

안드로이드 어플리케이션 개발에서 퍼미션 분석을 사용한 테스트 케이스의 다양한 테스트 환경 조건 생성 기법¹

송광식, 한아람, 정세훈, 차성덕
고려대학교 정보대학
서울 성북구 안암동 5가
{kwangsik_song, arhan, gifaranga, scha}@korea.ac.kr

요약: 최근 제조되는 스마트폰에는 다양한 주변장치가 집약되어 있어 어플리케이션 개발하는데 있어 개발 및 테스트 복잡도가 증가하였다. 따라서 사용자의 GUI 이벤트 외의 대상 어플리케이션과 상호작용하는 주변기기를 특정하여 이들로부터 구체적인 실행 환경 조합을 체계적으로 생성하는 연구가 필요하다. 본 연구에서는 안드로이드 어플리케이션과 함께 배포되는 퍼미션 (permission) 이라는 정보를 분석하여 어플리케이션의 기능에 영향 주는 필요한 외부 환경을 분석하여 실행 환경 조건을 자동으로 생성하는 방법을 제안한다.

핵심어: testing, permission, coverage, test precondition, Android.

1. 서론

최근 스마트폰 어플리케이션들 상황 인지 (context-aware) 어플리케이션 개발 환경에서는[1] 개발자들이 다양한 기능을 제공하는 어플리케이션을 제작할 수 있다는 장점이 있는 반면, 어플리케이션에 대하여 다양한 환경 및 스마트폰 하드웨어 구성의 조건에 대하여 테스트를 해야 하는 어려움이 있다[2].

GUI 이벤트 기반 테스트에 대한 연구는[3] 시스템 테스트와 자동화가 용이하지만, 환경 조건을 효과적으로 생성하지 못하며, 어플리케이션의 사용 형태 (user profiling) 를[4] 분석하여 환경 조건을 만드는 연구는 체계적으로 수행하기가 어렵고, 특히 고려해야 할 환경 조건이 많은 경우에는 수동으로 하기 어렵다는 단점이 있다. 따라서 각 어플리케이션에 영향을 줄 수 있는 환경 요인을 분석하여 다양한 실행 환경 조건을 자동으로 생성해 주는 기법이 필요하다. 본 연구에서는 안드로이드 어플리케이션의 퍼미션 (permission)의 분석 정보를 이용하여 체계적으로 다

양한 실행 환경 조건을 고려한 안드로이드 어플리케이션 테스트 방법을 제안한다. 여기에서의 퍼미션이란, 안드로이드 어플리케이션이 주변기기 사용과 같은 시스템의 기능을 사용하는 권한을 관리하는 정의된 선언을 나타낸다. 퍼미션 정보로부터 구체적인 실행 환경을 만들기 위해 각 퍼미션에 수록될 수 있는 주변기기 종류별로 기기들이 가질 수 있는 상태 후보 군을 미리 정의한 후, 이를 이용하여 테스트 대상 어플리케이션의 퍼미션 정보에 따라 각 주변기기들이 가질 수 있는 상태를 후보군에서 뽑아 조합하여 환경 조건을 생성한 후 상황 인지 테스트 케이스 집합을 생성한다. 실험에서는 오픈소스에 본 연구기법을 적용하여 코드 커버리지가 향상됨을 보임으로써 본 연구의 효과성을 검증하였다.

2. 관련 연구

사용자 입력과 더불어 외부 환경 이벤트를 고려하는 최근 연구로 D. Amalfitano 의 연구[5]를 들 수 있는데 모바일 어플리케이션의 다양한 기능을 테스트할 수 있는 외부 환경 이벤트 시나리오를 제안하고 이를 기존 GUI 이벤트 기반 테스트 케이스 자동 생성 도구[6]에 구현하였다. 이 논문에서는 결합 분석을 통해 생성한 이벤트를 소스의 인텐트 이벤트 핸들러 (intent event handler)를 분석하여 테스트 케이스에 추가해서 동작하게 하였다. 다양한 이벤트 처리하고 전문가가 수동으로 설정한 환경에서 수행하는 장점은 있으나, 코드 분석과정과 소스 코드가 필요한 단점이 있다.

3. 퍼미션을 이용한 환경 조건 생성 방법

¹ 본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학 IT 연구센터육성 지원사업/IT 융합고급인력과정지원사업의 연구결과로 수행되었음 (NIPA-2014-H0301-14-1023).

3.1 안드로이드 퍼미션 분석

안드로이드 퍼미션을 통해서 관련 주변기기의 종류와 각 주변기기가 어플리케이션에서 사용되는 상태 값을 분석하였다. 일례로, ACCESS_FINE_LOCATION 퍼미션은 관련 주변기기로 Wi-Fi, GPS, Radio 가 식별 되고, 각 장치의 상태 값을 ON, OFF, Fail 로 구별 될 수 있다.

3.2 환경 조건 조합 및 우선 순위화

앞서 분석한 퍼미션-주변기기 관계 및 각 주변기기가 가질 수 있는 상태 후보들을 이용하여 퍼미션 정보에 맞는 환경 조건 그룹을 생성한다. 예를 들어 주변기기로 Bluetooth, GPS, Wi-Fi, Radio 를 도출해 낸 경우에 상태 조건 가능성을 모두 조합하여 최대 54 가지 환경 조건 후보 군이 생성될 수 있다. 이러한 조합법은 구현과 자동화가 쉬우나 경우가 많으므로 우선 순위를 결정해야 하는 문제가 발생한다. 이때 우선 순위 분류의 계산 기준으로 각 주변기기 상태 후보 군의 세부 가중치가 사용될 수 있다. 정상적인 모든 설정을 확인하는 것이 비중이 높고, 그렇지 않은 것은 점수가 낮다. 정상인 설정을 고려한 검증을 먼저 수행하여, 기본 기능의 검증부터 수행하고 다양한 예외사항까지 순차적으로 실행이 가능하다.

4. 평가

4.1 실험 방법

테스팅 대상이 되는 안드로이드 어플리케이션으로 Open Camera[7]와 Angulo[8]를 선정하였다. Open Camera, Angulo 의 생성된 테스트 케이스 22, 43 개에 각각의 환경 조건 32 개와 4 개 중에 선택된 조건을 추가하여 합쳐진 테스트 케이스의 statement coverage 를 측정하여 기존 GUI Ripper 가 생성한 테스트 케이스의 statement coverage 와 비교하였다.

4.2 결과

GUI Ripper 는 Open Camera, Angulo 어플리케이션에 대해서 각각 22, 43 개의 테스트 케이스를 생성하였으며 테이블 1 와 같이 각각 32, 4 개의 실행 환경 조건을 생성하였다. Open Camera 에서 GUI Ripper 가 하나의 실행 환경을 가정하여 생성한 테스트 케이스는 37% statement coverage 를 보이는 반면, 하나의 실행 조건을 퍼미션을 기반으로 분석하여 32 개로 세분화하여 적용하여 coverage 를 38%로 향상시킬 수 있었다. 이와 같은 낮은 커버리지 개선은 기존 생성된 케이스가 coverage 가 낮아서 케이스를 보강이 필요하거나, 기능 코드의 부족을 의미한다. 또한

[테이블 1] 생성된 테스트 케이스의 수

어플리케이션	실행 환경의 경우의 수			GUI Ripper 테스트 케이스
	퍼미션	상태 후보	생성한 환경 조건의 수	
Open camera	외장 저장장치	2	32 개 (=2X8X2)	22 개
	세부 위치정보	8		
	카메라	2		
Angulo	진동	2	4 개 (=2X2)	43 개
	카메라	2		

Angulo 에 대해서 GUI Ripper 가 생성한 테스트 케이스에 퍼미션 분석을 통한 테스트 케이스를 추가하여, statement coverage 58%를 70%로 향상시킬 수 있었다. 그리하여 각각 어플리케이션의 Camera 관련 수행이 안되었던 코드들이 수행되었다.

5. 결론 및 향후 연구 계획

본 연구에서는 퍼미션 정보를 활용하여 어플리케이션 테스트에 활용될 수 있는 환경 조건을 생성하고 기존 GUI 이벤트 기반 테스트 케이스와 조합하여 환경 변화를 고려한 안드로이드 어플리케이션 테스트 케이스를 작성할 수 있었다. 제안한 기법의 효과성을 보이기 위해 오픈 소스 안드로이드 어플리케이션 2 종에 적용하여 코드 커버리지가 기존 GUI 이벤트만 고려한 테스트 케이스보다 향상되었다. 향후 연구로는 테스트 케이스를 생성해내기 위한 퍼미션들의 정의를 확장하고 개발자를 위한 자동화 분석 도구 개발을 계획 중이다.

참고문헌

- [1] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith and P. Steggle. "Towards a Better Understanding of Context and Context-Awareness." the 1st international symposium on Handheld and Ubiquitous Computing (HUC '99), Springer-Verlag, 1999, pp. 304-307
- [2] A. K. Maji, K. Hao, S. Sultana, S. Bagchi, Characterizing Failures in Mobile OSes: A Case Study with Android and Symbian. Proceedings of the 21st IEEE Int. Symposium on Software Reliability Engineering, 2010, IEEE CS Press, pp.249-258.
- [3] Monkey. DOI= <http://developer.android.com/tools/help/monkey.html>
- [4] Wenliang Du, Aditya P. Mathur, "Vulnerability Testing of Software System Using Fault Injection", Purdue University, 1998
- [5] Domenico Amalfitano, Anna Rita Fasolino, Porfirio Tramontana, and Nicola Amatucci, "Considering Context Events in Event-Based Testing of Mobile Applications" IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops, 2013.
- [6] D. Amalfitano, A.R. Fasolino, P. Tramontana, S. DeCarmine, G. Imparato. "A Toolset for GUI testing of Android Applications" Proceedings of the 28th IEEE International Conference on Software Maintenance (ICSM), 2012, IEEE CS Press, pp.650-653.
- [7] Open Camera. DOI = <http://opencamera.sourceforge.net/>
- [8] Angulo , DOI = <http://www.domob.eu/projects/angulo.php>