센서 활용 어플리케이션을 위한 동적 테스트 도구의 요구사항

최유림, 백두산, 이정원
아주대학교 전자공학과
e-mail: cyl5780@ajou.ac.kr, whitedusan@gmail.com, jungwony@ajou.ac.kr

Requirements of Dynamic Testing Tool for Applications using Sensor

Yoo-Rim Choi, Du-san Baek, Jung-Won Lee
Dept of Electronic Engineering, Ajou University

요 약

모바일 어플리케이션 개발자들은 센서를 활용하여 사용자를 위한 맞춤화 서비스를 제공하고 있다. 센서를 활용한 어플리케이션은 다른 소프트웨어와 마찬가지로 더욱 높은 품질을 위해 테스트를 필요로 한다. 그러나 기존의 모바일 어플리케이션을 위한 테스트 도구들은 센서에 대한 접근성, 센서 데이터 입력, 전력 소모에 대한 비고려 등의 문제를 갖고 있어, 동적 테스트를 수행하는데 사용하기에는 한계를 지니고 있다. 물론, 이러한 문제를 해결하기 위해 여러 연구들이 진행되어 왔지만 기존의 연구들은 일부의 특성만을 반영한 테스트 도구를 개발하였거나, 기법만 개발하였다. 따라서 본 논문에서는 이러한 특성들을 통합적으로 반영한 동적 테스트 도구를 개발하기 위해 센서를 활용한 어플리케이션의 특성을 도출하고, 이를 이용하여 기존의 동적 테스트 도구들의 한계점을 분석하였다. 그 후, 이를 기반으로 하여 동적 테스트 도구의 요구사항을 도출하였다. 그 결과, 총 3개의 특성과 5개의 한계점 그리고 6개의 요구사항을 도출하였다. 향후 본 논문은 센서를 활용한 어플리케이션을 위한 동적 테스트 도구의 개발에 기초가 될 것이다.

1. 서론

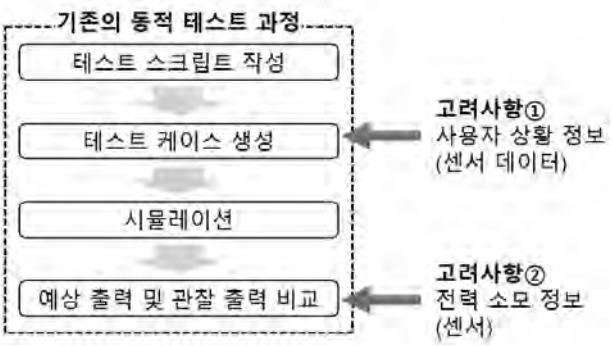
소프트웨어의 품질을 보장하기 위해 개발하는 과정 동안 테스트가 진행된다. 테스트를 진행하는 것은 추가적인 개발 시간과 노력을 필요로 하며, 이를 절약하기 위해서 다양한 테스트 도구들이 연구, 개발, 사용되었다. 하지만, 기존의 테스트 도구들은 PC 어플리케이션에 초점을 두고 있어, 모바일 플랫폼을 대상으로는 테스트를 수행할 수 없거나, 수행하여도 정확한 테스트 결과를 도출하지 못하는 한계를 지니고 있다. 물론, 이러한 도메인의 차이에서 오는 이질적인 문제를 극복하기 위한 연구들도 진행되고 있지만, 그럼에도 불구하고 센서를 활용한 어플리케이션을 대상으로 테스트를 수행할 수 있는 도구들은 부족한 현실이다.

센서를 활용한 어플리케이션은 센서를 통해 정보를 수집 및 분석하고 상황을 도출하여 이에 적합한 서비스를 제공하는 어플리케이션이다. 이러한 어플리케이션이 사용자에게 적합한 서비스를 제공하는지 검증하기 위해서는 동적 테스트를 수행해야 한다. 동적 테스트란, 프로그램을 실행하면서 결함을 찾는 테스트이다. 그러나 기존의 모바일 어플리케이션의 동적 테스트를 위한 도구들은 센서를 활용한 어플리케이션의 특성을 반영하지 못하기 때문에 이를 동적 테스트하기에 다음과 같은 미흡한 면이 있다.

첫째, 센서에서 측정되는 데이터에 대한 고려가 부족하다. 센서에서 측정되는 데이터는 GUI기반의 사용자 이벤트와 더불어 어플리케이션의 동작에 영향을 미친다. 따라서 일부 센서에 대한 접근성만을 가진 기존의 동적 테스트 도구들은 접근 권한이 없는 센서에서 측정되는 데이터에 대한 검증이 어렵다. 또한, 센서에서 측정되는 데이터를 고려하지 않은 테스트 케이스는 고정적인 시나리오만 반영하여 모든 상황에 대한 커버리지를 만족하기 힘들다. 둘째, 센서를 활용한 어플리케이션에서 소비되는 전력량에 대해 고려가 부족하다. 배터리와 같이 한정적인 전력을 가지는 모바일 플랫폼에서는 더욱 효율적인 전력 사용이 필요하다. 따라서 테스트 케이스의 예상 출력(expected output)과 관찰된 출력(observed output)을 분석할 때 기능적 요소뿐만 아니라 비기능적 요소인 전력에 대한 고려도 필요하다. 그림 1은 위에서 언급한 특성을 반영한 동적 테스트 과정을 도식화한 그림이다. 그림 1과 같이 센서를 활용한 어플리케이션의 동적 테스트는 기존의 동적 테스트 과정에 센서를 활용한

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(No. NRF-2014M3C4A7030504).

어플리케이션의 특성인 사용자 상황 정보와 전력 소모 정보를 반영해야 한다.



(그림 1) 특성을 반영한 테스트 과정

이에 본 연구에서는 센서를 활용한 어플리케이션의 특성을 분석하고, 이를 이용하여 기존의 동적 테스트 도구들이 센서를 활용한 어플리케이션에 적합하지 않음을 지적한다. 그 후, 특성과 한계점을 토대로 요구사항을 도출한다.

2. 관련 연구

2.1 테스트 입력으로서 센서 데이터

센서를 활용한 어플리케이션에서는 GUI기반의 사용자 이벤트뿐만 아니라, 센서에서 측정되는 데이터 또한 어플리케이션의 동작에 큰 영향을 미친다. 그러므로 어플리케이션의 모든 상황을 검증하기 위해서는 센서 데이터를 입력 할 수 있어야한다. 하지만 기존의 동적 테스트 도구들은 센서의 데이터를 입력하지 못하거나, 위치와 관련된 정보만 입력 할 수 있었다[1]. 이러한 한계점을 개선하기 위해 동적 테스트의 입력으로 가상의 데이터를 실측한 데이터처럼 입력하는 연구들이 진행되고 있다. 그 중 하나로 테스터가 어플리케이션의 외부에서 센서의 측정값을 입력할 수 있도록 하는 시스템에 관한 연구가 있다[2]. 입력에 관한 다른 연구로는 센서에 관련된 시스템 이벤트를 발생시킴으로써 간접적으로 센서 데이터를 입력하는 센서 에뮬레이터에 관한 연구가 있다[3].

2.2 상황정보 기반 테스트 케이스 생성

센서를 활용한 어플리케이션에서 센서를 이용하여 얻은 정보들은 언제라도 어플리케이션의 동작에 변화를 줄 수 있기 때문에 모든 상황에 대해 높은 커버리지로 검증하기 어렵다. 따라서 어플리케이션의 상황 정보를 이용하여 테스트 케이스를 생성해야 한다. 이와 관련된 연구로는 코드의 호출 구조를 통해 상황 정보를 도출하여 테스트 케이스를 생성하는 연구[4]와 사용자가 입력한 상황 정보를 이용해 테스트 케이스를 생성하는 연구가 있다[5].

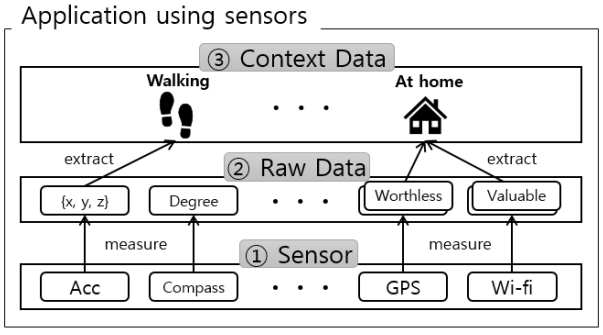
2.3 전력 소모 테스팅

모바일 기기는 한정적인 전력을 가지고 있기 때문에 전력을 효율적으로 사용해야 한다. 효율적인 전력 사용을 위해서는 정확한 전력 측정이 수반되어야 한다. 그러나 센서를 활용한 어플리케이션은 사용패턴에 따라 전력 소모량의 큰 차이를 보이기 때문에 전력 소모를 예측하기 어렵다. 이를 보완하기 위한 연구 중 하나는 PowerForecaster로, 비슷한 어플리케이션의 사용 패턴을 이용하여 전력 소모량을 예측한다[3]. 또 다른 연구로는 센서를 이용하여 데이터를 얻어내는 과정에 대한 기회비용을 다루는 연구가 있다. 센서에서 사용한 전력 대비 측정한 데이터의 사용도가 높지 않으면 이는 전력 낭비라고 할 수 있다. 해당 연구는 이를 분석하기 위해 taint 기법을 적용하여 센서 데이터가 얼마나 활용되고 있는지를 검증한다[6].

기존의 연구들은 일부의 특성만을 반영한 동적 테스트 도구를 개발하였거나, 기법만을 개발하였다. 센서를 활용한 어플리케이션의 모든 특성을 반영하기 위해 여러 종류의 테스트 도구들을 사용한다면, 동일한 시나리오를 테스트 도구 마다 반복해야 되기 때문에 테스트의 효율이 낮은 한계를 지니고 있다. 따라서 센서를 활용한 어플리케이션의 실효적인 동적 테스트를 위해서는 통합적인 테스트 도구 개발이 필요하다.

3. 센서를 활용한 어플리케이션의 특성 및 도구 분석

본 장에서는 관련 연구와 기존의 테스트 도구를 분석하여 센서를 활용한 어플리케이션의 특성을 추출하고, 이로 인해 발생하는 기존 동적 테스트 도구의 한계점을 분석하였다. 그 결과는 아래와 같으며, 그림 2는 추출한 특성 중 일부를 도식화 한 그림이다.



(그림 2) 센서를 활용한 어플리케이션의 특성

- 다양한 센서에서 측정한 데이터 : 그림2 에서 보이는 바와 같이 센서를 활용한 어플리케이션이 운용되는 모바일 플랫폼은 다양한 기기와 기기에 내장되는 센서(①)로 구성된다. 센서를 활용하는 어플리케이션은 내장되는 센서에서 측정되는 낮은 수준의 데이터(②)를 가공하여 사용자의 정보(③)를 추출한다. 이러한

과정을 통해 추출한 데이터와 GUI기반의 사용자 이벤트는 어플리케이션을 동작시키는 주요 요소이다. 이 특성을 반영하지 못해 생기는 첫 번째 한계점으로는 센서에 대한 접근성이다. 센서에서 측정되는 데이터를 검증하기 위해 센서에 대한 접근성이 필요하다. 하지만 기존의 동적 테스트 도구들은 접근성을 가지지 못하거나, 일부 센서에 대한 제한된 접근성을 가진다. 두 번째 한계점은 테스트 입력이다. 센서에서 측정한 데이터는 어플리케이션의 동작에 영향을 미치기 때문에 시뮬레이션에 대한 입력으로 포함되어야 한다. 하지만, 기존의 동적 테스트 도구들은 입력이 불가능하거나, 일부 센서에 대한 입력만 지원한다. 세 번째 한계점은 상황 정보를 반영하지 못한 테스트 케이스이다. 센서로 얻은 상황에 대한 정보는 언제든지 어플리케이션의 동작에 변화를 줄 수 있기 때문에 센서를 활용하는 어플리케이션의 커버리지를 검증하기 어렵다. 기존의 동적 테스트 도구들은 테스트 케이스를 생성할 때, 상황에 대한 정보를 고려하지 않는다.

- 전력 소모 : PC 플랫폼과는 달리 배터리와 같은 한정적인 전력을 가진 모바일 플랫폼에서는 전력 소모가 매우 중요한 문제로 다뤄지고 있다. 그러므로 모바일 기기의 전력은 효율적으로 관리되어야 한다. 그러나 센서를 활용한 어플리케이션은 센서를 이용하여 데이터를 얻어내기 때문에 기존의 모바일 어플리케이션보다 더 많은 전력을 소모한다. 이러한 전력 소모를 줄이기 위해 먼저 정확한 전력 소모량을 측정 할 수 있어야 한다. 하지만, 기존의 동적 테스트 도구는 전력 소모에 대한 분석을 지원하지 않거나, 이를 지원하여도 어플리케이션의 사용패턴을 반영하여 분석하지 않아 정확한 전력 소모량을 분석할 수 없다.
- 모바일 기기의 다양성 : 센서를 활용한 어플리케이션이 동작하는 모바일 플랫폼 시장에는 다양한 기기들이 계속해서 출시되고 있다. 또한, 기기에 탑재되는 OS도 짧은 주기로 업데이트 되고 있다. 기존의 동적 테스트 도구들은 일부 기기와 OS에 종속되어 일부 기기만 테스트 가능하다. 따라서 새로 출시된 기기나 OS에 대한 테스트가 어렵다는 한계점이 있다.

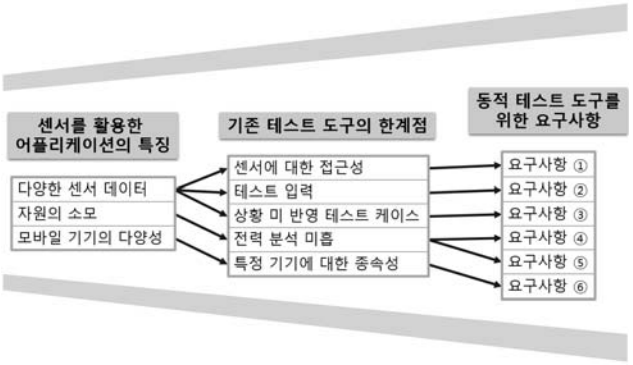
표 1은 실제로 유명한 모바일기기의 동적 테스트 도구들이 어떤 한계점을 내포하고 있는지 비교한 표이다.

<표 1> 기존 동적 테스트 도구의 한계점 비교 분석

	Appium[7]	MonkeyRunner [8]	Robotium[9]
센서 데이터 입력	일부분	X	일부분
센서 접근성	일부분	X	일부분
상황 정보를 반영한 테스트 케이스 생성	X	X	X
자원 소모 분석	X	X	X
모바일 기기 종속성	O	X	O

4, 센서를 활용한 어플리케이션을 위한 동적 테스트 도구의 요구사항

본 장에서는 3장에서 도출한 센서를 활용하는 어플리케이션의 특성과 분석한 기존 동적 테스트 도구의 한계점을 토대로 센서를 활용한 어플리케이션을 위한 동적 테스트 도구의 요구사항을 도출하였다. 그림 3은 특징 및 한계점과 요구사항의 관계를 나타내는 그림이다. 요구사항은 총 6가지가 도출되었으며, 각 요구사항은 표 2와 같다.



(그림 3) 특징, 한계점, 요구사항의 관계

<표 2> 테스트 도구의 요구사항

센서를 활용한 어플리케이션의 동적 테스트 도구를 위한 요구사항	
요구사항 ①	모든 센서에 대한 접근권한을 가져야 한다.
요구사항 ②	가상의 센서 데이터를 실측 데이터처럼 입력할 수 있어야 한다.
요구사항 ③	상황 정보를 기반으로 하여 테스트 케이스 생성이 가능해야 한다.
요구사항 ④	어플리케이션의 사용 패턴을 고려하여 자원 소모를 분석할 수 있어야 한다.
요구사항 ⑤	센서로 측정한 데이터가 사용되고 있는지 테스트 할 수 있어야 한다.
요구사항 ⑥	특정 모바일 기기나 OS에 종속되지 않아야 한다.

① 센서를 활용한 어플리케이션은 센서에서 측정한 데이터에 대한 의존성이 강하다. 따라서 센서가 올바르게 데이터를 측정하고 있는지 확인할 수 있어야 하므로 모든 센서에 대한 접근 권한이 필요하다.

② 센서를 활용한 어플리케이션의 동작에 영향을 미치는 요소로 GUI기반의 사용자 이벤트뿐만 아니라 센서에서 측정한 데이터 또한 해당된다. 따라서 센서를 활용한 어플리케이션의 동적 테스트를 위해서는 테스터가 원하는 가상 데이터를 실측 데이터처럼 입력할 수 있어야한다. 이와 관련하여 진행되는 여러 가지 연구들 중 하나로 가상 센서 시스템이 있다[2]. 이 시스템은 테스터가 원하는 가상 데이터를 실측데이터처럼 넣을 수 있다는 장점이 있다. 그러나 안드로이드의 센서 플랫폼을 수정해야 한다는 단점이 있다. 또 다른 연구로는 안드로이드에 센서와 관련된 시스템 이벤트를 발생시키는 센서 에뮬레이터가 있다[3]. 센서 데이터의 입력을 위해 따로 어플리케이션의 수정이

필요하지 않다는 장점이 있지만, 센서 이벤트만을 입력으로 받을 수 있기 때문에 상세한 상황설정은 불가능하다. 또한 센서 데이터의 입력 범위에서 생길 수 있는 결함은 발견이 불가능하다.

③ 센서를 활용한 어플리케이션에서 센서를 이용하여 얻은 정보들은 언제라도 어플리케이션의 동작에 변화를 줄 수 있기 때문에 모든 상황에 대한 높은 커버리지를 검증하기 어렵다. 따라서 테스트 케이스 생성 시, 어플리케이션에서 도출 될 수 있는 상황 정보에 대한 고려가 필요하다. 이와 관련된 여러 연구들 중 하나는 어플리케이션 코드에서의 호출 구조를 통해 상황 정보를 파악하여 테스트 케이스를 생성한다[4]. 이 연구는 실제로 context coverage를 만족하기 위한 테스트 실행 시간을 3배 단축하였다. 하지만, 비슷한 상황 정보 (예-멈춤, 걷기, 뛰기)를 활용하는 어플리케이션에 한해서만 동작한다는 단점이 있다. 또 다른 연구로는 사용자가 입력한 상황 정보를 이용하여 테스트 케이스를 생성하는 연구가 있다[5]. 테스터가 1차적으로 상황 정보를 사용하여 테스트 케이스를 직접 생성하면 이를 이용하여 mutation 기법을 통해 2차 테스트 케이스를 생성한다. 실제로 이 연구는 LOC code coverage를 5~15%정도 상승시켰다. 하지만, 테스터가 수동으로 1차 테스트 케이스를 생성해야 한다는 단점이 있다.

④ 센서를 활용한 어플리케이션은 다양한 센서 사용으로 인해 전력을 과도하게 사용할 수 있다. PC 플랫폼과는 달리 배터리와 같은 한정적인 전력을 가진 모바일 플랫폼에서는 전력 소모가 매우 중요한 문제로 다뤄지고 있다. 따라서 과도한 전력 소모를 방지하고, 효율적으로 전력을 관리하기 위해서는 먼저 정확한 전력 소모량을 측정 할 수 있어야 한다. 그러나 센서를 활용한 어플리케이션은 이를 사용하는 패턴에 따라 동작과 전력 소모량이 달라지기 때문에 사용 패턴을 고려하지 않는 전력 소모 분석은 정확도가 떨어질 수 있다. 그러므로 정확한 전력 소모 분석을 위해 사용 패턴을 고려하여 전력 소모를 측정해야 한다. 이를 위한 연구로 PowerForecaster가 있다[3]. 이 연구는 비슷한 어플리케이션의 사용 패턴 정보를 이용하여 사용자에게 특화된 전력 소모량을 예측한다. 그러나 어플리케이션을 테스트하기 위해서는 비슷한 어플리케이션을 장기간 사용한 정보가 있어야 한다는 단점이 있다.

⑤ 적은 비용을 드려 수집한 센서 데이터 일지라도, 이를 사용하지 않는다면 전력 낭비라고 할 수 있다. 따라서, 동적 테스트 도구는 센서 데이터의 사용률의 분석을 지원해야 하며, 개발자는 이를 활용해서 사용률이 낮은 데이터의 수집을 유예하거나 차단하여 전력 효율을 도모해야 할 것이다. 이와 관련된 연구로는 taint 기법을 적용하여 센서 데이터가 얼마나 활용되고 있는지를 검증하는 GreenDroid가 있다[6]. 하지만 이 도구는 안드로이드의 버전마다 어플리케이션 실행 모델을 생성해야 한다는 단점이 있다.

⑥ 현재 모바일 플랫폼 시장은 다양한 모바일 기기들이 계속해서 출시되고 있다. 또한, 모바일 기기에 탑재되는 OS의 경우에도 업데이트 주기가 매우 짧은 특성을 가지고 있다. 특정 모바일 기기나 OS에 종속된 테스트 도구는 새로 출시된 기기나 OS에 대한 테스트가 어렵다는 한계점이 있다. 따라서 테스트 도구는 임의의 모바일 기기 혹은 OS에 종속되지 않도록 확장 가능성을 염두하며 개발되어야 할 것이다.

5. 결론

기존의 모바일 어플리케이션을 위한 동적 테스트 도구들은 센서를 활용한 어플리케이션의 특성을 반영하지 못하여 센서를 활용한 어플리케이션의 동적 테스트에 사용하기에 한계를 갖고 있었다. 이를 보완하기 위해 본 논문에서는 센서를 활용한 어플리케이션의 특성을 분석하고 이 특성을 반영하는 동적 테스트 도구의 요구사항을 도출하였다. 그 결과, 총 3개의 특성과 5개의 한계점 그리고 6개의 요구사항을 도출하였다. 본 논문에서 도출한 요구사항은 센서를 활용한 어플리케이션의 동적 테스트를 돕는 테스트 도구의 개발에 기초가 될 것이다. 향후 본 논문의 요구사항을 반영하여 센서를 활용한 어플리케이션을 위한 통합적인 테스트 도구를 개발할 것이다.

참고문헌

[1] DDMS, https://developer.android.com/studio/profile/ddms.html
[2] Jo, Jang-Wu, and Hwan-Cheol Joeng. "An Effective Method of Testing Application Software of Smart Sensors." Journal of the Korea Society of Computer and Information 18.8 (2013): 105-111
[3] Min, Chulhong, et al. "Powerforecaster: Predicting smartphone power impact of continuous sensing applications at pre-installation time." Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems. ACM, 2015.
[4] Wang, Zhimin, Sebastian Elbaum, and David S. Rosenblum. "Automated generation of context-aware tests." Proceedings of the 29th international conference on Software Engineering. IEEE Computer Society, 2007.
[5]Amalfitano, Domenico, et al. "Considering context events in event-based testing of mobile applications." Software Testing, Verification and Validation Workshops (ICSTW), 2013 IEEE Sixth International Conference on. IEEE, 2013.
[6] Liu, Yepang, Chang Xu, and Shing-Chi Cheung. "Where has my battery gone? Finding sensor related energy black holes in smartphone applications." Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on. IEEE, 2013.
[7] Appium, http://appium.io/
[8] Monkeyrunner, https://developer.android.com/studio/test/monkeyrunner/index.html
[9] Robotium, https://en.wikipedia.org/wiki/Robotium