華中科技大學

本科生毕业设计（论文）参考文献译文本

译文出处：

Sumi Helal，Raja Bose，Wendong Li.

Mahadev Satyanarayanan, Carnegie Mellon University.

Mobile Platforms and Development Environments. 2012.

Page 55-72

院　　系　计算机科学与技术

专业班级　1201 班

姓　　名　王玉龙

学　　号　0121212370329

指导教师　孙伟平

2015 年 9 月

译文评阅

导师评语

评分：_____（百分制）　　指导教师（签名）：_____

年　　月　　日

第六章

内嵌平台：基于地理定位的服务

6.1 透视历史

基于定位服务的主要起源是 1996 年美国政府授权通过的 E911 委任书。这个授权书是为了移动网络用户对紧急呼叫者在给定精度下的定位，以便于操作者可以发送呼叫者的定位信息到公共安全应变中心。蜂窝技术不能满足那时的精度需求，所以操作者开始花费无穷的努力去介绍先进的定位方法。来得到在 E911 项目中的投资回报，经营者们建立了一系列的商业化基于定位的服务。在大多数情况下，它们包括寻找服务，对于需求的用户发送给他们周边有趣的地点，像餐厅或者是加油站。然而，多数用户对这种 LBS 并不感兴趣，所以许多经营者很快终止了他们的 LBS 服务和相关发展的努力。在 2005 年的时候又刮起了 LBS 风潮，这次正当时机而且方向正确，一些重大的发展和良好的环境使 LBS 复活了。

伴随着可用 GPS 移动设备的出现，网络 2.0 规范的来临，和新生 3G 宽带无线网络服务进入发展时期，同时，小型软件和硬件厂商意识到 LBS 技术在巨大市场和小型市场上的广泛可应用性，以及新一代 LBS 技术规范的建立。大多数重大发展和 LBS 里程碑事件的简短历史的时间线可以参见图 6.1

6.2 LBS 技术的演变

最早的 LBS 是需要用户的初始请求来做出反应的。他们是自参照的并且是单目标的，意味着只关心一个移动用户的定位信息。他们以内容为主导，只提供基于定位信息。最早的 LBS 设备是以用户为中心并且是特有的。网络用户控制服务限定在直接或间接内容提供者合作伙伴之间。例如，威瑞森通讯公司最早的 LBS 服务就为微软 MSN 的合作伙伴提供过短时间的服务。

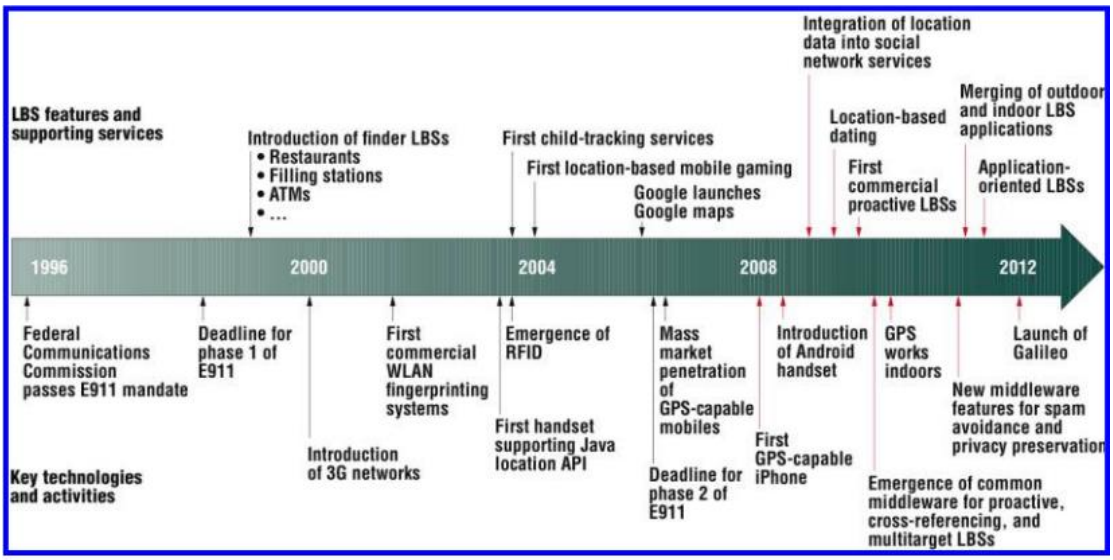

Figure 6.1: A brief history of Location-Based Services (Courtesy IEEE Pervasive Computing April-June 2008).

图 6.1 基于定位服务的简史

在 2004 年，经营者们和其他的供应商们开始为船队管理和追踪小孩和宠物的踪迹提供服务，这是第一个 LBS 技术交叉应用的一个案例。这些服务的初始版本是使用三角技术的基于单元识别号的定位，遭遇了定位精度低并且很快的被 GPS 所取代。伴随着 GPS 功能的出现和移动电话的可编程性（例如，java 2 微型可编辑电话和随后出现的 iPhone），有远见的用户开始写一些小型的应用程序发送定位信息到中心服务器，使得他们的位置信息可以被其他用户所知道。很快的，这些最初的创造转变成了创造广泛的积极性和综合目标服务的专业型商业，例如向移动游戏，市场营销和健康咨询提供服务。这些发展伴随着 Web 2 在社交网络成员之间进行内容交换的定位技术的发展，就是当今复合目标 LBS 技术位置信息共享基础功能的前身。

随着移动平台可编程技术的发展，GPS 技术的发展，WiFi 渗透到了全世界，第三方基于地理定位应用在成熟环境下的激增。除了低功率的 GPS 接收机（例如辅助 GPS 定位技术），还有由蜂窝技术和 WiFi 三角网覆盖组成的地理定位技术，使得在多数变化的时间中和可变定位精度下的定位信息成为可能。移动平台地图的涌现进一步加速了 LBS 技术的扩展，出现了成百个像现今这样基于地理定位/地图富有活力的应用。

回顾起来，网络经营者在以 LBS 技术为中心的发展似乎成为了它的主要限制，这也是早期失败的原因。第三方驾驭 LBS 发展的实施性将 LBS 技术的拥有权转移到了终端用户。许多应用提供积极主动地服务，根据地理定位背景追踪用户的踪迹来提供服务。此外，许多 LBS 提供以人为基础的社会影响服务，和以基于内容的开发形成对比。图片 6.2 描绘了 LBS 家族

通过积极的，以公众为导向的，以用户为中心的大爆炸式发展。

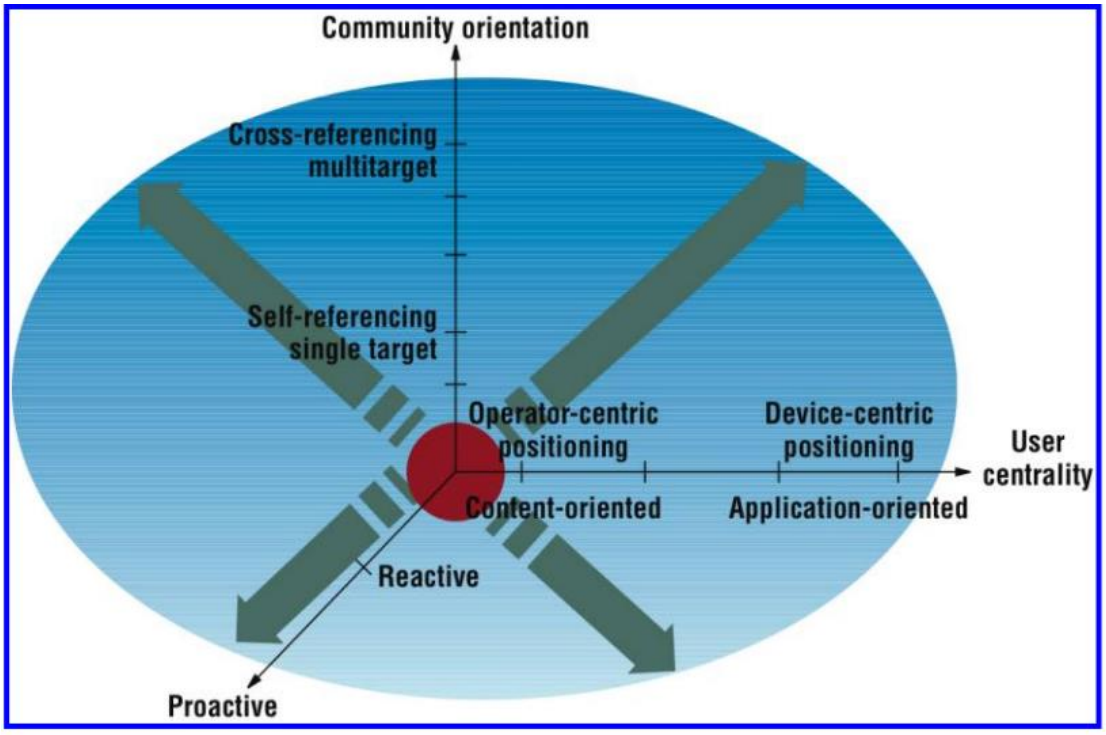

Figure 6.2: Location-Based Services evolution (Courtesy IEEE Pervasive Computing April-June 2008).

图 6.2 基于定位服务的进化

6.3 绘制全世界

交互式的数字化地图，对 LBS 用户看来拥有最有力且最自然的界面。在地图上通过极简的多点触控方式来表达一个目标的定位。找到了一种没有比近距离圆更好的表达用户现在在地图位置的方式。对于今天的移动平台，移动地图成了 LBS 技术必须的一部分。

6.3.1 户外地图

我们现在所习惯的世界导航电子地图（网络和移动）可追溯到 NAVTRQ 公司，是一家在地理地理信息系统和电子地图处于主导地位的公司。NAVTEQ 成立于 1983 年，拥有一个特殊商标的名字，并且在 2007 年全部称为诺基亚公司的一个附属机构。NAVTEQ 主要生产便携式 GPS 导航设备，许多基于网络的地图应用（包括雅虎地图，MapQuest 和必应地图），和其它移动端地图（诺基亚地图，必应微软手机地图，和之后（2011）的 iOS 地图），甚至谷歌在 2004 年转换之前也是用的是 NAVTEQ 地图，之后生产了它自己的地图资源。

Tele-Atlas，一个在 1984 年成立的荷兰公司，是 NAVTEQ 的主要竞争对手。在 2008 年，在诺基亚获得 NAVTEQ 之后，谷歌-接着的一个对于诺基亚在移动平台领域的新竞争对手——决定舍弃 NAVTEQ 而转换到 Tel-Atlas。在谷歌开始在美国本土从事快速生产自己的地图之后，这种转换持续不到一年的时间。并且在 2008 年，Tele-Atlas 被 Tom-Tom 所收购——一个

便携式 GPS 导航公司——一个只能限制的将 Tele-Atlas 应用到 Tom-Tom 自己和与之相随的一小部分汽车制造商的迁移。

谷歌地图成为主要的在线地图服务的动力在于它的 API，一个可以让第三方将谷歌地图嵌入到他们自己的网站上的接口。这个接口不论是作为 JavaScript 还是网络服务的集合，对于移动网络来说都是可用的。Adobe Flash 接口是被简单支持的，但是在 2011 年之后就陨落了。在 2006 年，谷歌发布了在当时只支持 J2ME 的移动端的谷歌地图。今天，谷歌移动端地图已经支持了许多平台，包括安卓，苹果和黑莓。

当比较诺基亚和谷歌地图移动端地图时可以发现二者在性能和功能上极其相似（例如，POI 视图，搜索，路径计算，交通，卫星图等等）。然而，诺基亚地图在完全离线的移动地图上实现多拐点导航上是开拓创新者。它的基于矢量移动地图最先在塞班操作系统上发布。现在这项支持已经扩充到其它诺基亚平台和 Windows Phone 系统上了。自从 5.0 版本发布后，谷歌地图提供了有多点触控支持的基于矢量 3D 的安卓版地图。拖动两根手指将会倾斜地图，并且扭曲手指会旋转地图，从而使其在 2D 和 3D 视图之间转换。

6.3.2 室内地图

就像是 Woold Is Not Enough 电影里一样（1999 年 james Bond 电影），这个世界的数字化地图（网络上或者移动端）将会随着在室内的地理信息的扩充而变化。眼看着全世界零售业，购物和旅游工业的宏伟发展，室内地图的发展以月以百计的速度被创造着。包括主要的飞机场，购物商场，体育场，度假地和其它复杂建筑的场所。随着室内地图转变成在室内 LBS 服务成为一个竞争对手追逐的新馅饼时，这个步伐将会跳跃到加速状态。

在 2009 年，Micello 在 iOS 移动端导航应用上发布了它的第一期展览馆室内地图。这个地图是通过人工运用公共可用的信息资源制作的，但是不包括在任何时候需要特殊外界资源支持的情况。这些地图和谷歌地图可以很好的叠加，使得用户可以很容易的在户内和户外地图之间转换。

Micello 地图考虑到加入了数据层来使室内子空间可以实现语义描述，搜寻和导航。例如，在商场的一个数据层可以允许搜索询问“男人的鞋子”和“洗手间”。室内商场 Micello 地图在图片 6.3 中被展示。

Figure 6.3: A Micello map of Prien Lake Mall, Louisiana.

图 6.3 Louisiana，Prien Lake 商场的 Micello 地图

Point Inside 是另一家提供室内地图的公司，但是不像 Micello 公司，它的地图目标主要定位在了零售业，通过几种不同的方式把零售业与顾客关联。除了移动应用（包括 iOS 和安卓平台），Point Inside 提供了一个零售平台——一系列的工具使得零售店可以深度分析他们的顾客并且在零售店内提供高度个性化，及时性的服务。

室内地图在 2011 年得到了重大意义的推动，当微软和谷歌官方建立了他们的室内地图并且加入了和游戏开拓者 Micello 和 Point Inside 公司的竞争。必应地图在 2011 年 8 月发布了提供美国一些机场和购物广场的室内地图应用。在 2011 年 8 月，这个地图对于桌面用户和从 Windows Phone 平台开始的移动端上都是可以使用的。必应地图是无缝嵌入并且融合到了必应室外地图中去了，不需要用户的转换动作——只有需要放大看到室内地图的细节时才需动作。在 Windows Phone 平台上的必应地图可以参看图 6.4.

图 6.4 必应移动端室内地图

在必应地图之后，谷歌地图 6.0 安卓版在 2011 年 11 月建成，包括了美国和国外的零售店，商场和国际机场。谷歌地图也是无缝嵌入的。他们也是根据用户所在的层而自动改变。在途 6.5 中展示的是在谷歌地图 6.0 版本前后相同位置的比较。

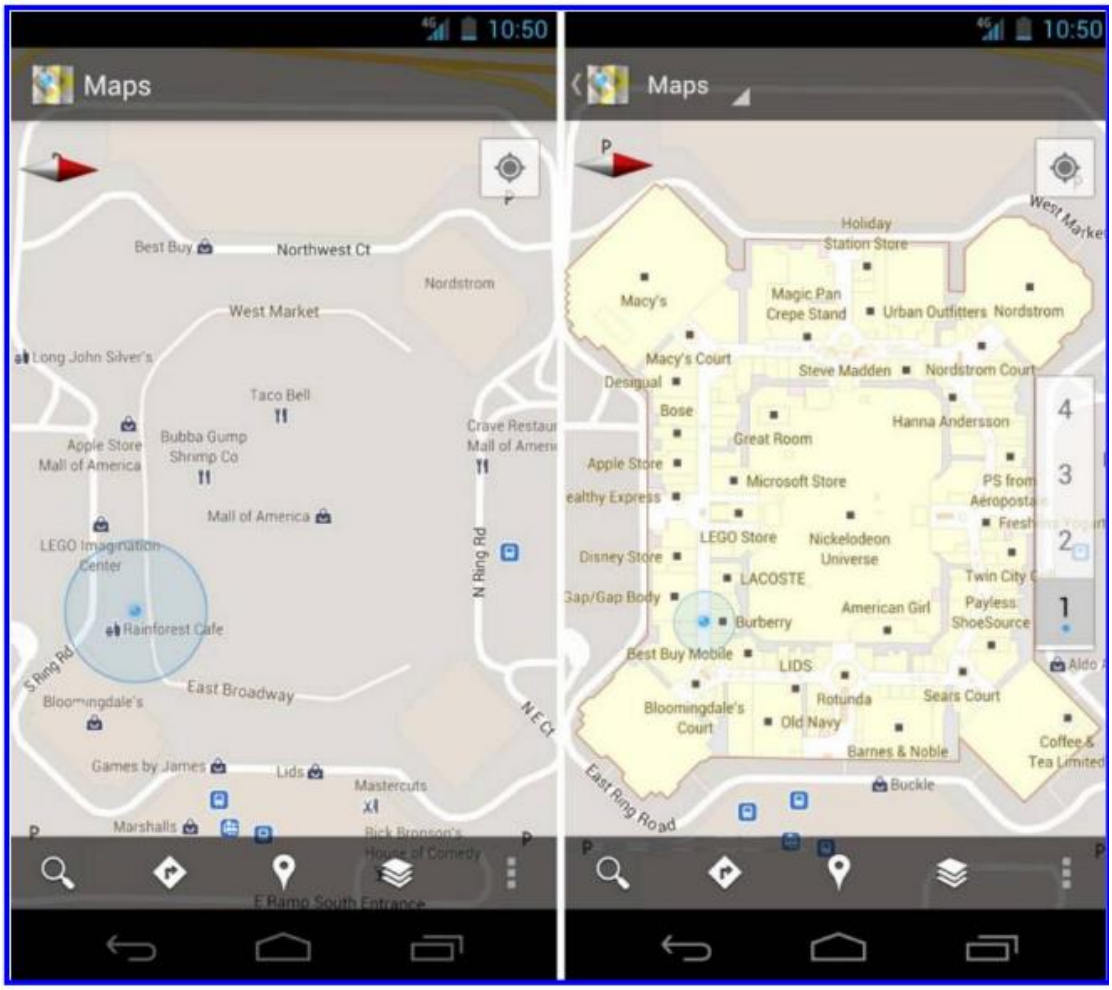

Figure 6.5: Mall of America in Minneapolis, before & after Google Map 6.0 for Android Release.

图 6.5 安卓版谷歌地图 6.0 前后在 Minneapoilis 的美国商场

6.4 支持 iOS 系统的 LBS

在 iOS 平台上的 LBS 支持包括对用户可用的位置定位框架核心和位置改变的通知，一个可用地图工具的框架和可控地图，和捆绑在平台上可以进行地图查看，浏览，搜索和导航到一个地址或 POI 的地图应用。

6.4.1 iOS 定位框架内核

iOS 除了用 GPS，还使用 WiFi 和蜂窝网络去使开发者自己选择定位信息的精度。如果定位服务被使用，定位框架核心必须连接到应用中去。这个框架赋予了 LBS 应用下列能力。

- **核实设备定位服务的能力。**这可以确保 LBS 应用在启动时可以成功的运行。应用商店也使用了相同的检验手法确保应用被下载到合式的设备上。验证包括通过 UI 请求设备能力码，一个应用的 Info.plist 文件来使验证得以完成。这个密钥的值是任何可定位服务的“定位服务”，或者是高精度下“GPS 的定位服务”。即使一个设备是可以使用定位服务的，并且 LBS 应用成功的启动了，开发者依然提供可定位服务的方法类使得位置信息可以优先

获得，以确保定位服务可用。当用户其他环境中选择了不可使用的定位服务时这个做法是有用的。

- **获得当前的位置。**这个框架提供了与多数应用可供配置的标准定位服务。使用这个标准服务，一个 CLLocation Manager 的实例可以被创建并且可以配置想要的精度和合适的渲染距离（两个连续更新的需求距离之间）。当开始接收定位通知时，必须特派一个代表到这个对象上去，并且 startupdatingLocation 方法必须被调用。当位置信息可用时，location manager 将会通知那个对象的特派代表。最新的位置信息也会直接从 CLLocationManager 对象上获得，而不会等待一个新的事件被送到特派员那。下面是开始一个标准定位服务的代码，是从在线 iOS 发展库摘录的。

```
- (void)startStandardUpdates
{
    // Create the location manager if this object does not
    // already have one.
    if (nil == locationManager)
        locationManager = [[CLLocationManager alloc] init];

    locationManager.delegate = self;
    locationManager.desiredAccuracy = kCLLocationAccuracyKilometer;

    // Set a movement threshold for new events.
    locationManager.distanceFilter = 500;

    [locationManager startUpdatingLocation];
}
```

接收位置更新的委托人代码没有摘录出来，但是应该被提供去实现 locationManager:didUpdate:fromLocation:方法的接收更新通知的功能，并且如果出错，这个 loc ationManager:didFailW ithError:方法会去接收通知。

- **获得关键改变位置的定位。**一个可以自定义的，低功率的，蜂窝无线电提供服务设备。服务提供定位和被许多应用接受的精度下随事件改变的定位。对于“常在“手机上的移动社交网络应用，这的确是一项低能耗且有吸引力的功能。这项服务不使用 GPS 而取代的工具是蜂窝网络 ID。这个服务仅仅在 iOS 4.0 及以上版本可用。要使用关键点定位的服务，就要创建 CLLOCAtionManager 类并且指定一个委任人；然后 startMonitoringSignificantLocationChanges 方法应该被调用。当位置信息可用时，location manager 就通知它的特定委任对象。像标准服务一样，最新的位置信息可以直接从 CLLocationManager 对象上获得，不需要等待新事件送到那个对象的委任那。在下面是开

始一个关键点定位服务的代码，是从在线 iOS 发展库上摘录的。

```
- (void)startSignificantChangeUpdates
{
    // Create the location manager if this object does not
    // already have one.
    if (nil == locationManager)
        locationManager = [[CLLocationManager alloc] init];

    locationManager.delegate = self;
    [locationManager startMonitoringSignificantLocationChanges];
}
```

- **区域监测。**在 iOS 4.0 及以上的版本中，档用户穿过地理边界的时候应用可以适用区域监控去通知。提供那些定义区域边界和开始接收区域边界穿越事件（进或者出）的类。

6.4.2 iOS 地图工具框架

地图工具是一个 iOS 3.0 及以上版本可以把地图通过全功能地图界面嵌入到应用中去的框架。这个框架支持显示街道视图和卫星视图地图，寻找地址和 POI，通过多点触控交互显示来进行放大和缩小。如今，通过谷歌地图客户端使用了地图工具包。因此，iOS 使用地图工具套件的开发人员受谷歌使用条款和与相关的服务谷歌地图的限制。

若要使用地图工具包框架功能，MapKit.framework 必须链接到 Xcode 项目中的应用。地图套件提供将地图添加到应用区域和课对地图属性进行配置的接口。它提供内置在地图中的图形位置查找器（地理编码）。它还提供反向地理编码能力。有关地图套件的各类框架的详细信息可以从苹果 iOS 开发者中心访问。

6.4.3 对于 iOS 的其他 LBS/地图的支持

除了在地图工具包架构上的谷歌地图外，微软提供必应地图在 iOS 上的软件开发工具包。这个开发工具包发布于 2011 年中期，给开发者提供了一套实际的 C 类，让他们在 Xcode 开发环境中开发 iPhone 和 ipad 应用。包括支持必应地面，空中和混合航拍模式的地图控制，包括添在地图上添加图钉和通过 GPS 在地图上定位电话获取用户位置的能力。iOS 地图开发包可以从微软下载中心下载。

6.5 安卓 LBS 的支持

在安卓平台上的 LBS 支持包括定位管理服务和地理编码服务，谷歌地图视图——一个通过谷歌接口老进入和操纵谷歌地图的谷歌地图库——在这个平台上非常有影响力的几个 LBS 应用（例如安卓版谷歌地图，地方和导航等）。

6.5.1 安卓定位管理服务

安卓利用位置提供者的重叠，现今包括 GPS 系统，WiFi 网络和蜂窝网络。这种重叠提

供在使用不同内容时可替代的定位资源（例如室内 WiFi 和室外 GPS）。它还提供了一个当所有源相结合时更可靠的定位方式，并给出了开发人员的定位精度和节能的选择。获取用户可靠位置的能力对于安卓提供 LBS 发展的关键服务。要想使用安卓位置管理服务，位置提供者和位置监听者必须被提供。从 LocationManager 来的位置请求需要调用 requestLocationUpdates()，传递一个 LocationListener，当用户位置或位置提供者状态改变时要使位置管理调用来完成几个回调。下面的代码段展示了一个对 LocationManager 和简单简单监听者的调用（安卓开发指南代码示范）。

```
// Acquire a reference to the system Location Manager
LocationManager locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);

// Define a listener that responds to location updates
LocationListener locationListener = new LocationListener() {
    public void onLocationChanged(Location location) {
      // Called when a new location is found by the network location provider.
      makeUseOfNewLocation(location);
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {}
    public void onProviderEnabled(String provider) {}
    public void onProviderDisabled(String provider) {}
  };

// Register the listener with the Location Manager to receive location updates
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER
, 0, 0, locationListener);
```

在调用的第一个参数是位置服务类型（NETWORK_PROVIDER，在这里，表示 WiFi 或者蜂窝网络），随后是通知之间的最小时间间隔和通知之间的最小位置变化——把两者设为零以获取尽可能快的更新频率。最后的参数是 LocationListener，接收位置更新的回调。

位置更新可以从多个提供者获得，每个提供者使用不同的获取频率。安卓开发者指南上有关于监听频率，从 GPS 系统或其他网络提供者上获取和使用位置信息的实践指导建议。ACCESS_COARSE_LOCATION or ACCESS_FINE_LOCATION 必须在应用层清单上被提供，否则这个应用将会获取位置信息失败。

6.5.2 安卓地理编码服务

地理编码和地理反编码是安卓应用有 LBS 功能的关键服务。对于地理编码，一个地址提供了它的经纬度。getFromLocationName 的调用（在下面展示）提供位置名称和指定的 maxResults——最大的地理参考返回值。经纬度的边界盒的值也是被指定的。

```
public List<Address> getFromLocationName (String locationName, int
maxResults, double lowerLeftLatitude, double lowerLeftLongitude, double
upperRightLatitude, double upperRightLongitude)
```

对于反向地理编码，是对于一个给定经纬度范围的返回地址序列。getFromLocation 代码在下面展示。

```
public List<Address> getFromLocation (double latitude, double longitude, int
maxResults)
```

6.5.3 谷歌地图视图

谷歌地图库是用来创建在应用的谷歌活动的。它不是安卓标准库的一部分，并且必须通过更新 AndroidManifest.xml 文件将其包含。第一步去使用谷歌地图视图在应用上创建一个地图视图层。通过创建或编辑 XML 布局文件去包含作为根结点的 com.google.android.maps.map view.main.xml，如下显示的是需要一个必须从谷歌开发先驱获得的 API 密钥来发展地图活动的代码。可点击表示如果这个地图是有意的/允许去让用户点击的。

```
<?xml version="1.0" encoding="utf-8"?>
<com.google.android.maps.MapView
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:id="@+id/mapview"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:clickable="true"
   android:apiKey="Your Maps API Key goes here"
/>
```

第二步通过扩展 MapActivity 来建设这个应用。后边的这个是地图库里 Activity 的子类，它包括查看和交互谷歌地图的各种功能。一旦建成，这个布局应该被 main.xml 地图视图布局所覆盖包含。这个被展示在下面的代码段里。

```
@Override
public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.main);
}
```

6.5.4 对安卓的其它 LBS/地图的支持

必应地图安卓软件开发工具包是一个开源工程，允许安卓开发人员在他们的 LBS 应用中去使用必应地图。这个 SDK 是以使用 Bing Map AJAX 控件的库形式表现出来的。

6.6 WINDOWS PHONE LBS 的支持

对于 WINDOWS PHONE 的 LBS 支持包括定位服务接口，必应地图和它的 Sliverlight 控件。

6.6.1 WINDOWS PHONE 定位服务

要想使用定位服务，WMAppMainfest.xml——这个应用的清单文件必须包含在<Capabilities>部分的定位服务功能 ID_CAP_LO CAT ION。

定位内核的接口主要收录在一个 System.Device 集合的 System.Device.Location 命名空间中类名叫 GeoCoordinateWatcher 的类中。为了接收用户的位置信息，应用需要创建一个 GeoCoordinateWatcher 的实例，订阅事件，并且调用它的 Start（）方法（看下面的代码片段）。

传递给 GeoCoordinateWatcher 的构造函数指定所需的精度的值。这两个可能值是 GeoPositionAccuracy.Default 和 GeoPositionAccurar.High。

应用可以监听到的两个 GeoCoordinateWatcherprovides 事件：

- GeoPositionStatusChanged——表示定位符的状态已经改变了
- GeopositionChanged——表明位置数据的经纬度已经改变了

通过监听 GeopositionStatusChanged，应用可以得到像服务不可用，可用，初始化等的改变通知，并且以当前模式和用户沟通（见下面的代码片段）。

```
public MainPage()
{
   InitializeComponent();

   if (geoCoordinateWatcher==null)
   {
   // Create GeoCoordinateWatcher with high accuracy
   geoCoordinateWatcher = new
GeoCoordinateWatcher(GeoPositionAccuracy.High);
   geoCoordinateWatcher.MovementThreshold = 40;

    // Subscribe to status changed event
    geoCoordinateWatcher.StatusChanged +=
            new  EventHandler<GeoPositionStatusChangedEventArgs>
(OnStatusChanged);

   // Subscribe to position changed event to receive GPS coordinates
   geoCoordinateWatcher.PositionChanged +=
            New EventHandler<GeoPositionChangedEventArgs<GeoCoordinate>>
(OnPositionChanged);

   geoCoordinateWatcher.Start();
   }
}
```

```
private void OnStatusChanged(object sender, GeoPositionStatusChangedEventArgs
e)
{
   switch (e.Status)
   {
      case GeoPositionStatus.Disabled:
         // The location service is disabled or unsupported
         break;
      case GeoPositionStatus.Initializing:
         // The location service is initializing.
      case GeoPositionStatus.NoData:
         // The location service cannot get location data
         break;
      case GeoPositionStatus.Ready:
         // The location service is working and is receiving location data
         break;
   }
}
```

PositionChanged 事件只有在设备位置改变到 MovementThredhold 指定改变量时才会

报警。没有开发人员设定定位服务获取位置信息的属性和方法。但是如果数据是从 GPS 获得的，水平精度和垂直精度属性将会一贯的为几米；和其它非 GPS 资源，在一些情况下，精度可达到 100 米或更高。下面的代码展示了如何获取和提取位置信息对位置改变事件的响应。

```
private void OnPositionChanged(object sender,
GeoPositionChangedEventArgs<GeoCoordinate> e)
{
   var location = e.Position.Location;

   // Access position information
   location.Latitude.ToString("0.000");
   location.Longitude.ToString("0.000");
   location.Altitude.ToString();
   location.HorizontalAccuracy.ToString();
   location.VerticalAccuracy.ToString();
   location.Course.ToString();
   location.Speed.ToString();
   e.Position.Timestamp.LocalDateTime.ToString();
}
```

6.6.2 必应地图控件

要想使用必应地图控件，应用必须获得必应地图控件，可以从必应地图门户网站上用微软在线 ID 创建（https://www.bingmapsportal.com/）。在 XML 布局文件里生产的线上创建一个必应地图对象：

```
<m:Map Name="bingMap" CredentialsProvider="YourBing-Maps-Key"
Height="500" Width="450" />
```

另外，必应地图密钥可以加载这个代码上。

```
bingMap.CredentialsProvider =
new ApplicationIdCredentialsProvider("Your-Bing-Maps-Key");
```

在这篇文章中的 XAML 文件中，下面的命名空间需要加入到这个应用的必应地图控件上。

```
xmlns:m="clr-
namespace:Microsoft.Phone.Controls.Maps;assembly=Microsoft.Phone.Controls.
```

6.6.3 必应地图网络服务

必应地图的 GeoCode 服务提供 Windows Phone 可使用的地理编码和反向地理编码服务。必应地图的路线服务提供基于地图上复合定位来计算路径方向的接口。必应地图交互软件开发包提供和这些必应网络服务交互的简单代码。

6.6.4 对于 WINDOWS PHONE 的其它 LBS/地图支持

没有对于 WindowsPhone 的官方的谷歌地图客户端或谷歌地图控键。但 Bing Maps 控制非常灵活，开发人员实际上可以将谷歌地图绑定到它。

诺基亚提供独家 windows phone 设备诺基亚品牌的地图和驱动客户端。诺基亚驱动提供完整的离线，免费的语音导航，这是一个和其它 Windows Phone 设备的主要的区分点。

6.7 移动网络 LBS 支持

对于 LBS 在移动网络应用上的主要支持途径是新兴的 HTML5 标准，在第 5 章有介绍。在 HTML5 的内部机理是 JavaScript 接口对地图渲染和地图控制的支持。例如诺基亚地图，提供支持 HTML5 的 JavaScript 接口，更确切的说是利用 HTML5 的特点去支持这个 API。更多的 API 细节可以从 Nokia’s Map API 在线指南上看到说明性的例子。诺基亚地图 JavaScript 的利用：

• HTML5 带有或不带有硬件加速的<canvas>元素。Canvas，是苹果公司在 2004 年提出来的想法，提供一个比点阵图更快的渲染图形和图片的方法。诺基亚地图 API 检测可用的 canvas，视图用 canvas 去渲染尽可能多的地图和图像注解。最终结果渲染速度是令人印象深刻的，达到了接近原生，非网页渲染的速度。

• HTML5 允许在 HTML 网页中嵌入可伸缩的矢量图形，提供了一个比点阵图展示的更高速渲染图形和图片的代替方法。诺基亚地图 API 也提供对图标和用户标注 SVG 使用的特殊支持。SVG 遵循文件对象模型意味着图形包括对象的可识别性（例如一个矩形图形）。尽管两者都是快速渲染机制，但是 SVG 和 canvas 的不同在于图形的生存周期：在 canvas 中，图形要在整个屏幕中回应对任何改变的修改（通过 JavaScript）。在 SVG，只有模型对象修改在本模型图形上的改变。对于对象外部的其它改变，不需要做任何改变。

• HTML5 支持的 W3C 地理定位接口。这个接口允许对浏览器位置的发现。它不需要任何关于定位服务的实际基础就可以找到位置。GPS，WiFi，蜂窝网络甚至浏览设备的地理参照 IP 地址都可以被独立或联合起来实现在 HTML5 上 W3C 地理定位接口的支持。这个 API 的初衷目标是为了通过客户端网络服务来追踪定位（为了人性化服务和定位目标），浏览者可以使用相同的接口实现定位和寻找当前位置。地图 API 的可用性和 HTML5 地理定位 API 的联合使得创建移动网络 LBS 成为可能。

诺基亚也提供了一个针对移动网络的有影响力的 HTNL5 地图应用。它支持挤捏缩放，驾驶或步行路由选择，伴随着喜爱位置和 POI 信息的保存。这个应用支持当用户在出行前不能下载地图时的离线模式，确保了没有网络覆盖服务的可用性。这个应用支持 iOS 4.3+和安

卓 2.2+的浏览器。

除了或外地图，移动网络也可使用户内地图。例如，Micello 的 JavaScript 接口对于室内地图支持 HTML5。Canvas 也被大量使用在那些不支持室内架构的几何图形 Micello API 中（商场和机场的站点）。

Table 6.1: Some Useful LBS References.

Micello	http://www.micello.com
Point Inside	http://www.pointinside.com
iOS Core Location Framework	https://developer.apple.com/library/ios/#documentation/CoreLocation/Reference/CoreLocation_Framework/_index.html
iOS Map Kit Framework	https://developer.apple.com/library/ios/#documentation/MapKit/Reference/MapKit_Framework_Reference/_index.html
Bing Maps for iOS	http://www.microsoft.com/download/en/details.aspx?id=1112
Android Location Manager	http://developer.android.com/reference/android/location/LocationManager.html
Android Google Map View	http://developer.android.com/resources/tutorials/views/hello-mapview.html
Bing Maps Android SDK	http://bingmapsandroidsdk.codeplex.com
Windows Phone Location Service	http://msdn.microsoft.com/en-us/library/system.device.location(v=vs.92).aspx
Bing Maps Portal	https://www.bingmapsportal.com
Bing Maps Geocode Service	http://msdn.microsoft.com/en-us/library/cc966793.aspx
Bing Maps Route Service	http://msdn.microsoft.com/en-us/library/cc966826.aspx
Bing Maps Silverlight Control Interactive SDK	http://www.microsoft.com/maps/isdk/silverlight/
Nokia Maps API	http://api.maps.nokia.com
W3C Geolocation API Specification	http://dev.w3.org/geo/api/spec-source.html