의 Use-Case 와 Message-Sequence Diagram 은 사용자 요구사항을 잘 명세할 수 있다.

Pre-Testing 은 어플리케이션을 개발하기 전(前)에 시스템 검증 방법으로 M&S(Modeling & Simulation) 기법을 사용한다. 이 방법은 실제 환경과 최대한 가까운 가상환경에서 가상 객체가 제대로 작동하는지 검증 할 수 있다. 본 논문은 Pre-Testing 을 위한 테스트케이스 추출방법에 관한 것이다.

제안한 방법은 M&S, UML Modeling, CED(Cause-Effect Diagram), DT(Decision Table), TC(Test Case) & Test Scenario 마지막으로 가상환경에서의 Pre-Test 로 총 6 단계로 구성되어있다. 첫번째 단계인 M&S 에서는 사용자의 요구사항에 맞게 가상환경을 구축한다. 두번째 단계인 UML Modeling 에서는 구축된 가상환경을 중 시간의 흐름에 맞게 쓸 수 있는 Message-Sequence Diagram 으로 표현한다. 세번째 단계 CED 에서 Message-Sequence Diagram 에서 Input 과 Condition, Output 을 찾아 나타내고 네번째 단계인 DT 은 CED 를 테이블로 나타낸 것이다. TC&Test Scenario 단계에서는 DT 를 통해 지정된 사용자 요구사항에 옳은지 맞는지를 명세하게 된다. 마지막으로 가상환경에서의 Pre-Test 을 통해 가상 검증을 수행 할 수 있는 프로세스이다.

기존의 Cause-Effect Diagram 을 이용한 Test-Case

[†] 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2011-0004203)과 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임.

발생 방법에 개선된 Message-Sequence Diagram 을 접목하여 보다 많은 요구사항을 테스트 케이스에 반영 가능하다. 가상환경에서 Pre-Testing 으로, 실제 환경에서의 검증에서 발생하는 비용과 시간을 줄일 수 있는 효과를 보려고 한다.

본 논문의 구성은 다음과 같다. 2 장에서는 모델 도구인 HIMEN ver.1.0 개발에 관해 언급한다. 3 장에서는 Pre-Testing Process 에 대한 전체적인 설명과 각 Process 의 Step 에 설명한다. 4 장에서는 결론과 향후 연구에 대해서 기술한다.

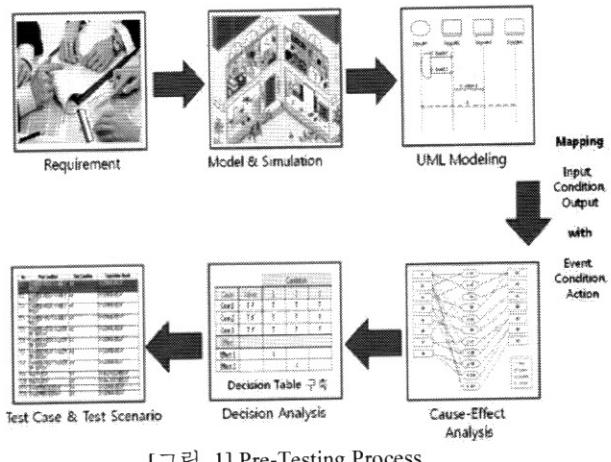
2. 관련연구

M&S 공학은 모델링/시뮬레이션을 이용한 문제 해결의 전 과정과 이들 전 과정에서 일어나는 각 단계별 주제를 연구하는 학문이다. M&S 와 관련된 학문은 총 5 가지로 분류된다. 첫 번째 요구공학(Requirement Engineering), 두 번째 모델링 이론(Modeling Theory), 세 번째 시뮬레이터 구현(Simulator Implementation), 네 번째 가설 시험(Hypothesis Test), 마지막 성능공학(Performance Engineering)분야이다. 이 중 네 번째 시뮬레이터 구현(Simulator Implementation)관련 분야에 대해서 소개하려고 한다. 시뮬레이터 구현은 모델링 이론으로부터 얻어진 모델들을 효율적으로 시뮬레이션 하는 알고리즘과 모델 및 이들 알고리즘의 프로그램 구현에 관한 것이다. 구현된 시뮬레이터가 모델 명세를 잘 반영하는지를 확인하는 과정인 모델 검증(Model Verification)은 프로그램 디버깅과 동일한 과정으로 모델 명세서로부터 테스트 입력과 대응출력을 구하여 시뮬레이터를 테스팅하는 것이다[3]. 본 논문은 SELab 에서 개발된 HIMEN ver1.0 를 통해서 자동적으로 Test Case 를 추출하게 된다. 이를 이용해 실제 시스템 개발 전에先 테스트로, 더욱 효율적으로 검증하기 위해선 테스트 케이스를 추출하여 개발 후 시스템을 테스트와 비교 분석하여 오차를 줄이고자 한다. 이 방법 하나로 Cause-Effect Diagram[3]을 사용한다.

3. Pre-Testing Process

본 장에서는 Pre-Testing process 을 통해 가상환경에서 발생되는 오류나 문제점을 미리 발견하여, 실제 환경에서 개발할 때 시간과 비용에 대해 절감할 수 있는 방법을 설명하려고 한다. 이를 위해 테스트케이스 도출방법은 총 5 단계로 나누어 진다. 첫 번째 단계는 M&S(모델 & 시뮬레이션)으로 실제 환경과 같이 가상환경상에서 구축하고, 두 번째 단계는 MSD(Message Sequence Diagram) 메시지 순차적 다이어그램으로 메시지를 순차적으로 드러내 보여주고 싶을 때 쓰는 방법이다. 세 번째 CED(Cause-Effect Diagram)은 메시지 순차적 다이어그램으로부터 도출

및 매핑이 된다. 네 번째 DT(Decision Table)결정 테이블은 Cause-Effect Diagram 부터 나온 결과를 테이블로 보여지는 단계이며, 마지막 다섯 번째 테스트케이스는 결정테이블에 의해서 자동적으로 생성한다.



[그림 1] Pre-Testing Process

Step 1 : M&S (Model Simulation)

M&S 에서의 모델링은 현실 세계의 모든 객체와 현상을(즉 실제 시스템) 모델로 표현하는 과정 또는 행위이다. 시뮬레이션은 이 표현된 모델을 이용하여 현실 세계 시스템의 현상들을 모의하는 과정을 나타낸다. 모델링은 현실세계 현상을 정량적(수학적)으로 표현한 것이고, 시뮬레이터는 소프트웨어, 하드웨어, 실제 장비 등을 나타내는 것이다.[4,5] M&S 에서는 가상 환경을 오브젝트로 만들어서 수행한다. 이 가상 객체에 대한 움직임과, 로그 객체의 행동과 움직임에 대한 오류를 시뮬레이션 상에서 알고, 제작 시에 최대한 오류가 없게 함으로서 시간과 비용에 대해 절감 할 수 있도록 한다.

Step 2: UML Modeling

UML Modeling 은 소프트웨어 중심의 시스템을 명세화, 구성, 가시화, 그리고 문서화하는 모델링 언어이다. 여러 가지 언어 중 메시지순차적 다이어그램으로 나타내려고 한다. 메시지 순차적 다이어그램은 객체간의 동적 상호작용을 시간적 개념을 중심으로 모델링하는 과정을 나타낸다.[6] 다이어그램의 수직방향은 시간의 흐름을 나타낸다. 요구사항을 통해서 UML Modeling 하는 것은 M&S 에서의 한 기능을 시간의 흐름에 맞게 순서대로 나타내는 것이다.

Step 3: CED(Cause-Effect Diagram)

Cause-Effect Diagram 은 일의 결과(특성)과 그것에 영향을 미치는 원인(요인)을 계통적으로 정리한 Diagram 이다.[7,8] Message-Sequence Diagram 에서 Input, Condition, Output 을 찾아 다이어그램으로서

나타낸다.

Step 4: DT(Decision Table)

결정 테이블은 Cause-Effect Diagram 을 표로 나타낸 것이다. 각각의 세로는 Condition 을 나타낸다.

Step 5: TC(Test Case) & Test Scenario

지정된 사용자 요구사항에 옳은지 맞는지를 검증하는 것을 의미한다. Decision table 을 패턴 기반으로 테스트 케이스로 전환하게 된다.

Step 6: 가상환경에서의 Pre-Test

위의 발생한 테스트 케이스를 통해 가상환경의 객체를 통해 가상 검증을 수행하게 된다. 이런 과정을 수행함으로써, 개발 전에 발생할 수 있는 오류와 mal-function 들을 발견 할 수 있다. 이를 인해 실제 개발 후, 테스팅 비용과 시간을 줄 일 수 있으리라 본다.

4. 결 론

본 논문은 사용자의 요구사항에 맞게 가상 환경 시스템 개발 되기 전 즉, Pre-Testing 을 통해서 시스템의 옳고, 그름을 Pre-Testing(선 테스트)를 하는 방법을 제안했다. UML Modeling 언어 중 메시지 순차적 다이어그램을 통해서 시간의 흐름에 따라 객체간의 상호작용을 나타냈다. 또한 메시지 순차적 다이어그램을 논리 게이트를 차용하여 여러 개의 Input 값이 Condition 을 통해서 여러 개의 Output 값을 Cause-Effect Diagram 으로 나타냈다. Cause-Effect Diagram 을 통해서 결정 테이블을 나타내고 Test Case 전환했다. 마지막 단계에서 발생한 Test Case 를 통해서 가상환경에서 검증을 수행하는 방법을 제안했다. 이 방법으로서 시간과 인력에 대한 비용을 최대한 줄이면서 효과는 최대로 끌어 올릴 수 있도록 하며, 더 나은 검증을 할 수 있으리라 본다.

향후 연구로 MSD 를 CED 로 자동변환 할 수 있도록 MSD 의 표기법을 확장하고, 변환규칙에 대하여 연구 중이다.

참고문헌

- [1] 손현승, 김우열, 김재승, 김영철, “가상환경상에서 다관절로봇의 테스트프로세스연구”, 소프트웨어 공학회, 2011.02
- [2] 김기윤, 나관식, 양동구, “요구사항 불확실성, 통제 표준화, 상호작용이 소프트 품질에 미치는 영향”, 경영정보학연구, 2002.12
- [3] 김탁곤, “모델링 시뮬레이션공학”, 정보과학회지, 2007.11
- [4] 김재수, 손현승, 김우열, 김영철, “A Study on M&S Environment for Designing The Autonomous Reconnaissance Ground Robot”, 한국군사과학학회, 2009.12
- [5] 김동우, 손현승, 김우열, 김영철, “Application of M&S(Modeling & Simulation) for The Autonomous Reconnaissance Ground Robot”, 국방과학연구소, 2008.10
- [6] 김우열, “Model Driven Architecture 기반의 임베디드 소프트웨어 모델링에 관한 연구”, 홍익대학교 대학원, 2005.12
- [7] Gary E. Mogyorodi, B.Math., M.B.A, “Requirements-Based Testing: Cause-Effect Graphing”, Software Testing Services, 2010.
- [8] Kyu Won Kim, Woo Yeol Kim, Hyun Seung Son, and Rober Young Chul Kim, “A Validation Process for Real Time Transactions”, Software Engineering Business Continuity and Education Vol.257