

시나리오기반의 모바일 어플리케이션 소프트웨어 GUI 테스트 방법

황선명^{1*}, 김정종²

A GUI Testing Method base on Scenario for Mobile Application Software

Sun-Myung Hwang^{1*} and Jung-Jong Kim²

요약 모바일 응용소프트웨어의 그래픽 유저 인터페이스는 모바일 상에서 모든 기능들을 작동시키며 정보를 획득하고 변경하는 작업 환경이며 도구이다. 모바일 소프트웨어 생산성을 높이고 신뢰성을 향상시키기 위하여 그래픽 유저 인터페이스의 테스트는 반드시 필요한 과정이다. 본 연구에서는 테스트 시나리오를 이용하여 그래픽 유저 인터페이스를 테스트 하는 방법을 제안하고 이를 구현한 자동화 도구를 제안한다. 자동화 도구(MoGuT)는 테스트 시나리오에 의해 기능적인 결함을 빠른 시간에 찾을 수 있으며 SKVM 환경에서 운영된다.

Abstract Graphic User Interface(GUI) of mobile application is the working environment exchanging informations and almost functions on mobile are operated by it. In order to increase the productivity and to develop reliable mobile software, GUI test is absolutely needed. In this paper, we introduce the scenario based GUI test method and automatic tool(MoGuT). The MoGuT can detect the functional faults using scenarios and operate in SKVM environment.

Key Words : Mobile GUI Test, Scenario, Mobile Application,

1. 서론

1.1 연구 배경 및 목적

모바일 소프트웨어의 생산성을 높이고 신뢰성 있는 소프트웨어를 개발하기 위해서 모바일 어플리케이션 소프트웨어의 테스트는 절실히 필요하리다 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 디자인 전달력이 보장된 사용자 인터페이스가 무엇보다 중요하기 때문이다.[1][2][11][12]

본 논문에서 테스트 하고자 하는 어플리케이션 소프트웨어의 GUI(Graphical User Interface)는 모바일 CP(Contents Provider) 어플리케이션 소프트웨어의 GUI들로서 휴대폰 상에서 동작하는 프로그램으로 모바일 게임과 같이 폰으로 다운로드하여 실행하는 프로그램들의 GUI들이다. 모바일 CP(Contents Provider) 어플리케이션 프로그램의 특징으로 촉박한 개발일정과 어플리케이션의 소프트웨어의 단발성 그리고 안정성보다는 시장성으로 볼 수 있다.[3][4][14][15]

모바일 어플리케이션 소프트웨어에는 안정성을 위한 테스트가 필요하리다 생각되며, 모바일 어플리케이션 소

본 연구는 정보통신부 및 정보통신연구진흥원의 IT신성장동력 핵심기술개발사업의 일환으로 수행하였음.

¹대전대학교 컴퓨터공학과

²경남대학교 컴퓨터공학과

*교신저자: 황선명(sunhwang@dju.ac.kr)

소프트웨어 촉박한 개발일정에 따라서 테스트 또한 자동화 도구에 의한 테스트가 필요하리라 본다.

본 연구는 서론에 이어 제2장에서는 모바일 플랫폼 및 개발환경에 대해서 설명하고, 제3장에서는 GUI 자동화 테스트 도구의 구성 및 시스템 설계에 대하여 설명한다. 제4장에서는 자동화도구 구현방법에 대하여 설명하며, 제5장에서는 기존 도구와의 성능평가에 대하여 서술하였으며 끝으로 결론은 보완점 및 향후 연구계획에 대하여 기술한다.

2. 관련연구

2.1 J2ME 플랫폼

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

휴대폰을 비롯한 각종 무선 정보기기들이 각자 서로 다른 사양을 가지고 쏟아져 나오기 시작하면서, 자바 프로그램이 모든 플랫폼에서 동일하게 동작한다는 원칙을 지키는 것이 사실상 불가능해지므로 결국, 각 기기별 특성과 한계 때문에 하나의 크기로는 모든 것을 만족시킬 수 없다. 라는 것을 깨닫고 J2ME가 새로 등장한다.[5][13]

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 다운로드는 무선 서비스를 통한 자바 애플리케이션들은 실시간으로 동적으로 다운로드 되므로 사용자들은 A/S 센터를 방문하고나, 업그레이드를 쉽게 받을 수 있다.[6][7]

2.2 CLDC

CLDC(Connected Limited Device Configuration)는 성

능이 제한된 CPU나 메모리가 한정적인 시스템을 대상으로 하는 spec이다. CLDC 관련 라이브러리는 시스템에 독립적인 고수준의 API, 그리고 네트워크관련 API로 되어 있다.[8][9]

다음은 CLDC에서 주목해야할 몇가지 특징들이다.

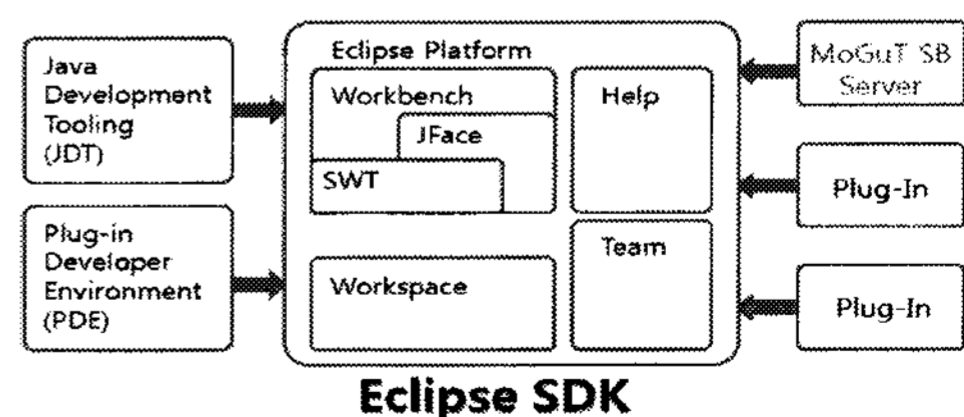
- 부동 소수점을 지원하지 않는다.
- 에러 처리가 제한적이다.
- 리플렉션을 지원하지 않는다.
- 사용자 정의 클래스 로더를 생성할 수 없다.

2.3 이미지 플로우 GUI 테스트 도구 분석

이미지 플로우 테스트 GUI 도구는 기존의 테스트터가 단계적으로 모든 화면에 대한 테스트를 하던 문제점을 극복하고자 구현한 도구이다. 이미지 플로우를 제공하여 개발자 또는 테스터가 오류부분을 명확하게 판별할 수 있는 기준을 제안하고 있다. 하지만 사용자가 원하는 화면 또는 화면의 흐름을 테스트를 할 수 없는 단점이 있다. [10]

3. GUI 테스트 자동화 도구

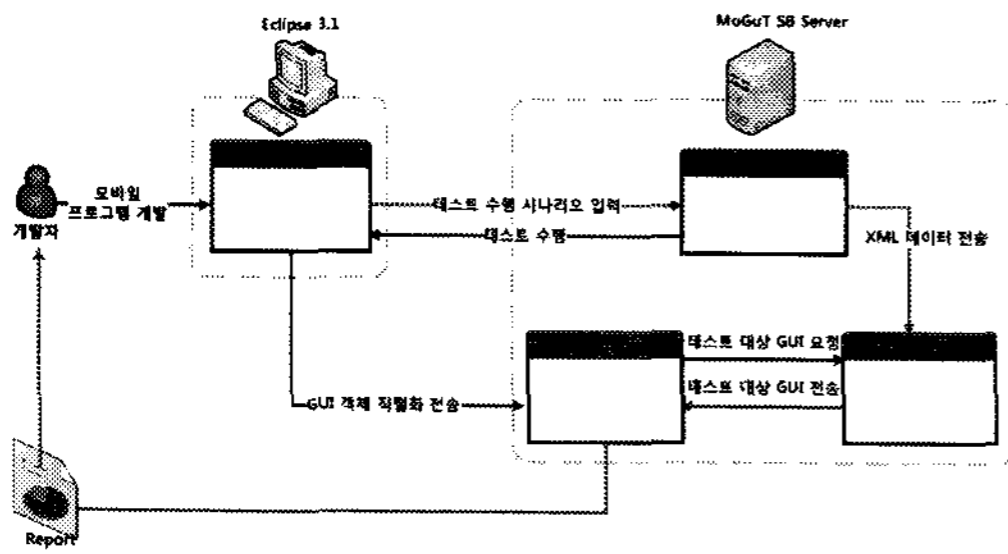
GUI 테스트 자동화 도구는 J2ME개발도구인 이클립스의 뛰어난 기능인 확장성을 이용하여 Plug-In기반으로 구성되었다. 이는 개발도구인 이클립스에 내장하여 개발 시초부터 테스트를 염두에 두고 개발하기 위한 방법으로 모바일 어플리케이션 소프트웨어 개발 당시에 테스트 수행에 필요한 파일들을 생성하여 테스트 수행 후 테스트에 성공하였다면 개발코드만을 제공하기 위해 J2ME 개발도구인 이클립스 Plug-In기반으로 구성하여 설계하였다. [그림 1]은 Eclipse SDK의 구성이다.



[그림 1] Eclipse SDK 확장성 Plug-In

3.1 GUI 테스트 자동화 도구 구성 및 설계

본 논문에서 제안한 GUI 테스트 자동화 도구의 전체 구성은 [그림 2]와 같은 환경으로 이루어져 있다.



[그림 2] GUI 테스트 자동화 도구 시스템

GUI 테스트 자동화 도구 시스템은 모바일 어플리케이션 소프트웨어 개발 후 GUI 테스트를 수행하기 위한 MoGuT(Mobile Application Software GUI Test) Framework와 데이터를 전송받아 테스트를 수행하는 MoGuT SB Server로 구성 된다.

3.1.1 MoGuT Framework 구성 및 설계

기존 모바일 개발환경 시 제공되는 MIDlet은 자동화 테스트 도구를 구현하는데 문제점이 있기에 MoGuT Framework Module을 구현하여 자동화 테스트를 수행하는데 필요한 Class와 Variable을 기존환경에 이식하여 개발 환경을 만들도록 설계하였으며, MoGuT Framework 적용하여 MoGuTMIDlet을 적용함으로써 차후 테스트가 끝난 후 구현 프로그램을 인수할 때 MoGuTMIDlet을 삭제하는 것으로 본래의 구현코드가 구현하고자 하는 기능들만 남게 된다.

MoGuT Framework는 MoGuTMIDlet을 바탕으로 MoGuT SB Server와 데이터를 전송하는 역할을 하는 Network 컴포넌트와 GUI 전체화면을 Capture하여 가공하는 Image Process 컴포넌트와 이 두 컴포넌트의 작업을 수행 할 수 있도록 실질적인 기능을 담당하는 Canvas 컴포넌트로 구성된다. MoGuTMIDlet을 사용하는 이유는 구현 MIDlet을 대신하여 개발자가 구현한 MIDlet을 수정하지 않고 본 논문에서 제시하는 MoGuTMIDlet을 적용함으로써 차후 테스트가 끝나고 구현프로그램을 인수할 때 MoGuT MIDlet을 삭제하는 것으로 앞에서 서술했던 것처럼 본래의 구현코드가 구현하고자 하는 기능들만 남게 하여 완성시키기 위해서이다.

MoGuT Framework는 GUI Image와 필요한 데이터들을 MoGuT SB Server로 전송하는 역할을 하는 부분을 모바일 상에 탑재하여 구동하기 때문에 최소한의 기능과 코드로 구성하였다. 또한 모바일 개발 시 사용되는 API인 J2ME의 경우 모바일 개발의 특성상 지원하는 기능이 극히 제한되어 있다. 데이터를 서버에 전송하기위

에서는 객체 직렬화 과정을 거쳐 보내야하는데 모바일 개발환경에서는 객체를 직렬화 하는 기능이 존재하지 않기에 해당 객체(GUI Image)를 int[]배열의 형태로 바꾼 후 byte[] 배열의 형태로 변환하여 서버로 전송해야한다.

3.1.2 MoGuT SB Server의 구성 및 설계

MoGuT SB(Scenario Based) Server는 Plug-In으로 3개의 모듈로 구성되며, 각 모듈은 테스트 시나리오 도구와 GUI 분석 테스트, Test Case로 구성된다.

가. 테스트 시나리오 도구 구성 및 설계

테스트 시나리오 도구는 개발자가 테스트 수행 시나리오에 의거하여 테스트 수행 시나리오대로 키 버튼 입력을 통해 수행하는 테스트 방법을 여러 번, 크게는 100번 이상 수행 시 소요되는 시간과 예산의 감소를 위해 자동화 도구로 개발되어졌으며, 입력된 키 버튼의 순서대로 Mobile Application Software의 실행화면인 Emulator를 통해 자동으로 수행되며, 입력된 키 버튼의 순서는 테스트 수행 시나리오로 저장된다. 입력된 시나리오의 정보를 분석하여 GUI Image와 State, event들의 정보는 XML 문서로 만들어져 MoGuT SB Server의 Test case로 전송되게 구성된다.

나. GUI 분석 테스트 구성 및 설계

GUI 분석 테스트는 MoGuT Framework에 의해 삽입된 Class 및 Variable에 의해서 개발 결과화면이 보여주는 Emulator에 변화되는 GUI들을 Capture하여 Image를 객체 직렬화를 통해 전송되는 GUI를 분석하여 전송되어진 GUI와 같은 GUI 정보를 Test Case에 요청하여 분석하고 테스트를 수행하여 결과를 보여주는 방법으로 구성된다.

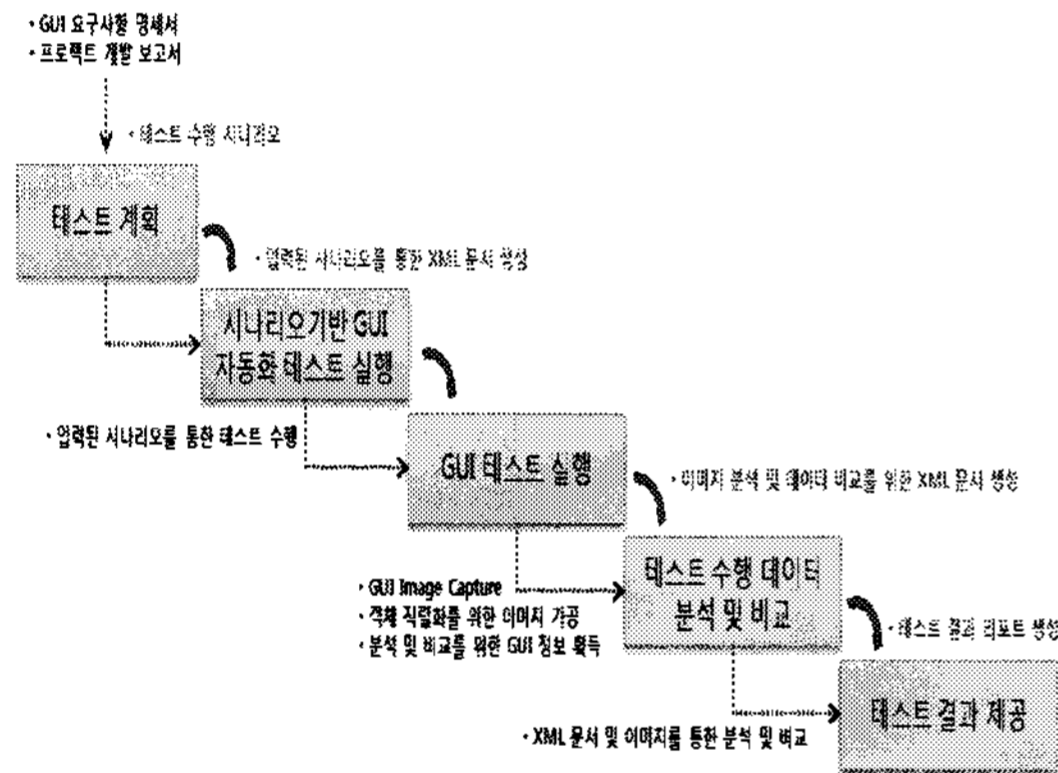
다. Test Case 구성 및 설계

Test Case는 테스트 시나리오 도구에서 입력되어 전송된 GUI Image, State, event들의 정보가 저장된 XML 데이터의 집합으로 구성되며, GUI 분석 테스트에서 GUI 분석 테스트를 수행하기 위해 대상 GUI와 같은 정보로 만들어진 XML 데이터 문서를 요청하게 되면 GUI 분석 테스트에 전송하는 방법으로 구성된다.

3.2 GUI 자동화 테스트 프로세스 설계

GUI 자동화 테스트 프로세스는 다음과 같이 설계 하였다. GUI 요구사항 명세서나 프로젝트 개발 보고서를

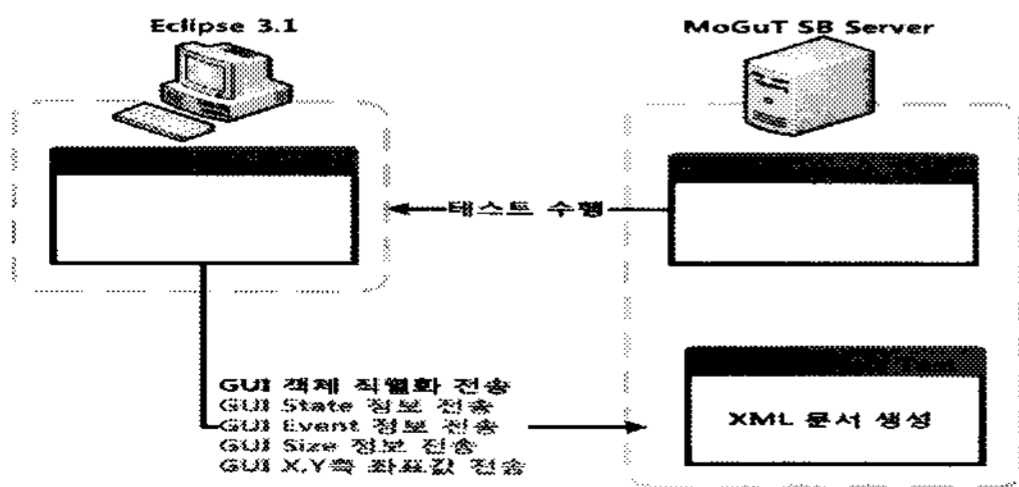
통해 테스트 수행 시나리오를 작성하게 되며, 작성된 수행 시나리오를 통해 시나리오 기반 GUI 자동화 테스트를 실행하게 된다. 이와 같은 과정에서 입력된 시나리오를 통한 XML 문서가 생성되며, 실질적인 테스트 수행에 들어가게 된다. 테스트 수행은 입력된 시나리오대로 Emulator에 자동으로 키 버튼을 입력하여 수행하며 변화되는 GUI Image를 Capture하여 객체 직렬화 과정을 거쳐 데이터를 보내게 된다. 이때 GUI Image의 State, Event정보와 Image Size, Image의 위치정보(X,Y축)들을 전송하여 테스트 수행 데이터 분석 및 비교에서 XML 문서로 파싱되어 입력된 시나리오를 통해 생성된 XML 문서를 호출하여 XML 데이터 비교 후 Image비교를 하여 결과를 제공하는 방법으로 구성된다.



[그림 3] GUI 자동화 테스트 프로세스의 구성

3.2.1 XML 문서

XML문서는 두 가지 방법으로 만들어지게 된다. 첫 번째 방법은 Scenario Test Tool에서 입력된 시나리오에 의해서 테스트 수행 시 객체 직렬화를 통해 GUI Analysis Test에 전송되는 GUI Image와 GUI의 State 정보, GUI Event 정보 GUI Size정보, GUI X,Y축 좌표값 등을 담아 XML 문서에 파싱되어 만들어 지게 된다.



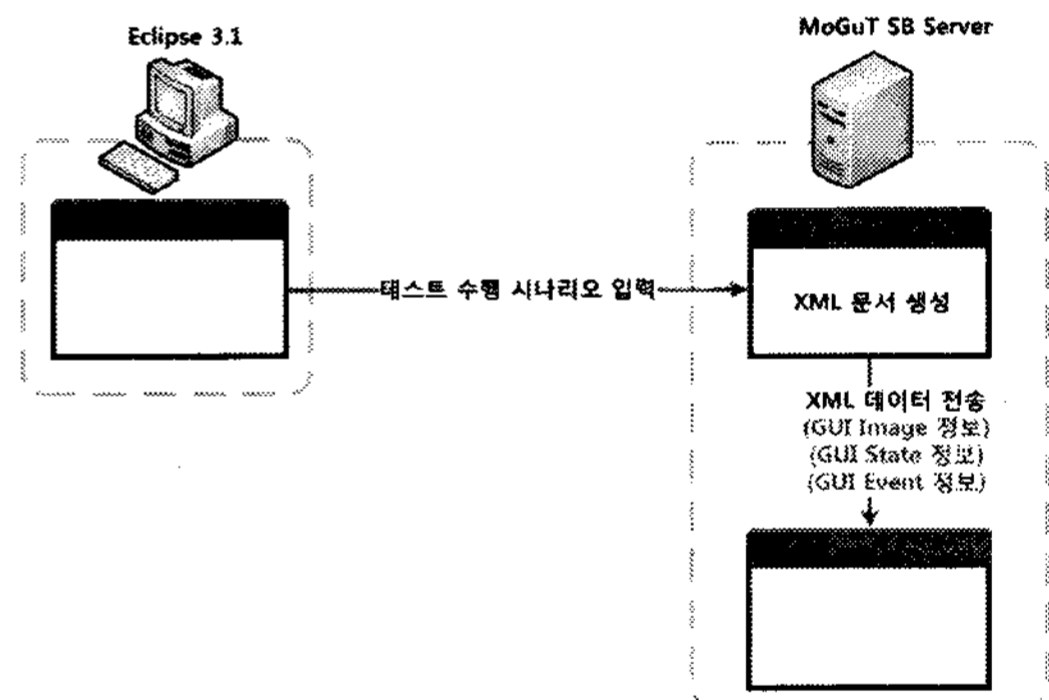
[그림 4] XML 문서 생성 시 필요한 정보 전송

```

- <GUI_Test1>
- <data no="1">
  <image>1.png</image>
  <state>start_menu</state>
  <event>none</event>
  <width>60</width>
  <height>80</height>
  <xposition>0</xposition>
  <yposition>1</yposition>
</data>
</GUI_Test1>
    
```

[그림 5] GUI 정보에 의해 생성된 XML 문서

두 번째 방법은 테스트 시나리오 도구에서 시나리오 작성 정보를 분석하여 XML 문서를 만들도록 구성하였다. XML 문서에는 3가지의 정보(GUI Image, State, Event)로 만들어진다. 3가지의 정보만으로 만들어지는 이유는 분석&테스트를 수행하기 위한 필요조건인 GUI Image 정보, GUI State 정보, GUI Event 정보만이 필요하기 때문이다. 구성은 아래 [그림 6]와 같으며, 각 XML 태그의 속성은 [표 1]과 같다.



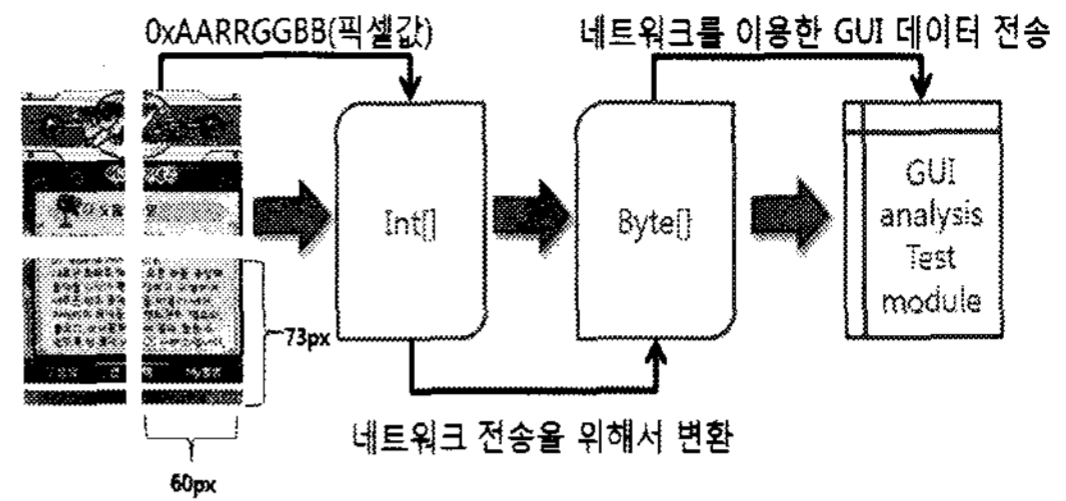
[그림 6] XML 문서 생성되어 Test Case에 전송되는 과정

[표 1] MoGuT Framework 저장용 XML 데이터 설명

태그 이름	속성	설명
GUI_Test	none	XML 시작 태그 이름
data	no	이미지 하나의 정보를 담고 있는 태그
image	none	이미지의 이름을 담고 있는 태그
state	none	상태 변수의 값을 담고 있는 태그
event	none	키 이벤트의 값을 담고 있는 태그
width	none	이미지의 넓이의 값을 담고 있는 태그
height	none	이미지의 높이의 값을 담고 있는 태그
xposition	none	이미지의 X축 시작 좌표 값을 담고 있는 태그
yposition	none	이미지의 Y축 시작 좌표 값을 담고 있는 태그

MoGuT Framework에서 GUI Analysis Test에 전송되는 정보로 만들어진 XML문서와 Test Scenario Tool에서 GUI Analysis Test로 전송되어 만들어지는 XML문서는 서로 다르게 구성되었다 기본적으로 사용되는 공통요소는 같은 MoGuT Framework에서 GUI Analysis Test로 보내져 만들어진 XML문서에는 기본정보인 State, Event 의로 Size 정보와 X,Y축 좌표 값에 대한 정보도 전송된다. 이는 객체 직렬화를 통해 데이터 전송 시 이미지를 분할하여 전송 후 이미지를 다시 조합하기 위해서 Size 값과 X,Y축 좌표 값이 필요하기 때문이다.

하게 되며 이와 같은 방법은 [그림 7]과 같다.



[그림 7] 객체 직렬화를 통한 전송 과정

4. GUI 테스트 자동화 도구 구현 기술

4.1 데이터 직렬화

모바일 플랫폼인 SKVM에서는 일반적으로 PC 보다 성능이 낮은 CPU나 메모리가 사용된다. 그렇기에 JAVA의 스펙 또한 많은 부분이 제외 되어 사용되어 진다. 그에 따른 모바일 스펙은 아래 [표 2]와 같다.

[표 2] 모바일의 CLDC 스펙

구분	CLDC (Connected, Limited Device Configuration)
Processor	16bit 또는 32bit
네트워크	저속 통신망을 사용
메모리	32KB - 512KB
가상머신	K Virtual Machine
목표시장	개인적이고 이동중에 사용하며 상호 연결된 정보 기기
적용 예	휴대폰, 무선통신기, 스마트폰, POS, 단말기

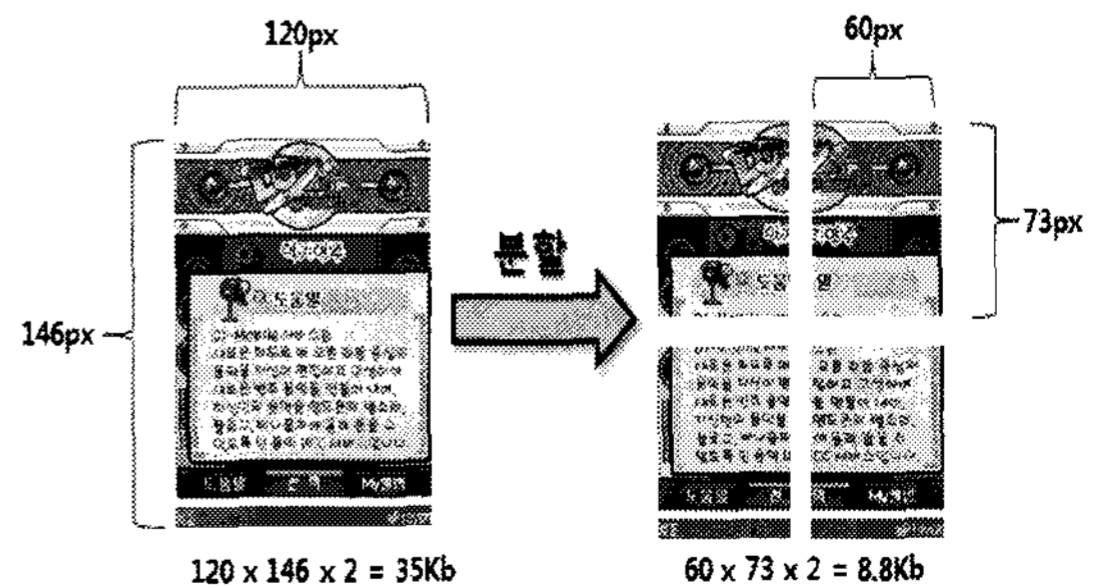
이러한 스펙 때문에 자바에서는 특히 객체 직렬화를 지원하는 리플렉션 기능이 빠져 있다. 본 논문에서 제시하는 프로그램에서는 객체 직렬화가 필수적이나 지원되지 않는 기능 때문에 객체 직렬화를 지원하는 기능을 개발하였다.

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 재변환을 시켜 다시 담는 과정을 함으로써 Encoder 역할을

4.2 이미지 분할

모바일에서는 메모리의 크기가 작기 때문에 화면 하나를 그리는데 드는 메모리 비용이 크다. 일반적으로 메모리의 크기가 모바일에서는 32Kb - 512Kb 인데 기본적으로 드는 메모리 비용이 240 x 320 픽셀 사이즈의 경우 화면 해상도와 색심도의 곱으로 계산하여 152Kb의 메모리가 사용되게 된다. 여기에 GUI 테스트를 위하여 이미지를 픽셀 화하여 저장을 하면 152Kb의 메모리가 더 필요하게 된다. 그리고 네트워크 전송을 위하여 byte 화 시키면 데이터의 양은 456Kb가 되기 때문에 화면을 그리는 데에서 바로 오버헤드가 발생하여 화면에 이미지가 그려지지 않는 일이 발생할 수 있다. 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 8]에 표현되어 있다.



[그림 8] 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. LCD의 크기에 대한 정보와 화면 한 장당에 대한 메모리 점유 크기에 대한 설명은 아래 [표 3]과 같다.

[표 3] 핸드폰의 LCD 사이즈와 메모리 점유량

넓이	높이	메모리 점유량
120 pixel	144 pixel	34,560 byte
120 pixel	146 pixel	35,040 byte
120 pixel	149 pixel	35,760 byte
128 pixel	112 pixel	28,672 byte
128 pixel	128 pixel	32,768 byte
128 pixel	144 pixel	36,864 byte
128 pixel	146 pixel	37,376 byte
176 pixel	178 pixel	62,656 byte
240 pixel	320 pixel	153,600 byte

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.

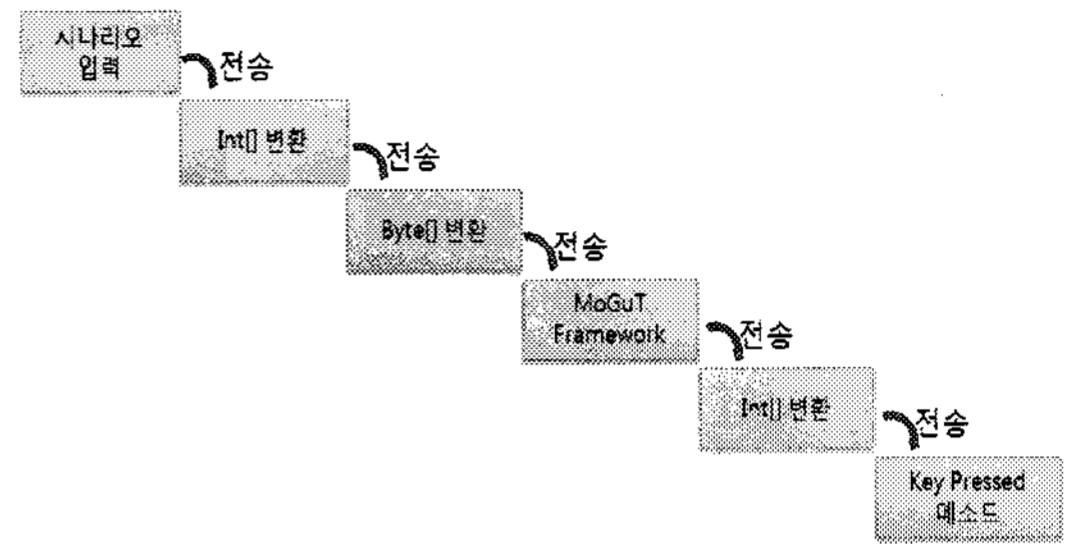
4.3 GUI 분석 테스트의 객체 데이터 복원 기술

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 불러오므로써 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.

4.4 테스트 시나리오 도구의 키 이벤트 처리 기법

테스트 시나리오 도구를 이용하여 테스트 수행 시 테

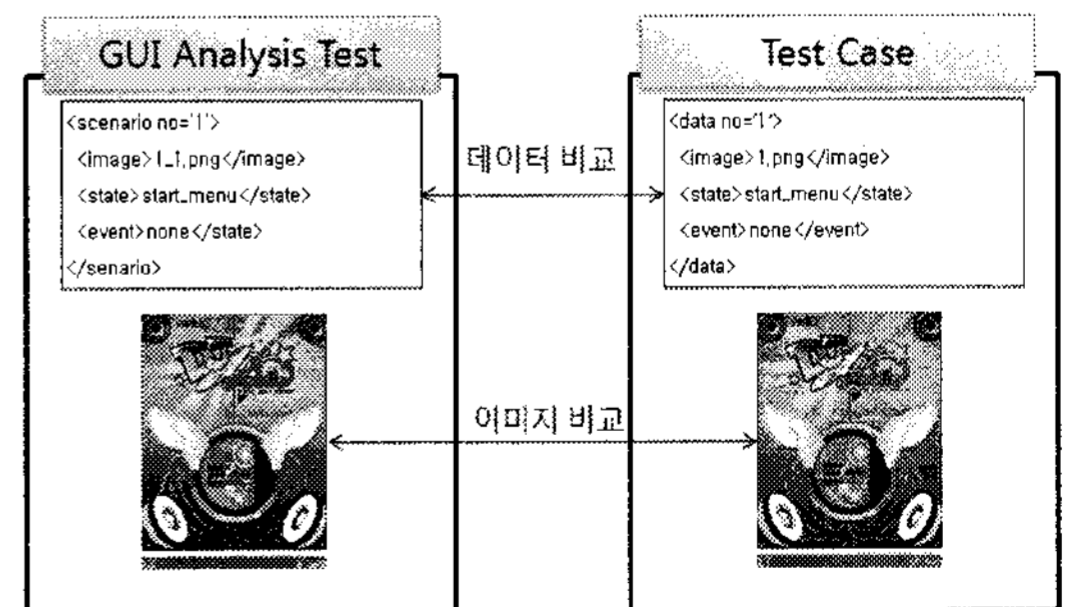
스트 수행 시나리오를 참고하여 개발자 및 테스트는 수행 시나리오를 입력하게 된다. 이때 등록된 정보에 키 이벤트를 발생하여 Emulator에 입력된 키 이벤트를 전송하여 자동으로 수행한다. 또한 이외의 문제점으로 어떠한 특정화면 상태에서 어떠한 키를 눌렀었는지에 대한 정보도 필요하게 된다.



[그림 9] 키 이벤트 처리 기법

4.5 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 10] 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.

5. 시나리오기반 GUI 테스트 자동화 도구 결과

5.1 실험 테스트

실험 테스트 대상으로는 Mobile DJ라는 Mobile Application Software로 새로운 화두로 떠오른 리듬 중심의 음악을 자신이 편집 및 구성하여 새로운 반주 음악을 만들어 내어, 자신만의 새로운 음악을 만들어 내어, 자신만의 새로운 음악을 핸드폰의 벨소리, 블로그, 미니홈피에 올려 들을 수 있도록 한 음악 UCC 서비스이다.

5.1.1 실험 테스트 환경

가. Test Mobile Software

- Mobile DJ

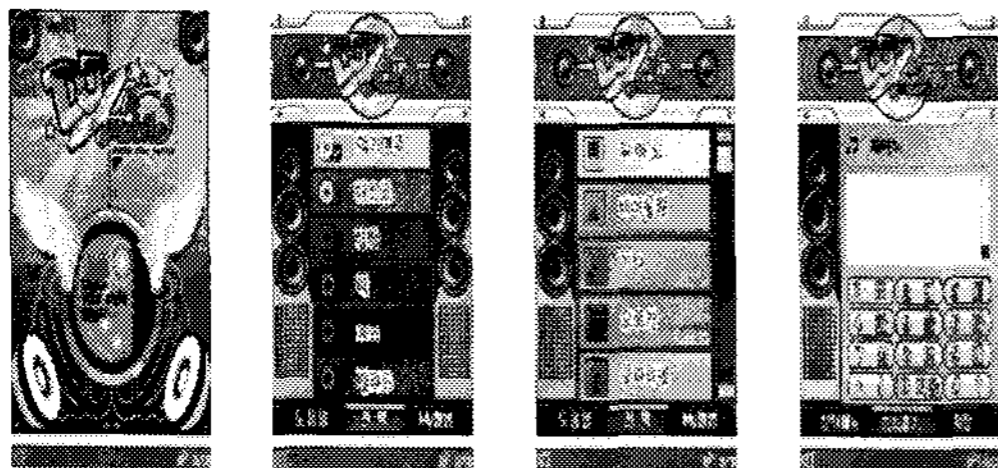
나. 개발 환경

- OS : Windows XP Professional
- 개발환경 : Eclipse 3.1
- 개발언어 : J2ME Java
- Emulator : XCE 2.0.2

다. 기능

- 여러 악기를 통한 음악 제작
- 제작된 음악파일 컴퓨터에 저장 기능
- 핸드폰 벨소리로 제공
- 인터넷을 통한 블로그, 미니홈피 업로드 제공

라. 화면 구성



[그림 11] DJ Mobile의 화면 구성

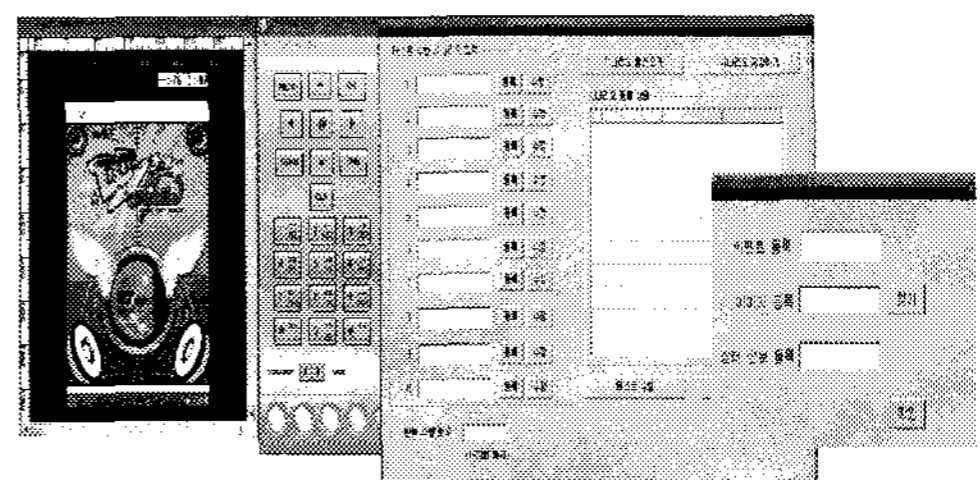
5.2 GUI 테스트 자동화 도구 테스트 시나리오

[표 4]는 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.

[표 4] GUI 테스트 자동화 도구 테스트 수행 시나리오

항목 명	시험 내용	이벤트	상태	파일 명	확인
1. 초기화면 확인	초기화면을 확인	확인	start_m enu	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	
2-5 DJ Mobile 메뉴 확인	메뉴 이동 색변환 확인	down	main_ menu5	6.png	

5.3 GUI 테스트 자동화 도구 수행



[그림 12] XCE2.0.2와 Scenario Test Tool

테스트 시나리오 도구를 이용하여 GUI 테스트 자동화를 수행하기 위한 실행 화면이다. 앞에서 서술한대로 설계되어 구현한 테스트 시나리오 도구를 수행 시나리오

오에 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.

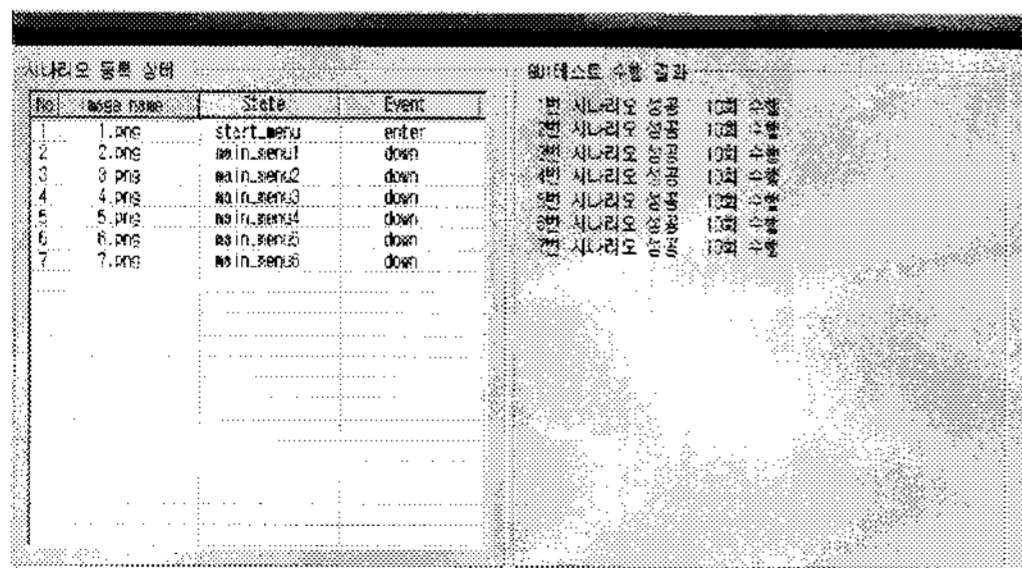
생성된 창에 이벤트 정보 등록 상태 정보 등록, 이미지 파일을 등록하면, 시나리오 등록상태에 순서대로 등록되게 된다. 등록된 시나리오는 수정 가능하며, 다시 등록 할 수 있다.

테스트 수행을 반복적으로 수행하기 위해서는 반복 수행횟수에 총 수행할 테스트 횟수를 입력하면 입력된 수만큼 테스트를 수행한다. 반복적 수행에 의해 1회에서 20회까지 가능하고 반복 테스트 수행 중 개발자나 테스트에 의해 중지 할 수 있다. 전체적인 시나리오의 흐름은 저장이 가능하며 저장된 시나리오는 로드하여 다시 재사용 가능하다.

[표 5] 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 5]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 수행 된 Test 결과 화면은 아래와 같다.



[그림 13] GUI 자동화 테스트 수행 결과

5.4 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 6]와 같다.

[표 6] 장점 및 단점 과 개선 사항

구분	내용
장점	<ul style="list-style-type: none"> - 간단한 시나리오부터 복잡한 시나리오까지 테스트 가능 - XML 데이터 사용으로 XML 데이터 보존 가능 - 기존 GUI 테스트 방식보다 짧은 테스트 처리 시간
단점	<ul style="list-style-type: none"> - GUI 테스트 자동화 도구는 특정 개발환경 - 테스트 대상의 정보를 알고 있어야 테스트 수행이 가능함 - 수행 테스트 환경(컴퓨터 환경)에 따라 테스트 수행 속도가 다를 수 있음
개선 사항	<ul style="list-style-type: none"> - 텍스트 형식의 결과 값에서 시나리오 진행 순서 GUI 표현 - GUI 테스트에서 벗어나 이벤트 테스트 가능토록 구현 예정 - 테스트케이스 10개 제한에서 10개 이상으로 변경

6. 결론

본 논문에서 제안하는 “시나리오기반의 모바일 응용 소프트웨어 GUI 테스트 방법에 관한 연구”는 모바일 소프트웨어의 생산성을 높이고 신뢰성 있는 소프트웨어를 개발하기 위해서 모바일 소프트웨어 테스트의 중요성이 부각되고 있다. 모바일 어플리케이션 소프트웨어에 자동화 테스트를 적용하므로 모바일 어플리케이션의 생산성 관 신뢰성을 향상 시킬 수 있을 뿐만 아니라 테스트 기간 및 비용을 줄일 수 있다.

일반적인 어플리케이션 소프트웨어의 GUI 테스트 도구들은 80년대 말에서 90년대 초 사이에 연구하기 시작한 분야로서 일반 어플리케이션 소프트웨어 시장에서 많은 연구가 진행 중이다. GUI는 일반 어플리케이션 분야뿐만 아니라 임베디드 시장, 인터넷시장, 모바일 시장에서 가장 많은 각광을 받고 있는 분야이다. 소비자의 제품의 선택이 제품의 질에서 벗어나 어플리케이션 소프트웨어의 디자인 면까지 보여 지고 있다.

현재 모바일 시장에서 가장 많이 사용하는 GUI 테스트 방법은 크게 4가지로 나뉘고 있다. Record-Playback 테스트, Capture-Playback방식, 명세기반 테스트, 베타테스트 방법 등이 가장 많이 사용되고 있다. 모바일 업계

에서 가장 많은 방법으로 사용되고 있는 베타테스트 방법은 많은 예산과 시간을 투자하여 전문성이 인증되지 않은 비 인증 테스트들로서 구성된 베타테스트보다 GUI만을 검출하여 분석하고 테스트하는 GUI 테스트 자동화 도구가 필요하다. 본 논문에서는 기존 방식보다 수월하고 반복적인 테스트를 수행가능한 자동화 도구를 구현하는 목적을 두고 연구하였다.

가장 많은 사용자를 확보하고 있는 이동통신사가 제공하는 콘텐츠형 어플리케이션 소프트웨어는 콘텐츠들이 더욱더 GUI화 되리라 보며 앞으로 더욱더 발전하는 GUI시장에 대비하여 시장성을 확보하기위해서 안정성이 확보된 GUI 테스트가 GUI 테스트가 필요할 것이다. 본 논문에서 제시하는 방법은 정적방식의 GUI테스트를 수행하는데 효과성이 높으나 동적방식의 테스트는 아직 불가피하다.

GUI 테스트 자동화 도구는 모바일 소프트웨어 개발 환경에 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 테스트 자동화에 한정을 두지 않고 이벤트 호출 및 존속여부까지 테스트 할 수 있도록 향후 연구가 필요하리라 생각된다.

참고문헌

[1] Myers, G, Sandler, C., Badgett, T., and Thomas, T., The Art of software Testing, Second Edition, John Wiley & sons, 2004

[2] IEEE Standard Glossary of Software Engineering Terminology, IEEE std 610.12-1990, The Institute of Electrical and Electronics Engineers, 1990

[3] Pressman, R., Software Engineering: A Practitioner's Approach, McGraw-Hill, 2003.

[4] 홍준성, "모바일 플랫폼의 기술현황 및 발전방향", 정보과학회지 제22권 21호 통권 제176호, pages 8-14, 2004, 01.

[5] BREW White Paper, BREW and J2ME-A Complete Wireless Solution for Operators Committed to Java, QUALCOMM.

[6] Mark Fewster & Dorothy Graham, Software Test automation, Addison-Wesley, 1999

[7] McGregor, J.D., and T.D. Korson, "Integrated Object-Oriented Testing and Development Process," CACM, vol. 37, no.9, September 1994

[8] Patton, R., Software Testing, Sams, 2000.

[9] 한국정보통신기술협회, 소프트웨어테스트 전문기술 기초분야, 한국정보통신기술협회, 2005

[10] 박상필. "이미지 플로우 기반의 모바일 GUI 테스트 도구에 관한 연구."

[11] 권원일. "모바일 소프트웨어 테스트 현황과 표준적인 테스트 케이스"

[12] NIST, "The Economic Impacts of Inadequate Infra-structure for Software Testing", 2002.5

[13] 이상윤, 김선자, 김홍남, 한국 무선 인터넷 표준 플랫폼(WIFI)의 표준화 현황 및 발전 방향, 정보과학회지 제22권 21호 통권 제176호, pages 16-23, 2004. 01.

[14] "QtUnit(Software development testing framework) Introduction Manual"

[15] Kent Beck, Erich Gamma, "JUnit Cookbook"

황 선 명(Sun-Myung Hwang) [정회원]



- 1984년 중앙대학교 소프트웨어 공학전공 (이학석사)
- 1987년 중앙대학교 소프트웨어 공학전공 (이학박사)
- 2000년~현재 한국S/W프로세스 심사인협회(KASPA) 이사
- 2000년~현재 한국정보처리학회 논문지 편집위원
- 1997년~현재 ISO/IEC J T C 7/ WG10 한국운영위원
- 1989년~현재 대전대학교 컴퓨터공학과 교수

<관심분야>

소프트웨어 프로세스 모델, 품질 매트릭스, 소프트웨어공학 표준화, 컴포넌트 품질측정, 테스트 방법론 등

김 정 종(Jung-Jong Kim) [정회원]



- 1977 중앙대학교 전자계산학과 (학사)
- 1981 중앙대학교 대학원 전자계산학과(이학석사)
- 1988 중앙대학교 대학원 전자계산학과(이학박사)
- 1981~현재 경남대학교 컴퓨터공학과 교수

<관심분야>

소프트웨어공학, 객체지향 방법론, 컴퓨터교육, 컴포넌트 등