모바일 앱 로그 분석 플랫폼 개발에 관한 연구

양국모* · 주재걸**
*고려대학교 컴퓨터정보통신대학원 빅데이터융합학과 · **고려대학교 컴퓨터학과
e-mail : yangkukmo@korea.ac.kr*, jchoo@korea.ac.kr**

A Study on the Development of Mobile App Log Analysis Platform

Kuk-Mo Yang* · Jae-Gul Choo**
*Dept of Big Data Convergence, Graduate School of Computer & Information Technology Korea University
**Dept of Computer Science and Engineering

요 약

미국 애플사의 아이폰 발표 이후 스마트폰은 PC 웹 기반 서비스를 빠르게 대체하며 산업 전반에 혁신을 가져왔다. 모바일 앱(이하 앱) 기술은 네이티브, 웹 방식이 융합된 하이브리드 방식으로 진화 되었다. 앱의 확산으로 디지털 마케팅, 사용성 분석에 대한 필요성이 증대 되었으나 시장에서 활용되는 분석 도구가 모든 기술 요소를 지원하지 못하여 분석가가 여러 도구를 번갈아 가며 사용하는 불편함이 발생하였다. 본 연구에서는 모바일 앱 네이티브, 웹 로그 데이터 통합 수집을 지원하는 로그 분석 플랫폼을 제안하고자 한다.

1. 서론

2007년 1월 미국 애플사의 아이폰 발표 이후 스마트폰은 웹 기반 서비스를 빠르게 대체하며 산업 전반에 혁신을 가져왔다. 전통적인 웹 기반 서비스의 로그 분석은 PC 브라우저 중심의 디지털 마케팅에 한정 되었지만 스마트폰의 모바일 환경은 이동성과 소형 화면 등을 고려하여 기존 디지털 마케팅과 사용성 분석 등을 포괄해야 되는 과제를 안게 되었다.
PC 기반 서비스는 웹 기반 기술이 표준으로 정착 되었지만 모바일 환경은 사용 단말의 제약으로 네이티브, 웹 방식이 융합된 하이브리드 방식으로 진화 되었다.

로그 분석의 필요한 사용자는 디지털마케팅을 수행하는 마케터, UI/UX 설계자이다.
시장에서 일반적으로 사용되는 로그 분석 도구는 구글 Analytics, Firebase, Facebook 통계 등을 사용하지만 모바일 네이티브, 웹 방식을 통합해서 지원해 주는 도구가 없어 사용자가 여러가지 도구를 번갈아 사용해야 되는 불편함이 존재한다.
또한 일반적으로 사용하는 분석도구는 로그 생성을 위해 프로그램 단위의 별도 태깅 작업(수정)이 필요하여 프로그램 수정 비용과 적용 시 운영 리스크 등으로 대규모 기업 환경에 적용하기에 어려운 점이 있다.

본 연구에서 제안하는 방식은 최소의 프로그램 수정으로 전체 프로그램에 적용하여 프로그램 수정 비용 절약하고 네이티브, 웹 두가지 방식의 로그를 통합 수집할 수 있는 방법을 제안하고자 한다.

2. 플랫폼 구성 요소

2.1 데이터 기획

앱 로그 수집 시스템 설계를 위해 수집 데이터를 정의하였다. 데이터 기획의 원칙은 사용자를 식별할 수 있는 민감 정보를 배제하고 자동 수집이 가능한 항목으로 기획하였다. 데이터 설계 시 CDP(Customer Data Platform) 사업자[1][2]의 앱, 디바이스 수집 항목을 참고하여 6가지 속성으로 분류하고 주요 항목을 설계 하였다.

구분	상세
Init	모바일 디바이스 정보
LoginInfo	로그인, 로그아웃 로그
Page	모바일 화면 접근 로그
Event	모바일 화면 內 이벤트 발생 로그
Error	모바일 앱 에러 로그
User	사용자 정보 (별도 DB 데이터)

<표 1> 수집 데이터 분류 속성 정의

구분	항목 영문명	항목 한글명
공통	ANONYMOUSID	사용자 식별키
	SESSIONID	세션 ID
	IP	디바이스 IP
	OSNAME	운영체제명
	OSVERSION	운영체제정보
	WIFIAPNAME	와이파이 접속 AP 명
	SCREENORIENTATION	디바이스 가로세로모드
	APPNAME	앱 명칭

Init	CARRIER	네트워크 사업자명
	LOCALECOUNTRY	디바이스 로케일 국가명
	STATE	디바이스 연결상태
	TIMEZONE	디바이스 시간지역
	USERAGENT	디바이스 브라우저 정보
LoginInfo	LOGINTYPE	앱 로그인 유형
	LOGINSTATUS	앱 로그인 상태
Page	SCREENNAME	화면명
	PREVIOUS	이전화면 주소정보
	CURRENT	현재화면 주소정보
	URL	화면 주소정보
	TITLE	화면 타이틀명
Event	EVENTTYPEID	이벤트 구분 아이디
	HEIGHT	이벤트 발생 X 좌표
	WIDTH	이벤트 발생 Y 좌표
	ACTIONNAME	액션명
	ACTIONID	액션 아이디
Error	ERROR MESSAGE	오류 메시지
User	GENDER	성별
	AGEGROUP	연령 그룹
	ADDRESSGROUP	거주지

<표 2> 수집 데이터 주요 항목 정의

2.2 앱 로그 수집 SDK

앱 로그 수집 SDK(이하 SDK)의 설계 목표는 앱에서 최소의 수정 작업으로 네이티브, 웹 로그를 통합하여 자동 수집하고 수집 서버의 부하 경감, 스마트폰 배터리 사용 최소화를 고려하였다.

1) 최소 수정, 로그 수집 자동화 구현

네이티브, 웹 로그 통합 로그 수집을 위해 네이티브는 모바일 OS에서 제공하는 API를 사용하고 웹은 공통 Java Script(이하 공통JS)를 웹 서버에 적용하여 웹 화면 정보는 공통 JS에서 수집 후 모바일 SDK에 전달하는 방식을 사용하여 구현 하였다.
테스트 진행 중 발생한 문제는 공통JS와 SDK 연동 시 URL 스키마 Call(공통JS 스크립트 內 SDK 함수 호출)을 사용 하였는데 iOS OS 특성 상 웹 화면에서 URL 스키마를 동시 2회 이상 호출 시 선행 호출된 스키마만 처리되는 현상이 발생하였다. 이 문제의 해결 방안으로 공통JS에서 iOS인 경우 별도 분기하여 화면 데이터 취득 후 변수에 저장만 하고 웹페이지 로드 완료 이벤트 webViewDidFinishLoad 발생시 SDK에서 변수에 저장된 값을 역으로 읽어오는 방식으로 문제를 해결하였다.

SDK를 호출하는 앱의 최소 수정을 위해 앱 실행 이벤트를 감지하여 SDK 인스턴스를 생성하고 Listener에 등록하여 전체 앱에 적용될 수 있도록 개발 하였다.

로그 수집을 자동화하는 2가지 주요 기술은 다음과 같다. 첫째, 화면접근 로그(이하 Page)는 모바일 OS 특성 상 아래와 같은 방식으로 구현 하였다.
Android OS는 onWindowFocusChanged 이벤트 발생 시 로그를 수집하고 iOS는 현재 뷰를 계층구조로 실시간 저장 후 기존 뷰와 비교하여 변경사항 발생 시 (현재 뷰 트리의 상태 변경) 로그를 저장하는 방식으로 개발 하였다. 둘째, 모바일 화면 내 이벤트 발생 로그(이하 Event)는 해당 이벤트 Listener를 등록하여 이벤트 발생 시 자동으로 로그를 저장하는 방식으로 구현 하였다.

로그 수집의 자동화 구현 후 발생한 문제는 앱 화면 내 이벤트 발생 시 실시간으로 스마트폰 내부에 저장할 경우 다수의 로컬 파일 IO가 발생하여 앱 속도 저하 현상이 발생하였다. 개선 방안으로 로그 수집 시 로컬 파일 실시간 저장이 아닌 메모리 저장 방식으로 변경하고 앱이 백그라운드 상태로 전환되는 경우에만 로컬 파일에 일괄 저장하는 방식으로 처리하여 문제를 해결하였다.

2) 서버 부하 경감, 스마트폰 배터리 사용 최소화

수집된 로그 데이터는 스마트폰 내부에 임시 보관하며 누적된 로그가 100건이 될 경우 서버에 JSON 타입으로 일괄 전송하여 서버 부하를 예방하고 스마트폰 배터리도 절약하게 개발 하였다. 전송 기준 디폴트값은 100건이며 SDK 최초 적용 시 가변(건수, 로그 Size) 적용이 가능하다.

2.3 앱 로그 수집 서버

로그 수집 서버로는 SDK 로그 수신 API 서버(Spring), ELK(ElasticSearch, Logstash, Kibana) Stack[3] 5.4을 사용하여 구축 하였다.
ELK Stack은 오픈소스 기반으로 로그 수집기, 빅데이터 검색 엔진, 시각화 도구를 패키지로 제공하며 대량의 로그 데이터 취합, 정제, 분석이 가능하도록 구성되어 있다.

구분	상세
API 서버	SDK 수신 데이터 파일 저장
Filebeat	수신 로그 → Logstash
Logstash	수신 로그 파싱 → ElasticSearch
ElasticSearch	앱 로그파일 DB 적재
Kibana	로그파일 시각화 도구

<표 3> 앱 로그 수집 서버 구성 요소

3. 시범운영 결과

2017년 7월 ~ 2017년 12월, 6개월간 특정 기업의 사내 메신저 앱에 시범 적용하여 파일럿 프로젝트를

진행 하였다. 집계된 자료는 2017 년 12 월 한달 간 수집된 데이터이며 시범운영 기간 중 앱의 특별한 성능적 이슈는 보고 되지 않았다.

구분	Init	LogInfo	Page	Event	Error	합계
A	0	93	735	2747	1	3,576
B	102	11,003	120,562	845,586	153	977,406
C	91	6,895	54,597	268,956	83	330,622
D	47	831	5,189	16,728	5	22,800
E	4	165	1,414	5,506	4	7,093
F	0	3	7	7	0	17
G	0	5	22	130	0	157
H	5	857	7,802	57,563	12	66,239
I	0	21	143	265	0	429
J	286	9,942	89,878	702,518	267	802,891
K	0	17	173	511	0	701
L	0	1	11	8	0	20
M	0	1	25	71	0	97
N/A	26,981	166	23,580	82,593	452	133,772
합계	27,516	30,000	304,138	1,983,189	977	2,345,820

<표 4> 시범운영 데이터 집계 현황

A~M 까지 13 개 그룹으로 분류 하였고 B,C,H,J 그룹이 사내 메신저 활용이 활발한 것을 확인할 수 있었다

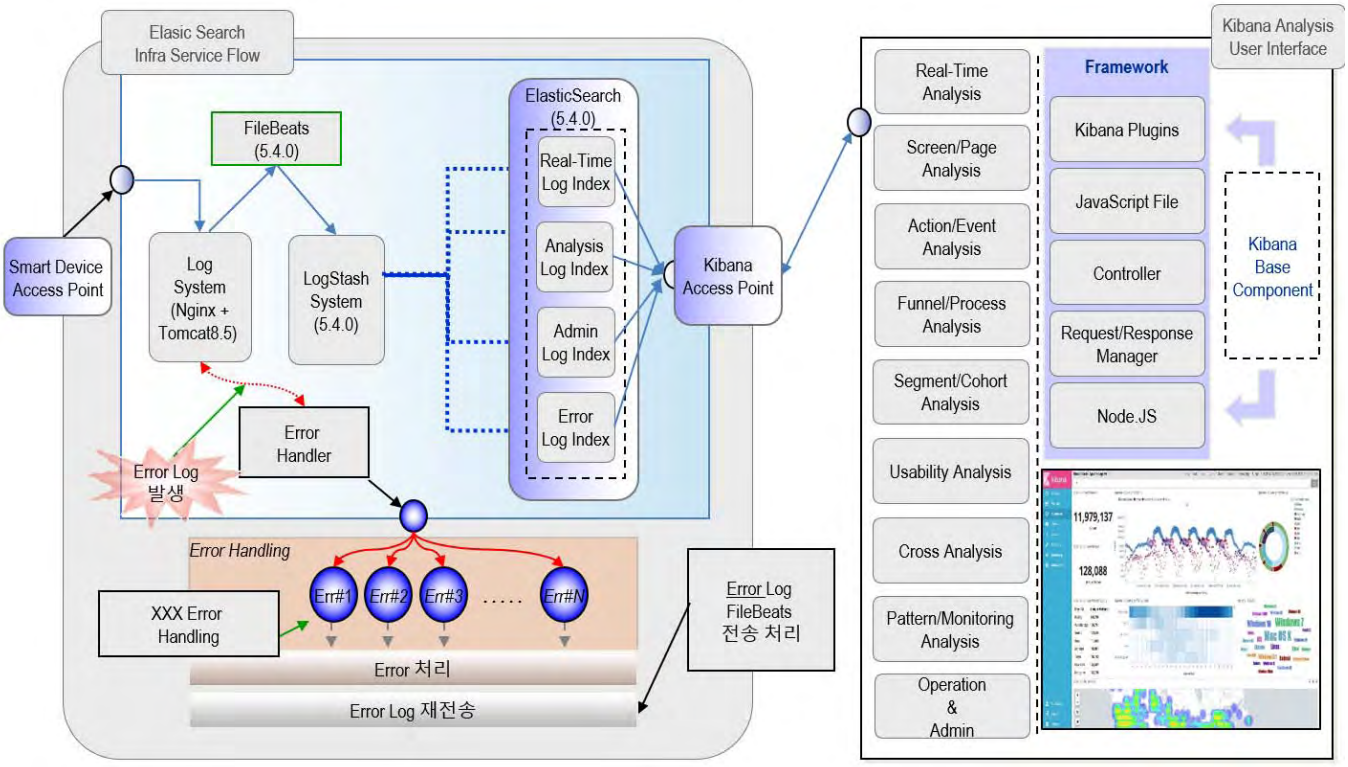
그룹이 분류되지 않은 미분류 데이터 133,772 건은 시범운영 초기에 그룹 플래그가 누락되어 발생한 건으로 추정 된다.

시범운영 시 사내 메신저 강제 업데이트는 진행하지 않았고 신규/재설치 건에 한정하여 시범운영을 실시하였다. 각 그룹 별 전체 인원 대비 참여율 현황은 아래와 같다.

구분	참여율	구분	참여율
A 그룹	8.3%	H 그룹	22.9%
B 그룹	14%	I 그룹	2.2%
C 그룹	9.4%	J 그룹	71%
D 그룹	7.9%	K 그룹	3.1%
E 그룹	2.4%	L 그룹	2.9%
F 그룹	0.9%	M 그룹	7.7
G 그룹	0.7%	미분류	-

<표 5> 그룹별 시범운영 참여율 현황

J 그룹의 참여율이 타 그룹 대비 월등한 이유는 사내 메신저를 개발/운영 중인 그룹으로 소속 구성원이 활발히 활용하고 있다.
데이터 집계 현황 분석 결과 앱에 화면 별 로그 수집 용도의 태깅 코드를 넣지 않고도 대용량의 로그 데이터가 간편하게 수집된 것을 확인할 수 있었다.



<그림 1> 모바일 앱 로그 분석 플랫폼 서버 아키텍처

4. 향후 과제 및 계획

시장에서 주류로 사용되는 구글 Analytics 는 범용 솔루션으로 많은 산업군에서 활용 되지만 화면 단위 태깅 작업은 엔터프라이즈 기업 환경에서는 매우 부담이 되는 요소임으로 이번 파일럿 프로젝트를 통해 개발한 SDK 는 효율적인 시도였다고 생각한다.
아쉬운 점은 6 개월 간 파일럿 진행 경험이 없어 여러가지 시행착오를 경험하였고 다음과 같다.

1) 시각화 도구 선정 관련

데이터 시각화를 위해 ELK Stack 에서 패키지로 제공하는 Kibana 도구를 커스터마이징 하여 새로운 기능 개발을 추진하였다. Kibana 가 오픈소스이긴 하지만 node.js 기반 One Page 어플리케이션 사상이 있어 화면 간 결합도가 매우 높다는 사실을 알게 되었다. 개발 진행을 할수록 작업자 간 conflict 이 발생하여 어려움을 겪었고 제공하는 chart component 가 미려하지 않은 문제로 고객용 서비스로는 적합하지 않다고 판단하여 이후 Spring 기반으로 재개발 하였다.

2) NOSQL DB 특성 파악

ElasticSearch 는 NOSQL Schemaless 사상의 DB 로 테이블 간 Join 이 매우 제한적인 특성이 있다. 데이터 모델 설계 시 NOSQL DB 의 특성을 충분히 검토하지 못하고 습관적으로 관계형 DB 설계 사상을 적용하여 분석화면 개발 시 혼란이 발생하였다. NOSQL DB 는 본래 취지인 대용량 데이터의 빠른 검색을 위해 역정규화하여 One Table 로 설계 했어야 했다.
또한 테이블 간 Join 이 매우 제한적이기 때문에 사용자 정보(별도 DB 데이터)와 결합하는 방법이 Join 이 아닌 Logstash 를 활용하여 ElasticSearch 에 데이터 병합 적재를 해야 되기 때문에 기업 내 DW 데이터와 결합하여 분석하기에는 적합하지 않다고 판단된다.

3) 특화 기능 개발

파일럿 기획 단계에서 구글 Analytics 와 같은 시장의 Major Player 와는 다른 특화 기능을 고민 하였다.
현재 활발히 사용중인 도구들의 특징은 데이터 기반 여러 분석지표를 제공하지만 분석에 의한 인사이트 발굴은 온전히 분석가의 휴리스틱 영역이다.
이러한 사항에 착안하여 사용자 행위 로그를 이용한 모바일 어플리케이션의 사용성 분석 기법[4], 모바일 앱에서 사용자 행동 모델 기반 GUI 사용성 저해요소 검출 기법[5]의 연구는 데이터 분석을 통한 이상징후의 자동 검출이라는 흥미로운 연구라고 생각한다.

4) 향후 계획

모바일 분야는 트랜드가 빠르게 급변하는 환경으로 특정 분석 기법, 룰 기반 특정 알고리즘이 지속적으로 유효하지 않은 환경이다.
시범운영으로 확보한 데이터 기반으로 데이터 마이닝, 기계학습 기법 등을 적용하여 사용자의 행동 패턴인지, 파악, 예측 분석 연구를 계속 진행 하겠다.

참고문헌

[1] https://segment.com
[2] https://www.treasuredata.com
[3] https://www.elastic.co/kr
[4] 김영욱, 변정원, 최순황, 박수용, 박수진 “사용자 행위 로그를 이용한 모바일 어플리케이션의 사용성 분석 기법”, 정보과학논문지: 소프트웨어 및 응용 39(2), 2012.2, 91-98 (8page)
[5] 마경욱, 박수용, 박수진, “모바일 앱에서 사용자 행동 모델 기반 GUI 사용성 저해요소 검출 기법”, 정보처리학회논문지/소프트웨어 및 데이터 공학 제 5 권 제 7 호(2016.7)