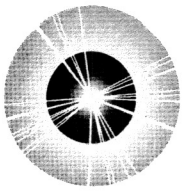
학술발표논문집 ISSN 1598-5164

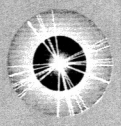


한국정보과학회
KOREAN INSTITUTE OF INFORMATION SCIENTISTS AND ENGINEERS

제39권 제1호

2012 한국컴퓨터종합학술대회

논문집(B)



한국정보과학회
KOREAN INSTITUTE OF INFORMATION SCIENTISTS AND ENGINEERS
2012년 6월 27일 ~ 6월 29일 · 제주도 휘닉스 아일랜드

UML 메카니즘과 원인-결과 다이어그램 기반 테스트케이스 생성을 위한 자동 도구 개발

우수정° 김영철
홍익대학교 소프트웨어공학연구실
woo@selab.hongik.ac.kr, bob@hongik.ac.kr

Automatic Tool Development for TestCase Generation Based on UML Mechanism and Cause-Effect Diagram

SuJeong Woo° R.Young-Chul Kim
Dept. of. CIC, Hongik University

요 약

본 논문은 Use Case 기반 개발에서 요구사항부터 테스트 케이스를 자동 추출하는 메커니즘을 제안 하고자 한다. 제안한 메커니즘은 기존의 테스트케이스 생성 메커니즘[3]을 Use Case 메커니즘과 접목[2]한 것이다. 그리고 그 기반으로 자동 도구 구현을 통해 모든 가능한 테스트 케이스 추출하는데 있다. 이는 최소의 테스트 케이스로 100%의 기능적인 요구사항 커버리지 만족[3]시킨다는 Gary 방법을 이용하고자 함이다. 이 도구의 단계는 확장된 UML 다이어그램으로부터 원인-결과 다이어그램을 전환 한 후, 결정 테이블화한다. 마지막 단계는 이를 통하여 테스트케이스가 자동적으로 발생 한다.

1. 서 론

최근 중요한 이슈는 소프트웨어 개발에서 사용자 요구사항을 어떻게 충족시키는 것이며, 요구사항에 맞지 않을 경우 소프트웨어 품질은 물론, 개발 비용과 개발 완료 시간에 부정적인 영향을 줄 수 있다[1]. 에러 수정하는 비용은 개발의 후반부로 갈수록 증가하게 된다. 특히 마지막 단계인 유지, 보수 단계에서의 에러수정 비용은 개발 초기인 요구사항 단계 보다 100~200배에 이른다[5]. 그러므로 올바른 요구사항 기반으로 시스템 개발해야 한다. 이런 시스템을 체계적으로 테스트하기 위해 정확한 요구사항으로부터 테스트케이스를 추출하여 개발 후 시스템 검증 테스트가 필요하다.

기존 테스트케이스 발생 방법으로는 여러 가지[3]가 있으나, 대부분 구조적 개발 방법상에서의 연구가 많다. 특히 객체지향 및 Use-Case 개발환경에서 State Machine을 통하여 테스트케이스 발생방법[6,7], 또한 Use-Case Diagram을 통한 테스트케이스 발생[9]방법 등등이 있다. 하지만 본 논문은 확장된 UML 파라다임에 원인-결과 다이어그램을 접목시켜 테스트케이스를 발생시키는 방법[3]에 대해서 설명하고자 한다. 원인-결과 다이어그램을 통한 테스트케이스 발생은 Gary E. Mogyorodi[3]가 제안한 검증된 방법이다. 이 방법은 최소의 테스트케이스로 100%의 기능적인 요구사항 커버리지를 만족 시킬 수 있는 방법[3]이다. 이 때문에 확장된 UML과 원인-결과 다이어그램의 접목을 통해 정확하게 구현을 위해 테스트케이스 생성하는 방법을 제안한다.

본 논문은 [1,2] 에서 제안한 UML 파라다임과 원인-결과 다이어그램의 접목을 통해 테스트케이스 발생을 위한 자동화 도구에 대해서 설명한다.

본 논문의 구성은 다음과 같다. 2장은 제안한 테스트케이스 추출 프로세스에 대해 설명하고, 3장은 자동 테스트케이스 생성도구에 대한 단계별 예를 들어 설명한다. 마지막 4장은 결론 및 향후 연구에 대해서 기술한다.

2. 제안한 테스트케이스 추출 프로세스

현재 객체지향 및 Use-Case 개발 경험자에게 쉽게 접근하고자 제안한 프로세스이다.

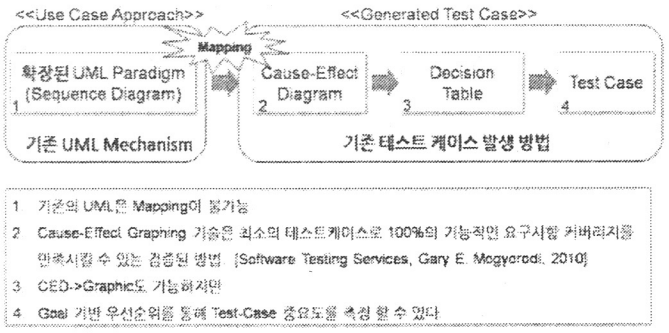


그림 1 테스트케이스 프로세스

그림 1은 테스트케이스 프로세스이다. 프로세스의 스텝

* 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업(NIPA-2012-(H0301-12-3004))과 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임.

은 다음과 같다.

첫 번째 단계는 UML 언어 중 메시지-순차적 다이어그램을 확장함으로써 테스트케이스 발생 가능한 원인-결과 다이어그램을 접목하는 단계이다.

두번째 단계는 원인-결과 다이어그램을 확장된 UML과 접목시킨다. Gary에 의하면, 원인-결과 다이어그램은 원인과 결과를 노드와 커넥션으로 표현한 다이어그램이다. 원인-결과 다이어그램은 최소의 테스트케이스로 100%의 기능적인 요구사항 커버리지를 만족시킬 수 있는 검증된 방법[3]이다.

세번째 단계는 원인-결과 다이어그램을 통해 결정-테이블을 생성한다. 결정테이블은 논리적인 조건을 포함하고 있는 시스템의 요구사항과 내부적인 시스템의 설계를 문서화 하는 방법이다. 이때 테이블이 아닌 트리 형식인 그래픽도 가능하다. 하지만 그래픽 형식의 단점은 nodes(causes and effects)나 branches은 잘못 설정이나, missing 된 노드와 branch를 찾을 수 없다. 그래서 결정 트리가 아닌 결정테이블을 사용한다.

네번째 단계는 자동적으로 테스트케이스가 발생된다. 테스트케이스는 테스트 입력, 수행조건, 기대 결과로 구성된 집합으로 특정한 목적을 위해 개발되었다. 또한 입력, 예상되는 결과, 테스트 아이템을 위한 수행조건의 집합으로 기술되어 있는 문서이다. 발생된 테스트케이스는 Goal 기반 우선순위를 통해 테스트케이스의 중요도를 측정[4] 할 수 있다.

3. 자동 테스트케이스 생성 도구

Step1 : Use-Case

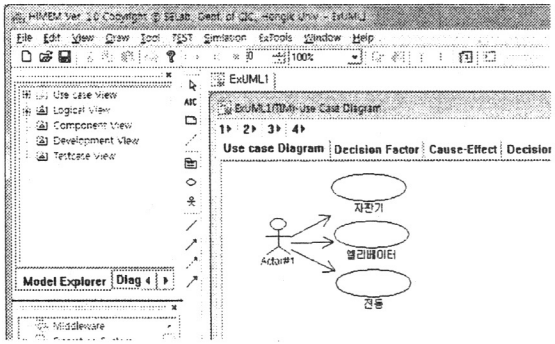


그림 2 Use-Case Diagram

그림 2은 유스케이스 다이어그램이다. Actor은 여러 가지 전자제품을 사용할 수 있다. 그 중 자판기 Use-Case를 선택하여 객체를 추출한다.

그림 3은 자판기 Use-Case의 요소들을 Input, Output, Condition에 맞는 단어들을 지정한다. Input 요소들은 '버튼을 누른다', '사이다', '콜라', '동전을 투입한다' 로 지정한다. Output은 '음료수가 나온다', '잔돈이 나온다', '사이다', '콜라', '아무것도 나오지 않는다' 로 지정한다. Input, Output을 이용하여 Condition을 지정한다. 이때 논리기호를 사용하여 같으면 '=', AND는 '&', OR은 '|', NOT은 '~' 으로 하여 지정한다.

그림 4는그림 3에서 뽑은 요소들을 테이블로 정리한 것이 결정요소(Decision Factor)이다. 4개의 Input, 5개의 Output, 그리고 6개의 Condition의 Decision Factor들을 나타낸다.

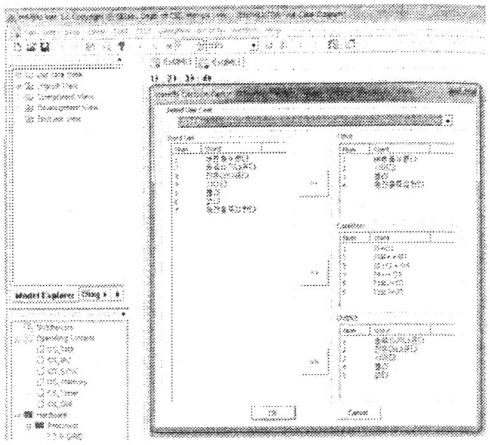


그림 3 Input/Condition/Output 정의

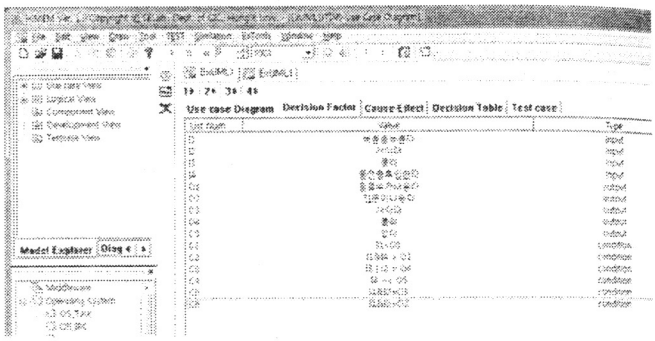


그림 4 Decision Factor

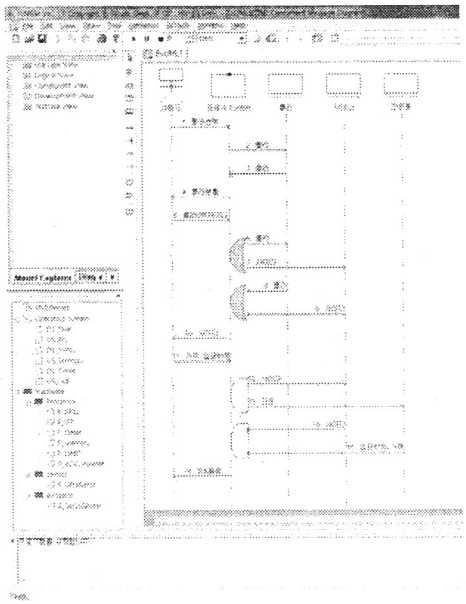


그림 5 메시지-순차적 다이어그램

그림 5는 메시지-순차적 다이어그램이다. Decision Factor을 통해 메시지-순차적 다이어그램의 컨트롤을 중심으로 Input메시지, Output메시지를 표시한다. 여러

개의 메시지가 들어갈 수도 있고 나갈 수도 있기 때문에 논리게이트를 차용[2]하여 확장한다.

Step 2 : Cause-Effect Diagram

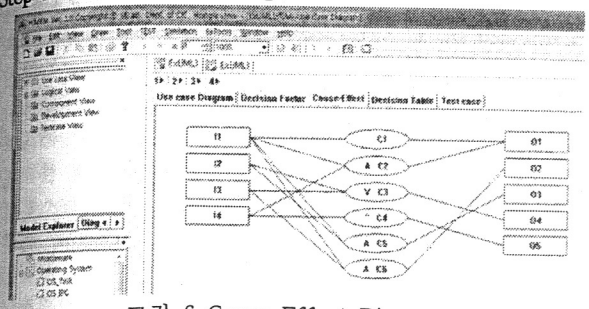


그림 6 Cause-Effect Diagram

그림 6는 원인- 결과 다이어그램을 나타낸다. 그림 4의 Decision Factor에서 Input(I1, I2, I3…)은 제일 왼쪽, Condition(C1, C2, C3…)은 중간, Output(O1, O2, O3…)은 오른쪽에 배치하고, AND Condition은 'A' 로 나타내고 OR Condition은 'V' 로 나타낸다. 또한 NOT Condition은 '~' 로 나타낸다.

Step 3 : Decision Table

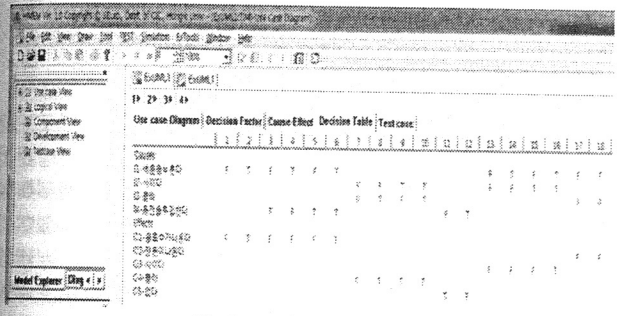


그림 7 Decision Table

그림 7는 원인-결과 다이어그램의 결정 테이블을 나타낸 것이다. Cause은 I1~I4이고, Effect은 O1~O5이다. Condition에 맞게 False/True를 나타낸다.

Step 4 : Test Case Generation

테스트 케이스는 그림 8과 같이 Decision Table을 통하여 자동적으로 발생된다.

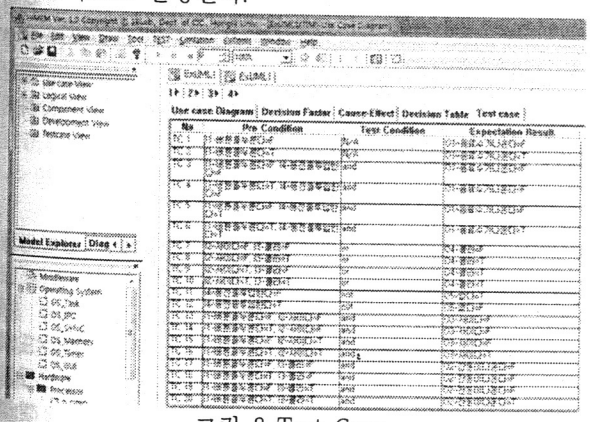


그림 8 Test Case

총 20개의 테스트케이스가 발생된다. 테스트케이스의 Pre-Condition은 결정 테이블의 Cause부분인 Input이고, Test Condition은 Condition이다. 마지막 Expectation Result은 결정 테이블의 Effect 부분인 Output을 나타낸다.

4. 결론 및 향후 연구

본 논문은 제안된 메커니즘 기반의 테스트케이스 자동화 도구의 개발에 있다. Gary의 원인-결과 다이어그램을 통해 테스트케이스 발생[3]은 최소한의 테스트케이스로 100%의 기능적인 요구사항 커버리지를 만족 시킬 수 있는 방법이다. 기존의 UML은 접목이 불가능 하므로 확장시켜 원인-결과 다이어그램에 접목을 시켰다. 이를 통해 확장된 UML 파라다임에 테스트케이스를 발생 시킬 수 있도록 했다.

본 논문에서 제시한 테스트케이스 자동화 도구는 총 4개의 단계를 가지고 있다. 첫번째 단계는 Use-Case 다이어그램을 작성한다. 작성된 Use-Case 다이어그램에서 Input, Output, Condition의 요소들을 정의한다. 세번째 단계는 원인-결과 다이어그램을 표로 나타낸 Decision Table로 나타냈다. 마지막 네번째는 Decision Table을 통해서 Test Case를 발생시킨다.

기존에 현존하는 도구는 UML1.x버전이다. 향후 연구에서는 최신버전인 UML2.4.1버전으로 개발할 예정이다.

3. 참 고 문 헌

[1] 우수정, 손현승, 김우열, 김재승, 김영철, "Pre-Testing를 위한 M&S 기반 Test-Case 추출 연구", 소프트웨어공학학회, 2012

[2] 우수정, 손현승, 김영철, "원인-결과 다이어그램과 접목을 위한 메시지-순차적 다이어그램 확장 연구", 한국정보처리학회, 2012

[3] Gary E. Mogyorodi, "Requirements-Based Testing Cause-Effect Graphing", 2010

[4] 박보경, 문소영, 김동호, 서채연, 김영철, "Goal 지향 유스케이스 기반의 요구사항 추출에 관한 연구", 소프트웨어공학학회, 2012

[5] 손현승, 김우열, 김재승, 김영철, "가상환경에서 다관절로봇의 테스트프로세스 연구", 소프트웨어공학학회, 2011

[6] Houde Fekih, Leila Jemni Ben Ayed, Stephan Merz, "Transformation of B Specification into UML Class Diagram and state Machines", ACM Vol.2

[7] WooYeol Kim, HyunSeung Son, Robert YoungChul Kim, "A Study on Test Case Generation Based on State Diagram in Modeling and Simulation Enviroment", CCIS199, 2011