



다이나믹 영향 분석 프로세스를 활용한 선택적 회귀 시험 방법

이 찬 · 윤 청*

충남대학교 컴퓨터공학과

Regression Test Selection Method using Dynamic Impact Analysis Process

Chan Lee · Cheong Youn

305-764 Dept. of Computer Science & Engineering, Chungnam National Univ., Gung-dong, Yuseong-gu, Daejeon, Korea

[요 약]

선택적 회귀 시험(regression test selection)은 기존의 테스트 케이스 가운데 일부의 케이스만을 선택하여 회귀 시험에 사용하는 방법이다. 선택적 회귀 시험을 설계하기 위해 사용되는 대표적인 방법은 변경에 대한 영향 분석(impact analysis)을 하는 것이다. 확인이 필요한 테스트 케이스를 빠짐없이 찾아 선택적으로 실행시키기 위해서는 변경에 대한 정확한 영향 분석 수행이 필요하다. 본 논문에서는 다이나믹 영향 분석 프로세스를 활용하여 회귀 시험의 테스트 케이스를 선택하는 방법을 제시한다. 이 방법은 유스케이스, UML 다이어그램에 대한 영향 분석 과정과 영향 분석 결과를 토대로 회귀 시험의 테스트 케이스를 선택하는 과정으로 이루어진다. 본 논문에서 제시하는 방법은 테스트 케이스 선택에 근거가 되는 요소들을 다이나믹 영향 분석 프로세스를 통해 명확하게 파악하여 정확성과 효율성 측면에서 장점을 보인다.

[Abstract]

Regression test selection is a method of testing by selecting some of the existing test cases. A typical method used to selective regression test is to perform an impact analysis of the change. It is necessary to accurately analysis the impact of changes to select all test cases that require confirmation. This paper proposes a method to select test cases for regression test using dynamic impact analysis process. This method consists of the impact analysis process on the UML diagram, and selecting the test case of the regression test based on the impact analysis result. The method presented in this paper clearly identifies the factors that basis of test case selection through dynamic impact analysis process and has advantages in terms of accuracy and efficiency.

색인어 : 영향 분석, 회귀 시험, 소프트웨어 변경, UML 모델

Key word : Impact Analysis, Regression Test, Software Change, UML Models

<http://dx.doi.org/10.9728/dcs.2018.19.10.1989>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 03 October 2018; Revised 10 October 2018

Accepted 22 October 2018

*Corresponding Author; Cheong Youn

Tel: +82-42-821-7452

E-mail: cyoun@cnu.ac.kr

I. 서론

소프트웨어에 변경이 가해지면 변경된 부분이 요구사항을 만족하는지 확인하기 위해 회귀 시험(regression test)을 필요로 한다. 회귀 시험 전략은 테스트의 재사용 범위에 따라 크게 다음과 같은 두 가지로 나뉜다[1].

- Retest All : 이미 준비되어 있는 모든 테스트 케이스를 재작동 시키는 방법
- Selective Retest : 변경된 부분과 변경이 영향을 미칠 수 있는 부분만을 테스트하는 방법

Retest All 방법은 기존의 테스트 케이스를 모두 재실행시키는 방법으로, 절차가 간편한 것이 장점이지만 실행시켜야 할 테스트 케이스가 너무 많아 시간과 비용 측면에서 비효율적이다.

Selective Retest 방법은 기존의 테스트 케이스 가운데 일부의 케이스만을 선택하여 회귀 시험에 사용하는 방법이다. 테스트 케이스의 효율적인 재사용을 위해 소프트웨어의 수정된 부분과 그로 인해 영향을 받는 부분을 찾고, 검출된 부분들을 검사하는 선택적 회귀 시험 방법이 흔히 사용된다[2].

선택적 회귀 시험을 설계하기 위해 사용되는 대표적인 방법은 변경에 대한 영향 분석(impact analysis)을 하는 것이다. 영향 분석은 변경에 의해 야기되는 파급 효과(ripple effect)를 인식하고 기록하여 변경의 범위를 파악하는 방법으로, 영향 분석 결과를 활용하여 변경이 필요한 부분을 충분히 확인할 수 있는 테스트 케이스를 선별할 수 있다[3]. 이 방법은 확인에 필요한 테스트 케이스를 빠짐없이 찾아 선택적으로 실행시키기 때문에 흔히 안전한 회귀 시험 기법(safe regression test)이라고 부른다[4]. 하지만, 규모가 크고 복잡한 시스템인 경우 회귀 테스트 케이스 선택을 위한 영향 분석 작업에 많은 시간과 노력을 소비할 수 있다. 따라서 선택적 회귀 시험 기법은 정확성과 효율성이라는 두 가지 측면에서 평가된다. 정확성은 시험에 필요한 테스트 케이스만을 얼마나 잘 선택하는지에 대한 평가 기준이며, 효율성은 영향 분석에 필요한 시간에 대한 평가 기준이다[5]. 결론적으로 정확성이 높고 효율성이 좋을수록 선택적 회귀 시험의 설계가 잘 이루어졌다고 할 수 있다.

회귀 시험 기법은 정확성과 효율성이라는 두 가지 측면에서 다양하게 시험의 단위를 구분할 수 있는데 객체지향 프로그램에서는 크게 클래스 단위, 메소드 단위, 문장 단위를 기본 테스트 단위로 하여 변경 영향 분석을 한다. 클래스를 기본 시험 단위로 하는 기법은 변경이 발생한 클래스와 그 클래스와 상속, 집합, 연관 관계에 있는 클래스를 분석하는 것을 통해 변경 영향 분석을 완성한다[6-8]. 이 기법은 클래스라는 큰 단위에서 분석을 함으로써 효율적인 이점을 갖고 있지만, 클래스 내에 변경이 발생한 부분이 매우 작을 때에도 클래스 전체를 시험하는 테스트 케이스를 선택하므로 정확성이 떨어질 수 있다. 문장 단위의 변경 영향 분석은 소프트웨어를 구현하는 가장 작은 단위

에서 변경 영향 분석을 진행하므로 정확성은 좋지만, 모든 문장에 대해서 비교 분석을 거쳐야 하므로 변경된 소프트웨어의 사이즈가 클 때, 효율성이 떨어지는 단점이 있다[9,10]. 따라서 정확성과 효율성을 고려해 볼 때, 클래스 단위와 문장 단위의 중간 사이즈인 메소드 단위에서 변경 영향 분석을 수행하는 것이 비교적 정확하고 효율적이라고 판단할 수 있다[11].

본 논문에서는 순방향 추적 및 역방향 추적이 혼합된 다이나믹 영향 분석 프로세스를 활용하여 회귀 시험의 테스트 케이스를 선택하는 방법을 제시한다. 이 방법은 요구사항 명세서의 구성요소인 유스케이스, 클래스 다이어그램, 시퀀스 다이어그램의 연관 관계를 활용한 영향 분석 과정과 영향 분석 결과를 토대로 회귀 시험의 테스트 케이스를 선택하는 과정으로 이루어진다.

다이나믹 영향 분석 프로세스는 유스케이스 시나리오의 변경을 식별하고 이를 토대로 각 모델이 받는 영향을 추적하는 순방향 추적 과정과 각 모델의 변경이 다른 요구사항에 미치는 영향을 파악하는 역방향 추적 과정으로 이루어진다[12]. 이 방법은 객체지향 개발에서의 기초적인 산출물을 활용하여 변경의 모든 영향을 파악할 때까지 지속적인 분석을 수행하고 변경의 누락을 방지하며 비용적인 측면에서도 장점이 있다. 또한, 영향 분석의 결과를 회귀 시험의 단위인 메소드와 연관 지을 수 있어 정확성과 효율성 측면에서도 장점을 가진다.

본 논문의 구성은 다음과 같다. 2장에서는 회귀 테스트 케이스 선택에 대한 관련 연구를 소개하고, 3장에서는 다이나믹 영향 분석 프로세스를 활용한 회귀 테스트 케이스 선택 방법을 제시한다. 4장에서는 회귀 테스트 케이스 선택의 사례 연구를 보여주고, 마지막으로 5장에서는 결론을 맺는 것으로 구성한다.

II. 관련 연구

회귀 시험의 선택은 일반적으로 소스 코드에 초점을 맞추고 있다. 소스 코드에 기반하여 변경 영향을 분석하는 방법은 최소한의 회귀 테스트 케이스를 식별할 수 있다는 장점이 있으나, 회귀 테스트 케이스를 선택하기 전에 변경의 구현이 미리 이루어져 있어야 하는 단점이 있다. 또한 요구사항에 변경이 발생했을 경우, 변경된 요구사항과 관련 있는 소스 코드를 직접 식별하기가 쉽지 않다는 문제가 있다. 소스 코드 기반 방식의 대안으로 Briand는 설계 단계에서 작성된 UML 모델을 기반으로 회귀 테스트 케이스를 선택하는 방안을 제시하였다[13]. 회귀 테스트 케이스를 설계 단계에서 미리 선택할 수 있다면, 회귀 시험 수행에 필요한 노력과 비용에 대한 측정이 일찍 이루어질 수 있다는 장점이 있다. 또한, 설계와 테스트 케이스 간에 추적성(traceability) 정보가 존재한다면, 설계 영향 분석의 마지막 단계에서 어떠한 회귀 테스트 케이스가 재수행 되어야 하고 어떤 테스트 케이스가 제거되어야 하는지 쉽게 결정할 수 있다.

Briand는 설계 단계의 클래스 다이어그램과 시퀀스 다이어

그램에 발생할 수 있는 변경의 경우들을 정의하고, 시퀀스 다이어그램의 변경으로부터 시스템 및 인수 시험 수준의 회귀 테스트 케이스를 선택하는 방법을 제시하였다. 이 방법은 시퀀스 다이어그램에서 액터가 경계 클래스(boundary class)를 통해 시스템과 상호작용하는 이벤트 시퀀스를 하나의 테스트 케이스로 보고 특정 시퀀스에서 변경이 발생하는 경우를 회귀 테스트 케이스로 선택한다.

Briand가 제시한 방법은 소스 코드 기반의 방식과 마찬가지로 설계 단계의 모델이 변경될 수 있는 배경과의 연결이 제시되지 않았다는 점에서 문제를 지니고 있다. 즉, 설계 단계의 클래스 다이어그램이나 시퀀스 다이어그램이 어떠한 이유로 변경되어야 하는지 그 배경에 대한 추적이나 분석 규칙을 정의하지 않고 있다. 일반적으로 변경은 고객의 요청에 의한 요구사항의 변경일 가능성이 높다. 따라서 요구사항의 변경과 관련하여 분석부터 설계까지 이어지는 추적성 분석이 이루어져야 설계 단계 모델의 변경 여부를 발견할 수 있다. 이러한 요구사항으로부터의 추적은 회귀 테스트 케이스 선택에 있어서 정확성을 높일 수 있는 주요한 근거가 된다.

Orest는 UML 모델의 구성요소 간의 연관 관계를 이용한 OMDAG(Object Method Directed Acyclic Graph) 방법을 제시하였다[14]. 이 방법은 구성요소의 변경 유형을 분류하고, 이들 간의 논리적 관계를 그래프로 나타내어 영향 분석에 활용한다.

Al-Refai는 Fuzzy 기법을 활용하여 기존 모델과 변경된 모델의 불일치를 구별하고, 테스트 케이스의 불확실성을 줄이는데 초점을 맞추었다[15].

Dahiya는 UML 모델의 변경 유형을 분류하고, 액티비티 다이어그램의 변경을 테스트 케이스 선택의 근거로 삼아 테스트 케이스의 정확성을 높이고, 변경 유형에 유연하게 대처할 수 있는 방법을 제시하였다[16].

Orest, Al-Refai, Dahiya의 방법들은 보다 정확한 회귀 시험 선택을 할 수 있다는 장점이 있지만, Briand와 마찬가지로 변경 요청에 포함된 요구사항들이 UML 모델에 어떠한 영향을 미치는지 구체적으로 언급하지 않아 추가적인 변경 영향에 대한 파악이 어렵다.

III. 회귀 테스트 케이스 선택 방법

3-1 영향 분석 프로세스

정확한 회귀 테스트 케이스 선택을 위해서는 변경 요청에 포함된 요구사항을 빠르게 파악하여 변경된 요구사항이 미치는 영향의 범위를 정확하게 분석하는 것이 필요하다. 유스케이스는 시스템이 수행하는 기능 요구사항에 대해 명확하게 표현할 수 있으며, 클래스 다이어그램과 시퀀스 다이어그램을 도출하는 근거가 된다. 유스케이스 시나리오 흐름(flow)에 나타난 입출력 데이터들은 클래스 다이어그램의 클래스, 속성, 관계를 식별하는 근거를 제공하며, 유스케이스 시나리오의 흐름을 구성

하는 유스케이스 이벤트(UC 이벤트)는 시퀀스 다이어그램 상의 메시지로 확장되어 클래스 다이어그램의 오퍼레이션으로 반영된다[17]. 즉, 유스케이스 시나리오의 변경이 다른 다이어그램들에 영향을 주므로, 변경 요청 시 유스케이스 시나리오의 변경으로부터 클래스 다이어그램 및 시퀀스 다이어그램에 대한 변경을 추적하여 변경의 범위를 파악해야 한다. 또한, 다이어그램의 변경이 다른 요구사항(유스케이스)을 변경시키지 않는지 검토하여, 추가적인 변경 가능성에 대해 파악해야 한다. 본 논문에서는 그림 1과 같은 프로세스를 이용하여 유스케이스와 UML 모델에 대한 변경 범위를 측정한다.

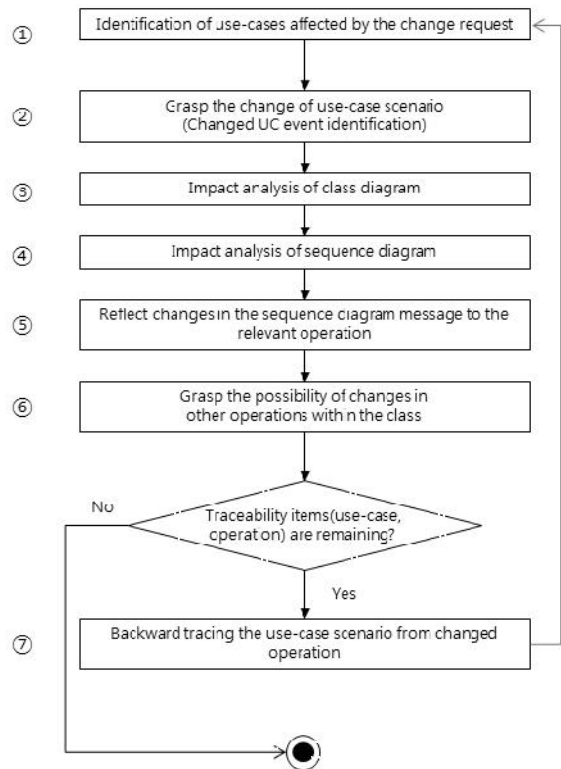


그림 1. 다이나믹 영향 분석 프로세스
Fig. 1. Dynamic Impact Analysis Process

① 변경 요청으로 인해 영향을 받는 유스케이스 식별
영향 분석 프로세스의 첫 단계는 변경 요구사항에 의해 영향을 받는 유스케이스를 식별하는 것이다. 유스케이스 식별 시 고려해야 할 사항은 다음과 같다.

- A. 식별된 유스케이스가 선행 또는 후행 관계를 맺는가?
- B. 변경 요구사항이 새로운 유스케이스를 요구하는가?
- C. 식별된 유스케이스가 하나 이상인가?

일반적으로 선행 유스케이스의 변경은 후행 유스케이스의 변경을 유발할 가능성이 높으므로 선행 유스케이스에 대한 분석을 먼저 수행해야 한다. B의 경우에는 새로운 유스케이스를

작성한 후 영향 분석 프로세스의 이후 단계를 수행한다. 그림 1에서 제시한 프로세스는 단일 유스케이스에 대한 추적 과정을 나타내므로, 하나 이상의 유스케이스가 식별되었을 경우 그림 1의 프로세스를 각 유스케이스 별로 적용해야 한다.

② 유스케이스 시나리오에 대한 변경 파악

유스케이스 시나리오를 이루는 흐름, 행위자, 사전조건 등에 대한 변경을 파악한다. 주요 고려 사항으로는 시나리오 흐름을 구성하는 UC 이벤트의 변경(입출력 데이터의 변경, 이벤트 순서의 변경)을 파악하여 클래스 다이어그램, 시퀀스 다이어그램의 변경을 식별할 수 있어야 한다.

③ 클래스 다이어그램에 대한 영향 분석

UC 이벤트의 입출력 데이터 변경이 클래스, 속성, 관계에 어떠한 영향을 미치는지 분석한다. 주요 고려 사항은 구조의 변경(클래스나 관계의 변경)이 발생하는지 파악하는 것이다. 구조의 변경은 관련된 클래스의 오퍼레이션 변경(매개변수 또는 반환 값의 변경)을 야기할 수 있으므로, 이에 대한 분석도 필요하다. 구조의 변경이 발생하지 않는다면 입출력 데이터의 변경 내역을 속성으로 반영한다. 또한, 속성이 변경될 경우에는 해당 클래스 내부 오퍼레이션의 매개변수나 반환 값이 변경될 가능성이 높으므로 이에 대한 영향 분석도 필요하다.

④ 시퀀스 다이어그램에 대한 영향 분석

UC 이벤트의 변경은 관계된 시퀀스 다이어그램 메시지의 변경으로 반영된다. UC 이벤트의 입출력 데이터의 변경은 메시지의 매개변수 또는 반환 값 변경의 근거가 된다. 시퀀스 다이어그램의 전체적인 메시지 흐름과 라이프라인은 변경된 유스케이스 시나리오 및 클래스 다이어그램을 참조한다.

⑤ 시퀀스 다이어그램 메시지의 변경 내용을 관련된 오퍼레이션에 반영

오퍼레이션에 대한 변경은 추가, 수정, 삭제로 구분할 수 있으며, 변경된 오퍼레이션이 내부적으로 복잡한 로직(logic)을 가진다면 액티비티 다이어그램에 대한 영향 분석을 수행한다. 액티비티 다이어그램의 변경 내역을 오퍼레이션으로 식별할지, 아니면 해당 오퍼레이션 내부에 구현될 로직으로 표현할지는 분석가나 설계자의 선택에 의해 결정된다.

⑥ 클래스 내의 다른 오퍼레이션에 대한 변경 가능성 파악

클래스 다이어그램, 시퀀스 다이어그램에 대한 변경을 파악하며, 그로부터 추가적으로 발생할 수 있는 변경 가능성에 대한 영향 분석이 필요하다. 이를 위해선 앞에서 이루어진 순방향 추적 과정에서 식별된 변경으로부터 영향을 받을 수 있는 클래스 내의 다른 오퍼레이션을 식별해야 한다.

③의 과정에서 클래스의 구조 또는 속성이 변경되는 경우 해당 클래스의 다른 오퍼레이션에 대해 변경 가능성을 파악해야 한다. 또한, 오퍼레이션의 매개변수 및 반환 값의 변경이 클레

스 속성의 변경을 유발한다면 해당 클래스의 다른 오퍼레이션에 대해서도 변경 가능성을 파악해야 한다. 변경 가능성이 있는 오퍼레이션의 실제 변경 여부는 해당 클래스와 오퍼레이션 정보를 기술하는 클래스 명세서에 대한 분석을 통해 이루어진다.

여러 오퍼레이션의 변경 가능성이 식별된다면 역방향 추적 순서를 정하여 분석가의 실수로 인한 오퍼레이션의 누락, 중복된 역방향 추적을 방지해야 한다. 따라서 본 논문에서는 역방향 추적을 위해 식별된 오퍼레이션들을 그림 2와 같이 큐(queue)로 관리하는 방법을 제시한다.

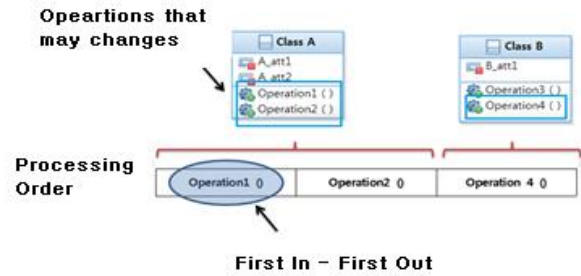


그림 2. 변경 가능성이 있는 오퍼레이션의 역방향 추적 순서
Fig. 2. Backward tracing order of changed operation

⑦ 변경 오퍼레이션으로부터 유스케이스 시나리오까지 역방향 추적

변경 가능성이 있는 오퍼레이션에 대한 실제 변경이 일어난다면, 변경이 요구되는 오퍼레이션과 관련된 시퀀스 다이어그램 메시지를 찾아내고 이에 해당하는 유스케이스 시나리오까지 추적을 수행한다. 이 과정은 변경이 요구되는 새로운 유스케이스를 역방향 추적을 통해 식별해낸 것이며, ①의 과정에서 변경 요청으로 인해 영향을 받는 유스케이스를 식별해 낸 것과 동일한 의미를 갖는다.

하나의 오퍼레이션은 여러 유스케이스 시나리오와 관련이 있을 수 있다. 따라서, 역방향 추적된 모든 유스케이스 시나리오들에 대해 순차적으로 영향 분석을 수행해야 하며, 추적된 유스케이스가 선행 또는 후행 유스케이스를 가진다면, 이에 대한 영향 분석도 수행해야 한다.

그림 1에서 제시한 과정을 반복적으로 수행한 후, 변경 가능성이 있는 오퍼레이션이 더 이상 식별되지 않는다면 단일 유스케이스에 대한 영향 분석은 종료된다. 변경 요청으로 인한 초기 ①의 과정에서 식별한 모든 유스케이스 목록에 대해 그림 1의 과정을 수행해야 하며, 최종적인 영향 분석의 종료는 다음 조건을 모두 만족해야 한다.

- A. 추적할 유스케이스가 없는 경우
- B. 그림 2에서 제시한 큐가 비어있는 경우

B의 조건은 변경 가능성이 있는 오퍼레이션이 더 이상 식별되지 않는 것과 동일한 의미이며, 두 조건 모두 그림 1의 ⑥의

과정 이후에 만족 여부를 확인한다.

3-2 회귀 테스트 케이스 선택

영향 분석이 종료된 후, 변경된 기능을 다시 검사하기 위한 회귀 테스트 케이스를 선택해야 한다. 본 논문에서는 회귀 시험의 단위인 메소드의 변경 유무를 파악하기 위해 시퀀스 다이어그램 메시지의 변경을 근거로 활용한다. 시퀀스 다이어그램 메시지에서부터 테스트 케이스를 식별한다면, 회귀 테스트 케이스를 선택할 때 시퀀스 다이어그램 메시지에 발생한 변화를 분석하는 것이 효과적이다. 시퀀스 다이어그램 메시지에서 발생할 수 있는 변경의 경우는 메시지의 추가, 수정, 삭제나 메시지 흐름의 변경과 관련이 있다. 이를 토대로 시퀀스 다이어그램 메시지에 발생할 수 있는 변경을 식별해보면 다음과 같다.

- a. 메시지 추가
 - 액터로부터 촉발되는 메시지가 추가되는 경우
 - 기존의 특정 시퀀스 내에서 메시지가 추가되는 경우
- b. 메시지 수정
 - 액터로부터 촉발되는 메시지가 수정되는 경우(매개변수나 반환 값의 변경)
 - 기존의 특정 시퀀스 내의 메시지가 수정되는 경우
- c. 메시지 삭제
 - 액터로부터 촉발되는 메시지가 삭제되는 경우
 - 기존의 특정 시퀀스 내에서 메시지가 삭제되는 경우
- d. 메시지 순서의 변경
 - 시퀀스에 포함된 메시지의 순서가 변경되는 경우
- e. 실행 결과의 변경
 - 시퀀스 다이어그램에 여러 시퀀스가 존재할 때, 특정 시퀀스의 이전 시퀀스 실행 결과가 변경되는 경우
- f. 시퀀스 선후행 관계의 변경
 - 시퀀스 다이어그램에 포함된 시퀀스의 선후행 관계가 변경된 경우

액터로부터 촉발되는 메시지가 추가되는 경우, 액터의 요청으로부터 시스템의 응답이 있기까지 새로운 시퀀스가 생성되기 때문에 이에 포함된 메시지들을 새로운 테스트 케이스로 식별해야 한다. 즉, 변경 후 수행되어야 하는 회귀 테스트 케이스가 되는 것이다. 특정 시퀀스 내에 새로운 메시지가 추가되는 경우, 기존 시퀀스 흐름에 변화를 발생시키기 때문에 기존 테스트 케이스를 회귀 테스트 케이스로 선택해야 한다.

액터로부터 촉발되는 이벤트가 수정되는 경우(매개변수나 반환 값의 변경), 기존 테스트 케이스의 입력 값이나 출력 값을 갱신하여 회귀 테스트 케이스로 선택한다. 액터로부터 촉발되지 않았지만 시퀀스 내에 포함된 메시지가 수정된 경우에도 시퀀스 내부에 변화가 발생한 것이기 때문에 기존 테스트 케이스를 회귀 테스트 케이스로 선택한다.

액터로부터 촉발되는 메시지가 삭제되는 경우는 기존의 기

능이 삭제되는 경우로 간주할 수 있으며, 이러한 기능에 대해 작성된 테스트 케이스는 더 이상 유효하지 않은 것이 된다. 즉, 회귀 테스트 케이스로 선택하지 않는다. 특정 시퀀스 내에 존재 하던 메시지가 삭제된 경우, 시퀀스는 여전히 유효하지만 시퀀스의 흐름에 변경이 발생하였으므로 관련된 메시지에 대해 기존 테스트 케이스를 회귀 테스트 케이스로 선택한다.

시퀀스 내에 포함된 메시지의 전달 순서가 변경된 경우 기존 테스트 케이스를 재실행해야 하므로 회귀 테스트 케이스로 선택한다. 시퀀스의 실행 결과에 변경이 발생한 경우, 그 시퀀스의 결과 값을 입력 값으로 이용하는 시퀀스가 영향을 받게 된다. 이렇게 영향을 받는 시퀀스에 포함된 메시지에 대해서도 기존 테스트 케이스를 회귀 테스트 케이스로 선택하고 테스트 케이스 입력 값을 갱신하도록 한다.

마지막으로 시퀀스 다이어그램에 여러 시퀀스가 포함되어 있을 때, 이들 시퀀스의 선후행 순서가 변경되는 경우에는 시퀀스의 실행을 위한 시작 조건이 달라질 수 있으므로 순서가 바뀐 시퀀스에 포함된 메시지들에 대해서 기존 테스트 케이스를 회귀 테스트 케이스로 선택한다.

지금까지 제시한 다이나믹 영향 분석 프로세스와 시퀀스 다이어그램 메시지 변경 유형을 통해 회귀 테스트 케이스를 선택하는 예를 인터넷 쇼핑몰 시스템을 통해 간단히 보이고자 한다.

IV. 사례 연구

인터넷 쇼핑몰 시스템의 여러 기능 요구사항 중, 상품 관리와 관련된 요구사항의 변경이 요청된 경우를 가정한다. 기존의 인터넷 쇼핑몰에서는 다음과 같이 상품 정보가 정의되어 있었다고 가정한다.

- 상품 정보는 상품명, 상품규격명, 상품사진, 가격, 상품설명의 속성을 포함한다.
- 위의 속성 가운데 상품명이 상품을 구별하는 키 속성으로 정의되어 있다.

기존에는 상품명을 상품을 구별하는 키 속성으로 정의하였으나, 기능 요구사항을 잘못 분석한 결과로 판명되어 이에 따른 변경 범위를 파악하고자 한다. 변경된 요구사항에 의해 상품규격 속성이 상품의 키 속성으로 포함되어 상품의 가격이 결정된다고 하자. 변경 요구사항은 표 1과 같다.

표 1. 변경 요구사항의 예

Table. 1. Example of change requirements

1. All products have different standards and price information may vary depending on the standard.
2. It is necessary to separately store attributes(product name, price) that depend on the standard.
3. Customer must be able to order the products according to the standard.

표 1에서 기술하는 내용은 상품 정보에 대한 변경을 필요로 한다. 따라서 그림 1의 ①의 과정에 의해 ‘상품등록’ 유스케이스를 변경 대상으로 식별한다. ‘상품등록’ 유스케이스의 변경 내역은 표 2와 같으며 입출력 데이터의 변경과 UC 이벤트의 추가가 발생하였다.

표 2. 상품등록 유스케이스 시나리오

Table 2. Use-case scenario of product registration

Before the change of product registration use-case scenario(Basic flow)
⑦ Input the information of the product(product name, product standard, price, product image, product description) and execute product registration function.
After the change of product registration use-case scenario
⑦ Input the information of the product(product name, product standard, price, product image, product description) and execute product registration function.
⑧ The standard information(standard name, price) of the registered product is input, and the standard information registration function is execute.

UC 이벤트의 변경 내역을 토대로 클래스 다이어그램에 대한 영향 분석을 수행한다. 입출력 데이터의 변경으로 인해 ‘상품’ 클래스의 상품규격명, 가격 속성이 삭제된다. 또한, 키 값(상품규격명)의 변경을 반영하기 위해 ‘상품’ 클래스에 종속적인 ‘상품규격’ 클래스를 추가하고 상품규격명, 가격을 속성으로 가지게 한다. 클래스 다이어그램의 변경 내역은 그림 3과 같다.

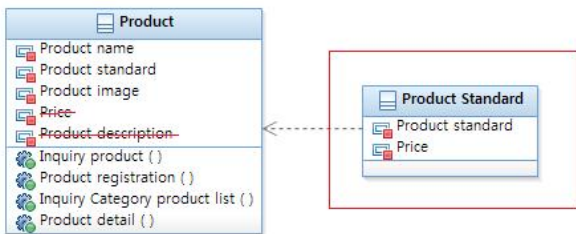


그림 3. 변경된 상품 및 상품규격 클래스

Fig. 3. Changed of product class and product standard class

클래스 다이어그램에 대한 영향 분석 후 그림 1의 ④의 과정을 수행한다. UC 이벤트의 변경으로 인해 ‘상품등록’ 메시지의 매개변수가 변경되었고, UC 이벤트 추가로 인해 ‘상품규격등록’ 메시지가 추가되었다. ‘상품규격등록’ 메시지는 추가된 UC 이벤트의 입출력 데이터를 매개변수로 사용한다(상품규격 라이프라인을 추가한다). 시퀀스 다이어그램의 변경 내역은 그림 4와 같다.

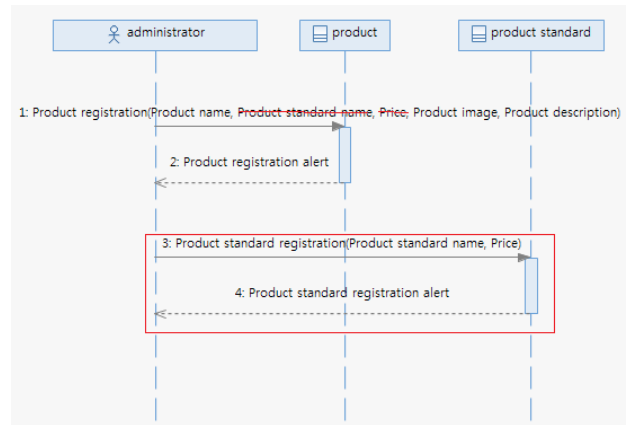


그림 4. 변경된 상품등록 시퀀스 다이어그램

Fig. 4. Changed sequence diagram of product registration

시퀀스 다이어그램 메시지의 변경으로 인해 ‘상품’ 클래스의 상품등록 오퍼레이션이 변경되고, ‘상품규격’ 클래스에 상품규격등록 오퍼레이션이 추가된다. 순방향 추적을 통해 산출물들의 변경이 파악되면 그림 1의 ⑥의 과정을 통해 그로부터 영향을 받을 수 있는 오퍼레이션이 없는지 파악해야 한다. ‘상품’ 클래스와 관련된 변경 가능성이 있는 오퍼레이션은 ‘상품내역조회’, ‘카테고리상품목록조회’, ‘상품상세조회’ 오퍼레이션이며 이들의 정보는 표 3과 같다.

표 3. 상품 클래스의 오퍼레이션 정보

Table 3. Operation information of product class

Operation	Parameter	Return Value
Inquiry product	Product name	Product name, Price
Inquiry Category product list	Category name	Product name, Registration date
Product detail	Product name	Product name, Product size, Product image, Price, Product description, Registration date

표 3의 오퍼레이션들을 본 논문에서 제시한 큐에 삽입한 후, 오퍼레이션의 매개변수 및 반환 값을 확인하여 실제 변경 여부를 판단한다. 클래스의 속성(상품규격명, 가격) 삭제로 인해 변경되는 오퍼레이션은 ‘상품내역조회’, ‘상품상세조회’ 오퍼레이션이며 그림 1의 ⑦의 과정에 의해 두 오퍼레이션을 기술하는 유스케이스까지 역추적을 하여 추가적인 변경 가능성에 대해 파악해야 한다. 큐에 삽입한 순서대로 ‘상품내역조회’ 오퍼레이션에 대해 역방향 추적을 수행하면 ‘장바구니상품담기’와 ‘온라인입금처리’ 유스케이스 시나리오가 변경되어야 함을 파악할 수 있다. 이 과정은 초기 변경 요구사항으로 인해 영향을 받는 다른 요구사항(미처 발견하지 못했던)을 식별해낸 것이며, 식별해낸 요

구사항(유스케이스)에 대해 그림 1의 ①의 과정 이후부터 영향 분석을 반복적으로 수행해야 한다. 표 1의 변경 요구 사항에 대한 최종적인 영향 분석 결과는 표 4와 같다.

표 4. 영향 분석 결과

Table. 4. Impact analysis results

Artifact	Changed Items
Use Case	Product registration, add product to shopping cart, online deposit processing, product details inquiry, product order
Class Diagram	Product, product standard, shopping cart, order details
Sequence Diagram	product registration, product inquiry, product detail, product standard registration, standard inquiry, standard information, add product to cart, save product to cart, delete product from cart, save order details, order details inquiry, deposit information check, payment completion

회귀 테스트 케이스 선택을 위해서는 영향 분석의 결과를 토대로 시퀀스 다이어그램 메시지의 변경 유형을 파악해야 한다. 시퀀스 다이어그램 메시지의 변경 유형과 변경 내용은 표 5와 같다.

표 5의 ‘상품등록’ 메시지는 액터로부터 촉발되는 메시지가 수정되는 경우로 ‘상품등록’ 메시지의 매개변수 가운데 ‘상품규격명, ‘가격’이 삭제되었다. 이 경우는 ‘상품등록’이라는 테스트 케이스의 입력 데이터가 변경된 경우로서 회귀 테스트 케이스로 선택되어야 한다. ‘주문내역저장’ 메시지는 기존의 특정 시퀀스 내의 메시지가 수정되는 경우로 액터가 시스템에게 입력하거나 시스템이 출력하는 데이터상에는 변화가 없지만, ‘주문등록’이라는 이벤트를 처리하기 위해 내부적으로 호출되는 메시지의 매개변수에 변경이 발생한 경우이다. 사용자가 이용 및 확인할 수 있는 데이터에 변경이 발생하지 않더라도 내부적인 로직에 변경이 발생했다면, 해당 기능을 재차 검사할 필요가 있기 때문에 회귀 테스트 케이스로 선택하는 것이 합당하다. 선택된 ‘상품등록’, ‘주문내역저장’의 테스트 케이스는 표 6, 7과 같다. 예로 보인 회귀 테스트 케이스 선택의 경우는 가장 전형적인 경우이며, 이 외에도 메시지가 삭제되거나 순서가 변경되는 등의 경우도 회귀 테스트 케이스 선택 대상으로 고려해야 한다.

표 5. 시퀀스 다이어그램 메시지의 변경 유형

Table. 5. Change types of sequence diagram message

Changed Item name	Change type	Changed detail(parameters / return values)
product registration	change parameters	product name, product standard name , product description, price
product inquiry	delete event	
product detail	change return values	product name, product standard name , product image, price ; product description
product standard registration	add event	product standard name , price
standard inquiry	add event	product name , product standard name / product name , product standard name , price
standard information	add event	product name / product standard name , price
add product to cart	change return values	name, recipient name, receiving address, receiving contact, product name, product standard name , price, quantity
save product to cart	change parameters/return values	member ID, product name, product standard name , quantity / product name, quantity, product standard name , price
delete product from cart	change parameters	member ID, product name, product standard name , quantity
save order details	change parameters	member ID, product name, product standard name , quantity
order details inquiry	change parameters	member ID, order number, product name, product standard name , quantity
deposit information check	change return values	product name, product standard name , price, quantity
payment completion	change return values	name, recipient name, receiving address, receiving contact, product name, product standard name , price, quantity

표 6. '상품등록' 회귀 테스트 케이스

Table. 6. Regression test case of product registration

No.	Test method	Classification	description	expected result
1	createItem(String item_name, String item_Standard, String ctgr_name, String cost, String description)	normal	test case with normal data	store the delivered parameter values and outputs a normal processing message
2	createItem(String item_name, String item_Standard, String ctgr_name, String cost, String description)	abnormal	parameter is null	print an error message
3	createItem(String item_name, String item_Standard, String ctgr_name, String cost, String description)	abnormal	duplicate item_name	print message that " item_name is duplicated"

표 7. '주문내역저장' 회귀 테스트 케이스

Table. 7. Regression test case of order detail

No.	Test method	Classification	Test case	expected result
1	createOrderDetail(String order_no, String user_id, String item_list)	normal	test case with normal data	stores the passed parameter values and returns true
2	createOrderDetail(String order_no, String user_id, String item_list)	abnormal	parameter is null	prints an error message and returns false

본 논문에서 제시한 방법은 다이나믹 영향 분석 프로세스를 통한 명확한 영향 분석 결과를 기반으로 하여, 회귀 테스트 케이스 선택의 정확성과 효율성 측면에서 장점을 보인다. 또한, 회귀 테스트 케이스 선택에서 활용한 영향 분석 결과(표 5)를 사용해 회귀 테스트 케이스의 입력 값과 기대 결과를 쉽게 설계할 수 있다.

회귀 테스트 케이스 선택의 관한 기존 연구[13-16]들은 UML 모델 간의 관계성을 정의하고 변경 유형을 분류하여 테스트 케이스 선택에 활용하고 있다. 기존 연구들은 테스트 케이스 선택의 정확성을 높이는 데는 장점이 있으나, UML 모델의 변경 원인과 테스트 케이스가 선택되어야 하는 근본적인 원인인 요구사항 변경에 대해서는 다루지 않고 있다. 본 논문에서는 다이나믹 영향 분석 과정과 영향 분석 결과를 토대로한 회귀 테스트 케이스 선택 과정을 통해 기존 연구에서 다루지 않았던 요구사항 변경에 대한 문제점도 해결하였으며, 시퀀스 다이어그램 메시지를 활용하여 테스트 케이스 선택의 근거를 명확하게 밝혀 정확성 측면에서도 장점을 보인다.

리오에 대한 분석을 통해 변경 요청에 포함된 요구사항을 빠르게 파악하였고, 이를 토대로 각 모델이 받는 영향과 역으로 각 모델의 변경이 요구사항의 미치는 영향을 명확하게 식별하였다. 또한, 회귀 테스트 케이스 선택의 정확성과 효율성을 높이기 위해 시퀀스 다이어그램 메시지를 테스트 케이스 선택의 기준으로 삼아, 메소드 수준의 회귀 테스트 케이스를 선택에 있어서 중요한 근거를 제공해줄 수 있다.

본 논문에서 제시한 방법은 설계, 구현 단계 산출물 작성의 근거가 되는 요구사항 명세서에 대한 변경을 식별하여, 모델이나 코드가 변경되는 근본적인 원인을 식별하였다. 따라서, 회귀 테스트 케이스 선택 시 변경 유형이나 변경 대상, 테스트 케이스 설계 방법에 제한을 받지 않는 확장성이 높은 방법이라 볼 수 있다. 또한, 테스트 케이스 설계의 기초가 되는 테스트 입력 값, 기대 결과, 테스트 대상 선정 시 영향 분석 결과를 활용할 수 있어, 회귀 테스트 케이스의 선택 뿐만 아니라 회귀 테스트 케이스 명세서 작성 시 시간과 비용 측면에서 장점이 있다.

VI. 결 론

회귀 시험은 변경된 소프트웨어의 완전성을 검증하기 위해 수행되는 것으로 정확한 테스트 케이스 선택이 매우 중요하다. 효율적이고 안전한 회귀 테스트 케이스의 선택은 영향 분석 과정의 정확함이 필수적으로 동반된다. 본 논문에서 제시한 방법은 다이나믹 영향 분석 프로세스를 통해 UML 다이어그램에 대한 변경 사항을 식별하고 분류하였으며, 식별한 변경 사항을 활용하여 회귀 테스트 케이스를 선택하였다. 영향 분석 과정에서는 유스케이스 시나

감사의 글

본 연구는 충남대학교 학술연구비에 의해 지원되었음

참고문헌

- [1] Y.R. Kwon, Software testing, *Life and Power Press*, 2010
- [2] G. Rothermel and M.J. Harrold, "A safe, efficient regression test selection technique", *ACM Trans. on Software Engineering and Methodology*, Vol.6, No.2, pp.173-210,

April 1997

[3] Filho A. O., "Change Impact Analysis from Business Rules", *ICSE'10*, pp.353-354, May 2010

[4] Harrold M. J., "Testing Evolving Software", *Journal of Systems and Software*, Vol.47, No.3, pp.173-181, July 1999

[5] G. Rothermel and M.J. Harrold, "Analyzing Regression Test Selection Techniques", *IEEE Trans. Software Engineering*, Vol.22, No.8, pp.529-551, August 1996

[6] D.C. Kung, J. Gao, P. Hsia, J. Lin and Y. Toyoshima. "Class firewall, test order, and regression testing of object-oriented programs", *Journal of Object-Oriented Programming*, Vol.8, No.2, pp.51-65, May 1995

[7] M.A. Chaumun, H. Kabaili, R.K. Keller and F. Lustman, "A change Impact Model for Changeability Assessment in Object Oriented Software Systems", *Science of Computer Programming*, Vol.45, No.3, pp.155-174, December 2002

[8] M.K. Abdi, H. Lounis and H. Sahraoui, "Analyzing Change Impact in Object-Oriented Systems" *32nd EUROMICRO Conference on Software Engineering and Advanced Applications*, pp.310-319, August 2006

[9] G. Rothermel, M.J. Harrold, and J. Dedhia. "Regression test selection for C++ software", *Journal of Software Testing, Verification, and Reliability*, Vol.10, No.6, pp.77- 109, June 2000

[10] M.J. Harrold, J. A. Jones , T. Li , D. Liang , A. Orso, M. Pennings, S. Sinha, S.A. Spoon and A. Gujarathi, "Regression test selection for Java software", *Proc. of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications*, pp.312-326, November 2001

[11] Y. K. Jang, Y. R. Kwon, "A Study on the Regression Testing of Object-Oriented Software", *Korea Information Science Society 24(2)*, pp.569-572, October 1997

[12] C. Lee, Y. Cheong, "Dynamic Impact Analysis Method using Use-case and UML Models on Object-oriented Analysis", *Journal of KIISE*, Vol.43, No.10, pp.1104-1114, October 2016

[13] Briand L. C., Labiche Y., Soccar G., "Automating Impact Analysis and Regression Test Selection Based on UML Designs", *Proceedings of the International Conference on Software Maintenance*, pp.252-261, October 2002

[14] Orest Pilskalns, Gunay Uyan, Anneliese Andrews, "Regression Testing UML Designs", *2006 22nd IEEE International Conference on Software Maintenance*, pp.254-264, September 2006

[15] Mohammed Al-Refai, Walter Cazzola, Sudipto Ghosh, "A Fuzzy Logic Based Approach for Model-Based Regression Test Selection", *2017 ACM/IEEE 20th International*

Conference on Model Driven Engineering Languages and Systems (MODELS), pp.55-62, September 2017

[16] Sumit Dahiya, Rajesh K. Bhatia, Dhavleesh Rattan, "Regression test selection using class, sequence and activity diagrams", *IET Software*, Vol.10, No.3, pp.72-80, June 2016

[17] D.Y. Kim, C. Youn, "Methodology for traceability Management and Impact Analysis for Efficient Change Management in Object-oriented Development", *Journal of KIISE*, Vol.42, No.3, pp.328-340, March 2015



이 찬(Chan Lee)

2015년: 충남대학교 컴퓨터공학과
졸업(학사)

2017년: 충남대학교 대학원 데이터 및
소프트웨어공학전공 석사

2017년~현재: 충남대학교 대학원 데이터 및 소프트웨어
공학전공 박사과정 재학

※ 관심분야 : 소프트웨어 변경관리, 요구사항 분석, UML
모델 등



윤 청(Cheong Youn)

1979년: 서울대학교 물리학과(학사)

1983년: Illinois State University
전산학과(석사)

1988년: Northwestern University
전산학과(박사)

1983년~1985년: Wayne State University 전산학과 전임강사

1985년~1987년: Northwestern University 전산학과 전임강
사

1988년~1993년 Bell Communications Research 선임연구원

1993년~현재 충남대학교 컴퓨터공학과 교수

※ 관심분야 : 소프트웨어공학, 객체지향 개발방법론 등