

시나리오기반의 모바일 어플리케이션 소프트웨어의 GUI 테스트 자동화 도구 설계 및 구현

정일재*, 황선명*, 신규상**

*대전대학교 컴퓨터공학과

**한국 전자통신연구원, 임베디드 S/W 연구단

e-mail : sky7681@nate.com

GUI test automation tool design of mobile application software based scenario and implementation

Il-Jae Jung*, Sun-Myung Hwang* GyuSang Shin**

*Dept of Computer Engineering, Daejeon University

**Embedded S/W Research Division, ETRI

요 약

모바일 어플리케이션 소프트웨어의 GUI는 눈으로 보고 교감하는 것에 끝나지 않고 나날이 발전하여 소프트웨어의 성공여부를 결정하는 경우로 많이 부각되고 있다. 소프트웨어의 성공에 중요한 요소로 자리 잡히고 있으므로 소프트웨어의 성공을 위한 GUI 테스트가 요구되나 GUI 테스트는 많은 노력과 시간이 투자되는 테스트 분야이다. 이러한 자원의 소비의 줄이기 위해서 GUI 테스트의 자동화가 필요하리라 본다.

1. 서론

1.1 연구 배경 및 목적

모바일 소프트웨어의 생산성을 높이고 신뢰성 있는 소프트웨어를 개발하기 위해서 모바일 어플리케이션 소프트웨어의 테스트는 절실히 필요하리라 생각된다. 모바일 소프트웨어의 많은 부분을 테스트하기에는 모바일 어플리케이션의 짧은 개발주기와 생명주기로 인해 테스트 또한 명확하며 빨라야 한다. 모바일 테스트는 개발 중에도 테스트가 가능해야 하며 개발 후에도 테스트가 가능해야 한다고 본다. 본 논문에서는 모바일의 어플리케이션 소프트웨어의 여러 부분 중 가장 급부상하고 있는 GUI시장에 발맞추어 테스트 하고자 한다. 모바일 어플리케이션의 소프트웨어의 GUI는 사용자가 그래픽을 통해 모바일 콘텐츠를 효율적으로 사용할 수 있도록 하는 디자인 설계를 일컫는 것으로 모바일 시장의 급성장에 힘입어 올해 들어 약 300억 규모 시장으로 성장했다. GUI시장의 급부상은 단순한 통화나 메시지 전송 외에도 모바일 환경에서 활용되는 콘텐츠가 대폭 늘어나면서 관련 메뉴가 많아진 데 따른 것으로 풀이된다. 늘어난 콘텐츠를 효율적으로 사용하려면 사용성과 기자인 전달력이 보장된 사용자 인터페이스가 무엇보다 중요하기 때문이다.

본 논문에서 테스트 하고자 하는 어플리케이션 소프트웨어의 GUI(Graphical User Interface)는 모바일 CP(Contents Provider) 어플리케이션 소프트웨어의 GUI들로서 휴대폰 상에서 동작하는 프로그램으로 모바일게임과 같이 폰으로 다운로드하여 실행하는 프로그램들의 GUI들이다. 모바일 CP(Contents Provider) 어플리케이션 프로그

램의 특징으로 촉박한 개발일정과 어플리케이션의 소프트웨어의 단발성 그리고 안정성보다는 시장성으로 볼 수 있다.

모바일 시장은 누가 먼저 제품을 출시하느냐에 따른 싸움에 있기에 다른 업체보다 먼저 소비자에게 콘텐츠를 제공하기 위해서 촉박한 개발일정을 두고 개발에 들어가며, 개발 후 바로 콘텐츠가 제공되므로 콘텐츠에 대한 추가적인 업데이트가 없는 시장이다. 추가적인 업데이트를 하기에는 기술적인 어려움에 한계에 있으며, 모바일 어플리케이션 소프트웨어는 안정성보다 시장성을 두고 개발에 하기에 촉박한 개발일정으로 인해 안정성을 위한 충분한 설계와 테스트가 요구되지 않고 있다. 경우에 따라서는 일부 모바일 폰의 문제가 있을시 해당 모바일 폰에 콘텐츠 다운로드를 막기도 한다.

모바일 어플리케이션 소프트웨어에는 안정성을 위한 테스트가 필요하리라 생각되며, 모바일 어플리케이션 소프트웨어 촉박한 개발일정에 따라서 테스트 또한 자동화 도구에 의한 테스트가 필요하리라 본다.

2. 관련연구

2.1 모바일 어플리케이션 소프트웨어

모바일 어플리케이션 소프트웨어는 모바일 상에서 사용하는 어플리케이션 소프트웨어로서 그 특징(CP, SP, EM)에 따라서 다음과 같이 3가지로 나뉜다.

2.1.1 모바일 CP(Contents Provider) 어플리케이션

시장 점유율이 높은 어플리케이션 소프트웨어로서 SK

이동통신의 Nate, KTF의 MagicN과 같은 이동통신사의 WAP(Wireless Application Protocol)서비스에 입점하는 서비스로 이동통신사의 CP(Contents Provider)들이 주로 개발을 하고 서비스를 한다.

2.1.2 모바일 SP(Service Provider) 어플리케이션

VM(Virtual Machine)상의 동작하는 특정 서비스를 위한 어플리케이션이며, 단기간의 매출보다는 장기적인 서비스를 위해서 개발되어지고 있으며, 이동통신사와 CP(Contents Provider)계약을 통해서 서비스 되고, 폰으로 다운로드 하여 사용한다.

2.1.3 모바일 임베디드 어플리케이션

휴대폰 출시와 더불어서 포팅 되어서 나오는 어플리케이션으로 주로 휴대폰 제조사의 요구에 의해 개발되어지고 있다. VM(Virtual Machine)기반일수도 있고, 폰의 API를 이용해서 C-Based의 어플리케이션일수도 있다.

2.2 모바일 어플리케이션 소프트웨어의 GUI

GUI는 Graphical User Interface의 약자로서 사용자가 그래픽을 통해 모바일 폰과 정보를 교환하는 작업 환경을 말한다. 이제까지 사용자는 모바일 폰의 키패드를 통해 메뉴에 접근하여 정보를 전달 받았으나 모바일 시장의 기술 발달로 인해 GUI(Graphical User Interface)를 키패드를 사용하는 접근 방법과 사용자의 GUI 터치로 인해 모바일 폰과 정보를 교환 할 수 있게 됐다.

2.3 모바일 어플리케이션 소프트웨어의 GUI Test

모바일 어플리케이션의 GUI 테스트방법으로 모바일 어플리케이션 소프트웨어 업체에서 가장 많이 사용되는 기술로는 Record/playback, Capture/playback, 명세기반, 베타 테스트를 사용한다.

2.3.1 Record/Playback 기법

사용자의 마우스 및 키보드 입력을 통해 선택된 GUI에 발생하는 이벤트를 기록하여 기록된 이벤트를 재생하여 소프트웨어의 GUI 테스트를 수행하는 방법으로 현재 GUI 테스트 도구들은 Record/Playback 기술을 적용하여 많이 개발되어지고 있다. Record/Playback 기법은 단순한 패턴을 가지며, 이 기술을 이용한 테스트 도구를 구현하기가 용이하다. Record/Playback기법의 테스트 도구들은 가능한 짧은 시간 내에 기존 GUI에 대한 테스트 슈트를 생성하고자 할 때 유용하며, 어플리케이션에 대한 변경이 발생하면, 영향을 받는 모든 테스트케이스를 재작성해야 한다. 수작업이 많고 오류에 취약하다는 단점을 가지고 있다.

2.3.2 Capture/Playback 기법

시험 하고자하는 모바일 어플리케이션 소프트웨어를 사용자가 시범적으로 사용하고 이때 사용 시나리오를 기록함으로써 GUI의 이상 현상을 기록하는 수행 방법이다.

2.3.3 명세기반 테스트 기법

명세 기반 테스트는 시스템의 GUI 명세를 기반으로 시스템을 테스트 한다. 명세 기반 기술 테스트는 다른 테스트 기술에 비교하여 볼 때 여러 조건 요소들이 있다.

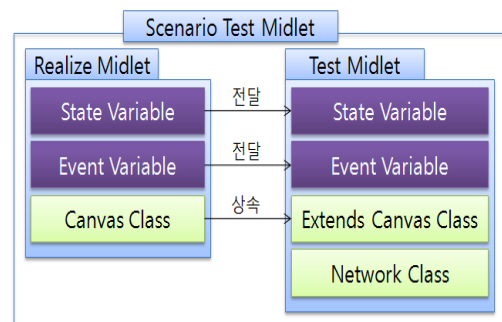
요구사항이나 디자인 명세가 일괄적이어야 하고 완벽해야 하며, 분명하게 기술되는 것을 요구한다. 명세기반 GUI 테스트는 높은 수준의 GUI 명세를 요구한다. 이런 명세를 얻는데 많은 노력이 필요하다.

2.3.4 베타 테스트 기법

베타 테스트는 많은 업체들이 사용하는 GUI 테스트 방법으로서 가장 인기 있는 테스트 방법이다. 베타 버전의 소프트웨어를 출시하여 일반 사용자에게 GUI 테스트를 수행하게 하는 방법으로 많이 사용되어지고 있으며, GUI 테스트 방법 중 가장 많은 오류를 찾을 수 있는 방법이나 비전문적인 테스터들로 구성되어 그 전문성이 떨어지고 많은 일반 사용자를 이해시키고 교육시키는데 어려움이 따른다.

3. 시나리오기반 GUI 테스트 구조 및 설계

3.1 Scenario Test Midlet 구조

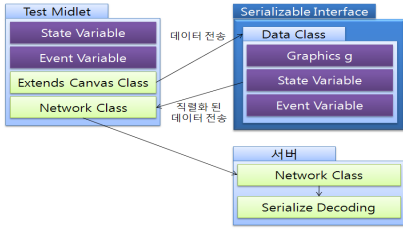


(그림 1) Scenario Test Midlet의 구조

Scenario Test Midlet의 구성은 Realize Midlet 과 Test Midlet 의 두 개의 Midlet의 구조로 되어있다. Realize Midlet은 중요 변수인 상태 변수와 이벤트 변수, 화면을 그리는 역할을 하는 Canvas Class 가 있고 네트워크 클래스, 로직 클래스 등 다양한 클래스들로 구성될 수 있다. Test Midlet의 구조는 (그림 1)과 같다. Test Midlet 에서 구현 코드에 테스트 코드를 삽입 하지 않고 Test Midlet 을 따로 두어 실제 구현 Midlet을 상속받아 제작을 하고 이때 paint() 메소드 부분에 (그림 2)와 같이 테스트 수행 코드를 작성한다. 이를 통하여 GUI의 상태, 이벤트 그리고 화면을 서버로 전송한다. 이때 발생할 수 있는 문제점으로는 상태 변수와 키 이벤트 변수를 담고 있는 변수를 지정해 주어야 한다는 점이다. 프로그래머가 사용하는 키 이벤트 저장 변수와 상태 변수의 네이밍 규칙 등이 다르기 때문에 테스트 미들렛 작성 시에 해당 변수들의 정보를 입력받아야 한다.

```
public void paint(Graphics g) {
    super.paint(g);
    testStart(g, keyPressed, appState);
}
```

(그림 2) Paint() 메소드

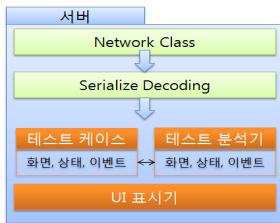


(그림 3) 직렬화 과정

모바일 프로그래밍에서는 객체의 직렬화가 존재하지 않기 때문에 객체 직렬화 클래스를 제작하여야 한다. 모바일 상에서는 Graphics 객체를 이용하는 데에 한계가 있으므로 모바일 상에서는 분석 및 가공할 수 있는 방법이 없기 때문에 서버로 Graphics 객체를 전송하여 서버 측에서 Graphics 객체를 이용하는 방법을 채택하였다.

객체를 직렬화 하기 위해서는 (그림 3)과 같이 직렬화를 하는 인터페이스를 만들고 직렬화 하고자 하는 클래스에 인터페이스를 구현하여 준다. Test Midlet에서 구현한 Test Canvas Class에서 직렬화를 하고자 하는 데이터를 전송하고 직렬화 클래스에서 직렬화를 한 후에 Network Class에 전송한다. 직렬화 된 객체를 서버에 네트워크를 통하여 전송을 하면 서버 측에서는 직렬화된 데이터를 풀어서 원상태의 Graphics g 객체와 상태 변수, 이벤트 변수를 얻는다.

3.2 Test Server 구조



(그림 4) 서버의 구성

Server의 구성은 Network Class와 Seialize Decoding 부분과 테스트 케이스, 테스트분석기, UI 표시기로 구성되어 있다. Server는 Network Class를 통하여 직렬화 되어 전송된 데이터를 디코딩하여 원상태의 데이터로 변환한 후 테스트 분석기에 전송하게 된다.

테스트 분석기는 테스트케이스에게 테스트 분석기에 전송되어진 데이터와 같은 데이터들은 요청하게 된다. 요청된 데이터들은 테스트 분석기에서 Bit-by-Bit방식으로 GUI 테스트를 수행하게 된다.

3.2 테스트 케이스

시나리오기반 테스트 케이스에서 입력된 화면, 상태, 이벤트에 대한 정보가 <표 1>과 같은 코드로 전송되어 서버에 테스트케이스에 저장된다.

3.3 테스트 분석기

테스트 분석기에는 시나리오 기반 GUI 테스트 수행도구에서 키 버튼 입력으로 작성된 테스트 수행 시나리오에 의해서 검출된 화면, 상태, 이벤트에 대한 정보를 <표 1>과 같이 XML데이터로 받게 된다. XML 데이터로 전송된 정보는 테스트 케이스에 테스트 항목들을 요청을 통해 GUI 테스트를 수행한다.

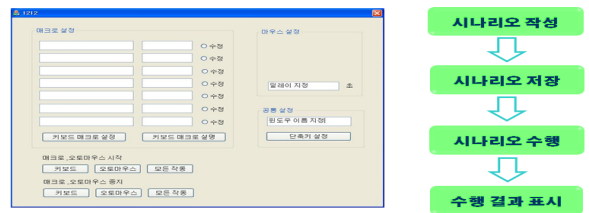
<표 1> XML 데이터

```
<data no='1'>
  <image>1.png</image>
  <state>start_menu</state>
  <event>none</event>
</data>
<data no='2'>
  <image>2.png</image>
  <state>main_menu</state>
  <event>Enter Key</event>
</data>
```

4. 시나리오기반 GUI 테스트방법

모바일 상에서 테스트는 화면단위로 설계되며 그에 따라서 화면이 바뀌었을 경우 해당 조건까지 키 입력으로 이동을 하여 테스트를 하는 방법이 있다. 그러나 본 시나리오 기반의 테스트 기법은 이미 테스트를 수행한 부분까지 테스트를 반복적으로 수행을 하여 그 이전에 발생될 수 있는 코드상의 GUI 변경 및 돌출 GUI 발생으로 인한 오류를 감지할 수 있고 또한 변경된 화면의 테스트를 새로 수행하여 GUI 테스트의 커버리지를 높이는 방법이다.

4.1 시나리오기반 GUI 테스트 절차



(그림 5) 시나리오 기반 GUI 테스트

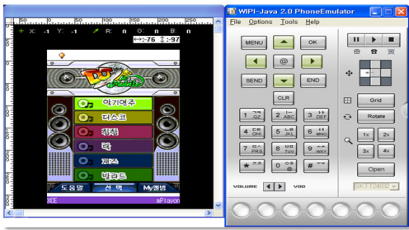
테스트 수행은 (그림 5)와 같은 도구를 이용하여 시나리오를 작성하게 된다. 시나리오 작성은 마우스와 키보드를 이용하여 키보드의 키 입력된 순서를 저장하여 시나리오를 작성하고 마우스 이용 시 마우스의 버튼입력에 대한 좌표값에 대한 정보를 순차적으로 저장하여 다시 재생시켜 보는 방법이다. 이는 관련연구에 표기된 Record/Playback

방식과 흡사하다고 보면 된다. 작성된 시나리오는 (그림 6)의 예물레이터 구동화면의 키 입력으로 발생하는 이벤트 처리 후 변경 되는 GUI 및 GUI의 상태, GUI의 이벤트에 대한 정보가 서버측의 테스트 분석기에 제공된다.

사용자는 시나리오를 작성 전에 비교대상 이미지와 상태 정보 그리고 다음 이벤트 정보를 입력 받으며, 작성된 시나리오는 XML 형태로 저장하여 서버에 테스트 케이스에 제공하게 되며, 서버에 테스트케이스에 입력될 비교 대상 XML 데이터다 <표 2> 비교대상의 이미지의 상태와 이벤트에 대한 정보로 구성되어졌다.

<표 2> XML 데이터

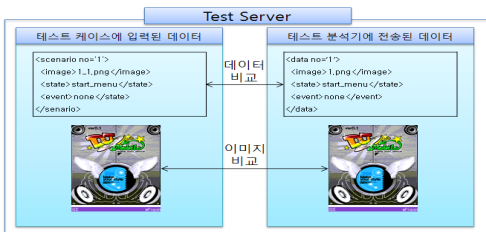
```
<scenario no='1'>
  <image>1_1.png</image>
  <state>start_menu</state>
  <event>none</state>
</scenario>
```



(그림 6) 예물레이터 구동 화면

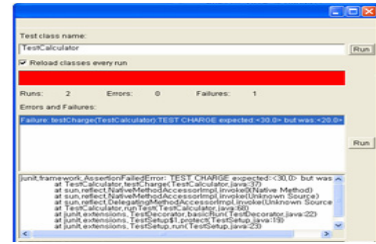
4.2. 시나리오기반 GUI 테스트 결과

테스트 수행 결과는 시나리오에서 작성된 데이터와 모바일상에서 전송되어진 데이터를 비교하여 동일 조건하에 같은 이미지가 전송되어졌는지에 대한 비교를 (그림 7)과 같이 하게 된다. 이를 통해 해당 시나리오의 수행 결과를 확인할 수 있다. 만약 테스트가 실패한다면 해당 부분에서 테스트는 중지되고 테스트에게 실패 시점을 출력하게 된다.



(그림 7) 시나리오기반 테스트 기법

4.3. 시나리오기반 GUI 테스트 결과 화면



(그림 8) 시나리오 수행 도구

테스트 수행 결과는 (그림 9)처럼 표기되며 GUI 테스트 결과를 테스트 케이스에 입력되어 있는 정보와 테스트 분석기에 전송된 정보와 일치 할 경우에는 Green Bar를 보여주며 불일치 할 경우에는 Red Bar를 제공한다.

5. 결론

본 논문에서 제시하는 시나리오 기반 모바일 어플리케이션 소프트웨어 GUI 테스트 자동화 도구는 모바일 시장의 짧은 개발주기와 생명주기에 따른 모바일 어플리케이션 소프트웨어의 GUI 테스트 자동화 도구가 필요하리라 본다. 모바일 업계에서 가장 많은 방법으로 사용되고 있는 베타 테스트 방법은 많은 예산과 시간을 투자하여 전문성이 인증되지 않은 비인증 테스트들로서 테스트들을 대상으로 시행하는 베타 테스트보다 짧은 개발주기와 생명주기에 맞는 자동화 도구가 더 좋은 방법이라 생각된다.

가장 많은 사용자를 확보하고 있는 이동통신사가 제공하는 컨텐츠형 어플리케이션 소프트웨어는 컨텐츠들이 더욱더 GUI화 되리라 보며 앞으로 더욱더 발전하는 GUI시장에 대비하여 시장성을 확보하기위해서 안정성이 확보된 GUI 테스트가 필요하리라 생각된다. 본 논문에서 제시하는 방법은 정적방식의 GUI테스트를 수행하는데 효과를 발휘하며 동적방식의 테스트는 아직 불가피하다. 향후 연구로 동적방식의 GUI테스트도 필요하리라 본다.

참고문헌

- [1] Pressman, R., Software Engineering: A Practitioner's Approach, McGraw-Hill, 2003.
- [2] Patton, R., Software Testing, Sams, 2000.
- [3] NIST, "The Economic Impacts of Inadequate Infra-structure for Software Testing", 2002.5
- [4] 한국정보통신기술협회, 소프트웨어테스트 전문기술 응용분야, 한국정보통신기술협회, 2005
- [5] SQA G. S/W Center, "Manual vs. Automated Test 에 대한 사례 연구 소개", 2003
- [6] 권원일. "모바일 소프트웨어 테스트링 현황과 표준적인 테스트 케이스"
- [7] 이정규. "Record-Playback 기술 기반의 GUI 테스트 케이스 자동생성, 한국컴퓨터종합학술대회 논문집, 2007