

# 모바일 앱 사용행태 분석을 위한 역컴파일 및 역어셈블 데이터 로깅

김명준\* · 남양희\*\*

## Disassembly and De-Compilation Based Data Logging for Mobile App Usage Analysis

Myoung-Jun Kim\* · Yanghee Nam\*\*

### Abstract

This study presents a logging method to trace the usage patterns of existing smartphone apps. The actual smartphone app itself, not a specially developed similar app with usage logging, would be used best for the experiment of observing the usage patterns. For this purpose, we used a method of injecting logging codes into existing smartphone app. Using this method, we conducted an experiment to trace usage patterns of a commercial IPTV app, and found that the method is very useful for acquiring detail usage log without influencing participants.

Keywords : Mobile App Usage Analysis, In-App Data Logging, Decompile, Disassemble

---

논문접수일 : 2014년 10월 21일      논문게재확정일 : 2014년 11월 05일

※ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음(10041891).

\* 교신저자, 이화여자대학교 디지털미디어학부 부교수, e-mail : mjkim@ewha.ac.kr

\*\* 이화여자대학교 디지털미디어학부 부교수, e-mail : yanghee@ewha.ac.kr

## 1. 서 론

사용자 모델링 또는 사용행태 분석은 서비스 기획, 디자인 및 마케팅 등 비즈니스 전반에 걸쳐 고객의 요구 및 경험을 파악하는 데 필요하며, 적절한 계획 및 디자인을 수립하고 실행의 세부 내용을 조정하는 데 매우 중요한 근거가 된다[Brodie 외 3인, 2011; Cao, 2010].

개인이나 조직의 현재 혹은 과거 행동 자료를 수집하는 기존의 주류 방법론은 크게 두 가지로 분류할 수 있다. 첫 번째는 고객 설문 등 주관적인 의견과 경험 수집에 의존하는 방법이다. 이 방법은 다양하고 구체적인 사례 파악을 가능하게 하지만, 인간 기억의 불완전성과 주관적 상황의 개입 등으로 객관적 데이터를 얻기 힘들다. 두 번째는 판매량의 추이 등 시장 현황과 사용 실태의 전반적인 동향을 유형별로 조사하여 산업적, 사회적, 인구 통계학적 데이터를 얻고 변인 간 관계를 조사 분석하는 방법이다. 이 경우 객관적인 분석을 가능케 하고 새로운 서비스나 유형에 따른 전반적 경향은 알 수 있으나, 고객들의 구체적인 경험이나 실제 사용행태에 드러난 문제점 등을 상세히 관찰하고 분석하기 어렵다. 이와 같이 전통적 사용행태 조사 방법들은 경험 증거나 대략적 관찰 증거에 기반한 통계에 의존하므로 데이터의 구체성, 관찰 내용의 한계를 벗어날 수 없다.

그러나 디지털 플랫폼이 많은 비즈니스와 서비스의 근간을 이루게 되면서 객관적이고 상세한 사용 행태를 기록하여 분석하는 데이터 로깅 기반 자료 수집이 가능하게 되었다[Robinson et al., 2008]. 여러 디지털 플랫폼 가운데 특히 스마트폰의 경우, 그 앱(app)들은 일반 사용자의 일상과 밀착된 사용패턴을 보인다. 따라서 별도로 분리된 실험 환경에서의 관찰이나 설문보다는 실생활의 맥락에서 자연스러운 사용 행태를

획득하는 것이 자료의 신뢰성을 높인다.

그러나 데이터 로깅을 수행하는 기존 연구들은 데이터 기록 코드를 삽입한 시험용 서비스 앱을 별도 제작하여 이를 통해 사용자 테스트 데이터를 기록하는 것이 일반적이다. 이 경우 테스트 앱과 실제 앱이 다를 수 있고, 테스트나 비교를 수행하려는 대상물이 자체 제작한 콘텐츠나 서비스가 아닌 경우 데이터 기록 자체가 어려워므로 사실상 로그 데이터 수집이나 분석을 수행할 수 없다는 문제가 있다.

본 연구는 피험자에게 테스트용의 별도 앱이 아닌 실제 서비스되고 있는 스마트폰 앱을 그대로 사용하게 함으로써, 일상적 사용의 맥락에서 데이터 로깅을 수행할 수 있게 하는 새로운 행태 자료 수집 방법을 제시한다. 즉, 관찰 대상 앱의 기능과 인터페이스(UI)를 완전히 동일하게 유지한 채 앱 내에 탐침용 기록 코드를 삽입하는 사용 패턴 기록법의 구현 체계를 보임으로써 객관적이고 상세하며 맥락적인 데이터 수집 방안을 설명한다. 끝으로, 제안된 방법을 스마트폰 TV 시청 앱인 티빙(TVing) 앱에 적용한 실험을 통해, 사용자 사용행태 데이터 로깅 결과와 이러한 데이터 획득의 의의를 확인할 수 있게 하였다. 관찰 대상의 스마트폰 앱은 안드로이드 운영체제로 한정하였다. 이는 마켓을 통하지 않고서도 사용자가 직접 설치 할 수 있기 때문이다.

## 2. 애플리케이션 사용행태 데이터 로깅 연구들

본 절에서는 여러 디지털 플랫폼에서의 데이터 기반 온라인 사용자 행태 로깅에 관한 선행 연구들을 검토한다.

데이터 로깅은 애플리케이션이나 웹과 같은 서비스를 제작하여 배포한 조직이나 기관에서 직접 사용자의 행위를 기록하는 경우와, 제 3의

기관이 사용행태 조사 목적으로 여러 서비스 사용 행태들을 기록하는 경우로 구분해볼 수 있다.

제품이나 서비스의 소유 기관이 직접 사용자 모니터링을 하는 경우는 제작되는 응용 제품 내에 로그 기록 기능을 상품 개발 단계에서 삽입할 수 있으므로 데이터 수집 및 기록에 기술적인 어려움이 거의 없다. 예컨대, 진종근 등[2006]은 웹사이트 방문 시 클릭을 통한 사용자 브라우징 액티비티가 기록된 클릭스트림(clickstream) 데이터를 활용하였는데, 이는 해당 사이트 서버에 남는 기록이므로 운영 기관에서는 쉽게 접근, 조사 가능한 데이터이다. 웹의 경우는 이와 같이 로그가 자동으로 서버에 남겨지므로 데이터 수집을 위한 별도의 기법이 거의 필요치 않다 [Asenjo, 2011; Chen et al., 2014]. 일반적인 웹사이트 외에 소셜네트워크 서비스인 페이스북도 최근 사용자 뉴스피드를 조작해 감정이나 정서의 감염 여부에 관한 심리 실험을 진행한 바 있다. 실험에 대한 동의를 얻지 않은 채 수십만 명의 데이터(뉴스피드) 조작 및 감성 데이터 수집을 수행하였다는 점에서 윤리적 문제가 제기되었으나, 이 역시 서비스 소유 기관의 자유로운 데이터 조작 및 로깅이 용이함을 보여주는 대표 사례이다[Kramer, 2014].

이에 반해, 사용행태 조사를 하려는 기관이나 조직이 자체 제작한 것이 아닌 앱이나 서비스를 대상으로 사용 데이터를 얻고자 하는 경우는 데이터의 획득이 쉽지 않다.

이 경우, 가장 쉽고 빈번하게 사용되는 방식은 자가 보고(self-reporting) 방법이다. 이는 사용자가 사용 행태 기록에 동의하고 스스로 일정 간격으로 데이터 로깅이나 다이어리 작성 등으로 실험에 참여하는 방식을 지칭한다. 그러나 Moller 등[2012]에 따르면 자가보고 피험자의 경우 요구되어지는 로깅 빈도 대비 70% 이상을 기록하는 경우가 거의 없고, 실제 앱 사용

의 40% 미만을 기록하는 경향이 파악되었다. 기록 빈도가 높을 때 불편감을 호소하는 것은 물론이고 심지어 그로 인해 사용 행태가 변화되기도 하므로 결과적으로 잘못된 행태 측정 결과로 유도될 수 있다.

이러한 단점 때문에 최근에는 기술적 난이도는 높으나 사용자 개입이나 방해(distraction)를 최소화하는 방식으로 데이터 로깅을 시도하는 연구들이 많다. 예컨대, Robinson 등[2008]은 개인들의 웹 브라우징 액티비티 기록을 통해 사용자 관심을 파악하고 범주화하기 위해 간헐적으로 네트워크 트래픽 데이터를 건져내는(sniff) 방식을 사용했다. 그런데 이런 방식은 브라우징 이력이 가장 중요한 사용자 액티비티 정보인 웹 서비스에 주호한 방법이고, 스마트폰 등 모바일의 앱 사용은 그 맥락이나 행태의 복잡도가 훨씬 높고 다양하다.

스마트폰 앱의 사용 패턴의 로그를 기록하는 방법은 다음과 같다. 먼저 관찰 대상이 되는 앱과 유사한 앱을 만들어 피험자로 하여금 사용하게 하는 방법이 있다[Falaki et al., 2010; Do et al., 2011]. 이는 새롭게 앱을 제작하는 것이므로 원하는 다양한 사용 정보를 자유롭게 로그로 기록할 수 있는 장점이 있다. 하지만, 사용 패턴 연구가 일반적으로 이미 존재하는 앱에 대한 것이므로 비슷한 다른 앱을 만들어서 관찰하는 것은 그 관찰 결과의 신뢰성을 떨어뜨린다. 또 다른 방법은 관찰대상이 되는 앱을 그대로 사용하게 하고 스마트폰에 별도의 관찰앱(custom logging tool)을 설치하여 사용 패턴을 기록하는 방법이다. 이 경우 피험자가 실험 중이라는 의식 없이 일상적 맥락에서 진행할 수 있는 것이 커다란 장점이다. 반면 자체 제작하지 않은 기존의 앱으로부터 얻어낼 수 있는 정보가 제한적이어서 사용 패턴의 상세한 기록이 어렵다. 특히 특정 앱 내에서 발생하는 고유한 내부 기능

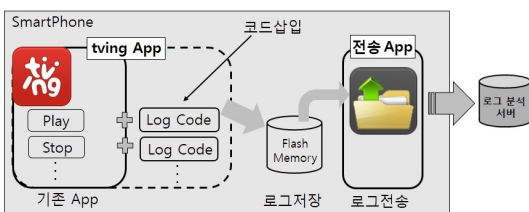
들의 사용 행태는 기록하기 어렵다.

이러한 기존 연구들의 맥락과 한계 분석을 통해 알 수 있는 것은, 사용 행태의 복잡도가 높은 스마트폰의 앱 내부적 기능에 대한 사용 패턴 분석은 기존 앱의 내부로 탐침을 넣어 관찰하듯 정보를 획득하는 인-앱(In App)형 관찰 도구가 필요하다는 것이다. 그러나, 앞 단락에 언급한 바와 같이 이는 내부를 알 수 없는 기존 앱의 데이터를 탐색해야 하므로 이를 가능케 할 구현 체계가 연구되어야 한다. 이에 다음 절에서는 본 연구가 제안하는 역어셈블(disassemble) 방식의 인-앱형 데이터 로깅 기법을 서술한다.

### 3. 역어셈블에 의한 인-앱형 데이터 로깅 방식

본 연구가 제안하는 인-앱 탐침형 데이터 로깅 방식은 피험자가 관찰 대상 앱을 그대로 사용하되 실제로는 필요 부분마다 기록용 로깅 코드가 삽입되어 수정된 앱을 사용하게 되는 방법이다.

<그림 1>은 이러한 코드 삽입 및 저장, 서버로의 전송을 포함한 인-앱 탐침형 데이터 로깅 기본 체계도이다. 로그 기능이 추가된 앱은 원래의 앱과 UI 및 기능이 100% 일치하므로 사용자가 별도의 앱을 설치했다는 느낌 없이 자연스럽게 사용할 수 있다. 또한, 스마트폰 단말에 기록된 로그 데이터를 서버에 전송하기 위해 별도의 전송 앱을 두었다.

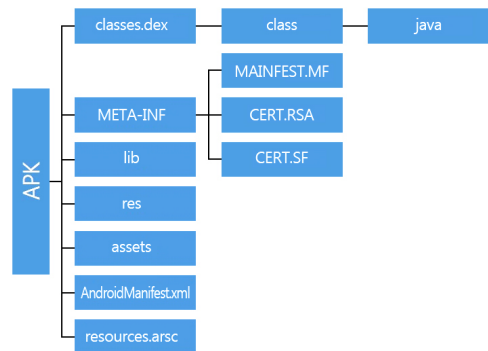


<그림 1> 로그 코드 삽입, 저장 및 전송 체계

### 3.1 APK의 역컴파일 및 역어셈블

본 절에서는 이미 존재하는 안드로이드 앱에 데이터 로깅을 위한 코드의 삽입 절차와, 그 과정에 필요한 도구들을 설명한다.

안드로이드 앱 설치 파일은 APK(Android Package)라 명명되어 있다. 일종의 앱 묶음이라 할 수 있으며, 이는 <그림 2>와 같은 디렉토리 구조의 압축 패키지이다. 안드로이드 앱들은 자바(Java) 언어를 사용하여 개발되지만 안드로이드를 만든 구글 사의 Dalvik 이라는 가상머신 환경에서 실행될 수 있도록 Dalvik 코드로 변환된다. <그림 2>의 APK 파일 구조에서 classes.dex 파일이 변환된 Dalvik 코드이다. 이러한 Dalvik 코드는 해독이 매우 어려우므로 본래 앱의 구조를 알 수 없어 원하는 관찰 기능을 위한 탐침, 즉, 로그 코드 삽입이 어렵다. 따라서 본래 앱의 구조를 알기 위한 일종의 해독 과정을 아래에 설명한다.



<그림 2> 안드로이드 APK 구조

앱에 로그 코드를 삽입하기 위해서는 우선 앱의 구조를 알아내야 하므로, 관찰 대상 앱의 APK 내 Dalvik 코드를 역컴파일 또는 역어셈블해야 한다. 컴파일이란 컴퓨터가 이해할 수 있도록 하는 자바언어의 문장 해독 과정이라 할 수 있으며, 어셈블이란 기계에게 친숙하도록 해독된 문장에 의미를 실어 컴퓨터에서 실행될 수 있는

형식으로 만드는 것이다. 역컴파일과 역어셈블은 이를 거꾸로 하여 본래의 앱 개발에 사용된 문장을 알아내려는 것이다.

이 두 방법을 시도해본 결과 다음과 같은 장단점이 있다.

• 역컴파일 방식

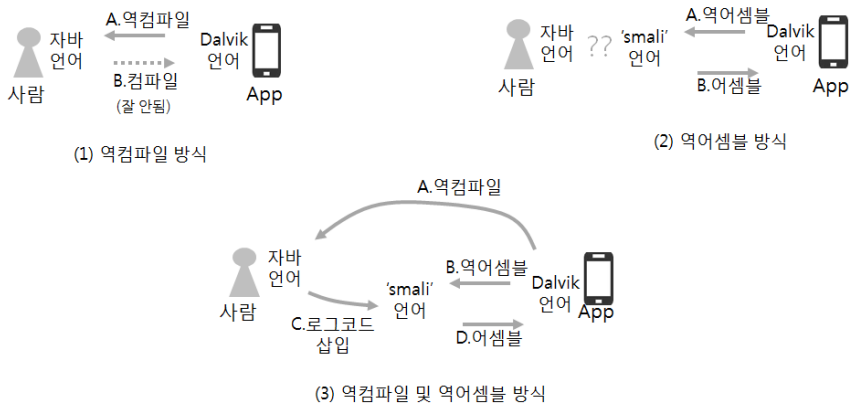
사람에게 친숙한 자바 코드로 변환되기 때문에 가독성이 높아 앱의 구조에 대한 해석이 용이하다. 반면 정상적으로 역컴파일이 안되거나, 이를 다시 컴파일하였을 때 원래의 앱이 제대로 재생성되지 않는 경우가 많다. 이는 관찰 대상이 되는 앱을 역컴파일하여 해독 가능한 프로그래밍 언어 문장을 통해 구조를 분석하게 하므로 로그 코드 삽입이 용이하지만, 정작 수정된 앱을 다시 컴파일하여 실행 가능한 정상적인 APK로 만들 수 없음을 의미한다.

• 역어셈블 방식

Dalvik 코드를 'smali'라는 일종의 Dalvik 용어 어셈블리 언어로 변환하는 것으로, Dalvik 코드를 좀 더 보기 편한 형태로 보여주는 것이다. 자바와 같이 완전히 개념이 다른 언어로 변화하는 것이 아니고, 원래의 Dalvik 코드 구조를 그대로

로 가지면서 표현만을 보기 좋게 바꾼 것이어서 이를 다시 어셈블하여 원래의 앱을 재생성하는데 문제가 없다. 즉, 로그 코드가 삽입된 앱을 재생성하는 데 문제가 없다는 것이다. 그런데, 어셈블리 언어 형태는 컴퓨터가 해독하기 좋은 형태이며 사람(개발자)가 언어를 이해하고 구조를 파악하는 것이 매우 어렵다는 것이 문제다.

따라서 본 연구에서는 역컴파일과 역어셈블을 동시에 사용하여 앱의 구조를 분석하고 데이터 로깅용 코드를 삽입할 수 있도록 하는 체계를 적용했다. 역컴파일과 역어셈블을 모두 수행한 후, 먼저 역컴파일된 자바 코드를 보면서 앱의 수행 흐름을 분석한다. 앱의 내부를 이해함에 따라 탐침 장치로서의 로그 코드를 삽입할 위치를 결정할 수 있게 되고 삽입할 로그 코드는 자바로 작성한다. 하지만, 수정된 것을 다시 컴파일 하여 원래의 앱을 생성할 수 없으므로 실제의 로그 코드 삽입은 어셈블리 언어에서 이루어진다. 이 때, 필요한 코드를 모두 어셈블리 언어로 작성하기는 어려우므로 로그 기록을 위한 모듈들을 자바로 작성하여 컴파일한 후 이를 별개의 파일로 삽입하고, 그 모듈을 사용하여 원래 앱의 특정 위치에 탐침용 로그 코드를 삽입하는 부분만을 직접 'smali' 어셈블리 언어로 작성한다.



<그림 3> 로그코드 삽입 방식. (1) 역컴파일 후 코드수정. 정상적인 APK사용이 어렵다. (2) 역어셈블 후 코드 수정. App의 이해가 어렵다. (3) 역컴파일/역어셈블 동시사용. App의 이해는 역컴파일로 실제 코드 수정은 어셈블리로 수행

### 3.2 APK 해석의 상세 과정

APK 해석을 위한 구체적인 역컴파일 및 역어셈블 과정은 다음과 같다. 우선 관찰 대상 앱의 APK는 안드로이드 폰에서 apkExtractor[16]를 이용하여 추출한 후 사용한다.

#### 3.2.1 APK 역컴파일 방법

역컴파일 시 필요한 도구는 'dex2jar'[13], 'jad'[17], 'jd-gui'[15] 프로그램이며 아래 순서로 역컴파일을 실행한다.

- APK 파일의 확장자를 apk 대신 zip으로 변경하고 압축을 해제한다.
- Dalvik 코드인 'classes.dex'를 'dex2jar'를 이용하여 자바 바이트코드로 변환하고, 'jar'파일을 생성한다.
- 자바 바이트코드인 'jar'파일을 역컴파일러인 'jad', 또는 'jd-gui'를 이용하여 자바언어로 변환할 수 있으며, 이 경우 성공적으로 역컴파일이 수행되면 <그림 4>와 같이 사람이 해독 가능한 코드 형태가 되어 구조를 살펴볼 수 있다.

#### 3.2.2 APK 역어셈블 방법

역어셈블의 경우는 'apktool'[12]이 필요하며, 다음과 같이 사용할 수 있다.

- 'apktool'을 이용하여 다음과 같이 apk를 압축 해제하고 'smali' 기계어로 변환한다.

```
java -jar apktool.jar d com.xxx.apk [압축해제폴더]
```

- 로그 코드를 삽입할 부분은 역컴파일한 자바 코드에서 찾고, 해당 파일과 같은 이름의 파일을 'smali' 어셈블리 언어 코드에서 찾아 수정을 한다. 역컴파일과 역어셈블의 결과의 디렉토리 구조 및 파일명은 거의 비슷하다.

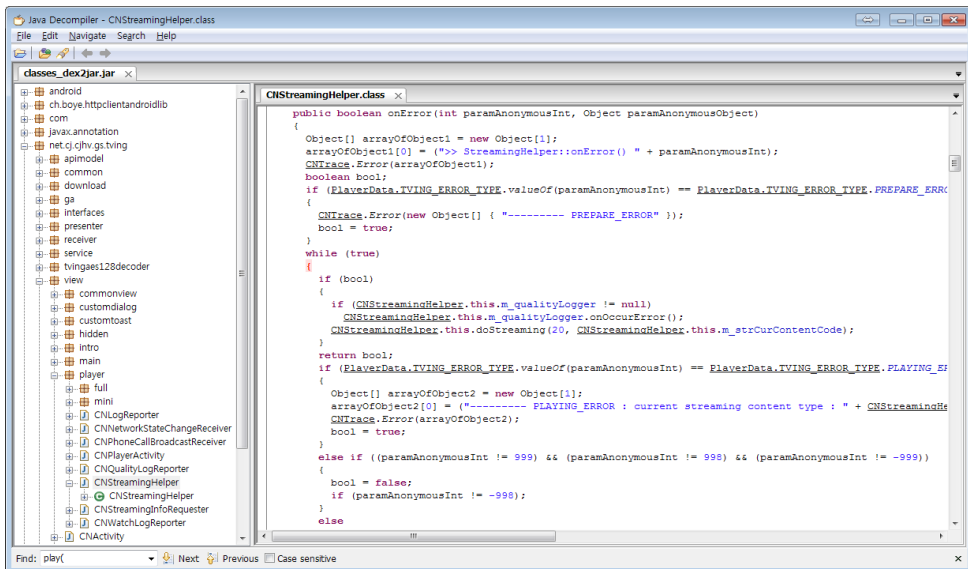
#### 3.2.3 'Smali' 파일에서 APK 재생성 방법

어셈블리 언어에 해당하는 'smali' 파일로부터 스마트폰에 설치 가능한 APK를 재생성하려면 역어셈블의 경우 사용했던 'apktool'이 마찬가지로 사용된다.

- 'apktool'을 이용하여 다음과 같이 APK를 생성한다.

```
java -jar apktool.jar b [수정된 smali 파일 폴더]
```

- APK 파일이 만들어졌으나 서명이 되어 있지 않



<그림 4> 'jd-gui'를 이용한 APK에서 자바로의 역컴파일 예

아 설치가 불가능하므로 다음과 같이 'signapk' [14]를 이용하여 재서명한다.

```
java -jar signapk.jar testkey.x509.pem
testkeypk8 [oldapkfile] [new apkfile name]
```

### 3.3 Smali 어셈블리어에서 로그 코드의 삽입

스마트폰 앱에 삽입되는 코드는 로그 기록을 위한 모듈과 이를 이용하여 실제 로그 기록을 생성하는 위치에 삽입되는 탐침용 로그 코드(실제 로그 writing 코드)가 있다.

먼저 로그 데이터 기록 모듈을 자바 언어로 작성한다. 예컨대, GPS 좌표 로깅 모듈이 필요하다면 GpsClass.java 와 같이 일반적인 안드로이드 자바 프로그래밍으로 작성하여 이를 APK 파일로 만든다. 그리고 이것을 다시 역어셈블 하여 'smali' 어셈블리어 파일을 생성한 후, 그 결과 생성된 GpsClass.smali를 삽입 대상인 앱의 역어셈블 폴더에서 탐침용 로그 코드를 삽입 할 파일과 같은 폴더 위치에 복사한다. 이 때 GpsClass.smali 파일 안에 있는 폴더 이름들을 앱의 폴더 구조에 맞게 수정해야 한다. 제 3.4절에서는 티빙(TVing) 앱에 적용한 실제 사례를 통해 상세한 구현 방법을 보일 것이다.

### 3.4 탐침용 로그 코드 삽입 예(클래스 모듈)

티빙앱 Player가 작동될 때 항상 해당하는 동영상의 스트리밍 주소를 받아오는데, 그 부분의 자바소스코드는 /view/player/CNStreamingHelper.java이다. 그 안의 CNStreamingHelper가 주소를 받아오는 함수인데 <그림 5>의 위치에 GPS 좌표 로그를 기록하는 코드를 삽입하여 동영상을 볼 때마다 GPS 좌표를 로그에 기록하고자 한다.

```
public CNStreamingHelper(PlayerMultiLayout ...)
{
    CNTrace.Debug(new Object[]
        {"CNStreamingHelper()"});
    this.m_context = CNApplication.getContext();

    <<탐침용 로그 코드 삽입 위치>>

    if (paramPlayerMultiLayout != null) {
        this.m_playerLayout = paramPlayerMultiLayout;
        this.m_playerLayout.setOnPreparedListener(...);
        ...
    }
    ...
}
```

<그림 5> 로그 코드 삽입 대상 함수 및 위치

만약 자바언어로 CNStreaming.java의 수정 사항을 표현한다면 다음과 같다. GPS 좌표를 기록하기 위해 제 3.3절에서의 모듈 GpsClass를 사용하는 데 다음과 같이 클래스 인스턴스를 생성하는 방식으로 동작한다.

```
gc = new GpsClass(this.m_context);
```

이것을 <그림 5>의 위치에 삽입하고 다음과 같이 필요한 선언을 해주면 로그 코드 삽입이 완료된다.

```
import view.player.GpsClass;
public GpsClass gc;
```

위와 같은 수정 사항은 실제로는 다음과 같이 smali 어셈블리로 구현되어야 한다. 해당 파일은 CNStremingHelper.smali이다. 먼저 변수 선언을 위해서는 '#instance fields' 주석을 찾아서 아래와 같이 변수를 추가한다.

```
#instance fields
.field gc:Lnet/view/player/GpsClass;
```

코드가 삽입될 위치가

```
this.m_context = CNApplication.getContext();
```

직후이므로 이를 찾기 위해서 'CNApplication'을 검색하여 다음과 같이 해당 smali 코드를 찾는다.

```

invoke-static {}, Lnet/CNApplication;
    ->getContext()Landroid/content/Context;
move-result-object v0
iput-object v0, p0,
    Lnet/view/player/CNStreamingHelper;
->m_context:Landroid/content/Context;

```

이 코드에 바로 이어서 'gc = new GpsClass(this.m\_context);'에 해당하는 다음의 smali 코드를 삽입하면 smali 어셈블리에서의 로그 코드 삽입이 완료된다. 이 코드는 해당 자바코드를 컴파일 한 것을 참조하여 작성한 것이다.

```

new-instance v0, Lnet/view/player/GpsClass;
iget-object v1, p0,
    Lnet/view/player/CNStreamingHelper;
->m_context:Landroid/content/Context;
invoke-direct {v0, v1}, Lnet/view/player/GpsClass;
-><init>(Landroid/content/Context;)V
iput-object v0, p0,
    Lnet/view/player/CNStreamingHelper;
->gc: Lnet/view/player/GpsClass;

```

### 3.5 탐침용 로그 코드 삽입 예(함수 모듈)

독립된 클래스가 아니고 수정하고자 하는 클래스에 멤버 함수를 추가하고 이를 호출하는 로그 코드를 삽입하는 경우가 있다. 예를 들어 현재 시각을 로그 파일에 기록하는 timestamp 함수를 다음과 같이 작성하였다고 하자.

```

public void timestamp()
{
    ...
}

```

마찬가지로 이것을 자바언어로 작성한 후 APK를 만들고 다시 disassemble하여 smali 파일을 만든다. GPS 좌표 로그 기록의 예와 마찬가지로 같은 위치에서 timestamp 함수를 호출하는 로그를 삽입하려고 한다. 먼저 로그 기록 모듈인 timestamp 함수를 추가하여야 하는 데, GpsClass의 경우와는 달리 timestamp는 함수이므로 별도의 smali 파일을 복사하지 않고 수정 대상인 CNS-

streamingHelper.smali에서 .method가 있는 부분에 timestamp.smali의 내용을 추가하여 멤버 함수로 만든다. 이제 제 3.4절에서와 같은 위치에 다음과 같이 timestamp 함수를 호출하면 로그 코드 삽입이 완료된다.

```

invoke-direct {p0},
    Lnet/view/player/CNStreamingHelper;
->timestamp()V

```

## 4. 티빙 앱 적용 실험 및 결과

본 절에서는 제안한 데이터로깅 방법을 안드로이드 티빙 앱에 적용하여 실험한 결과를 설명한다.

### 4.1 티빙(Tving) 앱 데이터 로깅 구조

먼저 티빙 앱의 주요 구조는 인터페이스(Interface), 수신자(Receiver), 서비스(Service), 시청(View)으로 구성된다. 인터페이스는 UI를, 수신자와 서비스는 티빙 앱과 서버 간의 네트워크 송수신을 관리한다. 시청(View)은 콘텐츠를 볼 수 있게 하는 디스플레이 모드를 포함한 플레이어의 모든 이벤트를 관리하는 부분이다. 이 시청(View) 부분이 허브(Hub) 역할을 하므로, 여기에 로그 코드를 삽입하여 원하는 데이터를 기록할 수 있다.

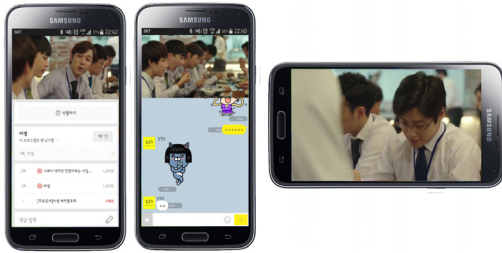
시청 패턴을 분석하기 위한 로그파일 포맷은 <표 1>과 같이 콘텐츠, 플레이어, GPS 그룹으로 분류하였다. 로그는 모든 그룹의 값들이 발생한 당시의 시각(Time)을 포함한다.

콘텐츠 그룹은 서버로부터 수신 받는 URL 주소인 콘텐츠 코드, 콘텐츠 제목, 콘텐츠가 시리즈물일 때의 에피소드 번호, VOD일 경우 중간에 멈춰 다시 돌아왔을 때의 이어보기 위치(Resume Play), 콘텐츠 유형을 포함한다. 이 가운데 콘텐츠 유형의 값은 광고(AD), 미리보기



(Preview), 짧은 클립(Short Clip), 실시간 방송(Live), 다시보기(VOD) 중 하나로 정해진다.

플레이어 그룹은 사용자가 어떤 시청 모드(Mode)와 상태(Status)로 이용하는지 기록할 수 있게 구분하였다. 모드(Mode)는 <그림 6>과 같이 디스플레이를 전체 창으로 띄우는 풀 모드(Full Mode), 시청 외에 멀티태스킹을 할 수 있는 팝업 모드(Pop-up Mode), 기본 모드인 미니 모드(Mini Mode)로 구분하였다. 상태(Status)는 재생(Play), 일시정지(Pause), 종료(Stop)로 구분된다.



<그림 6> 미니모드, 팝업 모드, 풀 모드

로그(Log)는 플레이어의 현재 상태변경, 다른 콘텐츠로의 변경, 실시간 콘텐츠 단위가 변경되는 것과 같은 이벤트가 발생할 때, 해당 시각(time stamp)과 더불어 기록되도록 하였고, 이벤트가 발생할 때의 위치 또한 따로 기록하도록 개발하였다. 그리고 이용자의 시청행위에 대한 편의성 측정을 위해, 현재 이용자가 무선망을 이용하여 시청을 하는지, 휴대전화망을 이용하여 시청을 하는지도 구분하여 기록한다.

#### 4.2 데이터 로깅 기반 시청패턴 추적 실험

본 연구 결과의 관찰과 확인을 위하여 2014년 2월 12일부터 3월 17일까지 약 4주 동안 서울 및 수도권에 거주하며 스마트폰을 통한 TV 시청 경험이 있는 20대, 30대 남녀 11명을 대상

으로 시청 패턴 추적 실험을 실시하였다. 연구 참여자들에게 연구목적 및 행태 로깅에 대해 설명하고 사전 동의를 구했으나, 실제 실험 기간 중에 참여자들이 가급적 실험을 의식하지 않도록 원칙적으로 1주일 단위로 데이터 전송 앱을 통해 전달받았다.

<그림 7>은 표집된 데이터의 예시이며, <표 1>의 콘텐츠와 플레이어 그룹의 정보들과 시청 장소(GPS) 정보는 별도의 로그로 기록된 후, 시간 정보(time stamp)에 의해 서로 연결된다. 이러한 시청 패턴 추적 및 관찰 결과 전체 연구 참여자의 한 달간 평균시청량, 프로그램별 시청량 추이, 평균 시청시간, 시청 시간대별 시청량, 시청 장소 정보, 모드 스위칭 패턴 등을 알 수 있었다.

<표 1> 티빙에 삽입된 로그 기록

Group	Field	Value
Content	Content Code	L000000/...
	Content Title	뉴스테스크/인기가요/...
	Content Episode	Null/1화/...
	Resume Play	89/125/...
	Content Type	AD/Preview/ShortClip/Live/VOD
Player	Mode	Full/Pop-up/Mini
	Status	Play/Stop/Pause
GPS	Current Position	37.4848, 126.8532 ...

<그림 8>은 로그 데이터 활용 예로서 일부 분석 결과를 보인다. (a)의 그래프는 피험자 1인의 콘텐츠 형식 및 유형별 시청량이며, (b)는 시청 로그 데이터와 GPS 좌표 기록을 매칭하여 시청이 발생한 이벤트 빈도에 따른 시청장소를 보였다.

```

2014-02-15-15-44-04,MODE,changeMode2MiniPlay
2014-02-15-15-44-04,WIFITEST,true
2014-02-15-15-44-04,FLAYSTATUS,start
2014-02-15-15-44-05,content,C00600,LIVE,개그콘서트, 737 화,null, 0: 0: 0,0
2014-02-15-15-44-10,FLAYSTATUS,stop
2014-02-15-15-44-12,MODE,changeMode2MiniPlay
2014-02-15-15-44-12,WIFITEST,true
2014-02-15-15-44-12,FLAYSTATUS,start
2014-02-15-15-44-12,content,C01763,LIVE,코미디 빅리그, 27 화,null, 0: 0: 0,0
2014-02-15-15-44-15,MODE,changeMode2FullPlay
2014-02-15-15-46-16,FLAYSTATUS,stop
2014-02-15-15-46-19,FLAYSTATUS,start
2014-02-15-15-46-18,WIFITEST,true
2014-02-15-15-46-19,FLAYSTATUS,stop
2014-02-15-15-46-20,FLAYSTATUS,start
2014-02-15-15-46-20,WIFITEST,true
2014-02-15-15-50-44,MODE,changeMode2MiniPlay
2014-02-15-15-50-45,FLAYSTATUS,stop
2014-02-15-16-22-22,MODE,changeMode2MiniPlay
2014-02-15-16-22-22,WIFITEST,true
2014-02-15-16-22-22,FLAYSTATUS,start
    
```

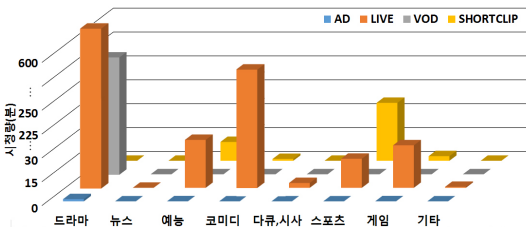
(a) 시청 콘텐츠와 플레이어 그룹 로그파일

```

2014-02-15-16-22-34,127.0491078,NetWork,37.591674
2014-02-15-16-27-33,127.0491125,NetWork,37.5917019
2014-02-16-10-20-00,127.0491095,NetWork,37.5916866
2014-02-16-10-20-00,127.0491095,NetWork,37.5916866
2014-02-16-10-21-20,127.0491038,NetWork,37.591698
2014-02-16-20-09-07,127.0095551,NetWork,37.5681522
2014-02-16-20-09-07,127.0095551,NetWork,37.5681522
2014-02-16-20-12-23,127.0095609,NetWork,37.5682175
2014-02-16-20-12-53,127.0095727,NetWork,37.5683556
2014-02-16-20-13-31,127.0095463,NetWork,37.5682224
2014-02-16-20-13-35,127.0095479,NetWork,37.5681583
    
```

(b) 시청 장소(GPS) 로그파일

〈그림 7〉 시청 로그 기록의 예시



(a) 피험자 1인의 콘텐츠유형 및 형식별 시청량



(b) 시청 이벤트 빈도에 따른 시청 장소

〈그림 8〉 로그 데이터의 분석 활용 예시

분석 결과 특이점 중 하나는 스마트폰이 이동성이 높은 기기이므로 교통수단을 이용한 장소 이동과 더불어 시청이 발생할 것이라 예측했으

나, 실제로는 동영상 시청이 데이터가 많이 소모되는 특징 때문에 교통수단과 같은 비장소(non-places)적 공간에서의 시청은 상대적으로 빈도가 낮았다는 것이다. 특히, 장소 중에서도 ‘집’에서의 시청 빈도가 가장 높았지만, 틈새 시간을 이용한 10~30초 단위 시청패턴들도 상당수 관찰되었으며 이러한 짧은 시청이 채널 스위칭이나 태스크 전환 등과 관련 있는 것으로 파악되었다.

콘텐츠 유형 선택의 경우도 광고를 끝까지 보거나 도중에 끊은 경우 등 상세 시청 패턴 추적이 가능했고, 콘텐츠 형식별 시청패턴에서는 시청총량에 따르면 케이블 드라마를 가장 많이 시청한 것으로 나타났으며, 드라마 시청 시간이 높은 3명의 연구 참여자는 같은 제목의 시리즈 드라마를 몰아보기 식으로 시청하는 공통적 특징을 보였으며, 몇몇 참여자의 로그데이터에서는 흥미로운 콘텐츠는 계속 돌려보는 ‘반복시청’ 패턴이 뚜렷하게 나타났다.

## 5. 결 론

본 연구는 스마트폰 데이터 로깅의 실질적인 솔루션을 제안한 데 의의가 있다. 기존 스마트폰 앱을 그대로 이용하면서도 삽입된 로그 기록 코드에 의해 다양하고 상세한 형태의 사용 패턴 로그를 기록하는 실질적 방법을 제시하였다. 이는 기존의 데이터로깅 방식들이 데이터 로깅을 위해 만들어진 인위적 앱을 사용하거나 앱과 무관한 스마트폰 기기 자체의 활동, 즉, 위치나 배터리 상태 등만을 기록할 수 있었던 점을 개선한 것이다. 즉, 이미 존재하는 상용 앱들의 인-앱(In-App) 사용 활동은 기록할 수 없었던 제약을 역컴파일 및 역어셈블 방식을 통해 해결해낸 것이다.

현재 안드로이드 플랫폼에 한정된다는 한계점은 있으나 향후 비즈니스 고객 경험 측정에 있어서 인터뷰나 관찰을 보완하거나 대체할 수

있는 방법이며, 데이터에 기반 한 객관적이고 구체적인 사용 행태 예측 및 사전 분석 등 다양한 용도의 활용을 기대할 수 있다.

향후 애플과 같은 다른 모바일 플랫폼에서도 같은 방식의 접근이 가능할 것으로 예상되지만, 앱 설치의 폐쇄성으로 인해 피험자의 기기에 설치가 불가능한 문제가 있다. 이는 별도의 탈옥된 기기에 앱을 설치하여 제공하는 방법으로 해결할 수 있을 것으로 보인다.

기존 상용 앱을 수정한 것이 영리의 목적으로 오용되거나 유출된다면 저작권 문제의 소지가 있으므로 주의를 요하여야 한다. 또한 사용 행태를 기록하는 것이 프라이버시 문제를 일으킬 수 있으므로, 기록내용과 용도에 대해 실험 전에 상세히 공지하고 피험자의 동의를 얻어야 하며 언제라도 원치 않는 경우 실험을 중지하고 데이터를 파기할 수 있음을 알려야 한다.

## 참 고 문 헌

- [1] 전종근, 박 철, “웹 로그 데이터를 이용한 온라인 소비자의 가격민감도 영향 요인에 관한 연구”, *Journal of Information Technology Applications and Management*, Vol. 13, No. 1, 2006, pp. 1-16.
- [2] Asenjo, P. E. R., “Web User Behavior Analysis”, Ph.D Thesis, Engineering System Dept., University of Chile, 2011.
- [3] Brodie, R. J., Ilic, A., Juric, B., and Hollebeck, L., “Consumer engagement in a virtual brand community : An exploratory analysis”, *Journal of Business Research*, Vol. 66, 2013, pp. 105-114.
- [4] Cao, L., “In-depth behavior understanding and use : The behavior informatics approach”, *Information Sciences*, Vol. 180, 2010, pp. 3067-3085.
- [5] Chen, M., Yang, X., and Liu, T., “A Research on user behavior sequence analysis based on social networking service use-case model”, *Int'l J. of U- and E-Service, Science and Technology*, Vol. 7, No. 2, 2014, pp. 1-14.
- [6] Do, T. M. T., Blom, J., and Gatica-Perez, D., “Smartphone usage in the wild : a large-scale analysis of applications and context”, Proceedings of 13th Int'l Conf. on multimodal interfaces(ICMI), Alicante, Spain, 2011, pp. 353-360.
- [7] Falaki, H., Mahajan, R., Kandula, S., LyMBERopoulos, D., Govindan, R., and Estrin, D., “Diversity in Smartphone Usage”, Proceedings of MobiSys, 2010, San Francisco, California, USA, pp. 179-194.
- [8] Kramer, A. D. I., Guillory, J. E., and Hancock, J. T., “Experimental evidence of massive-scale emotional contagion through social networks”, *PNAS*, Vol. 111, No. 24, 2014, pp. 8788-8790.
- [9] Moller, A., Kranz, M., Schmid, B., Roalter, L., and Diewald, Stefan, “Investigating Self-Reporting Behavior in Long-Term Studies”, Proceedings of CHI : Changing Perspectives, Paris, France, 2013, pp. 2931-2940.
- [10] Robinson D. J., Berk V. H., and Cybenko, G. V., “Online behavioral analysis and modeling methodology(OBAMM)”, Book Chapter In Social Computing, Behavioral Modeling, and Prediction, Springer US, 2008, pp. 100-109.
- [11] Smura T., “Access alternatives to mobile services and content : analysis of handset-

- based smartphone usage data”, ITS 17<sup>th</sup> Biennial Conf., Montreal, Canada, 2008, pp. 1-17.
- [12] <https://code.google.com/p/android-apktool>.
- [13] <https://code.google.com/p/dex2jar>.
- [14] <https://code.google.com/p/signapk/>.
- [15] jd-gui, <http://jd.benow.ca>.
- [16] APK Extractor, <https://play.google.com/store/apps/details?id=com.ext.ui>.
- [17] <http://varaneckas.com/jad>.

## ■ 저자소개



**김 명 준**

현재 이화여자대학교 디지털미디어학부 부교수로 재직하고 있다. 한국과학기술대학교 전산학과를 졸업하고, KAIST 전산학과에서 컴퓨터그래픽스 분야로 석사 및 박사학위를 취득하였다. 이후, 미국 University of Washington에서 방문연구원으로 재직하였고, 전자통신연구원에서 컴퓨터그래픽스 분야 선임연구원으로 재직하였다. 주된 관심분야는 컴퓨터그래픽스, 영상처리, 클라우드 컴퓨팅, 어플리케이션 개발 등이다.



**남 양 희**

현재 이화여자대학교 디지털미디어학부 부교수로 재직하고 있다. 이화여자대학교 전자계산학과를 졸업하고, KAIST에서 인공지능 분야 석사, 가상현실 분야로 박사학위를 취득하였다. 이후, 스위스 로잔 연방공대에서 컴퓨터 그래픽스 분야 박사 후 연구원으로 재직한 바 있고, 일본 국제 정보통신연구소(ATR)의 미디어 콘텐츠 연구부서에 초청연구원으로 재직하였었다. 주요 논문을 IEEE System, Man and Cybernetics와 Pattern Recognition Letters를 비롯한 국제 학술지와 한국콘텐츠학회 논문지, 한국정보과학회지 등 국내학술지에 게재해왔고, 주요 연구 분야는 증강현실, 인터랙티브 미디어, 모바일 미디어 콘텐츠, 디지털 스토리텔링 등이다. 현재 한국 HCI학회 부회장, 한국 정보과학회 이사, 그래픽스학회 이사, 여성정보인협회 이사로 활동 중이다.