

안드로이드 기반 동적 위치관리 기능 구현

김민정*, 박영호*

*숙명여자대학교 멀티미디어학과

e-mail : alswjd8814@nate.com

yhpark@sookmyung.ac.kr

An Implementation of Dynamic position sensing and display capabilities based on-Android Platform

Min-Jung Kim*, Young-Ho Park*

*Dept of Multimedia Science, Sookmyung University

요 약

최근 안드로이드 기반 어플리케이션 시장이 활성화됨에 따라 그중 가장 주목받고 있는 기술인 LBS에 대한 최근 동향과 관련연구를 소개한다. 이후 ‘위치기반 동적 위치감지 및 표시기능’이 가능한 보이스 채팅 어플리케이션을 개발하고, 이에 사용된 LBS이용 부분의 구현 내용 및 알고리즘을 소개하고자 한다. 현재 위치를 동적으로 감지하여 실시간으로 채팅방 생성이 가능하며 그 채팅방에 접속한 제한된 멤버간의 보이스 채팅이 가능한 SIC는 사용자들로 하여금 관심사가 같은 사람들 간의 간편한 의사소통을 가능하도록 제공하여 기존보다 손쉽고 간편하며 친근한 커뮤니케이션을 제공한다.

1. 서론

최근 안드로이드 기반 스마트 폰의 보급이 확대됨에 따라 모바일 어플리케이션 시장의 크기가 갈수록 커지고 있다. <그림1>은 모바일 플랫폼들의 최근 시장 점유율로서 그중 안드로이드는 2011년 1월 시점에서 31.2%를 기록하며 1위가 된 것을 확인할 수 있다. 또한 리서치 전문그룹인 Gartner사에서 오는 2012년을 이끌 모바일 애플리케이션 10대 기술로 위치기반서비스(LBS, Location Based Service)와 소셜 네트워킹(SNS, Social Network Service) 등을 손꼽았다.[2]

LBS란 이동통신사의 통신망이나 위성항법장치(GPS) 등을 통해 얻은 위치정보를 바탕으로 사용자에게 여러 가지 서비스를 제공하는 시스템이다. LBS 관련 매출도 지난해 9억9800만 달러에서 올해 22억 달러 수준으로 역시 두 배 이상의 성장이 접혀졌다.[1] 이와같이 LBS는 현재 가장 주목받으며 급속도로 성장하고 있는 기술이다.

해외에선 이미 다양한 LBS서비스(교통, 지도, SNS 등과의 결합)가 나와 있지만 국내에선 이제 걸음마 단계에 불과하다. 국내에서도 LBS와 다양한 서비스를 결합한 어플리케이션의 개발이 시급한 실정이다. 그 중 동적으로 위치를 감지하여 지도에 표시하는 기능을 이번 논문에서 소개하고자 한다. 이 논문은 앞으로 다방면에서 사용될 LBS이용에 많은 도움을 줄 것이다.

SIC(Space at ubiquitous Immediate communication based on CCG*)는 폐쇄형 채팅그룹 기반의 동적 위치감지 및 표시기능을 가진 보이스 채팅 어플리케이션으로 LBS기반의 커뮤니케이션 시스템이라고 볼 수 있다. 현재 LBS를 이용한 어플리케이션중 널리 활용되고있는 어플리케이션들은 일방적인 의사 소통 형태만을 가미한 SNS형

식이 대부분이다. 그러나 SIC는 실시간으로 생성 및 삭제되는 CCG와 멤버들의 위치를 google map을 통해 확인가능 하며 자신의 현재위치에 실시간으로 CCG를 형성하여 그룹 내의 멤버들 간에 보이스 채팅기능을 제공하여 새로운 커뮤니케이션 장르를 시도하였다. 이 논문은 앞으로 다양한 시스템에 결합되어 사용될 LBS를 구현하는데 도움이 되기 위한 논문이다.

Top Smartphone Platforms 3 Month Avg. Ending Jan. 2011 vs. 3 Month Avg. Ending Oct. 2010 Total U.S. Smartphone Subscribers Ages 13+			
	Share (%) of Smartphone Subscribers		
	Oct-10	Jan-11	Point Change
Total Smartphone Subscribers	100.0%	100.0%	N/A
Google	23.5%	31.2%	7.7
RIM	35.8%	30.4%	-5.4
Apple	24.6%	24.7%	0.1
Microsoft	9.7%	8.0%	-1.7
Palm	3.9%	3.2%	-0.7

<그림1> 안드로이드 시장 점유율[1]

2. 관련연구

이 장에서는 위치기반을 사용하는 시중의 대표적인 어플리케이션 두 가지를 소개하고자 한다.

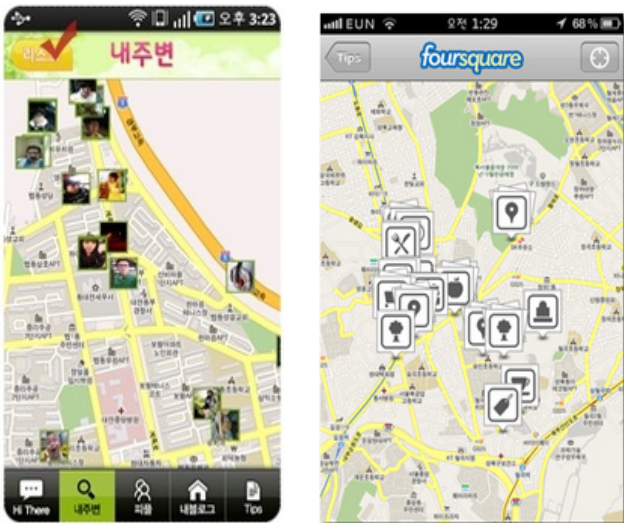
2.1 Hi,There

Hi,There 은 LBS+SNS형식의 안드로이드, 아이폰 어플리케이션이다. 자신의 블로그를 생성할 수 있고 위치기반 서비스를 이용하여 모든 접속자들의 위치확인이 가능하며 다양한 친구의 블로그를 구경 및 쪽지보내기 댓글달기가

가능하다. Hi,There에서 제공하는 기능은 다음과 같다. 첫째, 실시간으로 사람들이 올리는 talk(블로그의 글을 의미)을 통해 그 사람의 블로그를 볼 수 있고 그 talk에 댓글을 달거나 그 사람에게 쪽지보내기가 가능하다. 둘째, '내 주변'을 tap하면 <그림2>는 관련연구 어플리케이션 화면 예시이다. 현재위치를 중심으로 Hi,There에 접속중인 사람들을 지도로 확인 가능하다. 그러나, 블로그의 글 게시를 통한 일방적인 의사전달에 불과하며 이로 인해 실시간적인 의사소통은 불가능하다. 또한 내가 원하지 않는 사람들에게도 나의 신상이나 현재위치를 공개해야하며 상대방의 신분확인이 불가능한 상태에서 의사소통을 하게 되는 단점이 존재한다.

2.2 foursquare

foursquare[4]은 LBS+SNS 형식의 어플리케이션으로 아이폰,안드로이드,블랙베리,윈도우모바일 등 다양한 플랫폼에서 지원하고 있다. 스마트 폰을 이용하여 내가 위치하고 있는 장소에 발자취를 남길 수 있는데 그것을 'check-in' '한다는 표현을 사용하며 그 장소에 대한 다양한 정보들을 습득할 수 있다. 국내에 2010년에 들어왔으며 약 5만명의 회원 수를 가지고 있다. 그러나 이 어플리케이션 역시 일방적인 의사표현을 할 수 있도록 하며 의사소통은 불가능하다.



<그림 2> 관련연구 어플리케이션 예시

LBS를 제공하는 어플리케이션 중 실시간 커뮤니케이션이 가능한 어플리케이션은 존재하지 않았으며, 실시간 커뮤니케이션이 가능한 어플리케이션은 LBS와 관련없는 인터넷 전화기능이나 무전기 시스템 등에 불과하였다. 이 기능들은 네트워크의 제약이 있어 사용에 불편함이 있거나 상대방의 상황을 고려하지 않는 push형태의 일방적 의사 전달 시스템, 혹은 사용에는 제한이 없지만 상대방을 선택하지 못하고 무분별하게 채널 안에서 talk하여 실시간 커뮤니케이션에는 한계점이 존재하였다.

3. 위치기반 보이스채팅 SIC 어플리케이션

본 장에서는 본 논문이 제안하는 시스템의 개요와 그

시스템의 LBS 관련부분 설계에 대해 설명하고 전반적인 기능별 알고리즘을 소개한다.

3.1 개요

SIC는 기존의 단순 커뮤니케이션 어플리케이션(카카오톡, 네이트온 등)과 LBS 시스템을 접목시켜 동적 위치감지 및 표시기능을 통한 위치기반 실시간 보이스 채팅을 지원한다. 현재위치를 기준으로 CCG를 형성하고 그 CCG에 접속한 멤버끼리 보이스로 대화를 나눌 수 있다.

사용자(클라이언트)는 회원 가입, 수정, 삭제가 가능하며 CCG를 형성 및 관심사별로 모여 다른 사용자들과 대화를 나눌 수 있다. CCG에 입장하는 멤버를 제한하기 위해 비밀번호 설정도 가능하다. 화면의 지도상에서 현재 생성되어져 있는 CCG를 확인가능하며 CCG 입장 후에는 입장한 멤버들의 위치도 주기적 위치감지 및 표시기능을 통해 위치확인이 가능하다. 이 SIC 커뮤니케이션 시스템은 손을 사용하지 않고도 의사소통이 가능하며 입장멤버를 제한시킬 수 있다는 점에서 모바일 기기 상에서 원활한 의사소통이 가능하도록 한다.

3.2 LBS 적용부분 설계

본 절에서는 SIC 어플리케이션의 LBS관련 구현부분의 설계내용과 알고리즘에 대하여 설명한다.

3.2.1 Google Map API 적용

SIC는 Google Map API를 이용하는데 이를 위해서는 Google 계정이 필요하며 각 컴퓨터마다 다른 Google Map API Key를 발급받아야 한다. 이 key를 받기 위해서는 cmd창에서 Java SDK의 keytool을 실행시켜 MD5문자열을 생성한 후 그것을 google map key 인증사이트[5]에 입력하면 API key를 발급받을 수 있다. <그림3>은 google map 사용시 androidmanifest.xml설정 부분이다. 위치의 정확도와 map사용에 관한 permission을 설정한다. fine은 더 섬세한 정확도를 제공하며 coarse는 fine보다 조금 부정확한 정확도를 제공하나 둘 다 permission을 해두면 좀 더 정확한 정보를 얻을 수 있다. 또한 uses-library에서 <application> element안에 map library 를 작성하는 점이 중요하다. 이후 적용되는 화면의 xml파일에 <그림4>에서 볼 수 있듯이 google map API key를 입력하면 google지도 사용이 가능하다.

```
<uses-library android:name="com.google.android.maps" />
</application>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

<그림3> google map 사용시 androidmanifest.xml

```
<com.google.android.maps.MapView
    android:id="@+id/roomMapView" android:layout_width="fill_parent"
    android:layout_height="350px" android:enabled="true"
    android:clickable="true" android:apiKey="0g98dQO2TzOdIHHVlc98UJLcQk1SVbKp82vSw" />
```

<그림4> xml 소스에 google map key사용시 설정

3.2.2 CCG 모니터링 기능

본 절에서는 현재 개설된 CCG의 정보들과 나의 위치를 파악하여 marker표시하는 기능에 대하여 설명한다.

SIC로그인 후 보여 지는 지도에는 현재 시점에서 내 위치를 중심으로 개설되어져 있는 CCG의 팻말들을 확인 가능하다. 팻말에는 각 CCG의 방명이 적혀있고 팻말을 tap하면 그 CCG에 입장중인 현재 인원수/총 인원수 와 인사말 그리고 입장하기 버튼을 확인할 수 있다.

mapActivity에서 반복적으로 사용되는 함수들(mapView 사용을 위한 기본설정함수, 위치변경시 오버레이 삭제하는 함수, 위치값을 GeoPoint값으로 변환하는 함수, location manager사용을 위한 criteria설정(정확도, 고도 사용 유무, 비용 사용정도, 배터리 사용정도 설정 및 그 설정에 따른 provider 설정), location listener 클래스 (위치이동, 위치 탐색불가, 위치탐색가능 등 각 상황에 대해 자동 반응하는 클래스)) 구현하였다.

Location Manager 객체를 이용하여 위치감지service를 받을 수 있도록 설정한다. 이후 communitymain helper의 함수를 이용하여 위치탐색 시 설정할 criteria와 provider를 설정하고 현재의 위치정보를 GeoPoint값으로 받는다. 이후 MyPositionOverlay라는 overlay클래스의 객체를 생성하여 현재위치에 me라는 이미지를 marker표시하여 사용자의 현재위치를 보여지도록 제공한다. <그림5>는 현재 개설된 CCG들의 정보들 for문을 이용해 받아오는 소스이다. 서버의 database에 접속하여 각 CCG의 정보(CCG명, CCG가 생성된 위치, 접속중인 인원, 총 접속가능 인원, 인사말)를 받아와 My Position Overlay를 통해 해당위치에 marker를 표시해 주고 그 위에 itemizedOverlay를 이용하여 CCG명과 tap할 시에 발생하는 event를 설정해 준다. overlay를 두가지로 나누어 겹쳐지게 하는 이유는 MyPositionOverlay만으로는 onClick event 설정이 불가능하기 때문이다.

```
for (int i = 0; i < listCount; i++) {
    roomName = chatroom_hm.get("comName["+ i + "]");
    roomMember = chatroom_hm.get("comMember["+ i + "]");
    ovly_latitude = (double) Double.parseDouble(chatroom_hm
        .get("comLatitude["+ i + "]")) * 1000000;
    ovly_longitude = (double) Double.parseDouble(chatroom_hm
        .get("comLongitude["+ i + "]")) * 1000000;
    roomPwd = chatroom_hm.get("comPwd["+ i + "]");
    roomId = chatroom_hm.get("comId["+ i + "]");
    roomDes = chatroom_hm.get("comDescription["+ i + "]");

    GeoPoint gPoint = comhelper.updateWithNewLocation(ovly_latitude,
        ovly_longitude);
    MyPositionOverlay com_loc = new MyPositionOverlay(gPoint, roomName);
    mapOverlays.add(drawComLocation(roomId, roomName, roomMember,
        roomPwd, ovly_latitude, ovly_longitude, act,
        itemizedOverlay, roomDes));
    List<Overlay> overlays = myMapView.getOverlays();

    com_loc.draw(canvas, myMapView, false);
    overlays.add(com_loc);
}
```

<그림5> listCount값을 이용한 for문 구조

3.2.3 CCG 입장하기

본 장에서는 개설된 CCG중 하나를 선택하여 입장한 후

그 안의 멤버들의 위치를 표시하는 부분을 설명한다.

입장하기 버튼을 통해 그 CCG에 입장하게 되면 보이는 지도에는 현재 입장해 있는 CCG멤버들의 이미지팻말을 지도에서 확인 가능하다. CCG 입장시 넘겨받은 room Name 과 roomTotal(총 입장가능 인원수)을 화면의 지도 위에 setText한다. CCG의 id값을 받아와 자신의 정보 DB에 CCG방 id를 업로드한다. 이후 방id를 통해 현재 이 CCG에 접속중인 멤버의 인원수와 별명을 각각 DB에서 받아온다. 이 별명은 지도상에 멤버이미지를 marker로 표시하였을때 그 marker를 tap시에 보여지게 된다. 여기에서도 마찬가지로 이미지marker를 위한 My Position Overlay와 onTap시 적용되는 event를 위한 itemized overlay를 다르게 사용했는데 그 이유는 OnClick -event를 위해서이기도 하지만 inner-class이어야 event를 설정할 수 있기 때문이기도 하다. 또한 chatroom의 가장 큰 특징은 주기적인 동적 위치감지 및 표시가 가능하다는 점이다. <그림6>은 timer을 이용하여 5초당 자동적인 새로고침을 하는 소스이다. run 함수내의 update()함수는 멤버중 위치가 변경된 사항이 있을 경우 marker를 모두 지운 후 새로 marker한다.

```
timer_sec = 0;
count = 0;

second = new TimerTask() {

    @Override
    public void run() {

        Log.i("Test", "Timer start");
        Update();
        timer_sec++;

    }

};
timer = new Timer();
timer.schedule(second, 0, 5000);
```

<그림6> run함수를 이용한 동적 위치감지 timer

3.3 알고리즘 분석

<그림7>은 사용자가 SIC에 입장하는 순간 작동하는 현재위치 감지 알고리즘 이다. LocationManager 객체를 생성하고 criteria설정을 통해 provider를 지정받게 된다. 이후 그 provider를 통해 현재 위치값을 GeoPoint값으로 받아 overlay시 사용하게 된다. 또한 LocationListener는 inner클래스로 현재위치가 감지되지 않거나, 변했을 경우 등의 상황을 위해 상시 작동되고 있다.

```
# Algorithm for check My Position
Input : LocationManager L, Criteria C, Provider P, Location L
Output : Geopoint

Algorithm Start :
Step 1. L생성 및 location service 가능하게 설정
    L에 C 생성 및 비용, 에너지, 정확도 등을 설정
    C를 토대로 getBestProvider을 이용하여 P 설정
    L을 P를 사용하여 Current Location 찾음
    return Geopoint

END
```

<그림7> 사용자의 현재위치 감지 알고리즘

<그림8>은 사용자가 SIC에 입장하여 'CCG보기'화면으로 입장 한 경우 화면에 보여지는 CCG정보를 표시하는 과정을 설명하는 알고리즘이다. 'CCG보기'를 누름과 동시에 사용자의 현재위치를 감지하여 화면에 나타나는 지도 중심에 사용자의 위치가 표시되고 AllChatRoom함수를 통해 현재 개설되어있는 CCG의 모든 정보를 가져와 CCG명은 marker로 사용하고 나머지 정보는 marker를 ontap할 시에 보여지도록 한다.

```
# Algorithm for show CCG
Input : Current my location C
Output : CCG contents contents
        menu M

Algorithm Start :
Step 1. 'CCG 보기' 메뉴 입장
        C감지 및 DB에 update
        현재 개설된 CCG정보를 for문을 이용해 받음
        화면에 MyPositionOverlay.itemizedOverlay
        Return contents
Step 2. 'refresh'버튼
        현재위치 재 감지 및 DB에 update
        현재시점 개설된 CCG정보를 for문을
        이용해 받음
        화면에 MyPositionOverlay.itemizedOverlay
        return contents
Step 3. 'back'버튼
        메뉴화면으로 돌아감
        return M
END
```

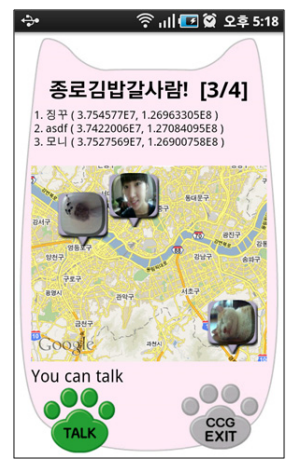
<그림8> CCG 개설정보 지도에 표시 알고리즘

4. 실행 결과

<그림9>는 SIC에서 구현한 'CCG보기' 결과화면이다. 구글맵 API를 통해 개설된 모든 CCG의 위치 및 정보를 표기하여 관심있는 유저들의 참여를 유도하고 지도 확대/축소(ZoomController)를 통해 CCG를 더 수월하게 찾기가 가능하도록 한다. CCG명을 클릭하면 입장이 가능하다. 또한 내가 CCG를 만드는 경우에는 Location Manager를 통해 알게 된 자신의 위치를 기반으로 CCG를 개설가능 하며 현 GPS 측정값을 Geo coder를 통해 주소형태로 변환하여 자동 입력하도록 되어있다. 개설된 CCG에 모든 사용자가 퇴장할 경우, 자동으로 CCG는 폐쇄된다. <그림 10>은 SIC에서 구현한 'CCG입장하기' 내부 화면이다. CCG에 입장하게 되면 그 CCG의 멤버의 위치가 표시되어 있고 멤버가 나가거나 새로운 멤버가 들어올 때, 그리고 멤버의 위치가 변경 되었을 때 자동으로 지도가 갱신된다.



<그림 9> 'CCG 보기'



<그림 10> 'CCG 입장하기'

5. 결론

본 논문에서는 보이스 채팅 어플리케이션(SIC)에서 LBS 관련 기능의 구현 방안에 대해서 소개하였다. LBS와 communication 시스템을 적절히 혼합한 SIC는 LocationListener과 LocationManager를 사용한 위치감지 시스템 그리고 Itemized Overlay, MyPositionOverlay를 사용한 사용자의 위치 marking 서비스를 통해 사용자는 SIC를 사용하는데 있어서 재미를 얻게 되고 좀 더 새로운 방식의 의사소통이 가능하다. 향후 연구로는 본 LBS 기능을 이용한 다양한 응용 프로그램의 구현을 시행할 예정이다.

참 고 문 헌

[1] 스마트폰 시장, 안드로이드 동향
<http://blog.naver.com/mamaiu?Redirect=Log&logNo=80125993658>
 [2] Gartner 그룹 선정 내년 10대 모바일 어플리케이션
<http://www.etnews.co.kr/201102110114>
 [3] Hi, There application 소개
<http://ydk012486.blog.me/80123109718>
 [4] foursquare application 소개
<http://blog.naver.com/spild2457/120120779558>
 [5] Google map key 인증
<http://code.google.com/intl/ko/android/maps-api-signup.html>