

Open API를 이용한 스마트폰 위치기반 모바일 매쉬업 개발

이용주, 서종원, 이정운
경북대학교 과학기술대학 컴퓨터정보학부
e-mail:yongju@knu.ac.kr, seo4088@gmail.com, olloxz@gmail.com

Location Based Mobile Mashup Development of Smart Phone Using Open APIs

Yong-Ju Lee, Jong-Won Seo, Jung-Woon Lee
School of Computer Information, Kyungpook National University

요 약

매쉬업은 공개된 Open API들을 이용하여 두 가지 이상의 서로 다른 자원을 섞어서 완전히 새로운 가치의 서비스를 만드는 것이다. 매쉬업은 미래 IT 융합 서비스의 효과적인 구현 방법으로써 그 관심도가 점점 높아지고 있으나 컴퓨팅 파워가 떨어지는 스마트폰 환경에서 매쉬업을 성공적으로 개발하기 위해서는 한정된 스마트폰 성능과 고성능 무선 인터넷 활용에 대한 이슈들을 고려해야만 한다. 본 논문에서는 다양한 Open API들을 이용하여 스마트폰에서 활용 가능한 위치기반 모바일 매쉬업을 개발한다. 이는 클라우드 컴퓨팅 개념을 접목하여 별도의 솔루션을 구축하지 않고도 저비용·고효율적으로 스마트폰 응용 프로그램을 제작할 수 있는 방법을 제시하고 있다.

1. 서론

Open API(Application Program Interface)는 Web 2.0의 근본 개념인 데이터의 공유·개방·참여 중심의 인터넷 환경을 구현할 수 있는 핵심 기술로 자사의 자원을 웹 서비스를 통해 외부로 공개한 것을 말한다. 매쉬업(Mashup)은 이러한 공개된 Open API를 이용하여 두 가지 이상의 서로 다른 자원을 섞어서 완전히 새로운 가치의 서비스를 만드는 것이다. 이는 기존의 자원을 활용하여 새로운 콘텐츠를 만들기 때문에 새로운 서비스를 구축하기 위한 개발 기간 및 비용이 절약될 수 있다. 매쉬업 서비스로서 가장 유명한 한 예는 Google 맵과 Craigslist 부동산 정보를 조합한 HousingMaps[1]로 지도 위에 부동산 매물 정보를 보여주는 서비스를 제공하고 있다.

매쉬업은 미래 IT 융합 서비스의 효과적인 구현 방법으로써 그 관심도가 점점 높아지고 있으며, 그들의 활용도 엔터프라이즈 비즈니스 분야로부터 과학적 e-사이언스 분야에 이르기 까지 매우 다양하다. 특히, 지난 5년간 다운로드 건수가 500억 건에 달하는 애플 앱스토어에서 알 수 있듯이 스마트폰을 이용한 매쉬업 활용은 앞으로 폭발적으로 증가될 것이다. 그러나 이러한 스마트폰 환경에서 매쉬업을 성공적으로 개발하기 위해서는 다음의 몇 가지 이슈들을 고려해야 한다.

첫째, 기존 대부분의 매쉬업들은 유선 환경을 기반으로 개발되었기 때문에 무선 및 모바일 환경에 이를 바로 적용하기에는 문제가 많다. 모바일 스마트폰 환경에서는

CPU 성능, 메모리 용량, 대역폭 등과 같은 다양한 제약조건들을 고려해야 하고, PC에 비해 작은 화면과 키보드와 같은 입력 장치가 없고 컨텍스트가 수시로 변화하는 모바일 환경을 고려해야 한다.

둘째, 스마트폰의 경우는 자체적인 컴퓨팅 파워가 부족하기 때문에 과부하 작업이나 방대한 자료를 활용하는 경우(예, 음성인식서비스) Open API를 통해 제약사항을 극복할 수 있다. 스마트폰에서는 단지 서비스를 제공하는 서버에 자료만 요청하고 복잡하고 계산이 오래 걸리는 작업은 클라우드 컴퓨팅(cloud computing) 환경을 이용하여 그 결과 값을 받아오는 형태로 구현될 필요가 있다.

셋째, 스마트폰은 모바일 기기로서 이동성·휴대성·즉시성 등의 장점이 있기 때문에 이러한 특성을 최대한 활용한 매쉬업 구현이 요구된다. 스마트폰의 위치기반서비스(LBS: Location Based Service)를 이용하면 현재 위치 탐색이 가능하며, 이는 즉각적인 피드백과 공간정보의 탐색 및 공유가 가능한 프로그램을 구현할 수 있다.

본 연구에서는 이러한 이슈들을 고려하고 기존의 다양한 Open API들을 이용한 스마트폰 위치기반 모바일 매쉬업을 개발한다. 본 논문에서 개발된 모바일 매쉬업은 안드로이드 상에서 GoogleMap API V2 최신 버전과 위치 API인 android.location, 그리고 Google Place API 등을 활용하여 장소 정보 디스플레이, 사용자 현재 위치의 건물, 설명 정보 업로드, 그리고 다양한 위치제공자 선택 기능 등을 제공한다. 이는 Web 2.0 환경에서 제공하는 다양한 Open API들을 활용하여 별도의 솔루션을 구축하지 않고도 저비용·고효율적으로 스마트폰 응용 프로그램을 제작할 수 있는 방법을 제시하고 있다.

이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(No. 2010-0008303).

2. 관련 연구

Web 2.0의 대표적인 산물인 매쉬업은 외국에 비해 국내에선 아직 활성화가 부족한 상황이다. 국내에서는 네이버, 다음, 옥션 등 포털사이트를 중심으로 Open API들을 제공하고 있지만 이들 API들로 실제 매쉬업을 구현한 사례는 그렇게 많지 않다. 더군다나, 스마트폰 기반 모바일 매쉬업 개발은 아직 초보 수준에 머무르고 있으며 스마트폰 환경을 고려한 체계적인 연구 개발이 되어지고 있지 않은 상태이다. 지금까지 발표된 관련 논문들로서, [2]에서는 Open API를 이용한 모바일 일정관리 매쉬업을 개발하였고, [3]에서는 스마트폰 기반의 위치, 날씨, 교통 매쉬업 서비스를 구현하였다. 또한 [4]에서는 구글맵 기반 사용자 참여형 스마트폰 GIS 매쉬업을 개발하였다.

외국의 경우 스마트폰 환경에서 매쉬업을 개발한 예로는 소셜 네트워크 게임인 WeRule, 위치기반 지도 서비스 Playmap 등 많은 종류의 매쉬업들을 볼 수 있으며, 모바일 매쉬업에 관한 논문으로는 Cho[5], Maximilien[6], Jin[7], Peng[8] 등이 있다. Cho[5]는 모바일 웹 애플리케이션을 위한 동적 매쉬업 플랫폼을 새롭게 제안하였고, Maximilien[6]은 모바일 매쉬업에 대한 기본 개념, 개발 방향, 그리고 향후 이슈들에 대해 정리하였다. Jin[7]은 모바일 매쉬업에 관한 현황 분석과 시스템 구조에 관한 설계를 제안하였다. 그리고 Peng[8]은 웹 서비스 조합을 위한 시맨틱 기반 모바일 매쉬업 플랫폼을 소개하였다. 그러나 이들 논문들은 비교적 단순한 서비스만 처리함으로써 현실성이 부족하고 구체적인 구현 내용은 서술되어 있지 않다.

3. 모바일 매쉬업과 Open API

3.1 모바일 매쉬업

최근 스마트폰의 사용이 대중화되었고 무선 인터넷 환경의 눈부신 발전으로 인해 모바일 매쉬업에 대한 수요도 폭발적으로 증가하고 있다. 초기의 모바일 환경은 유선 인터넷 망에 비해 확연히 네트워크 속도가 느렸고, WAP (Wireless Application Protocol)을 사용했기 때문에 기존의 HTTP와 호환 되지 않고 게이트웨이를 통한 한정된 콘텐츠만 제공받을 수 있었다. 그러나 최근에는 HSDPA (High Speed Downlink Packet Access), WiBro 등과 같은 3.5세대 이동통신 기술과 스마트폰의 성능이 대폭 향상되면서 사용자들은 언제든지 다양한 콘텐츠의 지속적인 접속이 가능한 환경이 조성되었고 실시간적으로 정보의 생산 및 공유도 가능하게 되었다.

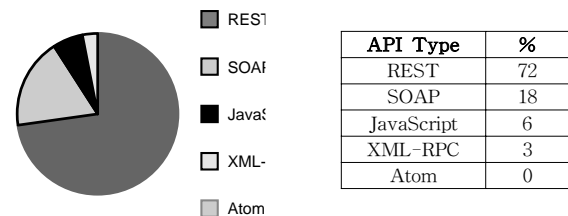
AnyTime, AnyWhere, AnyDevice 개념을 도입한 모바일 매쉬업은 근래의 초고속 무선 인터넷 환경과 매쉬업이 결합된 새로운 형태의 매쉬업으로서 사용자가 자주 이용하는 서비스를 스마트폰에 유/무선 통신을 통해 미리 또는 실시간으로 제공 받아 시공간의 제약 없이 이동 중에 검색, 갱신, 분석 작업을 수행할 수 있다. 또한, 모바일 매쉬업은 스마트폰의 위치정보, 사진, 음성, 음악 등의 컨텐

츠를 웹 환경에서 제공하고 있는 Open API들과 합쳐 새롭고 뛰어난 컨텐츠를 쉽게 만들 수 있는 장점이 있다. 이러한 모바일 매쉬업은 스마트폰 기술, 무선통신 기술, 매쉬업 기술, 그리고 Open API 기술과 융합되어 함께 발전되고 있다.

3.2 Open API

Open API의 개념은 포털의 개방성을 높이기 위한 기술로 공개된 응용 프로그램 인터페이스로 정의될 수 있다. 대부분 일반 사용자는 제공자에게 인증키를 요청하고 해당 인증키로 Open API에 접근하여 사용할 수 있다. 일반적으로, 구현하기 어려운 웹 서비스 기능이나 기업에서 보유하고 있는 데이터를 Open API를 통해 제공함으로써 특정 기능을 쉽게 구현할 수 있을 뿐만 아니라 사용자들에게 폭 넓은 콘텐츠를 제공할 수 있다. 따라서 Open API들을 얼마나 잘 활용할 수 있느냐가 매쉬업의 핵심이라 할 수 있다. 국내에서는 네이버, 다음, 옥션 등 많은 기업에서 Open API를 공개하고 있다.

한편, Open API 포털 사이트들 중 가장 유명한 사이트인 ProgrammableWeb.com에서는 API들에 대한 프로토콜 타입별 그 분포를 보여주고 있다. (그림 1)에서 알 수 있듯이 제공되는 API들의 약 2/3(72%)가 REST이고, 18%가 SOAP, 6%는 JavaScript, 3%가 XML-RPC, 그리고 Atom은 0%다. 따라서 본 논문에서 취급하는 Open API 프로토콜 타입은 REST, SOAP, JavaScript, 그리고 XML-RPC로 한정할 수 있다.



(그림 1) Open API 프로토콜 타입 분포

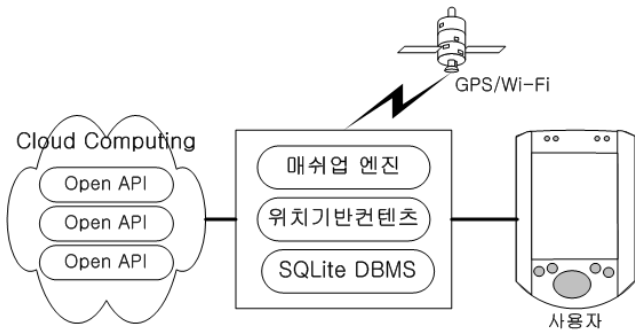
4. 위치기반 모바일 매쉬업 개발

본 논문에서 개발한 위치기반 모바일 매쉬업은 장소 검색 매쉬업 서비스로서 사람들이 스마트폰을 통해서 이동하면서 원하는 장소와 정보를 찾을 수가 있다. 사람들은 스마트폰 GPS 모듈을 이용하여 현재의 위치를 알려주고 구글맵 상에 그 사람의 현재 위치의 건물, 설명 정보를 업로드 할 수 있다.

4.1 시스템 구성도

(그림 2)는 제안하는 위치기반 모바일 매쉬업 서비스의 전체적인 구성도이다. 기존 모바일 매쉬업 서비스들의 문제점으로는 화면 로딩 시간이 늦고 서비스의 요청이 빈번

하게 발생하는 경우 웹 서버에 과도한 병목현상이 발생하는 것이다. 이는 데이터 량이 많아질수록 더 심각한 문제를 야기한다. 본 논문에서는 이러한 스마트폰 제약사항을 극복하기 위해 클라우드 컴퓨팅 개념을 접목시켰다. 스마트폰에서는 정보 요청 및 결과 디스플레이 등 간단한 로직만 처리하고 실제 서비스는 웹 상의 Open API들로부터 이루어지도록 한다.



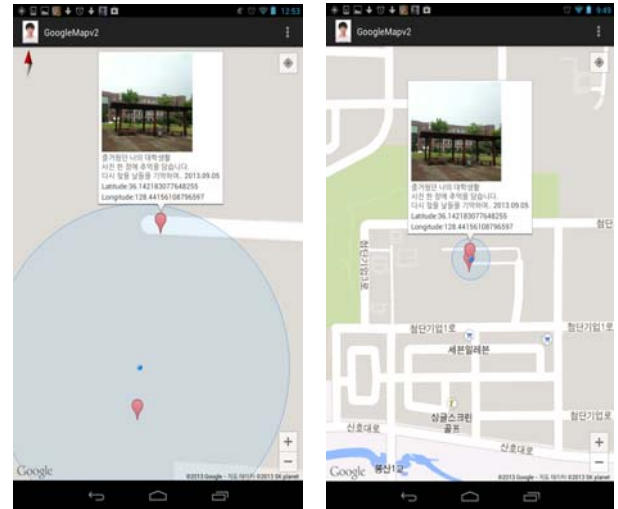
(그림 2) 클라우드 기반 매쉬업 아키텍처

안드로이드에서 위치기반서비스 기능은 주로 두 가지 API를 활용한다. 하나는 맵 API인 com.google.android.gms.maps이며, 다른 하나는 위치 API인 android.location이다. 안드로이드 맵 API는 맵을 표시하고 조작할 수 있는 기능들이 들어 있으며 본 논문에서는 구글의 새로운 버전인 GoogleMap API V2를 사용한다. 위치 API는 GPS나 무선랜 등의 정보를 이용하여 현재 위치 정보를 얻기 위한 기능이 포함되어 있다. 안드로이드폰에서 구글맵의 현재 위치를 표시하기 위해서는 LocationManager와 LocationListener 객체 설정이 필요하며, GPS를 통해 수신한 위치 정보가 변경될 때 마다 현재 위치 값이 Location 객체로 전달된다.

GoogleMap API V2에서는 안드로이드 Fragment 확장자인 MapFragment 클래스를 선보이면서 스마트폰과 같은 작은 화면과 태블릿과 같은 큰 화면의 복잡한 UI를 동시에 지원할 수 있는 유연한 화면을 만들 수 있다. 또한 V2에서는 벡터 타일을 사용함으로써 맵이 더욱 빠르고 더욱 작은 bandwidth를 사용하며 3D도 지원한다. 본 논문에서는 Google Place API로부터 받아온 장소 정보를 구글맵의 디스플레이 경계에 포함되는지 조사하여 경계 값 안에 표시되는 마크들과 경계 값 밖에 표시되지 않은 마크들을 분리하여 처리한다(그림 3.a).

단계별 흐름은 GPS로부터 위치 정보를 수신 받는다. 이 위치 정보는 바로 사용하기는 힘들기 때문에 구글 API와 통신을 거치게 되며 구글맵 상에 자신의 위치를 나타내는 정보를 받게 된다. 만일 사용자의 현재 위치해 있는 곳의 건물, 설명 등의 정보를 웹 서버에 업로드 하려면 카메라 연동 Intent를 정의하고 결과를 호출하여 SQLite 데이터베이스에 사진, 텍스트 설명 등을 저장한다. 이 정보는 웹

서버에 제공되어 다수의 사용자들이 공유할 수 있다(그림 3.b).



(a) 디스플레이 경계 (b) SQLite 사진과 설명

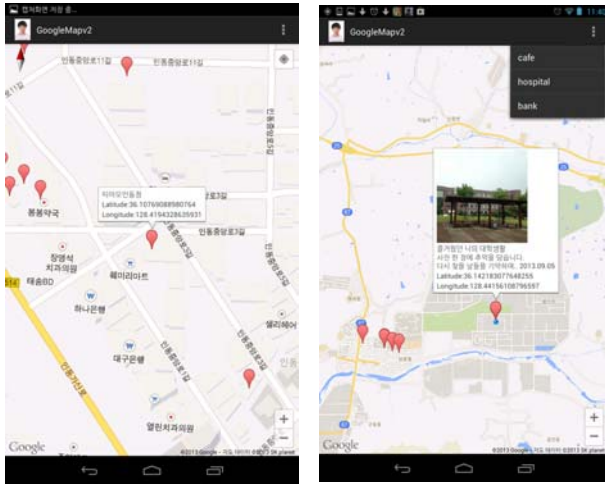
(그림 3) 경계 표시 및 SQLite 저장

안드로이드 SQLite는 데이터베이스 생성 및 오픈 처리를 담당하는 SQLiteOpenHelper 도우미 클래스를 제공하는데, 이 클래스를 생성하면 데이터베이스가 필요한 시점에 onCreate, onUpgrade, onOpen 메서드들이 호출되고 onCreate 메서드에서 처음 데이터베이스가 생성된다. 본 DB 테이블은 T_Name(id, latitude, longitude, image, subtext)로 구성되어 있으며 데이터의 추가, 삭제, 수정 처리 메서드가 구현되어 있다. 또한 내장 카메라로 찍은 사진과 위치 정보를 데이터베이스로부터 검색·저장·관리하는 모듈을 구현하였다.

안드로이드폰을 이용한 위치기반서비스 개발에서는 고려해야 할 요소들이 많이 있다. 특히, 지하나 건물 내에서는 수신이 잘 되지 않고 GPS 사용으로 인한 배터리 소모가 많다는 단점이 있으며, 현재 위치와 해당 위치정보에 대한 정밀도가 높은 빠른 탐색이 요구된다. 현재 안드로이드에서 제공하는 위치 제공자는 크게 GPS, Network, Passive의 3가지로 나누어져 있지만 GPS가 가장 현실적이고 정밀도가 높은 방법 중에 하나이다. 그러나 배터리 소모량이 크고 건물 내에서는 전파를 수신하지 못하는 한계가 있으므로 본 시스템에서는 건물 내에서는 Wi-Fi를 활용하여 정확한 위치를 획득하고 배터리 소모량을 줄일 수 있도록 한다. 이를 위해 사용자가 직접 위치 제공자를 선택할 수 있도록 설계하였다.

구현된 위치기반시스템의 전체 작업 흐름은 다음과 같다. 먼저 GPS를 중심으로 위치정보를 수신 받는다. 이는 구글 맵 API에서 사용자 위치 주변의 다양한 건물들을 검색하고 위치를 확인하는데 사용된다. 사용자가 검색 메뉴를 선택하면 현재 위치에 있는 주변의 카페, 병원, 은행

등 장소 정보를 HttpClient 클래스를 이용한 통신 모듈을 통해 이들을 찾아 화면에 마크들을 디스플레이 해 준다 (그림 4.a). 이때 원하는 곳을 터치하면 맵 상에 자세한 관련 정보를 보여 주고(그림 4.b), 이동경로 메뉴를 선택하면 이용자의 이동 경로가 추적된다. 본 시스템의 작동 스텝의 일부분은 다음 그림과 같다.



(a) 주변 장소 찾기 (b) 관련정보 디스플레이

(그림 4) 안드로이드폰 위치기반시스템

5. 결론

본 논문에서는 Open API를 이용한 스마트폰 위치기반 모바일 매쉬업을 개발하였다. 이는 기존 모바일 매쉬업의 문제점인 늦은 로딩 시간과 과도한 트래픽 병목 현상을 해결하기 위해 클라우드 컴퓨팅 개념을 접목시킨 것이다. 매쉬업의 활용은 현재 계속 그 영역이 확장되고 있으며, 특히 모바일 매쉬업은 향후 발전 가능성이 큰 서비스임에 틀림없다. 그러나 현재 국내의 매쉬업 개발은 아직 초보 수준에 머무르고 있으며 매쉬업 기술 또한 체계적인 연구가 되어있지 않은 상황이다. 이러한 문제점들을 보완한다면 앞으로 매쉬업은 더욱 각광받는 미래 서비스 기술이 될 수 있을 것이다. 본 연구의 향후 과제로는 공개된 Open API들과 스마트폰 자체 콘텐츠 자원을 섞어서 매쉬업을 만들 때, 이때 발생할 수 있는 보안 문제에 대한 더욱 깊은 연구가 필요하다.

참고문헌

[1] <http://www.housingmaps.com>
 [2] 고윤미, 오기남, 권경희, "GPS와 Open API를 이용한 모바일 일정관리 매쉬업 서비스 구현," 한국컴퓨터종합학술대회 논문집, Vol. 35, No. 1(D), pp. 281-284, 2008
 [3] 방태호, 양정진, "매쉬업 개념을 적용한 스마트폰 기반의 서비스 모델에 관한 연구," 한국컴퓨터종합학술대회논문집, Vol. 38, No. 1(B), pp. 140-142, 2011
 [4] 김병수, 김종훈, "구글맵기반 사용자 참여형 안드로이드폰 GIS 2.0 응용프로그램 개발," 한국컴퓨터교육학회 논문지,

제14권, 제4호, pp. 11-20, 2011

[5] S. Cho, H. Kim, D. Jung, and H. Park, "Dynamic Mashup Platform for Mobile Web Applications," Digest of Technical Papers International Conference on Consumer Electronics, pp. 1-2, 2009
 [6] E. M. Maximilien, "Mobile Mashups: Thoughts, Directions, and Challenges," In Proceedings of the IEEE International Conference on Semantic Computing, pp. 597-600, 2008
 [7] L. Jin, M. Song, and J. Song, "Mobile Mashup Architecture Solution, Direction and Proposal," In Proceedings of the IEEE 2nd Symposium on Web Society, pp. 698-704, 2010
 [8] Z. Peng, H. CHen, J. Rao, Y. Liu, L. Wang, and J. Chen, "Semantic-based Mobile Mashup Platform," The 9th International Semantic Web Conference, Posters & Demonstrations Track, 2010