KIPS Tr. Software and Data Eng.
Vol.5, No.4 pp.165~170 pISSN: 2287-5905

http://dx.doi.org/10.3745/KTSDE.2016.5.4.165

Android Intent Based Component Interaction Diagram Generation and Test Scenarios Design Techniques

Tae-San Baek† · Woo Jin Lee††

ABSTRACT

Using the intent of the Android application, an application can execute other application's components. However, if interaction between these components are not processed normally, such problems as incorrect component execution and unhandled system broadcast may be occurred. In this paper, to generate test scenarios for inter application interaction, a testing approach is proposed using a merged intent list and a single merged diagram. The proposed method can effectively be carried out to check the abnormal interaction among the applications which was not considered in existing testing approaches.

Keywords : Android, Interaction Testing, Automated Generation of Test Scenario, Automated Testing

안드로이드 인텐트 기반 컴포넌트 상호작용 다이어그램 생성 및 테스트 시나리오 설계기법

백 태 산† · 이 우 진††

요 약

안드로이드 어플리케이션은 인텐트를 이용하여 다른 어플리케이션의 컴포넌트를 호출하여 실행시킬 수 있다. 하지만 컴포넌트들 사이의 상호작용이 정상적으로 처리되지 않으면 잘못된 컴포넌트가 실행되거나 전화 수신과 같은 시스템 브로드캐스트를 처리하지 못하는 문제점이 발생할 수 있다. 본 논문에서는 이와 같은 상호작용 문제점을 검사하기 위해 서로 연동하여 동작하는 어플리케이션들로부터 컴포넌트 기반의 다이어그램들을 생성하고 이를 하나의 병합된 다이어그램으로 변환하여 테스트 시나리오를 생성하는 방법을 제안한다. 제안된 방식은 기존의 테스팅 기법에서 고려하지 못한 어플리케이션간의 상호작용 검사를 효율적으로 수행할 수 있다.

키워드 : 안드로이드, 상호작용 테스팅, 테스트 시나리오 자동생성, 자동화 테스팅

1. 서 론

전 세계 스마트폰 사용자 중 10명 중 8명은 안드로이드 플랫폼 기반의 단말기들이 사용되고 있으며, 비교적 간단한 인증절차와 다양한 경로를 통해 한 달에도 약 20만 개의 어플리케이션들이 출시되고 있다[1, 2]. 이런 대부분의 어플리케이션들은 전문 검증팀에 의한 검증과정을 거치지 않고 출시되어 신뢰성을 확보하지 못하고 있다. 안드로이드 어플리케이션은 4개의 컴포넌트(액티비티, 서비스, 브로드캐스트 리시버, 컨텐트 프로바이더)로 구성되어 있으며 이들 컴포넌트들 간의 상호작용은 인텐트를 통해 이루어진다. 인텐트를 통해 하나의 어플리케이션내의 컴포넌트뿐 아니라 다른 어플리케이션의 컴포넌트와도 연동될 수 있다. 어플리케이션간 상호작용 테스팅은 잘못된 인텐트 호출로 발생할 수 있는 어플리케이션의 오동작, 보안상의 문제점 등을 검사하여 발견할 수 있다.

인텐트는 어플리케이션내의 컴포넌트뿐 아니라 다른 어플리케이션의 컴포넌트들도 호출할 수 있다. 어플리케이션간의 상호작용 테스팅을 수행하기 위해서는 하나의 어플리케이션이 아닌 서로 관련이 있는 2개 이상의 어플리케이션들

※ 본 논문은 2014년도 정부(교육부)의 재원으로 한국과학재단의 지원을 받아 수행된 기초연구사업(No. NRF-2014R1A1A2058733)과 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임[No. 10041145. 자율군집을 지원하는 웰빙형 정보기기 내장 소프트웨어 플랫폼 개발].

† 준 회 원 : 경북대학교 컴퓨터학부 박사과정
†† 정 회 원 : 경북대학교 컴퓨터학부 교수

Manuscript Received : January 7, 2016
First Revision : March 17, 2016
Second Revision : March 30, 2016
Third Revision : April 6, 2016
Accepted : April 7, 2016

* Corresponding Author : Woo Jin Lee(woojin@knu.ac.kr)

의 컴포넌트와 인텐트 정보를 모두 고려하여야 한다. 본 논문에서는 각 어플리케이션 소스코드로부터 컴포넌트 정보와 인텐트 목록을 추출하여 각 어플리케이션의 인텐트 기반의 컴포넌트 다이어그램을 생성한다. 각각의 다이어그램을 인텐트 목록에서 다른 어플리케이션을 호출하는 인텐트 정보를 이용하여 하나의 결합된 다이어그램을 생성할 수 있다. 그리고 결합된 다이어그램으로부터 어플리케이션간 상호작용을 검사할 수 있는 테스트 시나리오를 생성한다. 이와 같인 생성된 테스트 시나리오는 기존의 테스트 시나리오 생성기법보다 소스코드 커버리지 향상된 테스팅을 진행할 수 있다.

2. 관련 연구

안드로이드 어플리케이션은 C언어나 Java언어로 구현된 데스크톱 또는 다른 플랫폼 기반의 모바일 소프트웨어와 달리 4개의 컴포넌트가 모두 진입점으로 사용된다. 즉, 안드로이드는 메인함수와 같은 유일한 진입점이 따로 있는 것이 아니라 처음으로 생성되는 인스턴스 생성자가 실질적인 진입점 역할을 한다. 진입점으로 사용되는 4개의 컴포넌트는 사용자 인터페이스의 기본 단위인 액티비티, 사용자 인터페이스 없이 백그라운드에서 반복적인 작업에 사용되는 서비스, 시스템 또는 다른 어플리케이션에서 발생되는 신호를 전달받는 브로드캐스트 리시버, 폴더내의 데이터 또는 데이터베이스에 접근할 때 사용하는 컨텐트 프로바이더 등이 있다[3]. AndroidManifest.xml 파일에는 어플리케이션내의 컴포넌트를 기술하고 다른 어플리케이션이 가지도록 요구되는 허가 선언, 시작 액티비티 정보, 링크되어야 할 라이브러리, 컴포넌트 각각을 구현하는 클래스 지정 및 기능 정보 등을 포함하고 있다.

기존의 안드로이드 어플리케이션 테스팅은 GUI 기반 테스팅, 보안 테스팅, 성능 분석을 위한 테스팅 등이 있다. GUI 기반 테스팅 방법에는 JUnit을 이용한 단위 테스팅, 명세 기반 테스팅, 액티비티 라이프사이클 기반 GUI 테스팅 등이 있다[4, 5, 6, 7]. 이러한 기존의 테스팅들은 하나의 어플리케이션만을 고려하고 어플리케이션 내부의 액티비티 호출 인텐트만 고려하고 있어 다른 어플리케이션과의 상호작용으로 발생될 수 있는 문제점들을 검사 및 발견할 수 없다. 기존의 인텐트를 이용한 상호작용 테스트에 대한 연구는 어플리케이션 내부 컴포넌트들의 상호작용에 대한 연구와 외부 어플리케이션의 컴포넌트와의 상호작용에 대한 연구로 나눌 수 있다. 기존 대부분의 안드로이드 어플리케이션 테스팅에 대한 연구는 어플리케이션 내부 컴포넌트들의 상호작용만을 고려하고 있고 외부 어플리케이션의 컴포넌트와의 상호작용에 대한 연구는 어플리케이션 외부의 컴포넌트를 호출할 때 발생할 수 있는 보안 취약점을 분석하기 위한 연구가 있다. 하지만 보안 취약점을 분석하는 연구에서는 호출되는 어플리케이션의 정보는 고려하지 않고 호출하는 행위만을 고려하고 있어 어플리케이션이 연동되어 동작하는 부분에서 발생되는 오류를 검출할 수 없다. 본 논문에서는 어플리케이션 외부의 컴포넌트들에 대한 상호작용 테스팅을 위해 상호작용하는 어플리케이션들의 모든 컴포넌트와 인텐트 정보를 고려한다.

3. 상호작용 테스팅을 위한 테스트 시나리오 생성

본 논문에서는 안드로이드 어플리케이션간의 상호작용 테스팅을 위한 테스트 시나리오 생성을 위해, 각 어플리케이션의 설치파일인 APK파일에서 Java 바이트 코드 분석 도구인 soot을 이용하여 Dalvik 바이트 코드를 역컴파일 변환하여 Java 바이트 코드를 생성하여 어플리케이션의 컴포넌트 정보와 인텐트 목록을 추출해낸다. 추출한 컴포넌트 정보와 인텐트 목록으로 각 어플리케이션의 컴포넌트 다이어그램을 생성하고 인텐트 목록에서 어플리케이션 외부의 컴포넌트를 호출하는 인텐트 정보를 고려하여 각각의 컴포넌트 다이어그램을 하나의 컴포넌트 다이어그램을 생성한다. 생성된 다이어그램에서 어플리케이션 상호작용 테스팅을 위한 테스트 시나리오를 추출해낸다. Fig. 1은 어플리케이션간 상호작용 테스팅을 위한 테스트 시나리오 생성 절차를 보여준다.

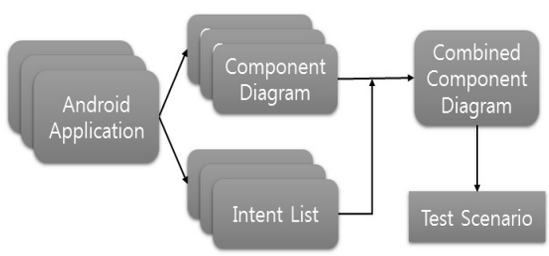

Fig. 1. android app interaction testing process

3.1 APK파일을 Java 바이트코드로 변환

APK(Android application PacKage)파일은 어플리케이션에 대한 정보 및 실행권한 등의 정보를 담고 있는 AndroidManifest.xml 파일과 Dalvik 가상머신에서 동작하는 바이트코드인 classes. dex 파일 등으로 구성되어 있다. 바이트코드 분석 도구인 soot API를 활용하여 APK내의 androidManifest.xml 파일과 classes. dex 파일을 분석 가능한 자바 바이트 코드로 변환한다[9].

3.2 어플리케이션별 컴포넌트 다이어그램 및 인텐트 목록 생성

변환된 자바 바이트 코드로부터 컴포넌트들 간의 호출 정보를 추출하여 Fig. 2와 같이 안드로이드 컴포넌트 기반의 다이어그램을 생성한다. 컴포넌트 기반의 다이어그램에는 기존의 생성기법에서는 고려하지 않았던 외부 컴포넌트와의 관계도 함께 표시되어있다. 그리고 APK 파일내의 AndroidManifest. xml 파일의 컴포넌트 속성 정보를 이용하여 Table 1과 같이 인텐트 목록을 생성할 수 있다.

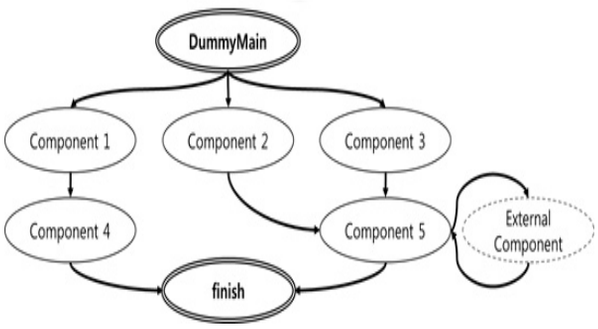

Fig. 2. Android component based diagram

Table 1. Intent list

Category	Intent Name
activity	com.android.appA.ComponentA
activity	com.android.appA.ComponentB
…	…
receiver	com.android.appA.ComponentMProvider
receiver	com.android.appA.ComponentNProvider

3.3 외부 컴포넌트 호출 인텐트 반영 컴포넌트 다이어그램 연동

어플리케이션간의 상호작용 테스팅을 위해서는 하나의 어플리케이션이 아닌 상호작용하는 모든 어플리케이션의 인텐트 정보와 컴포넌트 다이어그램을 이용하여야 한다. 이를 위해 Table 1과 같이 각 어플리케이션으로부터 생성된 인텐트 목록으로 Table 2와 같이 인텐트의 종류와 이름, 호출대상 어플리케이션의 종류 정보를 포함하고 있는 병합된 인텐트 목록을 생성한다.

Table 2. Combined Intent list

Category	Intent Name	App
activity	com.android.appA.ComponentA	A
activity	com.android.appA.ComponentA	A
activity	com.android.appB.ComponentB	B
activity	com.android.appB.ComponentB	B
…	…	
receiver	com.android.appA.ComponentMProvider	B
receiver	com.android.appA.ComponentNProvider	B

각 어플리케이션에서 생성된 컴포넌트 기반의 다이어그램을 하나로 병합하기 위하여 Table 2의 정보를 활용하여 A 어플리케이션의 다이어그램에서 B 어플리케이션의 컴포넌트를 호출하는 인텐트를 포함하고 있는 컴포넌트(Application A-Component 5)를 호출되는 대상인 B 어플리케이션의 컴포넌트(Application B-Component 1)와 연결하여 Fig. 3과 같이 병합된 컴포넌트 다이어그램을 생성한다.

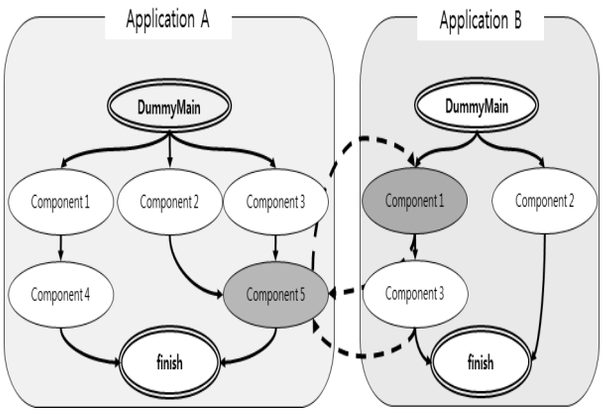

Fig. 3. Combined Component Diagram

3.4 상호작용 테스팅을 위한 테스트 시나리오 생성

병합된 컴포넌트 다이어그램은 다른 어플리케이션과의 상호작용과 관계가 있는 컴포넌트뿐만 아니라 어플리케이션 내부의 컴포넌트만 호출하는 컴포넌트도 포함되어 있다. 이와 같이 어플리케이션 내부의 컴포넌트만 호출하는 컴포넌트는 기존의 단위 테스팅, GUI 테스팅 등과 같이 성능 및 효율성이 검증 된 다양한 테스팅 기법이 존재한다. 본 논문에서 제안하는 어플리케이션 상호작용 테스팅에서는 다른 어플리케이션의 컴포넌트를 호출하는 컴포넌트만을 테스팅 대상으로 고려한다. 이를 위해 다른 어플리케이션의 컴포넌트를 호출하는 컴포넌트만을 고려한 추상화된 컴포넌트 다이어그램을 생성해야 한다. 이를 위해 어플리케이션 B의 컴포넌트를 호출하는 어플리케이션 A의 컴포넌트 5 상태로 갈 수 있는 경로와 어플리케이션 B의 호출되는 컴포넌트(Application B-Component 1)와 호출하는 컴포넌트(Application B-Component 3) 정보만을 고려한 추상화된 컴포넌트 다이어그램에서 Fig. 4와 같이 생성한다.

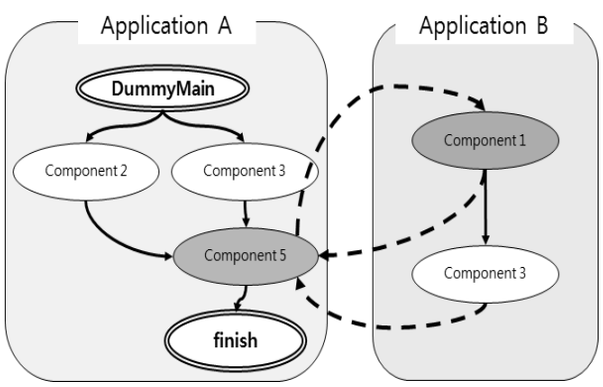

Fig. 4. Abstracted Combined Component Diagram

Fig. 4와 같이 생성된 상호작용 기반의 컴포넌트 다이어그램에서 테스트 시나리오는 유사한 형태의 다이어그램인 상태 챠트 다이어그램에서 테스트 케이스를 생성하는 기법[10]을 활용하여 Table 3과 같이 생성한다.

Table 3. Generated Test Scenario

No	TestScenario
TS1	A.DummyMain->A.Component2->A.Component5 ->B.Component1->A.Component5->finish
TS2	A.DummyMain->A.Component2->A.Component5 ->B.Component1->B.Component3->A.Component5 ->finish
TS3	A,DummyMain->A.Component3->A.Component5 ->B.Component1->A.Component5->finish
TS4	A,DummyMain->A.Component3->A.Component5 ->B.Component1->B.Component3->A.Component5 ->finish

4. 사례 연구

이 절에서는 제안한 상호작용 테스팅을 위한 인텐트 기반의 테스트 시나리오 생성 기법을 안드로이드 어플리케이션 중 안드로이드 플랫폼의 어플리케이션 중 Gallery 어플리케이션[11]과 카메라 어플리케이션[12]의 상호작용에 적용하여 효용성을 나타낸다. Fig. 5는 사례연구에 사용되는 안드로이드 어플리케이션인 Gallery와 Camera의 실행화면을 보여준다. 먼저 각 어플리케이션의 APK 파일에서 자바 바이트 코드와 manifest.xml 파일을 추출하여 각 어플리케이션의 컴포넌트 기반의 다이어그램과 인텐트 목록을 생성한다. 생성된 모든 컴포넌트 다이어그램과 인텐트 목록에서 Fig. 6과 같이 통합된 컴포넌트 기반의 다이어그램과 통합 인텐트 목록(Gallery : 16개, Camera : 3개)을 생성할 수 있다.

(A) Gallery (B) Camera

Fig. 5. Android Applications for Case Study

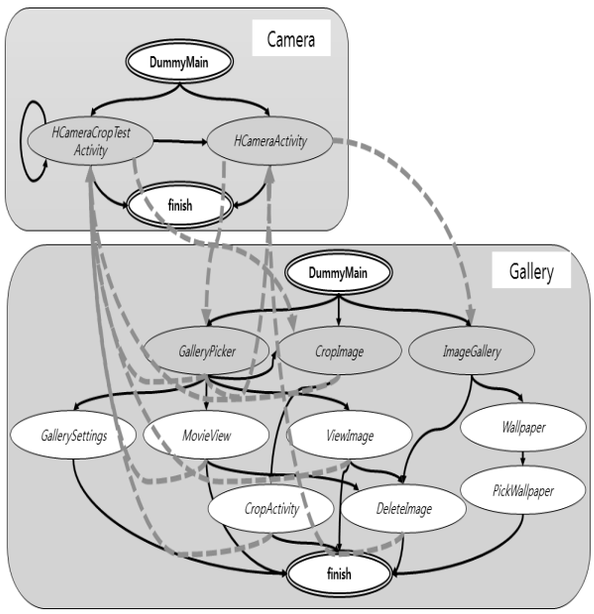

Fig. 6. Generated component diagram of Case Study

위와 같이 생성된 병합된 컴포넌트 기반 다이어그램에는 Gallery 어플리케이션의 GallerySettings와 같이 다른 어플리케이션과 상호작용하지 않는 부분이 존재한다. 상호작용만을 고려한 테스팅을 위하여 두 어플리케이션이 연결되는 상태만을 고려하여 Fig 7과 같이 추상화된 다이어그램을 생성한다. 추상화된 다이어그램으로부터 Table 4와 같이 Gallery와 Camera 어플리케이션의 상호작용 테스팅에 사용할 수 있는 14개의 테스트 시나리오를 생성한다.

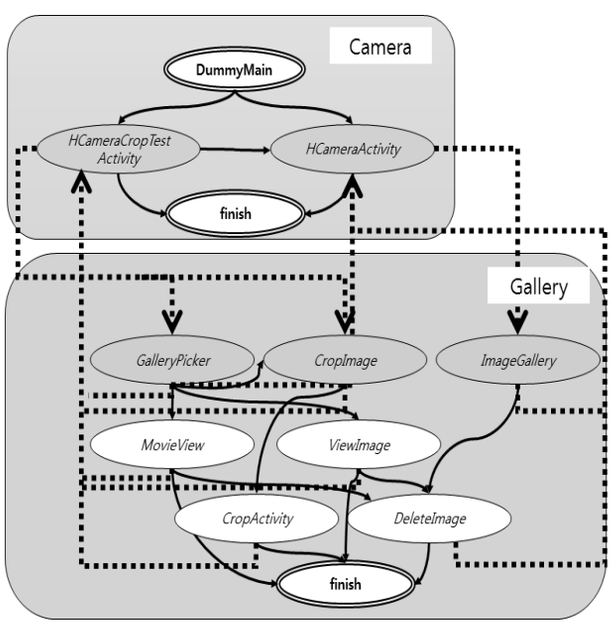

Fig. 7. Abstracted Combined Component Diagram of Case Study

Table 4. Test Scenario for Case Study

No	TestScenario
TS1	DummyMain->HCameraCropTestActivity->GalleryPicker->HCameraCropTestActivity->finish
TS2	DummyMain->HCameraCropTestActivity->GalleryPicker->MovieView->HCameraCropTestActivity->finish
TS3	DummyMain->HCameraCropTestActivity->GalleryPicker->MovieView->DeleteImage->HCameraCropTestActivity->finish
TS4	DummyMain->HCameraCropTestActivity->GalleryPicker->ViewImage->HCamearaCropTestActivity->finish
TS5	DummyMain->HCameraCropTestActivity->GalleryPicker->ImageView->DeleteImage->HCameraCropTestActivity->finish
TS6	DummyMain->HCameraCropTestActivity->CropImage->HCameraCropTestActivity->finish
TS7	DummyMain->HCameraCropTestActivity->CropImage->CropActivity->HCameraCropTestActivity->finish
TS8	DummyMain->HCameraActivity->GalleryPicker->HCameraActivity->finish
TS9	DummyMain->HCameraActivity->GalleryPicker->MovieView->HCameraActivity->finish
TS10	DummyMain->HCameraActivity->GalleryPicker->ViewImage->HCameraActivity->finish
TS11	DummyMain->HCameraActivity->GalleryPicker->MovieView->DeleteImage->HCameraActivity->finish
TS12	DummyMain->HCameraActivity->GalleryPicker->ViewImage->DeleteImage->HCameraActivity->finish
TS13	DummyMain->HCameraActivity->ImageGallery->HCameraActivity->finish
TS14	DummyMain->HCameraActivity->ImageGallery->DeleteImage->HCameraActivity->finish

5. 결 론

본 연구에서는 기존의 검증 기법에서 고려하지 못했던 어플리케이션간의 상호작용에 대한 테스팅을 하기 위해 기존의 테스팅 기법에서 고려하지 못한 상호작용하는 어플리케이션들에서 하나의 컴포넌트 기반 다이어그램을 생성하고 이로부터 상호작용 테스팅을 위한 테스트 시나리오를 생성하는 기법을 제안하였다. 그리고 사례연구를 통해 기존의 테스팅 방법에서 생성하지 못한 상호작용 테스팅을 위한 테스트 시나리오를 생성할 수 있음을 보였다. 향후 연구에서는 안드로이드 어플리케이션들간의 외부 컴포넌트들의 상호작용뿐 아니라 각 어플리케이션 내부 컴포넌트들의 상호작용을 함께 고려한 통합 테스트를 위한 테스트 시나리오 생성에 대한 연구가 필요하다.

References

[1] Smartphone OS Market Share, 2015 Q2 [Internet], http://www.idc.com/prodserv/smartphone-os-market-share.jsp.

[2] Number of available applications in the Google Play Store from December 2009 to July 2015 [Internet], http://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/.

[3] Application Component [Internet], http://developer.android.com/guide/components/fundamentals.html.

[4] JUnit [Internet], http://www.junit.org/taxonomy/term/6.

[5] Jung-Gyuew Lee, Hyeon-Soo Kim, Seung Hak Kuk, and Dae-Wan Cho, "Record-Playback based Automatic test case generation for GUI test," *Proceedings of the KIISE Korea Computer Congress 2007*, Vol.34, No.1, pp.96-100, 2007.

[6] Tae-San Baek and Woo Jin Lee, "A GUI Testing Technique based on Activity Lifecycle for Android Applications," *Journal of IEMEK*, Vol.8, No.6, pp.319-327, 2013.

[7] Tae-San Baek, Sung Hee Lee, and Woo Jin Lee, "Generation of Test Scenarios based on Intent List and Abstract Activity Diagram for Android Application," *Journal of KIISE Transactions on Computing Practices,* Vol.20, No.7, pp.386-391, 2014.

[8] Erika Chin, Adrienne Porter Felt, Kate Greenwood, and David Wagner, "Analyzing inter-application communication in Android," *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services. ACM*, pp. 239-252, 2011.

[9] Raja Vallee-Rai, Etienne Gagnon, Laurie J. Hendren, Patrick Lam, Patrice Pominville, and Vijay Sundaresan, "Optimizing Java Bytecode Using the Soot Framework: Is It Feasible?" *Proceedings of the 9th International Conference on Compiler Construction*, Springer, London, UK, pp.18-34, 2000.

[10] Ranjita Swainm Vikas Panthi, Prafulla Kumar Behera, and Durga Prasad Mohapatra, "Automatic test case generation from UML state chart diagram," *International Journal of Computer Applications,* Vol.42. No.7, pp.26-36, 2012.

[11] Android Gallery [Internet], http://developer.android.com/reference/android/widget/Gallery.html.

[12] Android Camera example [Internet], http://www.androidpub.com/2282206.

백 태 산

e-mail : sans79@gmail.com
2005년 위덕대학교 컴퓨터공학과(학사)
2008년 경북대학교 컴퓨터과학과 (이학석사)
2011년~현 재 경북대학교 컴퓨터학부 박사과정
관심분야 : 임베디드 소프트웨어 테스팅, 소프트웨어 품질관리, 테스트 시나리오 생성 자동화 등

이 우 진

e-mail : woojin@knu.ac.kr
1992년 경북대학교 컴퓨터과학과(학사)
1994년 KAIST 전산학과(공학석사)
1999년 KAIST 전산학과(공학박사)
1999년~2002년 ETRI 선임연구원
2002년~현 재 경북대학교 컴퓨터학부 교수
관심분야 : 임베디드 소프트웨어 테스팅, 임베디드 소트웨어 개발환경 등