

논문 2012-2-7

스마트 앱 프로그램 유사성 비교

이규대*

A comparative verification of the android application program on object code

Lee Kyu-Tae*

요 약

정보기기용 휴대용 단말기를 대표하는 스마트폰의 확산으로 안드로이드기반 앱 프로그램이 활성화 되고 있다. 스마트폰 프로그램은 인터넷과 연동되는 특성상 유용한 기능의 앱은 유사 프로그램의 제작이 용이해 저작권 피해가 우려되는 상황이다. 본 연구에서는 앱프로그램의 복제에 의한 분쟁이 발생시, 소스코드 및 실행코드의 유사성 분석을 위한 방안을 제시한다.

Abstract

The smart phone is the one of the best mobile IT devices. It is a hardware system which has special processor and interface modules for it's own function under Android Operating System. An application program called App. has been a useful program utility by seamless internet connection. The App. program created by an Eclipse tool is exposed in general. On the property right verification of these software, it should be evaluated on the source code. This paper shows how to make App program and the a verification procedure with object code.

한글키워드 : 스마트폰, 앱프로그램, 유사도 측정, 목적코드

1. 서 론

IT 정보기기는 구성면에서 프로세서를 기반으로 하는 입출력 하드웨어와 운영체제를 기반으로 하는 응용소프트웨어의 일체형으로 제작되고 있다. 스마트 폰은 이러한 정보기기의 모든 기능을 하나로 일체화한 특징을 가지며, 네트워크의 연

결기능으로 활용성이 증대하는 장점이 있다. 스마트 폰의 기능은 앱이라는 아이콘을 클릭하는 방법으로 사용되며, 이러한 앱은 자바언어를 안드로이드 운영체제에 동작하도록 작성된 프로그램을 말한다. 프로그램의 개발은 이클립스로 알려진 개발도구를 사용한다.[2] 프로그램의 작성 컴파일, 실행코드 전송 등 앱 개발을 위한 대부분의 기능을 제공하며, 프로그램 개발자는 이 개발환경에서 자바 언어를 사용하여 개발을 수행한다.

* 공주대학교 정보통신공학부
(email: ktleee@kongju.ac.kr)
접수일자: 2012.11.25 수정완료: 2012.12.16

안드로이드OS의 장점은 리눅스 기반의 Open OS라는 점으로, 휴대폰 제조사에 맞게 수정이 가능하도록 설계가 되었고 자바 개발환경에서 어플리케이션을 만들 때 통신사 휴대폰 기종에 맞게 따로 개발해야 하는 번거로움이 없다.

앱의 활성화에 따라 유사기능의 앱이 등장하면서 저작권 침해에 대한 관심이 증대하고 있다. 기존의 감정과 같이 소스코드가 제공되면 감정평가가 용이하다. 그러나 소스코드 기준의 감정을 어렵게 하는 부분이 소스코드확보가 어려운 경우이다. 분쟁 당사자들의 자료제공이 선행되어야 하나, 복제 의심을 받는 측에서 소스코드를 거부하는 경우, 또는 감정의 특성상 정보기기만 확보되어, 제품의 내부에 저장된 실행코드만으로 검증해야 하는 상황들이 발생한다. 또한 소스코드가 확보되지 못하고, 실행 앱화일이 확보된 경우, 유사성 비교에 있어 기능적 동작의 유사성 및 기술적 유사성에 대한 판단에 공지기술의 적용 범위를 검증한다.

본 연구는 앱 실행코드에 대한 유사성비교에 서의 과정 및 유사도의 공정성에 대해 다룬다.

2. 안드로이드 플랫폼

안드로이드는 자바 프로그램언어의 코어 라이브러리에서 대부분의 기능을 제공하는 코어 라이브러리의 집합이다. 어플리케이션은 자신의 프로세서에서 하나의 인스턴스와 함께 동작하며, 하나의 프로세서에 VM 인스턴스가 생성된다. Dalvik 는 디바이스가 다중 VM을 효과적으로 동작하도록 작성되며, 작은 메모리에 최적화 된 Dalvik Executable(dex) 포맷의 파일을 실행한다. VM은 레지스터 기반으로 (dex) 포맷으로 변환하는 java 언어 컴파일러에 의해 컴파일된 클래스를 실행한다.

VM 은 쓰레드, 저수준의 메모리관리 등, 기초적인 기능에 대해 리눅스커널을 사용한다.

2.1 안드로이드 프로그램 구조

프로그램은 안드로이드운영체제에 액티비티 어플리케이션 으로 구성된 "프리젠테이션 계층"이다. 사용자 어플리케이션에서 모든 화면은 액티비티(Activity) 클래스의 확장이 된다. 일반 어플리케이션의 윈도우 대화상자와 유사한 구조이지만 윈도우나 대화상자처럼 화면 구성요소가 있어야 되는 것은 아니며, 화면이 없는 "헤드리스(headless)" 프로그램은 콘텐츠 프로바이더나 서비스 형태로 사용 가능하다.

- 콘텐츠 프로바이더 [Content Provider] 콘텐츠 프로바이더(Content Provider)는 공유할 수 있는 데이터의 저장소이다. 어플리케이션 데이터베이스를 관리하고 공유하며, 각종 데이터에 대한 추상적인인터페이스 제공 및 여러 어플리케이션을 사용한다. 콘텐츠 프로바이더는 데이터에 대한 통제권을 확보한 상태에서 개발중인 어플리케이션뿐만 아니라 다른 어플리케이션에서도 접근 가능하도록 기능을 제공 할 수 있다.



그림 1. 안드로이드 플랫폼 구조

2.2 안드로이드 앱 개발환경

안드로이드 앱 개발을 위한 프로그램 환경은 다음과 같다.

1) JDK (Java Development Kit):

JDK[제이디케이]는 썬 마이크로시스템즈에서 나온 자바 소프트웨어 개발 환경으로, 자바 가상머신(JVM; Java Virtual Machine)과 컴파일러, 디버거, 그리고 자바 애플릿 및 응용프로그램 개발을 위한 도구들이 포함된다.

2) 이클립스 (Eclipse):

이클립스는 다양한 플랫폼에서 사용가능하며, 자바를 비롯한 다양한 언어를 지원하는 프로그래밍 통합 개발 환경(IDE; Integrated Development Environment)을 목적으로 시작하여, 현재는 OSGi를 도입하고, 범용 응용 소프트웨어 플랫폼으로 진화하였다. 자바언어로 작성되어 있으며, 자유 소프트웨어이지만 식(蝕)을 뜻하는 이클립스란 이름은 자바를 개발한 회사의 Sun이란 이름을 염두에 두었다는 설도 있다. 초기 IBM의 WebSphere Studio Application Developer란 이름으로 개발되었던 것인데, 엔진부분을 오픈소스로 공개한 것을 기반으로 지금의 이클립스로 발전해 왔다.

3) 안드로이드 SDK

(Android Software Development Kit): Android SDK란 ‘Android Software Development Kit’의 약자이고, SDK는 Software Development Kit를 어원으로 한다.

SDK 에는 아래와 같은 것들이 포함되어 있다.

- (1) 안드로이드 API - 개발자들이 안드로이드 개발을 손쉽게 하도록 접근을 허용하는 안드로이드 API들
- (2) 안드로이드 응용 프로그램 개발에 필요한 각종 도구들 - 안드로이드 소스 코드를 실행 가능한 어플리케이션으로 만들기 위한 컴파

일러나 에뮬레이터 등

- (3) 샘플 코드들 - 안드로이드 API사용법을 이해시키기 위한 예제 프로그램 코드들
- (4) 관련 문서들 - SDK를 사용하기 위한 각종 패키지나 클래스에 대한 설명 등

2.3 안드로이드 앱 프로그램 작성

단말기 화면에 <hello world>를 표현하는 과정을 통해 앱프로그램의 개발 과정을 추적한다.

그림과 같이 android application project을 지정한다.

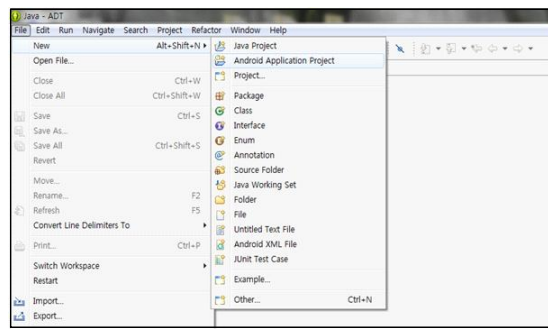


그림 2. 프로젝트 지정

클릭 후 아래 화면이 나타나면, 창에 그림의 내용을 입력한다.

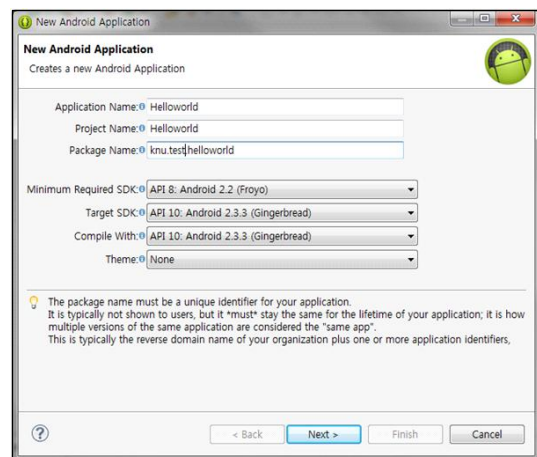


그림 3. 어플리케이션 지정

next 버튼과 finish 버튼을 동작하여 프로젝트를 완성한다.

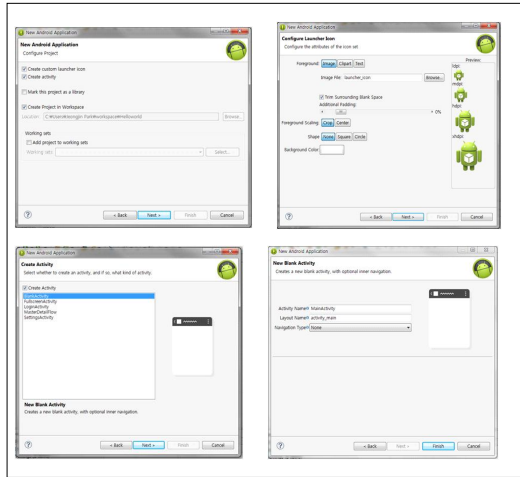


그림 4. app 아이콘 작성

위와 같이 프로젝트 설정과정에서 앱의 이미지를 작성하고, 설계가 가능하다.

2.4 안드로이드 앱 프로그램 동작

개발과정에서 앱을 실행하고 테스트하는 방법에는 두 가지가 방법이 있다. 하나는 PC에서 가상 안드로이드 기기를 만들어 그 안에서 실행하는 방법과 실제 폰이나 태블릿에 PC와 연결하여 실행하는 방법이 있다. 안드로이드 기기를 소유하고 있지 않다면 위의 방법으로 해야겠지만, 부팅부터 실행까지 너무 시간이 많이 걸리고 시스템이나 앱의 조작도 시간이 소요되어 실제 기기에서 테스트 하는 방법이 많이 사용된다.

여기서는 실제 기기를 이용하여 테스트하는 방법을 설명한다.

1. 자신이 소유한 기기의 제조사 사이트에 접속하여 USB드라이버를 다운로드하고 설치 .
2. 안드로이드 기기의 USB디버깅을 활성화

안드로이드 버전 2.X의 기기: 시스템설정
→ 애플리케이션 → 개발 → USB 디버깅 체크

안드로이드 버전 4.X의 기기: 시스템설정
→ 개발자옵션 → USB 디버깅 체크

3. 그림과 같이 이클립스에서 프로젝트폴더 우클릭 → Run As → Android Application 클릭
그림과 같이 단말기의 화면이 나타나고, 화면 속에서 실행된 결과가 표현된다.

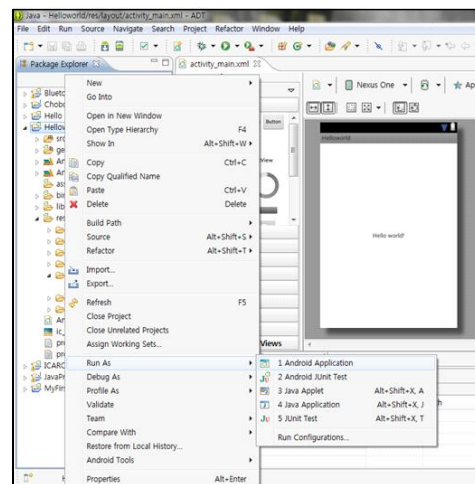


그림 5. 작성된 앱의 동작 확인

3. 앱프로그램 비교

안드로이드 운영체제 기반 스마트폰의 구조는 프로세서 인터페이스를 참조하여 작성되는 자바 소스코드(원시 프로그램)로 해당 프로세서가 인식하는 기계어(object code)로 번역되어 실행코드로 작성된다.

이 과정을 담당하는 프로그램을 컴파일러라고 하며, 일반적으로 java 언어 형태의 프로그램이 ecclips 개발환경에서 작성된다. 컴파일러는 해당 프로세서의 모델, 프로그램 코드 저장 위치, 클럭 속도 등을 고려하여 프로세서가 인식하는 기계어를 생성한다. 작성된 기계어는 프로세서내의 프로그램 메모리에 복사된다.

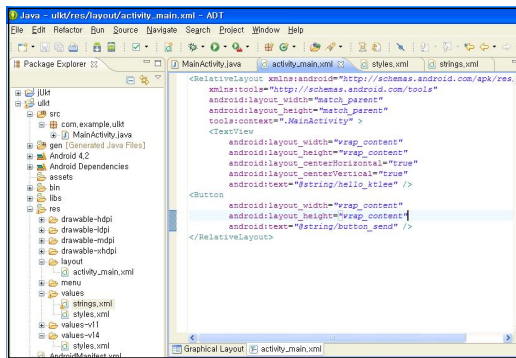


그림 6. eclipse 작업 환경

실행코드는 USB 케이블이나, 앱 다운로드 사이트를 사용하여, 기계에 전송되고, 설치됨으로써 사용이 가능하다. 에뮬레이터는 그림과 같이 사용하는 스마트 기기와 유사한 화면과 기능을 제공하여 작성된 앱 프로그램의 실행 상황을 그림과 같이 확인할 수 있다.

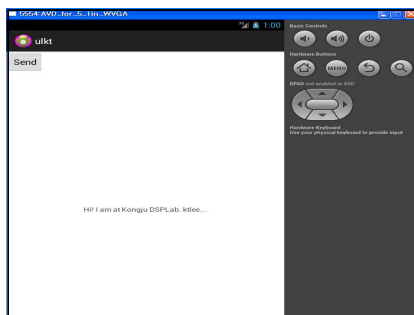


그림 7. App 실행 화면(에뮬레이터)

본 연구에서는 프로그램 소스가 개작된 상황을 보이기 위해 아래 그림과 같이 화면에 표시되는 문자열을 추가 수정하고, 수정되기 이전의 파일과 유사성 비교를 시도하였다.

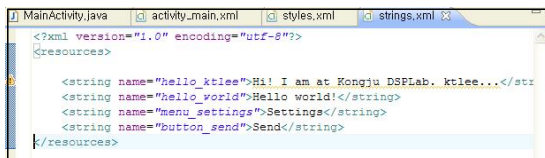


그림 8. 프로그램 수정 화면

프로그램을 작성하고, 컴파일을 하여 실행화일을 작성하면, *.apk 파일이 생성된다. 이 파일이 안드로이드 운영체제에서 실행되는 기계어 파일로, 목적코드와 같은 의미를 갖는다.

목적코드는 일반적인 프로세서의 기계어와 같은 용도로 사용되고, 그림과 같이 hex 파일의 구조를 갖는다.

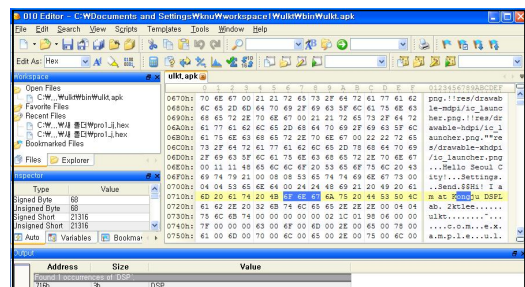


그림 9. “seoul city” 수정부분

기계어로 작성된 유사화일을 비교하는 방법은 beyond compare 와 같은 소스코드 유사성을 검출하는 상업용 비교도구를 사용하여 유사도를 검출한다.

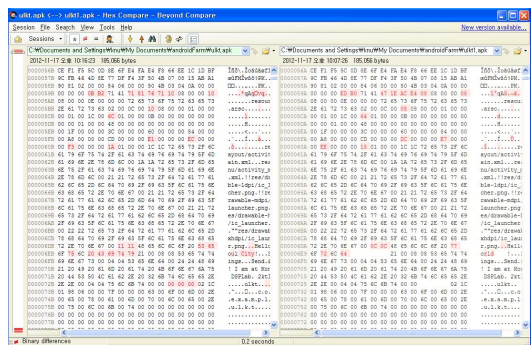


그림 10. 원본과 수정본의 apk 파일 비교

그림 8과 같이 기계어로 작성된 두 개의 화일은 수정이전의 파일과 수정이 가해진 파일의 비교결과를 보인다. 비교작업이 수행되고, 유사한 부분이 발견되면 적색으로 표시하여 위치를 표시한다.

이번 실험에서의 유사도 측정결과는 전체

185066bytes 중에서 183828 bytes 의 유사도가 검출되어 99.3%의 유사성을 보인다. 이는 java 언어의 특성상 기본적으로 생성되는 동일코드가 많아서 나타나는 현상이다.

따라서 안드로이드 앱프로그램의 유사성 비교 시, 자동 생성되는 공지기술부분은 제외하고, 저작권자가 직접 개발하는 부분에 대한 유사성 검출 과정이 필요하며, 판별의 정확성을 위해서는 실행 앱코드보다 소스코드의 비교를 위한 자료 확보가 중요한 것으로 나타났다.

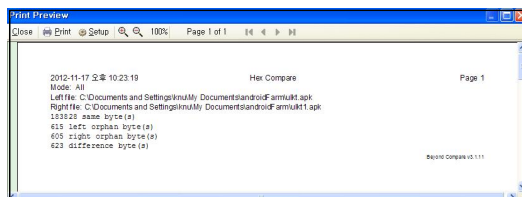


그림 11. 유사도 리포트

프로세서 인터페이스를 참조하여 작성되는 소스코드(원시 프로그램)은 해당 프로세서가 인식하는 기계어(object code)로 번역되어야 한다. 이 과정을 담당하는 프로그램을 컴파일러라고 하며, 일반적으로 C 언어 형태의 프로그램언어가 사용되고 있다. 컴파일러는 해당프로세서의 모델, 프로그램 코드 저장 위치, 클럭 속도 등을 고려하여 프로세서가 인식하는 기계어를 생성한다. 작성된 기계어는 프로세서내의 프로그램메모리에 복사되어야 하는데, 이 과정을 다운로드(download) 한다고 한다.

4. 결 론

스마트폰의 활용성이 증가하면서, 다양한 기능의 앱이 개발되고 있다. 스마트기기의 프로세서 성능을 고려하여, 단순하고, 적은 용량의 프로그램을 작성하는 관계로 활용성이 높고, 부가가치가 있는 기능의 앱은 복제의 가능성을 갖는다.

본 연구에서는 앱프로그램의 저작권침해 발생시, 유사도를 검출하는 과정 및 방법을 보였다. 가상환경을 설정하고, 유사성을 검출한 결과, 자바 언어의 공지기술이 포함되어, 유사도가 낮게 검출되고, 기계어(목적어)의 특성상 프로그램의 유사성이 낮게 검출되는 특징을 보였다.

정보기기 분쟁에서 소스코드 기준의 감정을 어렵게 하는 부분이 프로그램 소스의 불성실한 제공이다. 앱 프로그램의 분쟁에서도 소스코드의 확보를 통해 감정결과와 정확성을 향상하는 방향의 제도가 요구된다.

참 고 자 료

- [1] ko.wikipedia.org/wiki/이클립스
- [2] 전병태, "프로그램 복제도 감정기법 및 감정비 산출에 관한 연구" 프로그램심의조정위원회 결과보고서 2002.
- [3] 박선희, 오명환, ARM Cortex-M3 시스템 프로그래밍, D&W WAVE, 2010
- [4] www.eclipse.org
- [5] http://developer.android.com
- [6] 이규대, "임베디드시스템의 이진코드 추출 및 분석", 한국소프트웨어감정평가학회 논문지, 5권1호, pp27-38, 2009.5.

저 자 소 개



이규대(Kyu-Tae Lee)

1984 고려대 전자공학과 졸업
1986 고려대 전자공학과 석사
1991 고려대 전자공학과 박사
2001 미 조지아텍 교환 교수
2006 미 일리노이주립대 교환 교수

2007~2009 : 한국전자통신연구원 이동통신연구소 초빙연구원

1992.3~현재 : 공주대 정보통신공학부 교수
<관심분야> 회로 및 시스템, 신호처리, VLC