

GUI 애플리케이션의 단위 테스트를 위한 테스트 자동화 플랫폼의 설계

박환석, 류호동, 이우진
경북대학교 전자전기컴퓨터학부
e-mail : alwaysgohs@gmail.com

A design of test automation platform for unit testing on GUI application

Hwanseok Park, Hodong Ryu, Woo Jin Lee
School of EECS, Kyungpook National University

요 약

최근 IT 기기는 GUI를 지원하여 사용자들에게 편의성을 제공하였지만, 개발자들에게는 테스트에 어려움을 겪게 만들었다. 기존의 GUI 자동화 테스트기법인 Record Play-back 기법과 새로운 GUI 테스트 도구인 Sikuli가 등장 하였지만 기능이 추가되고 변경되는 소프트웨어를 테스트하기 부족하다. 본 논문에서는 GUI 애플리케이션에서 점진적인 테스트를 위해 단위 테스트인 각 테스트 케이스를 만들고 연결하는 시나리오 편집기와 시나리오 수행과 테스트 케이스 결과를 검증하는 수행 및 검증기로 구성된 자동화 테스트 도구를 설계한다.

1. 서론

최근 IT기기가 지원하는 GUI 환경은 사용자 편의성을 증가시켰지만 개발자에게는 GUI 테스트라는 새로운 과제를 던져 주었다. 기존의 테스트와 달리 GUI 프로그램에서는 특정한 위치와 매치되는 키보드 혹은 마우스 입력 이벤트를 입력을 받고 이에 대한 결과로 화면상의 아이콘, 다이얼로그박스안의 GUI요소 및 메시지 박스로 나타낸다. 이는 마우스를 이용하여 단순 반복적으로 테스트를 해야 하며 테스트 결과 검증이 어려운 문제점이 있다.

이런 문제를 해결하기 위한 노력으로 키보드 및 마우스 입력 이벤트들을 기록하고 재현하는 Record-Playback기법과 이미지 비교를 위해 컴퓨터 비전을 이용한 테스트 자동화 도구인 Sikuli[1]가 등장했다. Record-Playback기법은 기능 구현이 쉬워 많은 GUI 자동화 테스트 도구에 사용되었으나, 결과 검증을 할 수 없어 재현 후에는 사람이 직접 결과를 검증해야 한다는 단점이 존재한다. Sikuli는 컴퓨터 비전을 이용한 새로운 GUI 테스트 도구로써 컴퓨터 비전을 통해 GUI 요소의 특징점을 참고하여 화면상의 동일한 GUI 요소들을 찾는다. 하지만 적절한 실행기를 제공하지 않아서 다수의 시나리오를 관리할 수 없는 단점이 있다.

본 연구에서는 이러한 문제를 해결하고자 단위 테스트 관점에서의 GUI를 테스트를 하기 위해서, 사용자의 입력 이벤트를 기록하고 검증을 위한 예상이미지를 지정하여 테스트 케이스를 만들고 시나리오를 만드는 편집기와 시나리오를 수행하고 각 테스트 케이스를 검증하는 테스트

수행 및 검증기로 구성된 테스트 도구를 제안한다. 이를 이용하여 결과 검증이 가능한 GUI의 단위 테스트와 다수의 단위테스트로 구성된 시나리오를 점진적인 테스트과 다수의 시나리오를 수행하며 시나리오 수정과 테스트 케이스의 재사용으로 효율적인 테스트가 가능하다.

2. 기존 GUI 자동화 테스트 기법

Record-Playback기법은 기능 구현이 쉬워 많은 GUI 자동화 테스트 도구에 사용되었으나, 다양한 시나리오를 테스트하기 위해서 시나리오의 수만큼의 기록과정을 거치고 그에 대한 명세가 필요하며, 소프트웨어 개발과정에서 계속적으로 변화하는 UI에 대해서 기존의 이벤트 입력을 재사용하지 못하고 다시 입력을 반복해야 한다[2]. 그리고 결과 검증이 없어 재현 후에는 사람이 직접 결과를 확인해야 한다는 단점이 존재한다.

Sikuli는 컴퓨터 비전을 이용한 새로운 GUI 테스트 도구로써 테스트 대상 GUI 요소를 캡처하여 영역을 지정한 후 컴퓨터 비전을 이용하여 GUI 요소의 특징점을 참고하여 화면상의 동일한 GUI 요소들을 찾는다. Record-Playback에는 어려웠던 결과 검증이 가능하다는 장점이 있다. 기본적으로 키보드 혹은 마우스 입력 이벤트에 대한 좌표정보를 활용하지 않고 비전을 활용함으로써 개발 과정에서 변경되는 UI 및 GUI 요소의 위치 변화에 대한 입력을 반복하지 않고 재사용할 수 있다. 하지만 별도의 명세가 필요하고 스크린에 동시에 다수의 GUI 요소를 캡처하여 영역을 지정할 수 없다. 그리고 컴퓨터 비전의 한계로 의도하지 않는 이미지를 검색할 수 있어 정확한 테스트

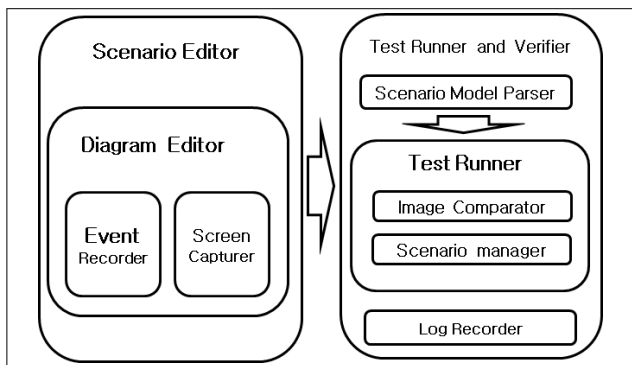
트가 어렵다. 시나리오에 대한 적절한 실행기를 제공하지 않아서 다수의 시나리오를 수행할 수 없다.

3. GUI 단위 테스트 : GUnit

단위 테스트는 특정 모듈이 의도된 대로 정확히 작동하는지 검증하는 절차이다.[3] xUnit의 단위 테스트와 달리 GUI의 단위 테스트의 테스트 케이스는 GUI 다이얼로그에서 어떤 모듈의 동작의 입력은 버튼에 대한 클릭이 되며 그 결과는 아이콘, 새로운 다이얼로그 또는 텍스트박스의 텍스트로 나타난다. 그리고 결과는 기존의 예상결과와 검증하고 문제가 없으면 단위 테스트는 종료된다. 개발과정이 계속될수록 단위 테스트의 수가 증가하게 되며 각각의 단위 테스트들은 관계를 가지게 된다. GUnit에서는 시나리오 편집기를 통하여 다수의 단위테스트인 테스트 케이스들의 관계를 정의하여 시나리오를 만든다. 테스트 수행 및 검증기에서는 시나리오 편집기에서 만든 각각의 테스트 케이스와 시나리오를 수행하며 그 결과를 검증한다. 시나리오의 각 단계에서 각각의 검증은 자동적으로 수행되며, 검증의 결과 역시 자동적으로 저장된다, 아울러 각각의 검증 결과는 다음 시나리오 선택과 같은 방법으로 수행 과정에 적용된다.

4. GUnit의 구조

GUnit은 다수의 테스트 케이스와 그 관계를 정의하는 시나리오 편집기와 정의된 시나리오를 수행하며 그림1과 같이 각 테스트 케이스를 검증하는 테스트 수행 및 검증기로 나누어진다.



(그림 1) GUnit의 구조

4.1 시나리오편집기

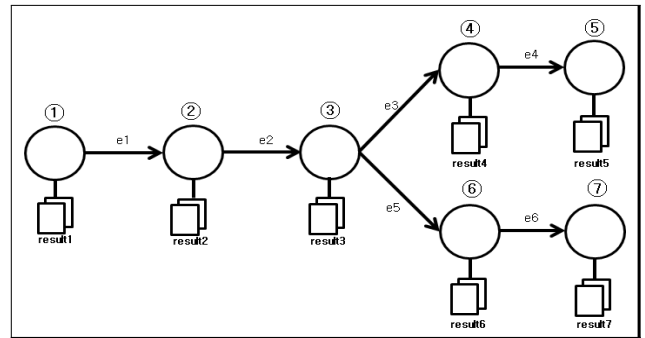
시나리오 편집기에서는 테스트 케이스 작성을 위해 이벤트 기록기를 이용하여 사용자의 마우스 또는 키보드 이벤트를 기록하고 테스트검증을 위해 화면캡처기를 이용하여 각 테스트 케이스에서 통과 여부를 판단하는 조건이 될 이미지를 캡처하고, 이들 이미지들이 실제 수행과정에서 비교될 화면상의 특정 영역, 즉 이미지와 비교될 관심영역을 지정한다. 이 관심영역은 하나 이상이 될 수 있으며 두 개 이상의 관심영역에 합연산과 곱연산과 같은 논리연산의 개념을 적용하는 것도 가능하다. 각 단계에서 테스트 케이스의 입력 이벤트기록과 예상이미지의 관심영역이 결정되면 테스트 노드들이 수행될 순서를 정의하여 하

나의 시나리오로 작성한다.

4.2 테스트 수행 및 검증기

테스트 수행 및 검증기에서는 시나리오 편집기에서 작성된 시나리오 모델을 읽어서 파서를 이용하여 분석한 후 정의된 시나리오를 시나리오 매니저에 의하여 순차적으로 테스트를 수행한다. 각 테스트 케이스의 이벤트들을 재현하면서 관심영역이 정의된 예상이미지와 실제이미지를 비교한 후 그 결과가 일치할 경우와 불일치할 경우를 시각적으로 구분하고 일치할 경우 시나리오를 진행하며 불일치될 경우 시나리오 목록에 있는 다음 시나리오를 수행한다. 마지막으로 시나리오 수행 과정에서 생성된 결과는 로그로 저장된다.

아래의 그림 2는 시나리오 편집기에서 ①→②→③→④→⑤, ①→②→③→⑥→⑦로 정의된 시나리오를 시나리오 매니저에 의하여 순차적으로 테스트케이스를 수행하는 과정을 도식화하고 있다.



(그림 2) GUnit의 시나리오 수행

그림에서는 테스트 시나리오①→②→③→④→⑤를 수행 후 다음 시나리오 ①→②→③→⑥→⑦을 수행한다.

5. 결론

기존의 단위테스트를 GUI 환경에 적용한 GUnit은 시나리오 편집기와 테스트 수행 및 검증기를 이용하여 테스트 케이스와 시나리오를 생성하여 GUI환경에서 점진적인 단위 테스트를 수행할 수 있다. 차후 이를 구현하기 위한 노력으로 시나리오 수행 알고리즘과 이미지 비교를 위한 비전 연구가 수행되어야 한다.

ACKNOWLEDGMENT

본 연구는 지식경제부 및 정보통신산업진흥원의 IT융합 고급 인력과정 지원사업의 연구결과로 수행되었음(NIPA-2012-C6150-1202-0011)

참고문헌

- [1] Tsung-Hsiang Chang, Tom Yeh, Robert C. Miller "GUI Testing Using Computer Vision", CHI 2010:Pixels and Perception, 2010
- [2] 마영철, 최은만, "XML을 이용한 안드로이드 GUI 테스트 자동화 방안", 한국컴퓨터종합학술대회 논문집 Vol.38, No.1(B), 2011
- [3] Unit testing, http://en.wikipedia.org/wiki/Unit_testing