

# 모바일 응용 S/W GUI 자동화 테스트를 위한 방법 및 도구 구현

채현철\*, 황선명\*\*

\*대전대학교 컴퓨터공학과

\*\*대전대학교 컴퓨터공학과

e-mail:chaehc@gmail.com, sunhwang@dju.ac.kr

## Implementation of Method and Tool for Mobile S/W GUI Automated Test

<sup>1)</sup>Hyeonl-Cheol Chae\*, Sun-Myung Hwang\*\*

\*Dept of Computer Engineering, Dae-jeon University

\*\*Dept of Computer Engineering, Dae-jeon University

### 요 약

모바일의 소프트웨어 GUI 테스트는 매우 중요하다. 그러나 모바일 특성상 일반 어플리케이션과 달리 많은 제약사항이 따르며 빠른 출시 등의 이유로 테스트를 소홀히 하고 있다. 현재 대부분의 기업에서 GUI 테스트는 프로그래머가 매뉴얼대로 검사하는 수동적 방식을 사용하고 있는 실정이다. 본 논문에서는 GUI 테스트를 자동화 하는 방법 제시 및 도구를 구현 함으로써 좀더 편리하게 GUI 테스트를 할 수 있도록 하고자 한다.

### 1. 서론

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 개발을 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.

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 프로그램의 특징으로 촉박한 개발일정과 어플리케이션의 소프트웨어의 단발성 그리고 안정성보다는 시장성으로 볼 수 있다. 따라서 모바일 어플리케이션 소프트웨어에는 안정성을 위한 테스트가 필요하리다 생각되며, 모바일 어플리케이션 소프트웨어 촉박한 개발일정에 따라서 테스트 또한 자동화 도구에 의한 테스트가 필요하리다 본다.

1) 본 연구는 정보통신부 및 정보통신연구진흥원의 IT신성장동력 핵심기술개발사업의 일환으로 수행하였음. [2007-S032-01, 다중플랫폼지원 모바일 응용 S/W 개발환경 기술 개발]

### 2. 관련연구

#### 2.1 J2ME 플랫폼

J2ME는 Java2 Micro Edition의 약자로서 휴대폰이나 스마트폰과 같은 소형 기기에서 사용가능한 자바 애플리케이션을 목표로 제안되었다.

J2ME 비록 휴대폰이나 임베디드 기기를 위한 플랫폼이지만, 기본적으로 자바가 가지고 있는 특징인, 객체지향방식의 프로그래밍, 코드의 높은 이식성, 안전한 네트워크 보안 지원 및 J2SE와 J2EE와의 상위 호환성은 그대로 유지하고 있다. 여기에 더하여 J2ME는 다음과 같은 특징으로는 다중플랫폼호환성과 보완성, 동적 애플리케이션 다운로드 등이 있다. 다중플랫폼호환성은 J2ME API를 사용하여 개발하였다면 단말기 모델에 상관없이 시간과 장소에 관계없이 강력한 호환성을 가지고 있으며, 보완성은 J2ME로 작성된 애플리케이션은 기본적으로 디바이스의 하드웨어나 다른 리소스에 직접 접근할 수 없기 때문에 바이러스나 다른 악성 프로그램을 만들어낼 수 없으리라 본다. 마지막으로 동적 애플리케이션 다운로드드는 무선 서비스를 통한 자바 애플리케이션들은 실시간으로 동적으로 다운로드 되므로 사용자들은 A/S 센터를 방문하고나, 업그레이드를 쉽게 받을 수 있다.

#### 2.2 모바일 어플리케이션 소프트웨어

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

2.2.1 모바일 CP(Content Provider) 어플리케이션

시장 점유율이 높은 어플리케이션 소프트웨어로서 SK 이동통신의 Nate, KTF의 MagicN과 같은 이동통신사의 WAP(Wireless Application Protocol)서비스에 입점하는 서비스로 이동통신사의 CP(Content Provider)들이 주로 개발을 하고 서비스를 한다.

2.2.2 모바일 SP(Service Provider) 어플리케이션

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

2.2.3 모바일 임베디드 어플리케이션

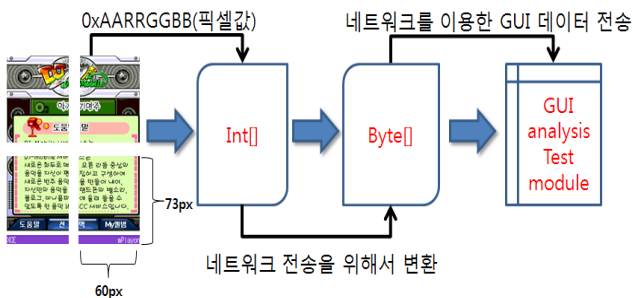
휴대폰 출시와 더불어서 포팅 되어서 나오는 어플리케이션으로 주로 휴대폰 제조사의 요구에 의해 개발되어지고 있다. VM(Virtual Machine)기반일수도 있고, 폰의 API를 이용해서 C-Based의 어플리케이션일수도 있다.

3. GUI 테스트 자동화 도구 구현 기술

3.1 데이터 직렬화

모바일 플랫폼인 SKVM에서는 일반적으로 PC 보다 성능이 낮은 CPU나 메모리가 사용된다. 그렇기에 JAVA의 스펙 또한 많은 부분이 제외 되어 사용되어 진다.

이러한 스펙 때문에 자바에서는 특히 객체 직렬화를 지원하는 리플렉션 기능이 빠져 있다. 본 논문에서 제시하는 프로그램에서는 객체 직렬화가 필수적이거나 지원되지 않는 기능 때문에 객체 직렬화를 지원하는 기능을 개발하였다. GUI 의 데이터를 네트워크로 전송하고 처리를 하려면 이미지 객체를 직렬화하여 처리를 해야 하기 때문에 MoGuT Framework에 Encoder 역할을 하는 기능을 제작하고 GUI 분석 테스트에 Decoder 역할을 하는 기능을 제작하여 이를 해결하였다. 화면 이미지 객체는 이미지 객체에 있는 모든 픽셀 값들을 0xAARRGGBB 형태의 값으로 변환하여 int[]형 배열에다가 담았다가 네트워크로 전송할 수 있는 데이터의 형태인 byte[]형 배열로 재 변환을 시켜 다시 담는 과정을 함으로써 Encoder 역할을 하게 되며 방식은 [그림 3-1]과 같다.

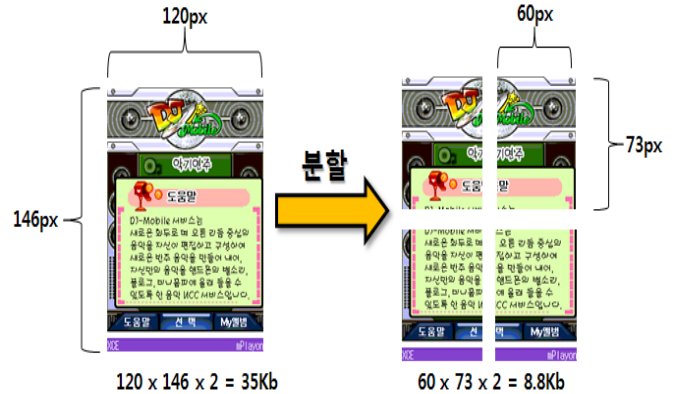


[그림 3-1] 객체 직렬화를 통한 전송 과정

3.2 이미지 분할

모바일에서는 메모리의 크기가 작기 때문에 화면 하나를 그리는데 드는 메모리 비용이 크다. 그렇기 때문에 최소한의 메모리를 사용하는 방식으로 이미지를 분할하여 전송하는 방법을 채택하였다.

이미지를 분할하는 기준은 이미지를 그리는데 필요한 메모리 크기를 10KB로 제한하여 자르게 되는데 width/2 와 height/2 의 기준으로 이미지를 나누게 된다. 이에 대한 설명은 아래 [그림 3-2]에 표현되어 있다.



[그림 3-2] 이미지 4분할하여 자르기

이미지를 분할할 때 width/2 와 height/2의 계산 공식을 사용한 이유는 핸드폰의 경우 LCD의 크기가 폰마다 다르고, 또한 폰에 따른 이미지의 크기들도 다르기 때문이다.

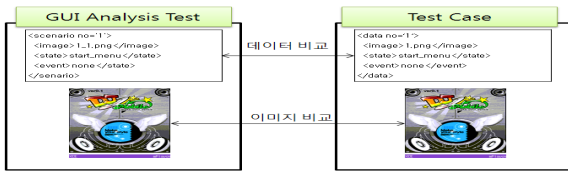
이렇게 분할된 이미지를 byte[] 배열에 한 번에 한 장의 분할 이미지를 넣게 되고 그와 함께 이미지의 좌표인 x축 시작좌표, y축 시작좌표를 함께 전송하고 이미지를 다시 int[]형 배열로 변환 할 수 있도록 이미지의 픽셀 사이즈 크기를 알려주어야 하기 때문에 분할 이미지의 넓이와 높이도 함께 전송이 된다. 또한, 이미지와 함께 현재 화면의 화면 상태 변수의 값과 어떠한 이벤트를 통하여 화면이 변하였는지에 대한 데이터를 얻기 위하여 키 이벤트를 발생시키게 되는 keyPressed() 메소드의 값을 함께 전송하게 된다. MoGuT GUI 분석 테스트에서는 이들 데이터를 순서와 형식에 맞게 Decoding 하여 사용하게 된다.

3.3 GUI 분석 테스트의 객체 데이터 복원 기술

MoGuT Framework에 의해서 객체 직렬화 과정을 거쳐 프로세싱 된 데이터를 이미지와 각 데이터로 복원하는 기술이다. 이미지 객체를 int[]형 배열로 변환하기 전에 먼저 이미지의 사이즈에 대한 정보를 먼저 불러오으로써 int[]형의 배열의 크기를 설정하고 그 배열에 픽셀 데이터들을 담는 역할을 하게 된다. 그 이후에 나머지 값들인 키 이벤트 값과 화면 상태 값, 이미지의 x 시작 좌표, 이미지의 y 시작 좌표들을 변수에 담게 된다. 이러한 모든 작업들은 스레드를 통하여 이루어진다.

### 3.4 GUI 데이터 비교방법

테스트 시나리오 도구에서 테스트 수행 시나리오에 의해서 시나리오를 이용하여 MoGuT Framework에 테스트를 수행 시 변화되는 GUI 정보를 객체 직렬화 과정을 통해 GUI 분석 테스트에 데이터를 전송된다. 전송된 데이터는 GUI 분석 테스트에서 XML 문서로 만들어지며, Scenario Test Tool에서 시나리오 등록을 통해 만들어진 XML 문서는 Test Case에 전송되어 저장되므로 GUI 분석 테스트는 Test Case에 테스트 대상 XML 문서를 요청하여 XML 문서를 받게 된다. 전송된 XML 문서는 아래와 같은 방법으로 분석 비교하여 결과를 제공한다.



[그림 3-3] GUI 데이터 비교 방법

각 XML 문서에 데이터(Image name, state, event)를 추출하여 데이터 비교를 거친 후 이미지에서 픽셀정보를 추출하여 픽셀 비교를 통해 테스트를 수행한다.

## 4. 시나리오기반 GUI 테스트 자동화 도구 결과

### 4.1 GUI 테스트 자동화 도구 테스트 시나리오

[표 4-1] GUI 테스트 자동화 도구 테스트 수행 시나리오

항목 명	시험 내용	이벤트	상태	파일명	확인
1. 초기화면 확인	초기화면을 확인	확인	start_menu	1.png	
2-1 DJ Mobile 메뉴 확인	메뉴 이동 색변환 확인	down	main_menu1	2.png	
2-2 DJ Mobile 메뉴 확인	메뉴 이동 색변환 확인	down	main_menu2	3.png	
2-3 DJ Mobile 메뉴 확인	메뉴 이동 색변환 확인	down	main_menu3	4.png	
2-4 DJ Mobile 메뉴 확인	메뉴 이동 색변환 확인	down	main_menu4	5.png	

[표 4-1]은 GUI 자동화 테스트 도구 사용 시 개발자나 테스터에게 제공되는 테스트 수행 시나리오이다. 테스트 항목과 시험 내용, 이벤트, 상태, 이미지 파일명에 대한 정보와 GUI Image를 제공한다. 개발자 및 테스터는 수행 시나리오에 의거하여 GUI 테스트 자동화 도구에 이벤트, 상태, 이미지에 대한 정보를 입력한다.

### 4.2 GUI 테스트 자동화 도구 수행



[그림 4-2] XCE2.0.2 와 Scenario Test Tool

테스트 시나리오 도구를 이용하여 GUI 테스트 자동화를 수행하기 위한 실행 화면이다. 앞에서 서술한대로 설계되어 구현한 테스트 시나리오 도구를 수행 시나리오에 입력된 시나리오 절차에 의해서 자동화 테스트를 수행하도록 구현되었다. 시나리오 등록을 위해 등록 버튼 입력 시 이벤트 정보와 상태 정보, 이미지 파일을 등록하기 위한 창이 하나 생성된다. 생성된 창에 이벤트 정보 등록 상태 정보 등록, 이미지 파일을 등록하면, 시나리오 등록상태에 순서대로 등록되게 된다. 등록된 시나리오에는 수정 가능하며, 다시 등록 할 수 있다. 테스트 수행을 반복적으로 수행하기 위해서는 반복 수행횟수에 총 수행할 테스트 횟수를 입력하면 입력된 수만큼 테스트를 수행한다.

[표 4-2] 시나리오 입력에 따른 이벤트

시나리오 입력	이벤트	시나리오 입력	이벤트
up	↑	1	1
down	↓	2	2
left	←	3	3
right	→	4	4
enter	확인	5	5
clear	취소	6	6
*	*	7	7
#	#	8	8
0	0	9	9

[표 4-2]에 나열된 문자는 시나리오 입력 시 up, down, left, right 등 이와 같은 문자열을 입력 시 옆과 같은 이벤트를 발생하여 자동으로 Emulator에서 키 이벤트를 발생시켜 테스트를 수행하게 된다. GUI 자동화 테스트 도구에 의해 수행된 Test 결과 화면은 아래와 같다.

No	Image name	State	Event
1	1.png	start_menu	enter
2	2.png	no_in_menu1	down
3	3.png	no_in_menu2	down
4	4.png	no_in_menu3	down
5	5.png	no_in_menu4	down
6	6.png	no_in_menu5	down
7	7.png	no_in_menu5	down

[그림 4-3] GUI 자동화 테스트 수행 결과

4.3 GUI 테스트 자동화 도구 장단점

GUI 테스트 자동화 도구는 기존 모바일 개발 환경에 본 논문에서 제안하는 MoGuT Framework를 적용하여 GUI 테스트 자동화 도구를 사용할 수 있도록 구현되었다. 기존 GUI 테스트는 블랙박스 테스트 방식으로 소스 코드 없이 결과물을 통하여 GUI 테스트를 수행한다. GUI 테스트 자동화 도구를 적용한 장단점을 요약하면 [표 4-3]와 같다.

[표 4-3] 장점 및 단점 과 개선 사항

구분	내용
장점	<ul style="list-style-type: none"> <li>- 간단한 시나리오부터 복잡한 시나리오까지 테스트 가능</li> <li>- 수행 시나리오 데이터 저장으로 시나리오 재사용 가능</li> <li>- XML 데이터 사용으로 XML 데이터 보존 가능</li> </ul>
단점	<ul style="list-style-type: none"> <li>- GUI 테스트 자동화 도구는 특정 개발환경(SKVM)에서 적용 가능하며, 타 개발 환경인(GVM, WPI)등에서는 불가능 함</li> <li>- 테스트 대상의 정보를 알고 있어야 테스트 수행이 가능함</li> </ul>
개선 사항	<ul style="list-style-type: none"> <li>- 텍스트 형식의 결과 값에서 시나리오 진행 순서 GUI 표현</li> <li>- GUI 테스트에서 벗어나 이벤트 테스트 가능토록 구현 예정</li> <li>- 테스트케이스 10개 제한에서 10개 이상으로 변경</li> <li>- 보존 XML 데이터 사용으로 웹 환경 구축</li> </ul>

5. 결론

본 논문에서 제안하는 목적은 모바일 소프트웨어의 생산성을 높이고 신뢰성 있는 소프트웨어를 개발하기 위한 것이다. 또한 모바일 소프트웨어 테스트의 중요성이 부각되고 있다. 모바일 어플리케이션 소프트웨어에 자동화 테스트를 적용하므로 모바일 어플리케이션의 생산성관 신뢰성을 향상시킬 수 있을 뿐만 아니라 테스트 기간 및 비용을 줄일 수 있다.

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 필요할 것이다. 본 논문에서 제시하는 방법은 정적방식의 GUI테스트를 수행하는데 효과성이 높으나 동적방식의 테스트는 아직 불가하다.

GUI 테스트 자동화 도구는 모바일 소프트웨어 개발환경에 프레임워크를 이식하여 GUI 자동화 테스트를 수행할 수 있도록 구현 되었다. 모바일 소프트웨어 개발환경(SKVM)에 한정되어 개발되었지만 앞으로 다른 모바일

개발환경에도 호환 될 수 있도록 연구가 필요하다. GUI 자동화 테스트에 한정을 두지 않고 이벤트 호출 및 존속 여부까지 테스트 할 수 있도록 향후 연구가 필요하리라 생각된다.

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  
 [8] Binder, R.V., "Testing Object-Oriented Systems: A Status Report," American Programmer, vol.7, no.4, April 1994  
 [9] NIST, "The Economic Impacts of Inadequate Infra-structure for Software Testing", 2002.5  
 [10] Sykes, D., and McGregor, J., Practical Guide to Testing Object-Oriented Software, Addison-Wesley, Addison-Wesley Pub Co, 2001.