키워드 기반 자동 테스트 구현 및 적용 사례

이영석 하영민
삼성SDS(주) 품질경영팀
youngss@samsung.co.kr spped@samsung.co.kr

Software Testing by a keyword driven test automation method and Effects

Young-Seok Lee Young-Min Ha
SAMSUNG SDS Co., LTD.
Quality Management Team

요 약

소프트웨어의 본질인 변경 문제로 인해 상용 테스트 도구의 단순 적용만으로는 자동 테스트의 실질적인 효과를 기대하기 어려운 것이 현실이다. 이러한 문제를 해결하기 위해 변경에 영향 받지 않는 다양한 자동 테스트 기법이 시도 되어 오고 있다. 그 중에서 가장 주목받는 것으로 키워드 기반 자동 테스트(Keyword Driven Automated Test)를 들 수 있으며 이 방법의 궁극적인 목적은 대상 소프트웨어의 변경에 따른 테스트 자원의 유지보수를 쉽게 하기 위한 것이라 할 수 있다. 테스트 자원의 구축, 실행 및 발생하는 변경으로 인한 유지보수까지의 전 과정에서 키워드 기반 자동 테스트가 보다 효율적이며 다른 프로젝트, 제품 테스트에서도 일부 함수 추가, 재정의만으로 재사용 가능한 장점을 가지고 있다. 키워드 기반 자동 테스트를 위해서는 기존의 테스트 도구를 사용하여 대상 소프트웨어에 맞게 테스트 도구가 제공하는 자체 프로그래밍 언어를 사용하여 개발해야 한다.

1. 서론

최근 들어 개발되는 정보 시스템의 규모가 점점 커지고 복잡해짐에 따라 소프트웨어 자동 테스트의 필요성이 증가하고 있으며 자동 테스트 도구에 대한 관심이 높아지고 있다. 하지만 소프트웨어의 본질인 변경 문제로 인해 상용 테스트 도구의 단순 적용만으로는 자동 테스트의 실질적인 효과를 기대하기 어려운 것이 현실이다. 이러한 문제를 해결하기 위해 변경에 영향 받지 않는 다양한 자동 테스트 기법이 시도 되어 오고 있다. 그 중에서 가장 주목받는 것으로 키워드 기반 자동 테스트(Keyword Driven Automated Test)를 들 수 있으며 이 방법의 궁극적인 목적은 대상 소프트웨어의 변경에 따른 테스트 자원의 유지보수를 쉽게 하기 위한 것이라 할 수 있다.

이 논문에서는 기존 자동 테스트 도구를 사용하여 키워드 기반 자동 테스트를 구현한 사례와 그 적용 효과를 제시하고자 한다.

2. Record & Playback의 문제점

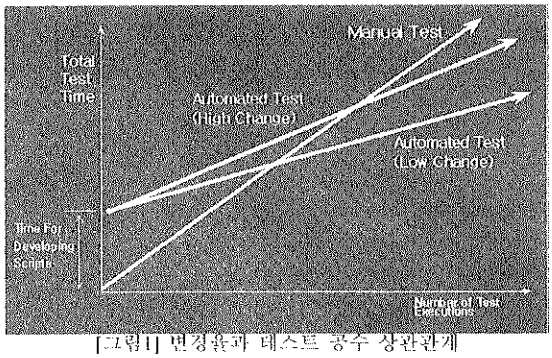
2-1. 스크립트 유지보수 어려움

현재 상용 소프트웨어 자동 테스트 도구가 일반적으로 Record & Playback 방식으로 되어있어 GUI화면상에서 테스트 내용을 그대로 수록한 스크립트는 손쉽게 생성 할 수 있으나 다수의 시나리오가 다수의 스크립트로 만들어짐에 따라 테스트 대상 소프트웨어의 기능, 사양이 변경되면 변경 이전에 만들어진 다수의 스크립트를 새로운 내용으로 수정, 변경해야 한다. 이렇게 추가로 발생하는 스크립트의 유지보수 작업은 자동 테스트 자원 구축 이후 테스트 비용을 증가 시키는 직접적인 원인이 된다.

2-2. 자동 테스트 적용 시점의 제한

대상 소프트웨어의 개발단계에서 프로그램 변경율이 너무 잦은 경우에는 위에 언급한 유지보수 비용으로 인해 자동 테스트가 수작업 테스트보다 비효율적이기 때문에 프로그램 변경율이 극히 작은 경우(통상 5%이하)에만 자동 테스트를 적용할 수 있다. 즉, 개발 후반 이후에만 적용할 수 있다.

3. 소프트웨어 변경율과 자동 테스트와의 관계



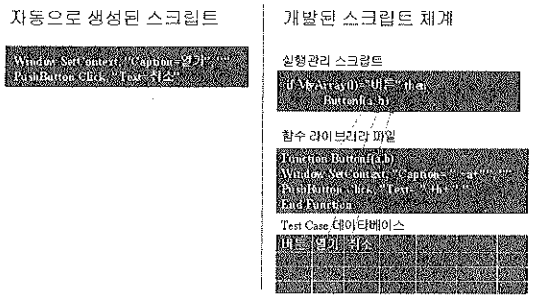
[그림1] 변경율과 테스트 공수 상관관계

위 그림1에서와 같이 변경율이 높을수록 위 그림의 자동 테스트 그래프의 기울기가 커지게 되며 수동 테스트와 유효 반복 실행 횟수가 높아지게 된다. 대상 소프트웨어의 변경이 자동 테스트에 심각한 영향을 미친다는 것을 알 수 있다.

4. 키워드 기반 자동 테스트(Keyword Driven Automated Test) 개념

키워드 기반 자동 테스트에서는 스크립트의 내용을 키워드와 함수로 분리하는 함수화 전략을 지향 한다.

항상 키워드와 함수는 일대일 관계이다. 키워드 부여에 규칙이 있는 것은 아니지만 사용자가 쉽게 테스트 내용을 알 수 있도록 부여함이 바람직하다. 그림2에서 보는 바와 같이 좌측(자동으로 생성된 스크립트)이 Record & Playback 방식에 의해 만들어지는 스크립트의 내용이며 우측(개발된 스크립트 체계)은 키워드 기반 방식을 나타내며 우측 중간의 함수 라이브러리 파일에 저장된 함수는 모두 두개의 값에 의해 테스트를 수행할 수 있다. 즉, 우측 하단의 "열기"와 "취소"가 값이며 맨 앞에 나타나는 "버튼"은 키워드 이다.

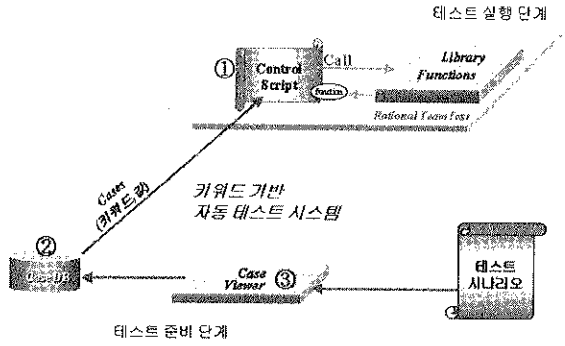


[그림2] 키워드 기반에서의 데이터 수록 유형

특정 함수에 실제로 빈번히 사용되는 대표적인 값의 유형은 특정 윈도우의 Caption값, 메뉴 ID값,특정 버튼의 Text값(또는 Caption값),좌표값(Rational TeamTest에서는 Coords값)등 이다.

5. 키워드 기반 자동 테스트 시스템 개요

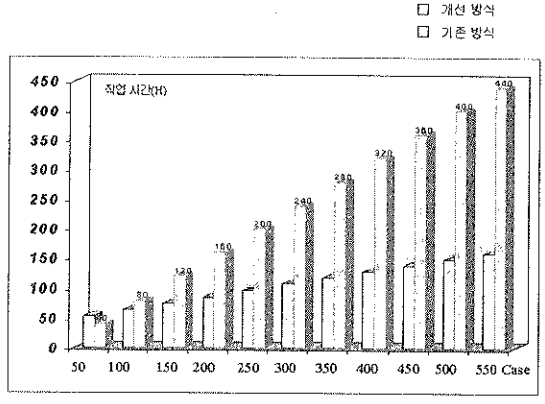
그림3 에서 보는 바와 같이, 시스템은 크게 자동 테스트 실행 단계와 자동 테스트 준비 단계가 필요하며 자동 테스트 준비 단계에서는 읽기 쉬운 문장으로 작성된 테스트 시나리오를 위 그림의 3번 Case Viewer를 사용하여 테스트 케이스를 하나씩 위 그림의 2번 데이터베이스에 수록 저장한다. 이때 부분적으로 테스트 도구에 의해 스크립트를 생성하고 그 값과 함수유형을 확인 한다. 이 테스트 케이스 데이터베이스에는 테스트 동작의 최소 단위인 케이스를 행(Row)단위로 저장된다.



[그림3] 키워드 기반 자동 테스트 시스템

6. 키워드 기반 자동 테스트 효율 분석

6-1. 테스트 자동화 효율



[그림4] 키워드 기반 자동 테스트 적용 효과

테스트 자동화 공수는 자원 구축부터 테스트 실행까지의 전체 테스트 공수를 의미하며 테스트 자원의 유지보수를 고려하지 않은 상태에서의 순수 테스트 공수만을 의미 한다. 자동 테스트 대상 시스템은 MIS 업무를 구현한 A시스템으로 시스템 규모면에서 전체 화면수(프로그램 본수)가 총 780개 였다.

위 그림에서 보는 바와 같이, 550개 케이스를 자동화 하는데 들어간 공수를 비교해 보면,

케이스 수	Record & Playback	Keyword Dirven Method
550개	440 시간	159.5 시간

(표1) 테스트 자원 구축 투입 공수 비교표

으로 나타나며 효율을 측정해 보면,

(440-159.5)/440 = 0.6375 = **63.75%**

약 64%의 정량적인 공수 절감 효과를 나타내고 있으며 이러한 효과의 정성적인 원인으로는

가. 대상 어플리케이션의 UI가 정형화 되어 있었다.

나. 단순 반복적인 레코딩 작업에 비해 값과 키워드의 입력 작업이 보다 수월 하였다.

6-2. 테스트 자원의 유지보수 효율

이 효율을 측정하기 위해서는 기존의 Record & Playback 방식에서의 아래값을 구하고 Keyword Driven 방식에서의 유지보수 공수를 구한 다음 비교 하여야 한다.

① 스크립트 재개발에 드는 시간(RD)

발생한 어플리케이션 변경에 따라 기존의 스크립트를 폐기하고 다시 만드는데 드는 시간

② 어플리케이션 변경율(C)

어떤 테스트 대상 어플리케이션에서 화면 단위로 만든 화면별 스크립트의 재개발을 유발시킬 만한

변경이 발생하는 빈도. 화면 변경율과 같음.

RD = 어플리케이션 변경율(C) × 평균 단위 스크립트 생성 시간 × 전체 스크립트수

C = 화면변경율 = 변경된 화면수 ÷ 전체 화면수

실제로 반복 테스트를 실행하다가 어플리케이션에 변경이 발생한 비율은 전체 화면수 780개 중에서 변경이 발생한 화면은 23개 화면으로 2.94% 로 나타났으며 위 수식을 사용하여 스크립트 재개발 공수를 계산해 보면

항목	측정값
어플리케이션 변경율(C)	0.029
평균 단위 스크립트 생성 시간	0.3시간
전체 스크립트수	550

(표2) 항목별 측정값

0.029 × 0.3 × 550 = **4.785시간**

으로 측정되었으며 약 5시간 소요됨을 알 수 있다.

Keyword Driven 방식에서는 Test Case 데이터베이스에서 변경 발생에 해당되는 행(ROW)를 찾아 변경된 내용으로 값을 바꾸거나 행전체를 삭제, 새로운 내용으로 삽입하는 작업만 수행하는 데 드는 전체시간이 0.75시간(약 45분) 소요 되었다.

위 결과를 정리 비교 하면 표3에서 보는 바와 같이 Record & Playback방식의 유지보수 공수를 1로 가정했을 때 Keyword Driven 방식에서는 0.156(약 16% 수준)으로 공수 절감 효과가 나타난다.

(1-0.156)/1 = 0.844 = **84.4%**

구 분	Record & Playback	Keyword Dirven Method
2.94% 변경시 테스트 자원 유지보수 공수	4.79 시간	0.75 시간
효율	1	0.156

(표3) 유지보수 효율 비교표

7. 결론

항 목	Record & Playback Method	Keyword Dirven Method
테스트 자원 구축비용	1	0.362
테스트 자원 유지보수 비용	1	0.156

(표4)자동 테스트 적용시 효율 비교표

위 효율 분석 내용을 보면 자동 테스트 자원의 구축부터 실행중 발생하는 변경으로 인한 유지보수까지 전 과정에서 Keyword Driven 방식이 효율적임을 알 수 있다. 이외에도 한번 사용한 실행 관리 프로그램, 함수, 키워드 등은 다른 프로젝트, 제품 테스트에서도 일부 함수 추가, 재정의만으로 재사용 가능한 장점을 가지고 있다. 키워드 기반 자동 테스트를 위해서는 기존의 테스트 도구를 사용하여 대상 소프트웨어에 맞게 테스트 도구가 제공하는 자체 프로그래밍 언어를 사용하여 개발해야 한다. 또한 데이터베이스에 키워드와 값을 입력하기 위하여 부분적으로 도구를 사용한 스크립트를 생성한 다음 값을 일일이 확인, 입력해야하는 번거로움이 있다. 테스트의 자동화율을 향상시키기 위한 향후 연구과제로 이러한 번거로운 키워드, 값 수작업 입력 방식을 스크립트로부터 또는 테스트 도구의 실행 엔진 부분에서 부터 자동으로 데이터베이스에 저장되는 방안을 연구할 예정이다.

8. 참고문헌

[1] Mark Fewster, Software Test Automation : Effective Use of Test Execution Tools, 1999

[2] 保田勝通, 소프트웨어 품질보증의 개념과 실제, 1992

[3] 이영석, 테스트 자동화 최적 적용시기 결정 방법 연구, 1999

[4] Paul C. Jorgensen , Software Testing : A Craftsman's Approach, 1995

[5] Elfriede Dustin, Automated Softwate Testing : Introduction, Management, and Performance, 1999