Department of Computer Science and Engineering

The Chinese University of Hong Kong

2003-2004 Final Year Project

Location-based Services using GSM Cell Information over Symbian OS

LYU0301

Supervised by Professor Michael LYU Rung Tsong

Group members: Mok Ming Fai
Lee Kwok Chau

Prepared by Mok Ming Fai (01713554)

Abstract

This project is subjected to the final year project LYU0301. The project title is "Location-based Services using GSM Cell Information over Symbian OS".

This report contains ten chapters. They include the introduction of this project, the introduction of Symbian OS, some basic GSM technologies, currently existing LBS solutions, location-based service using location area identifier (LAI) and cell identifier (CI), LBS in the 2-dimensional space, LBS in the 1-dimensional space, description of the programs that we developed and the approach we will take in the future.

Our final year project tries to make use of LAI and CI to provide location-based service in 2-dimensional and the 1-dimensional space using simple hardware which is going to be widely distributed in the near future - mobile phones with Symbian OS.

We performed investigations and experiments for all the possible ways to accomplish this project. This report includes the descriptions and results of all the experiments done. Programs developed are also explained.

Table of Content

1. Introduction

1.1 Motivation

With the rapid development of information technology, the need of various types of information service emerges. Among them the location sensitive applications are of the utmost importance. There are quite a number of researches being done in this field recently to look into a way to provide location-based service to the general public.

Knowing the location where one is at can be of great help in daily life. For example, one can find out nearby restaurants or fast food chain shops for lunch; one can also find out where to take a bus or change for MTR. There is really a need of location-based service that can be used by the general citizens. Therefore we would like to try to have a study in this field.

1.2 Project Objective

There are currently a number of existing location-based services (LBS) available. However, they require some special hardware in order to provide services. This make the service not easily distributed among the general citizens. We would like to work on a system that requires the minimum special hardware requirement and system overhead while providing LBS.

Our approach is to use the location area identifier (LAI) and cell identifier (CI) to approximate our current location. These information can be retrieved from the signal emitted by GSM base station. However, retrieving these information is not easy, since most mobile phone does not support this operation. This is why we turn to use mobile phones that utilize Symbian OS, which is power enough for us to carry out this operation.

With the help of Symbian OS, we tried to implement an LBS that only requires a mobile phone to operate. In the following chapters, we will try to explain the basic knowledge of Symbian OS, GSM Technologies first. After that, we will describe and explain our work in this semester.

2. The Symbian OS

2.1 The Rise of Symbian

Symbian, founded in June 1988, is a software licensing company jointly owned by several wireless industry leaders including Ericsson, Nokia, Panasonic, Psion, Samsung Electronics, Siemens and Sony Ericsson. It is a private independent company developing advanced, open, standard operating system for data-enabled mobile phones. Its product, Symbian OS (Symbian Operating System), is becoming popular in the mobile phone market.

There are currently quite a number of mobile phone using Symbian OS. They include:

- Nokia 6600, 7650, 3650, N-Gage and 9210 Communicator
- Sony Ericsson P800, P900
- Motorola A920
- Fujitsu F2051, F2102V

There will be more mobile phone producers turning to this new operating system. It is announced that the following mobile phones, which use Symbian OS, will be released very soon.

- Samsung SGH-D700
- Siemens SX1
- Sendo X
- BenQ P30

2.2 Operating System for Mobile Devices

Symbian OS is a robust multi-tasking operating system specially designed for real-world wireless environments and the constraints of mobile phones like limited amount of memory and limited speed of CPU. It is 32-bit, little-endian operating system working with ARM architecture chips with V4 instruction set or higher. Supported platforms include PrimeXSys platform from ARM, the StrongARM, Xscale architectures form Intel, the OMAP platform from Texas Instruments and the Dragonball platform from Motorola.

Symbian OS not only works on mobile phone, but also work on any mobile devices that satisfy the requirement of the operating system. This operating system contains an extensive and rich collection of libraries and for implementing many industry standards.

Industry Standards	Examples
Networking	TCP/IP, PPP, TSL, SSL, IPSec, FTP
Communications	Bluetooth, IrDA, Obex
Security	DES, RSA, DSA, DH
Messaging	POP3, IMAP4, SMTP, SMS, BIO
Browsing	HTML, HTTPS, WAP, WML
Telephony	GSM, GPRS, fax
Multimedia	WAV, AU, WVE, JPEG, BMP, MBM, GIF

Table 2.1: Industry Standards Supported by Symbian OS

Symbian OS is so powerful that, besides supporting a number of industry standards, it also support a lot of useful functions in various field for developers to develop application for both computation and telephony.

For example, Symbian OS allows developers to retrieve cell information and information of telcos via the frame sent out from base station; There is a DBMS in the phone for better data storage efficiency; There is also APIs for developers to use the camera.

All these strength are the reasons for us to turn to use Symbian OS as our working platform in our FYP.

2.3 Characteristics of Symbian OS

Symbian OS has a number of characteristics that make it suitable for mobile computing and better against other operating system.

1. **Integrated Multi-Mode Mobile Telephony**
   Symbian OS integrates the power of computing with mobile telephony, supporting complex computations, advanced data service and telephony service on one single mobile machine

2. **Open Application Environment**
   Applications written in different programming languages, like C++ and Java, can be deployed onto Symbian OS

3. **Open Standards and Interoperability**
   Symbian OS provides a core set of APIs that is shared by all Symbian OS phone. Thus, a program written using the core set of APIs can be used in many Symbian OS phones without the need of modifying the source and recompiling.

4. **Multi-tasking**
   Symbian OS fully implements multi-tasking and threading. Its special feature active object allows program require multi-threading to be written with only one thread with several active objects.

5. **Fully Object-oriented**
   Symbian OS is designed completely using object-oriented techniques. It also allows program to be written using OO technique.

6. **Flexible User Interface**
   Graphical user interface in Symbian OS is flexible, users can design their own user interface easily. For example, Nokia developed its own specific GUI called “Avkon” for use in its mobile phone.

2.4 System Structure of the Symbian OS

The system structure of the Symbian OS is quite straight forward. The operating system and its applications can be divided into 4 types of components. They are kernel, engines, servers and applications.

The kernel is used to manage the hardware inside the system. They may include RAM, display device and keypad etc. In Symbian OS, hardware devices can only be accessed by privileged APIs, which in turn can only be executed by the kernel. Other applications have to access these system resources by calling the kernel APIs.

A server is a program that do not have graphical user interface. It manages resources that may be needed by clients. Examples of server include file server, which helps to manage the operations of the system, and window server, which helps to manage the display area on the screen.

An engine is part of an application that only manipulates data and do not interact with the user. An engine helps to separate the program core from the part that manipulates the GUI. This makes the program more flexible. For example if we want to change the interface, we don't need to have any modification on the core engine.

An application is just a program that interacts with users with an interface. It may need to access to servers for certain resources.

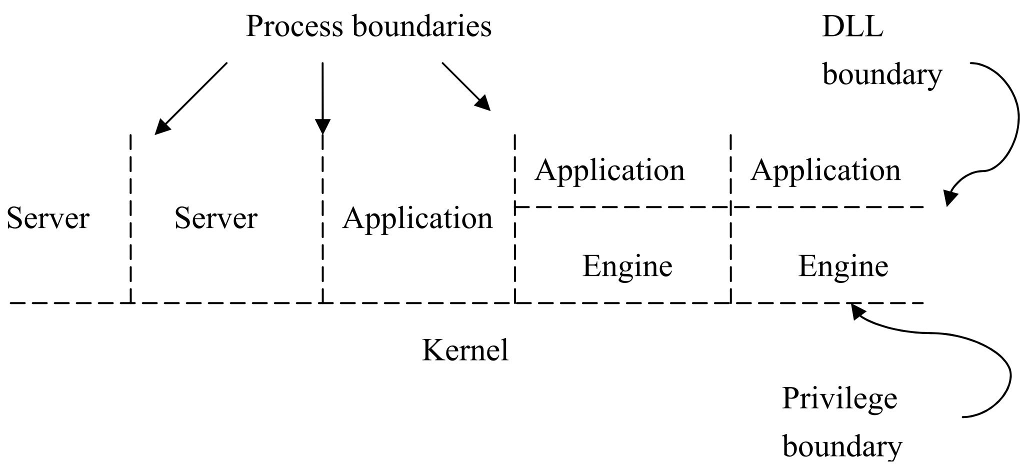


Figure 2.1: The components of the Symbian OS and its applications

2.5 Special Features of Symbian OS

There are a number of special implementation on the Symbian OS that make it a robust and suitable operating system for mobile devices with limited memory and computing powers. These special features include Descriptors, Cleanup Stack, Two-phase Construction, Active Object.

2.5.1 Descriptors

2.5.1.1 Descriptor Overview

In C, strings processing is inconvenient. Strings are NULL-terminated, so programmers have to allocate the extra byte to accommodate the terminator. A slightly wrong in arithmetic while manipulating the string may cause undesired memory overwrites. Therefore using strings in C may easily create bugs.

In Symbian OS, descriptors are used instead for string handling. Descriptor provides safe and consistent mechanism in dealing with strings and even binary data. Descriptor object maintains pointer and length information to describe data. It is used to access and manipulate the data. Descriptor object provides safe access and manipulation to the data it refers. Attempts to access memory outside the data area represented by the descriptor object are caught. In the current version of Symbian OS, two different sizes of descriptors are provided. They include 8-bit version and 16-bit version.

2.5.1.2 Classification of Descriptors

We can divide descriptors into different types from different point of view. In term of the location where the data and the descriptor object exist, we may classify descriptors into Buffer Descriptor, Heap Descriptor and Pointer Descriptor. In term of the modifiability, we may classify descriptors into modifiable and non-modifiable.

1. **Non-modifiable Descriptor**

   Non-modifiable descriptor can be accessed but cannot be changed (except complete replace). Examples are TBufC, TPtrC and HBufC. All descriptor class ends with a capital letter ‘C’ indicates that the descriptor object is non-modifiable. The length information stored in non-modifiable descriptor is the current data length.

2. **Modifiable Descriptor**
   Modifiable descriptor can both be accessed and changed. They include TBuf, TPtr. Besides the current data length, modifiable descriptor contains an extra length information, the maximum length of data that can be represented by the descriptor. This extra information is needed so as to make changes to the data within.

We may also classify descriptors in term of their structure:

1. **Buffer Descriptor**
   Data is part of the descriptor object and the descriptor object lives in the program stack.

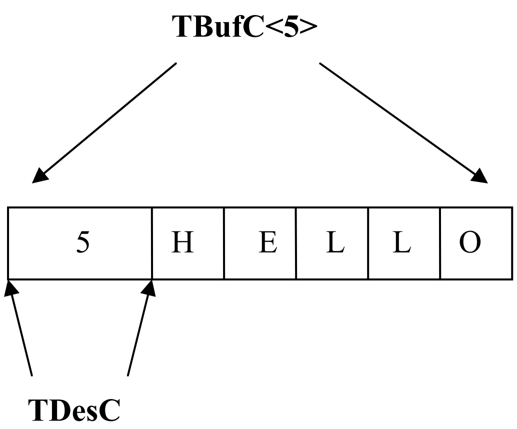


Figure 2.2: Representation of a Buffer Descriptor

2. **Heap Descriptor**
   Data is part of the descriptor object and the descriptor object lives in the heap.

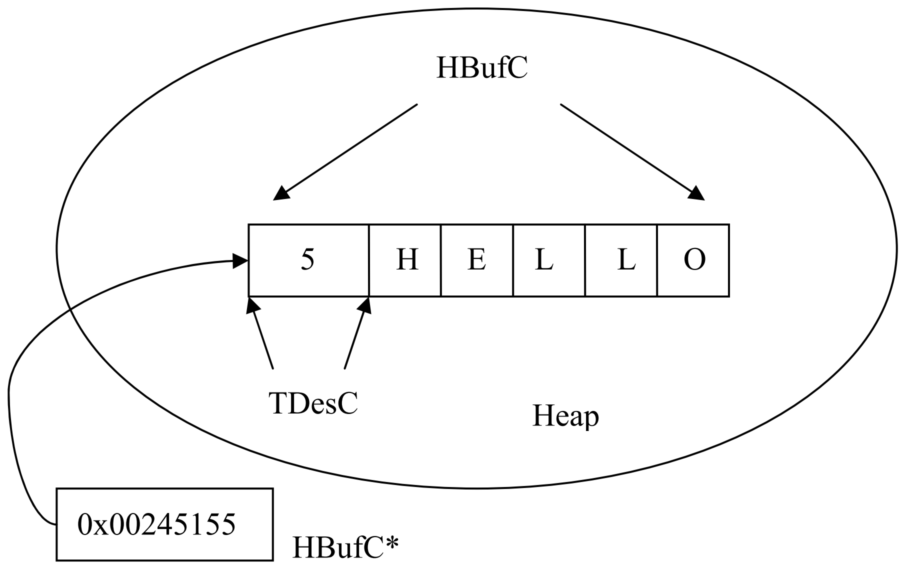


Figure 2.3: Representation of a Heap Descriptor

3. **Pointer Descriptor**

   Data is separated from the descriptor object. The descriptor object lives in program stack while the data may lives in heap or program stack.

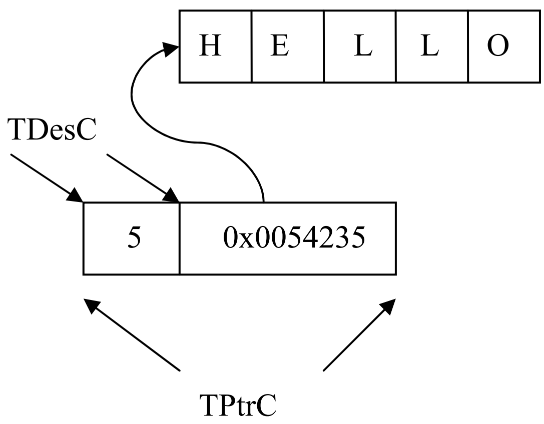


Figure 2.4: Representation of a Pointer Descriptor

2.5.1.3 The Hierarchy of Descriptors

Figure 3.4 shows the hierarchy of descriptors. All the descriptors, modifiable, non-modifiable, buffer-based, heap-based and pointer-based, are all derived from the abstract base class TDesC. As implied by the name, TDesC is non-modifiable. Other modifiable descriptor classes have some extra information and functions to make them modifiable. The abstract class facilitates the design of function and parameter passing.

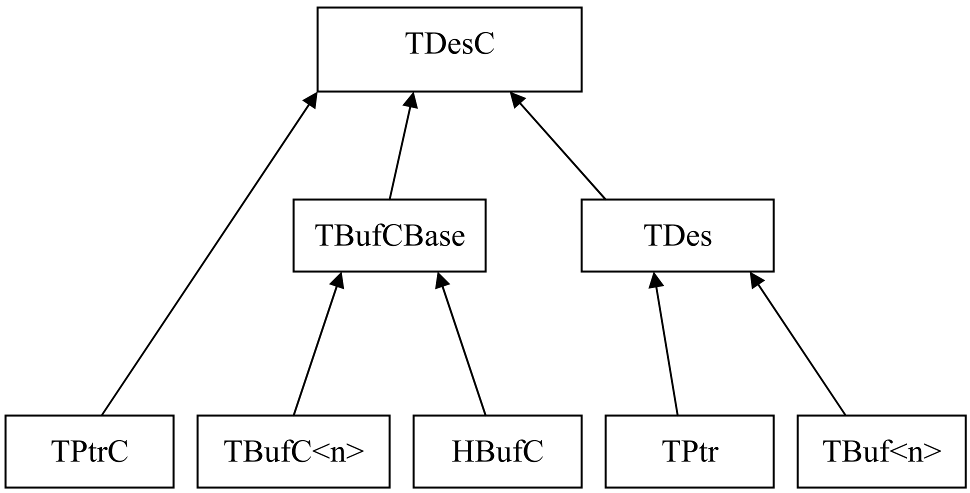


Figure 2.5: The Hierarchy of Descriptors

2.5.2 Cleanup Stack

Memory is of very limited size in most mobile device. It is even more treasurable in mobile phones. If we allocate memory and do not free them after finished using them, we may run into out-of-memory problem. Unlike desktop machines, which could be readily restart when there is out-of-memory problem caused by extensively not freeing memory after use, mobile devices are expected to work for a long period of time before they are switched off or restarted. Therefore, freeing unused memory becomes an important issue. If memory is continuously being allocated without freeing up, the mobile device is under the risk of out-of-memory problem.

In C++, a **delete** function is provided so as to free up heap memory that is allocated by objects. It seems nice with the function to free up unwanted memory, but there is still the chance that **delete** cannot free those unwanted memory. Consider the following code fragment:

```
CX* x = new (ELeave) CX;
x->UseL();
delete x;
```

Code fragment 2.1: Possible cause of Heap Failure

where CX is a class, x is an automatic variable allocated from heap by the identifier **new** and UseL() is a function that **leave** (terminates upon encountering unexpected error). In the above implementation, if UseL() does leave when it is executed, the automatic variable x will never be able to be freed since the pointer pointing to that memory is destroyed. This will cause a heap failure (heap memory cannot be freed) and may result in out-of-memory problem in a long run. This introduces a cleaning mechanism using the Cleanup stack.

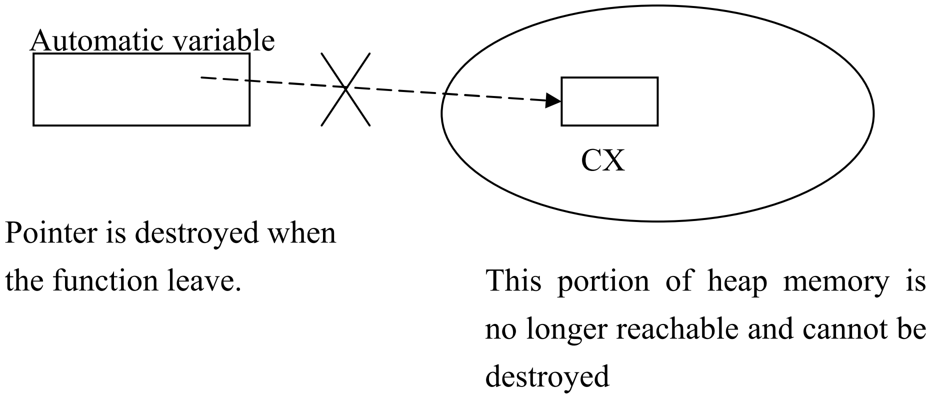


Figure 2.6: Heap Failure caused by a Leaving Function

The problem of heap failure is caused by automatic variables. Cleanup stack pinpoint this problem by providing a place for holding these automatic variables. With the existence of Cleanup stack, code fragment 3.1 can be rewritten as follow:

```
CX* x = new (ELeave) CX;
CleanupStack::PushL(x);
x->UseL();
CleanupStack::PopAndDestroy();
```

Code fragment 2.2: The use of Cleanup Stack

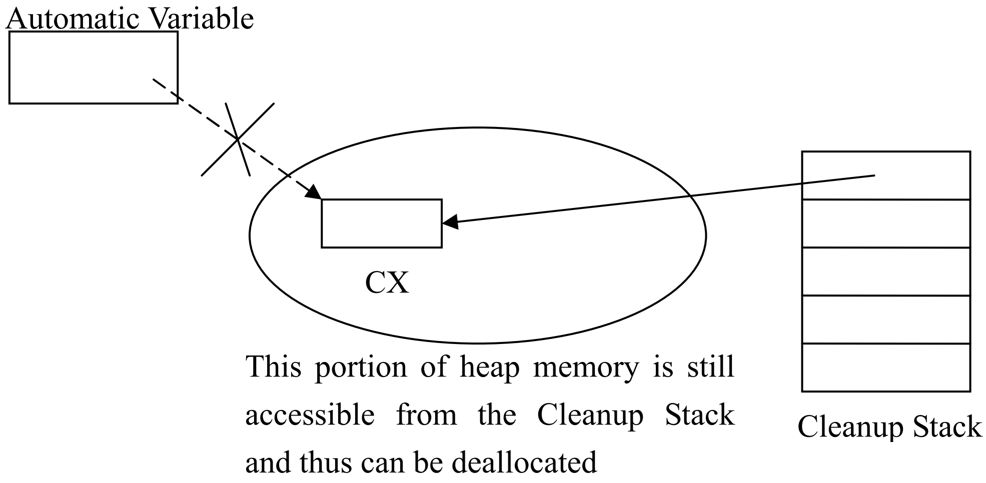


Figure 2.7: How Cleanup Stack help to ease Heap Failure

Cleanup stack will free up all the heap memory that are addressed by the pointer stored in the Cleanup stack. Thus, after we created an automatic variable, we push it onto the Cleanup stack. If the program leaves, then the allocated heap memory will be freed by the Cleanup stack. Therefore, using Cleanup stack can ease the problem of heap failure.

2.5.3 Two-phase Construction

The Cleanup Stack provides a safe mechanism to prevent memory leak. However, there is still some situation that memory may leak.

```
CY::CY()     // Constructor of CY
{
    iX = new (ELeave) CX;
}
```

Code fragment 2.3: Construction that may leave

When an instance of CY is being instantiated, there may be some problem about memory leak:

1. If the allocation of CY fails, everything leaves and there is no problem

2. If the allocation of CX fails, the codes leaves, but CY is not yet pushed onto the Cleanup Stack, the memory allocated to CY leaks.

3. After the initialization of CY and CX, any call to function that leave causes no problem, since CX and CY are on already on the Cleanup Stack.

The problem occurs when a leaving function, including the construction of another object, is executed. Immediately after CY is instantiated, there is not yet a chance to push it onto the Cleanup Stack. If a leaving function is called here and it does leave, CY will not be able to be de-allocated. Therefore, Symbian OS introduce another rule to prevent memory leaks. The rule is called the Two-phase Construction. This rule, together with the use of Cleanup Stack, fully prevent memory leak from occurring in Symbian OS.

Two-phase construction states that all constructors must not leave. This means that no leaving function, including the construction of other object, can be performed in the first phase of the construction process. All leaving function can only be called in the second phase of the construction process.

Two-phase construction divides the construction process into two phases. The first phase is the normal construction, denoted by the object’s name, for example CY::CY(). The second phase is usually denoted by CY::ConstructL(). In fact, it is a leaving function. All the operations that might leave should only be done here.

The construction process with two-phase construction rule is like follow:

1. Create the object, for example CY, using the first phase constructor
2. Push the object onto the Cleanup Stack
3. Call the second phase constructor to perform operation that leaves

In this way, if any of the operation in the second phase of the construction process leaves, CY will still be able to be deallocated since it is already on the Cleanup Stack.

2.5.4 Active Object

Symbian OS is a multi-tasking operating system. However, unlike most operating system, it does not use multi-threading to achieve the purpose of multi-tasking. Instead, it utilizes a mechanism called Active object to perform multi-tasking.

With a view to the system overhead of multi-threading, Symbian OS turned to another approach that requires lesser system overhead. In using active object to perform multi-tasking, there is only one thread. In the thread there is an active scheduler together with one or more active objects. The active objects send out requests. These requests will then be scheduled by the active scheduler so that these requests can be served in an interleaved manner.

The active scheduler schedules to serve the requests sent out from active object in an non-preemptive manner. Therefore there is no mutual exclusion code needed in using active objects. This further reduces the code complexity while achieving multi-tasking.

2.6 Summer Work on Symbian OS

In the summer, we learnt to program for the Symbian OS. We wrote two little applications to demonstrate what we have learnt and the computing power of this OS.

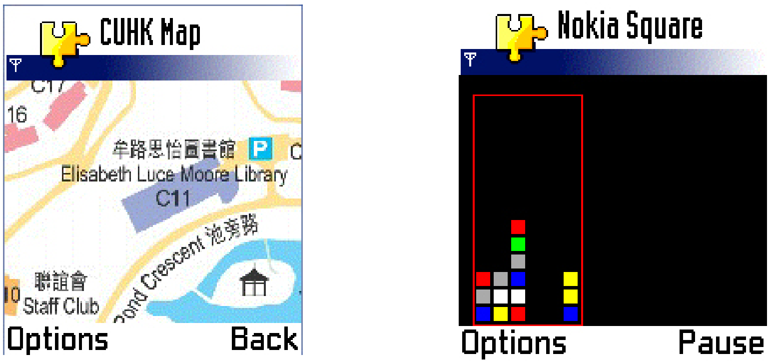


Figure 2.8: Programs written for the Symbian OS

The CUHK Map shown in figure 2.8 is a program that displays the map of the CU campus. User can scroll in the four directions to view the map. Nokia Square on the right hand side of figure 2.8 is a game that looks like Tertis. The one shown in figure 2.9 is a program written to show the various GUI components that can be used in Symbian OS.

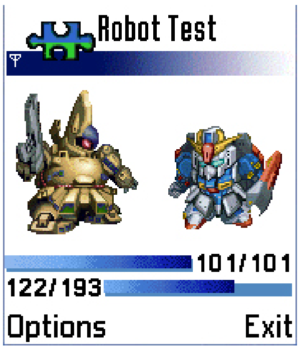


Figure 2.9: A program for illustrating the power of GUI components of Symbian OS

Having introduced the operating system we are going to use, we will next describe the basic GSM knowledge we will use in our project.

3. The GSM Technologies

3.1 Cellular Technology

GSM networks are hierarchically structured. In a GSM network, there is at least one administrative region, which is the Mobile Switching Centre (MSC). Each administrative region consists of at least one Location Area (LA). Each location area in turn is made up of several cell groups. Each of these cell groups is assigned to a Base Station Controller (BSC). Figure 3.1 shows the system hierarchy of the GSM network

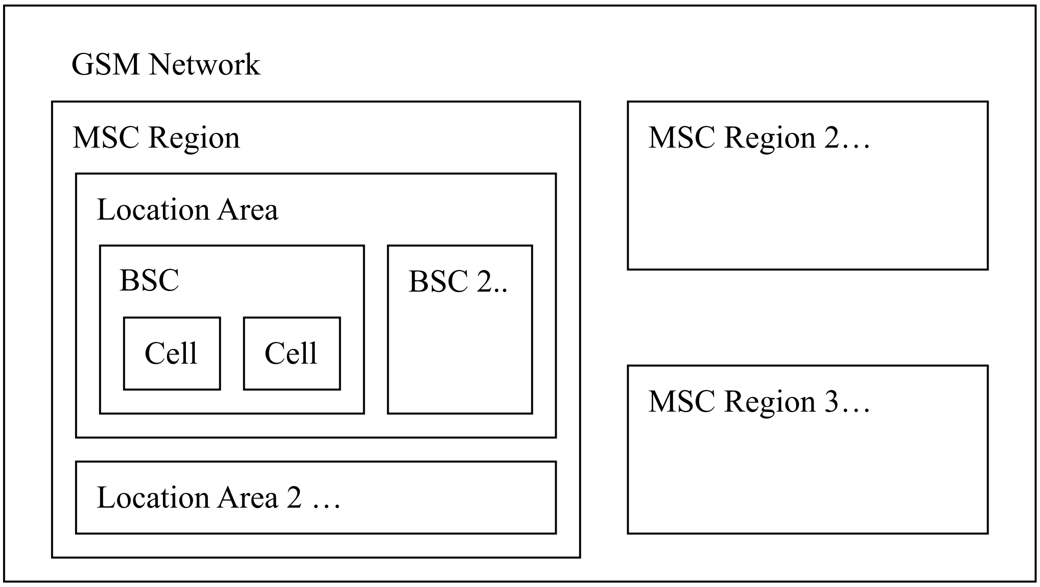


Figure 3.1: System hierarchy of the GSM network

Because of the limited frequency bands, GSM system only allocated 25MHz in the 900MHz frequency range. Maximum of 125 frequency channels each with a carrier bandwidth of 200kHz. Therefore GSM service providers must reuse their assigned frequency repeatedly. This give rise to the formation of cell clusters. For example, if a GSM operator is assigned k number of different frequencies that it can use to provide service, it has to reuse the frequencies. In order to prevent inter-channel cross-talk, cells using the same frequency band should be placed as far apart as possible. Figure 3.2 shows how frequencies can be reused in three cases, in each case the GSM operator is assigned different number of frequencies that it can use (indicated by k). Different cell clusters are formed in each case.

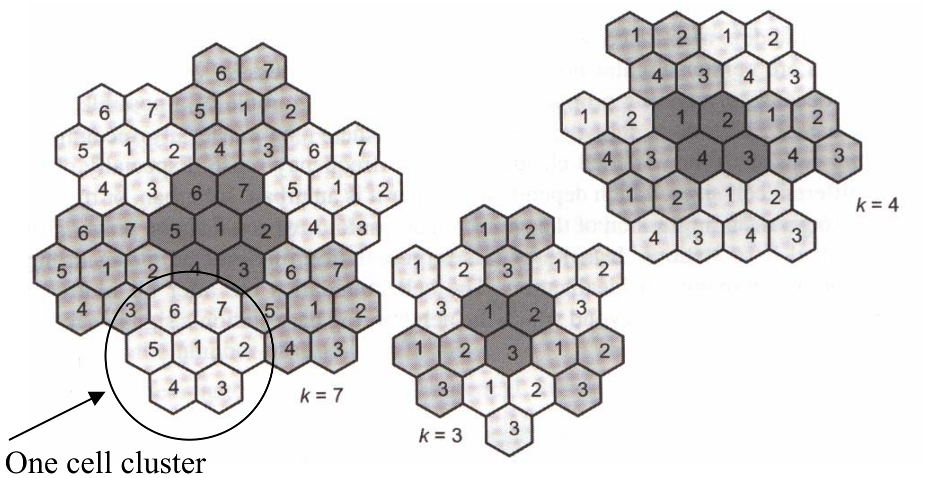


Figure 3.2: Frequency reuse and cell cluster formation

3.2 System Architecture and Addressing

As we mentioned in the previous subsection, GSM system is structured hierarchically. Figure 3.3 shows the physical structure of the GSM system architecture with essential components. Some important components are described in the previous subsection, while others are not very important to our project so we just ignored them.

Since the system is in a form of hierarchy, there should be some way to identify each component among the whole system. There are some international standard in addressing each mobile station, for example the International Mobile Station Equipment Identity (IMEI), which utilizes the Type Approval Code, Final Assembly Code and Serial Number to uniquely identify a mobile station internationally. However, in this project we only concentrate on how to identify each mobile station locally, but not internationally, so we do not need the IMEI, which is too detailed to be used. Instead, we concentrated at the Location Area Identifier (LAI) and the Cell Identifier (CI), which when combines together, can identify a local mobile station uniquely.

LAI consists of two portions, the Country Code (CC) and the Mobile Network Code (MNC). These two codes together forms an identifier which of maximally a 5 decimal digits number. For the CI, it is of maximum length of 16 bits in size.

LAI uniquely identify a location area in a GSM network while the CI uniquely identifies a cell in a base station controller's domain.

3.3 Cell Selection and Reselection

A mobile phone must periodically measure the strength of the signals emitted from the base stations nearby. Base on the measurement a mobile phone select a cell which gives the best reception. After connected to a cell, accessing a service become possible. However, when the mobile phone moves along a path, the strength of signal received from the base station that it is connected to may drops, which may due to several reasons like it is moving away from that base station or there is some environmental factor that distort the reception. In this case, a mobile phone must reselect another cell which gives a better strength of signal. This process is called the cell reselection. Cell reselection is very common, when we travel around some place we usually experience quite a number of cell reselection event, since cell are of limited size. When we reach the edge of a cell, a mobile phone will reselect another cell that gives a better reception in order to guarantee the quality of service provided to a user. When a cell reselection event occurs, the mobile phone will transit from one cell to another cell, thus receiving signal from another cell which is of different LAI and CI.

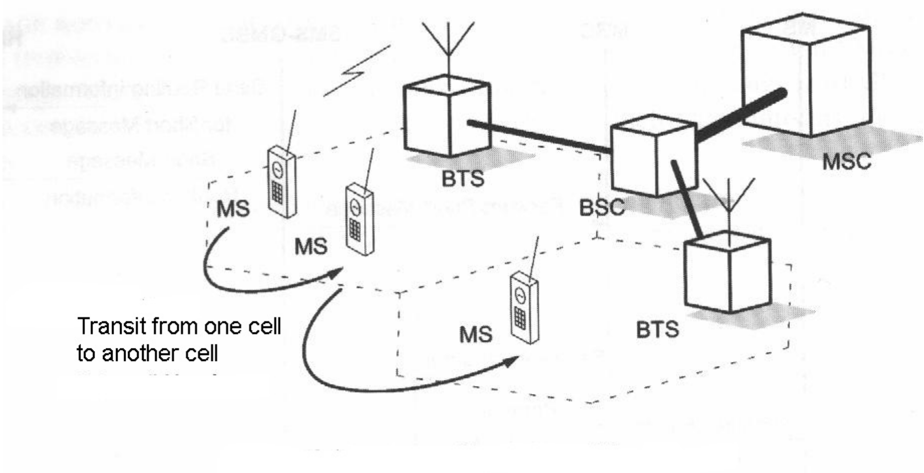


Figure 3.3: Cell reselection when traveling from one cell to another cell

4. Currently Existing LBS Solutions

There are currently two major streams of LBS solution available. They are the Global Positioning System (GPS) and SMS Query using specially designed SIM card. GPS is already very mature and is widely used in industrial and commercial fields, while the latter one is still very new and is still under the development process. In the following subsection we will give a brief introduction on the two methods and compare their advantages and disadvantages.

4.1 The Global Positioning System

The Global Positioning System is a satellite-based navigation system made up of a network of 24 satellites. The satellites orbit around the earth in 12 hours. There are totally 6 orbital planes which are equally spaced (separated at about 60 degree from each other).

Figure 4.1: 24 satellites and 6 orbital planes for GPS system

21 GPS satellites and three spare satellites are in orbit at 10,600 miles above the Earth. They are set to be in those positions so that there will be four satellites above the horizon at any particular location on the Earth. Each satellite continuously broadcast a signal representing its position. Then GPS receivers on the Earth receive any three of the four satellites that is above the horizon and calculate it position by using triangulation. The result is then represented in form of geographic

location, which includes longitude and latitude. The signal from the fourth satellites can be used to find out the attitude.

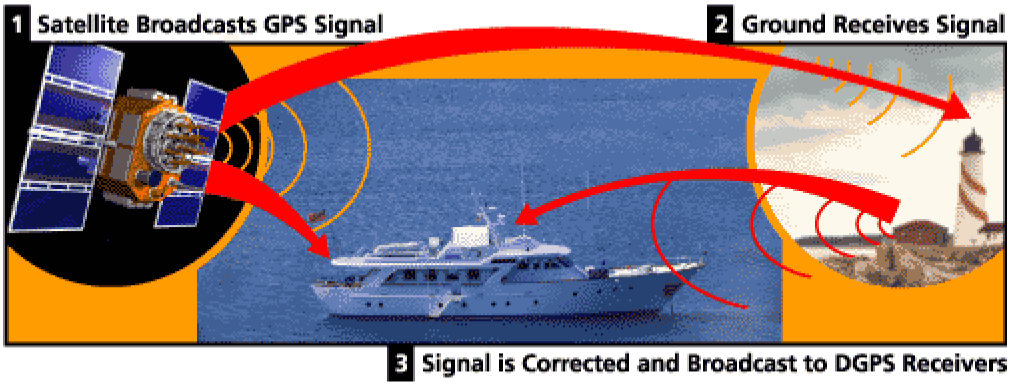


Figure 4.2: How GPS works

GPS is widely used in science, in sea navigation and in air (flight) to determine the current location and the route to take to destination. Furthermore, GPS is used militarily. There are still many applications but mostly not targeted to the general citizen.

Figure 4.3: PDA equipped with GPS receiver for location determination

4.2 SMS Query using Specially Designed SIM Card

This system is much simpler than the GPS system. It involves no satellite. What it needs are specially designed SIM card and some GSM base stations. It works as follow: The application drive the mobile phone to constantly send out SMS query message. Then the nearest base station will reply the query with a location string and some information of the current location of the user. Since the GSM base stations are deployed in a systematic manner, each base station can represent a specific region of area. So, when a base station replies an SMS query, it is sure that the user is inside that region, otherwise the SMS query would be replied by another base station which is nearer to him. A specially designed SIM card is needed so as to accomplish this task. This specially designed SIM card is used to send out special SMS query to GSM base station. This LBS solution uses simple equipment and would be more easily accepted by the general public.

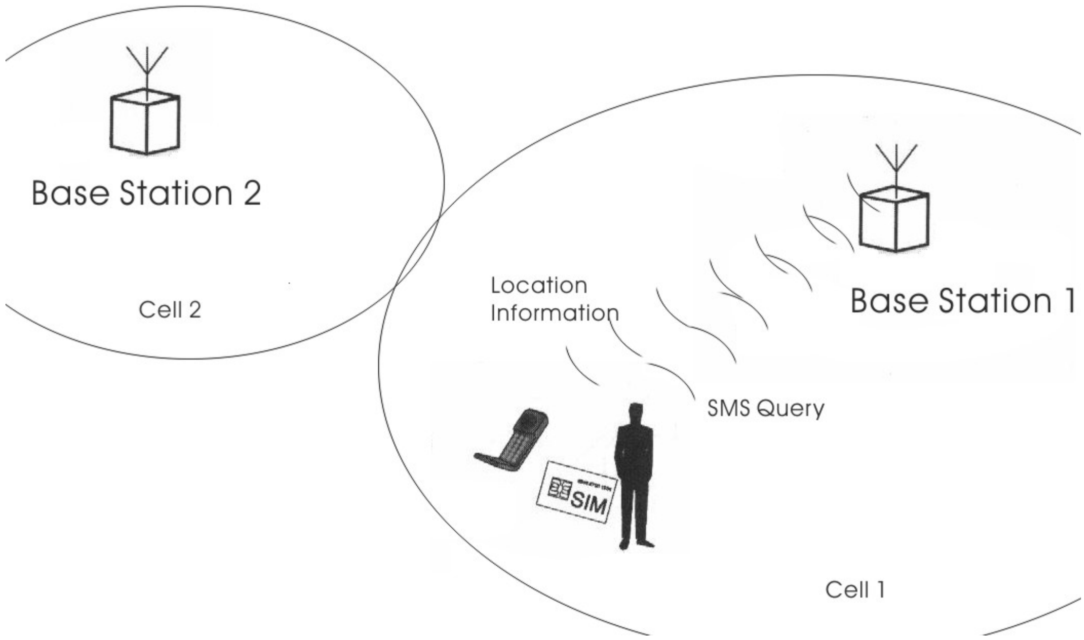


Figure 4.4: The operation of SMS Query for location information. The mobile phone actively sends out queries

4.3 Comparison of the Two Methods

The above methods mentioned are nice. They have their relative advantages and disadvantages.

The accuracy of GPS is generally high. It is about 10 to 100 metres with most equipment. This accuracy can be improved to 1 metre with military-approved equipment. For the second method, the accuracy greatly depends on the cell size and the signal strength of the base station. If the cell size is small, the accuracy can be high (within 50 metres). However if the cell size is large, the accuracy can be reduced to about 500 metres.

Both methods require extra hardware support. GPS users have to get a GPS receiver installed while the latter method requires user to have a specially designed SIM card.

The draw back of using GPS is that GPS signal would be seriously distorted by severe weather condition. Since signal has to be sent from space to the Earth, a severe weather condition, for example thunderstorm, would distort the signal, making the location-determination process inaccurate. Secondly, GPS signal is not reachable in closed environment, like inside a building. Furthermore, three satellites must be visible by the GPS receiver in order to provide the service. It is not very probable in densely populated urban area.

The SMS Query method does not have this constraint, since GSM signal can reach most part of the building, users can still query their current location. However, the major draw back is that the active role of the mobile phone requires it to constantly send out SMS query, which will occupy the communication channel of the mobile phone, other communication, for example making a phone call, will not be available then.

5. Location-based Service using LAI and CI

The two methods mentioned in the previous section are nice, but they require special hardware device like GPS receiver and specially designed SIM card. In order to provide location-based service with the minimal use of special hardware and overhead, we turned to an approach that tries to retrieve the Location Area Identifier (LAI) and Cell Identifier (CI) from the base station our mobile phone is connected to.

5.1 The Principle

In order to provide service with good quality, mobile phone service providers deploy GSM base station in a way that most of the area people may go to will be covered by one or more than one GSM cell. Each GSM cell covers a region of area which may overlap with other cells. Each mobile phone, when being switched on, will try to connect to one and only one GSM base station in order to access their service. Mobile phones would connect to a base station that gives the strongest signal strength to the location where the mobile phone is at. In most of

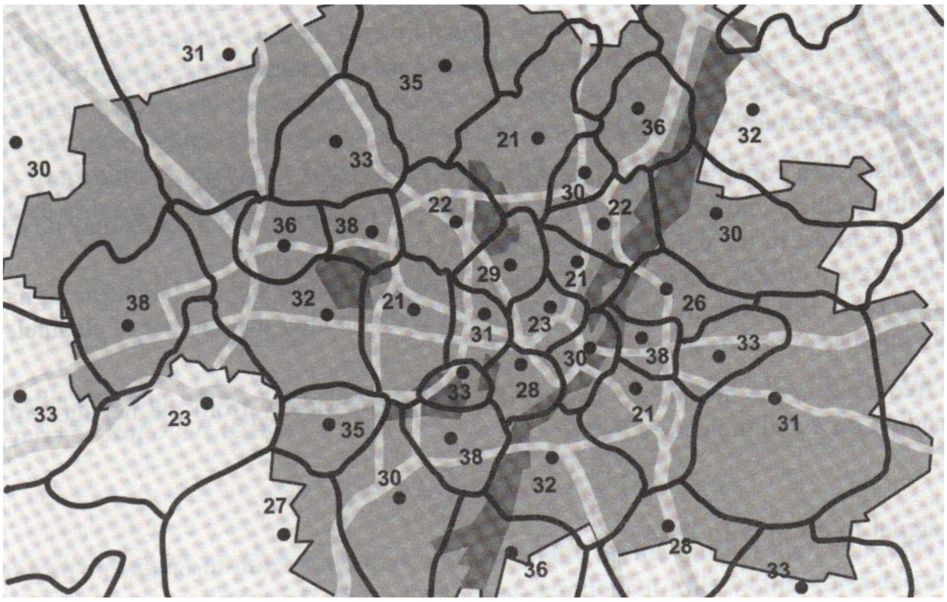


Figure 5.1: Area covered by cells. The dots inside cells represent the base stations

the case, the nearest base station would give the strongest signal strength to the mobile phone. Therefore, retrieving the LAI and CI of the base station that a device is currently connected to can tell us which cell region we are currently in. This in turn gives us an approximation of the current location of the mobile phone.

A GSM base station continuously sends out data and control frames. In the header of each frame, there are fields storing the information of the base station which send out the frame. One of the information is the base station identification which includes the location area identifier (LAI) and cell identification (CI) of that station. These information are transparent to mobile phone users. Using Symbian OS, a mobile phone can retrieve these data for special use. We try to use this approach to determine our current location.

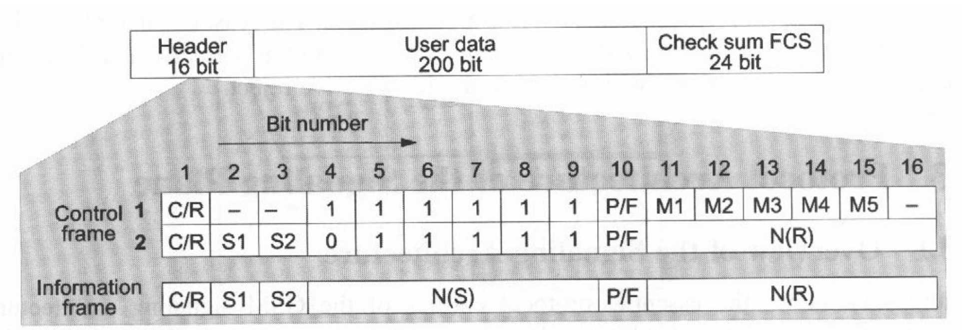


Figure 5.2: Structure of a GSM frame under RLP protocol. Header includes base station information

We try to discover the LAI and CI from the signal received by the mobile phone. Since a mobile phone can at most connect to one GSM base station which gives the strongest signal (which usually implies the nearest GSM base station) by retrieving the location area identifier and cell identifier pair of that GSM base station, we could approximately determine our current location. In the following sections, we will explain our main principle and what have to be done in order to provide this location-based service.

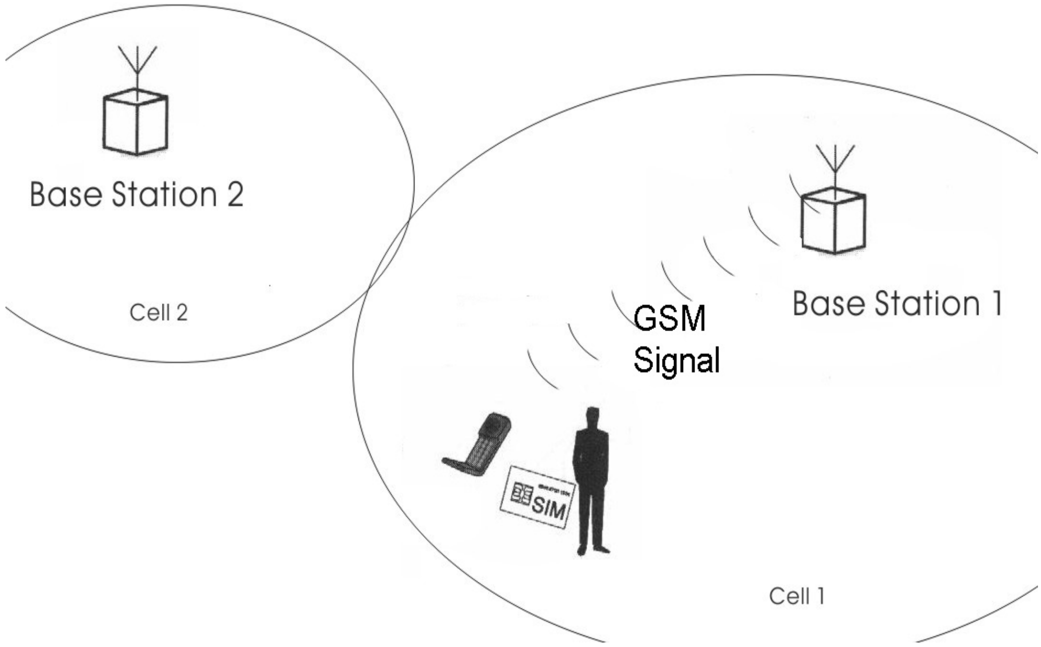


Figure 5.3: Retrieve LAI and CI passively from GSM broadcasting signal.

5.2 Development of the Data Collection Kit – GSM Status

To retrieve the currently LAI and CI from the GSM signal, we need to write a program which runs on Symbian OS and perform the required task.

We used Nokia Series 60 platform, which is in fact Symbian OS version 6.1 with a graphical user interface specially designed by Nokia, to perform our task. Inside the libraries provided by the system development kit, there is one function which can perform the task of retrieving the cell identification numbers. However, owing to some special reason, Nokia removed the header of this function from the system development kit, making it inaccessible by developers. In order to use this function, we have to re-activate this function by adding the header back to the program. In the older Communicator 9200 SDK, the function was not yet hidden from developer's access. Therefore, by copying the header files etel.h and etelbgsm.h from this SDK to our SDK and link them to our program, we can re-activate the function for our project to use.

With this function, we wrote a program that retrieve the current LAI and CI pair.

Figure 5.4: GSM Status that retrieve current cell information

The program retrieves the current cell information on every one-second interval. It displays the data onto the screen. Then it compares the pervious data with the newly-retrieved data, if a cell change occurs, it pops up a dialog box telling the user that a cell change has occurred. From figure 5.4, we can see that three information are retrieved. They are the LAI (displayed as “Location”), CI (displayed as “Cell ID”) and relative signal strength (displayed in them of percentage). Whenever any of the information changes, the program updates the data in the one-second interval’s time.

5.3 The Data Collection Methods

In order to investigate whether the use of LAI and CI is appropriate to serve as the foundation of a new direction of LBS, we have to look into the cell distribution, signal strength and the accuracy over an area.

To figure out the cell distribution within an area, we have to use our data collection kit, the GSM Status, together with some data collection rule, to achieve our goal.

GSM Status returns us the LAI, CI and relative signal strength at the particular point where the mobile phone is at. However, as we all know, there is overlapping of cells over the whole region of area, a data retrieved at that particular point in the area may only represent part of the actual data that we need to collect. Therefore, to figure out the cell distribution and boundaries within the whole area, we need some rule in collecting the data.

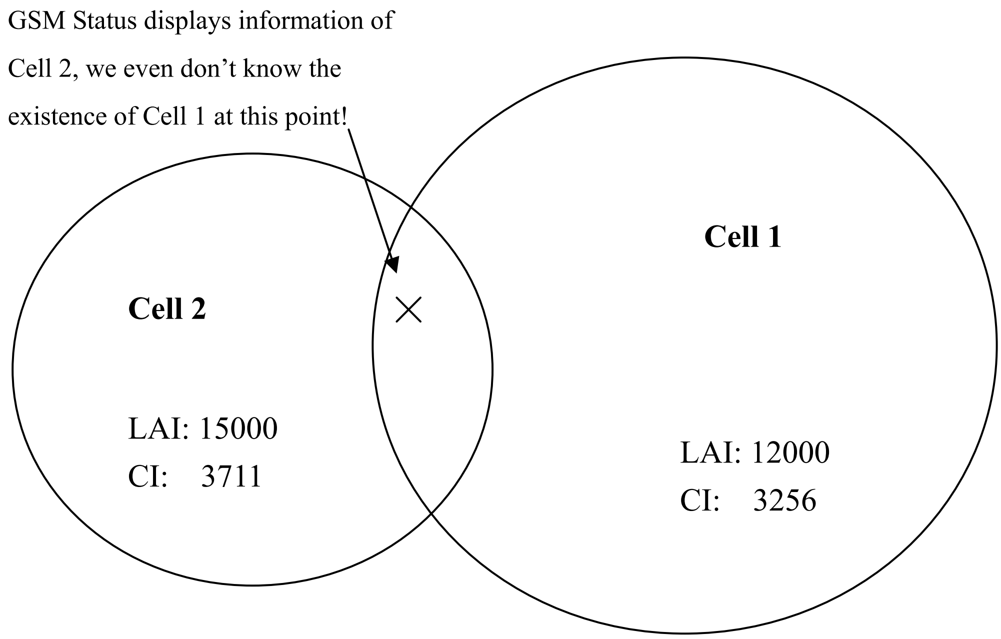


Figure 5.5: We need a rule to collect data so as to figure out all the hidden data

We proposed two rules to collect cell information. They are the static method and the cell change method. The two methods are explained in the following subsections.

5.3.1 The Static Method

The static method is a rule for use to collect cell information using GSM Status. In this method we first define a set of important points on the map of the area that we are interest. Then we use GSM Status to collect data at those set of points. At those points, we wait for a sufficiently long period of time to observe the cell information. If the point is at the overlapping region of two or more cells, we may experience cell change event during this period of time since the strength of the signal emitted by base station is dependent on the environment, for example the existence of an moving object like car. Any environmental change would have some effect of the signal strength. If the signal strength of the base station we are currently connected to drops below the strength of signal emitted by another near by station, the mobile phone will connect to that station instead, to guarantee communication quality. Therefore we may be able to determine whether we are at the overlapping region and can discover the existence of both cells.

By defining points dense enough and observing the data for a longer time, we can accurately determine the distribution of all cells within a region.

In fact, we can make the result even more accurate. We can repeat the experiment several times at those selected points and find out the probability of getting particular cell information. Then we may draw a probability contour line map to figure out the region covered by a particular cell with different probability. However, since this method requires a very long experimental time, we did not perform this (plotting probability contour line map) in our experiment.

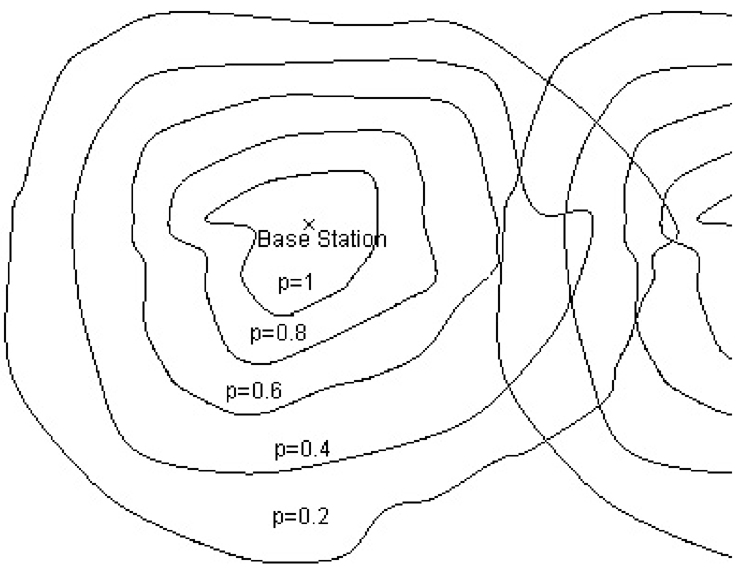


Figure 5.6: Probability contour line map with probability p that we are in a particular cell

5.3.2 The Cell Change Method

Another method we propose is the cell change method. In this method, we do not define any set of points to be taken as experimental points. Instead, we walk around the whole area to detect cell change events. Since our data collection kit is capable of detecting cell change event, we can use the cell change event to figure out the boundaries of the cells inside the area to be investigated. By walking along all the main path in the area, we may find out most of the place where cell change take place. By repeat the experiment several times we may improve the accuracy of the experiment and can figure out the boundaries of cells.

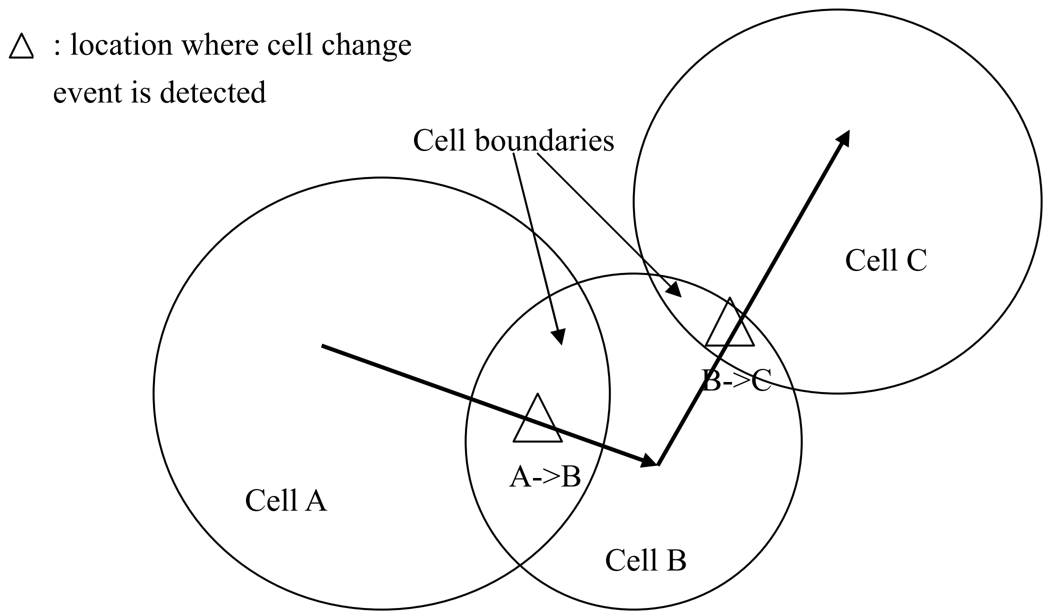


Figure 5.7: By walking along the main path in the area, most cell boundaries can be detected

5.3.3 The Comparison of the Two Methods

The two data collection rules have their advantages and disadvantages. We summarized them in the following table

:

	Static Method	Cell Change Method
Advantages	Results very accurate at those selected points	Can figure out the boundaries of different cells in the area
	Experiments only done on those selected points	Fast, no need to wait for a long time to get the result
Disadvantages	Takes a longer time	Boundaries detected are regions instead of sharp lines
	Cannot figure out the distribution of cells clearly without dense selected points	Have to walk through the whole area several times

Table 5.1: Comparison of the two data collection rules

There are relative advantages in each method we mentioned, in order to completely find out the cell distribution in an area, we used both the methods in our project.

6. Location-based Service in 2-dimensional Space

6.1 Motivation

Most of the current LBS applications are targeted in the 2-dimensional space. We would also like to work on the 2-dimensional field, which dominates the market of LBS application.

Hong Kong is an international city. Many tourists from all over the world enjoy traveling in Hong Kong. With technology advancement, it is supposed that we should have an electronic tour agent to guide tourists while they are traveling around Hong Kong. However, tourists may not have a GPS receiver to utilize the existing LBS. Therefore, we would like to develop an electronic tour agent running on mobile phones that utilize the Symbian OS. Therefore, we first choose the campus of The Chinese University of Hong Kong as a testing site. We would like to know whether the application of LAI and CI would be capable in helping us to build our system.

6.2 The Experiment

In order to investigate the capability of using LAI and CI for determining the current location, we have to figure out the cell distribution, cell size, degree of overlapping in the campus first. Then, we will plot a cell-to-location map for future use in building our system.

In this experiment, we investigated the cell distribution of two local telcos. They are SmarTone and Peoples. We used different approach mentioned in the previous section in different telcos. For SmarTone, we used the static method in collecting the cell information; for Peoples, we turned to the cell changed method. We repeated the experiment five times in order to obtain a better accuracy.

6.3 Expectation of the Experiment

Before we do our experiment, we have the following expectations:

- Cells are of similar size
- The portion of cell overlapping is small and only occur at the edge of the boundary
- No large cell completely covering a smaller cell

- Cells can be modeled as hexagonal shape cluster covering the whole area.

6.4 Results and Statistic

Distribution of cell identifiers in the CU campus for SmarTone:

(LAI is the same throughout the CU campus, number represents CI)

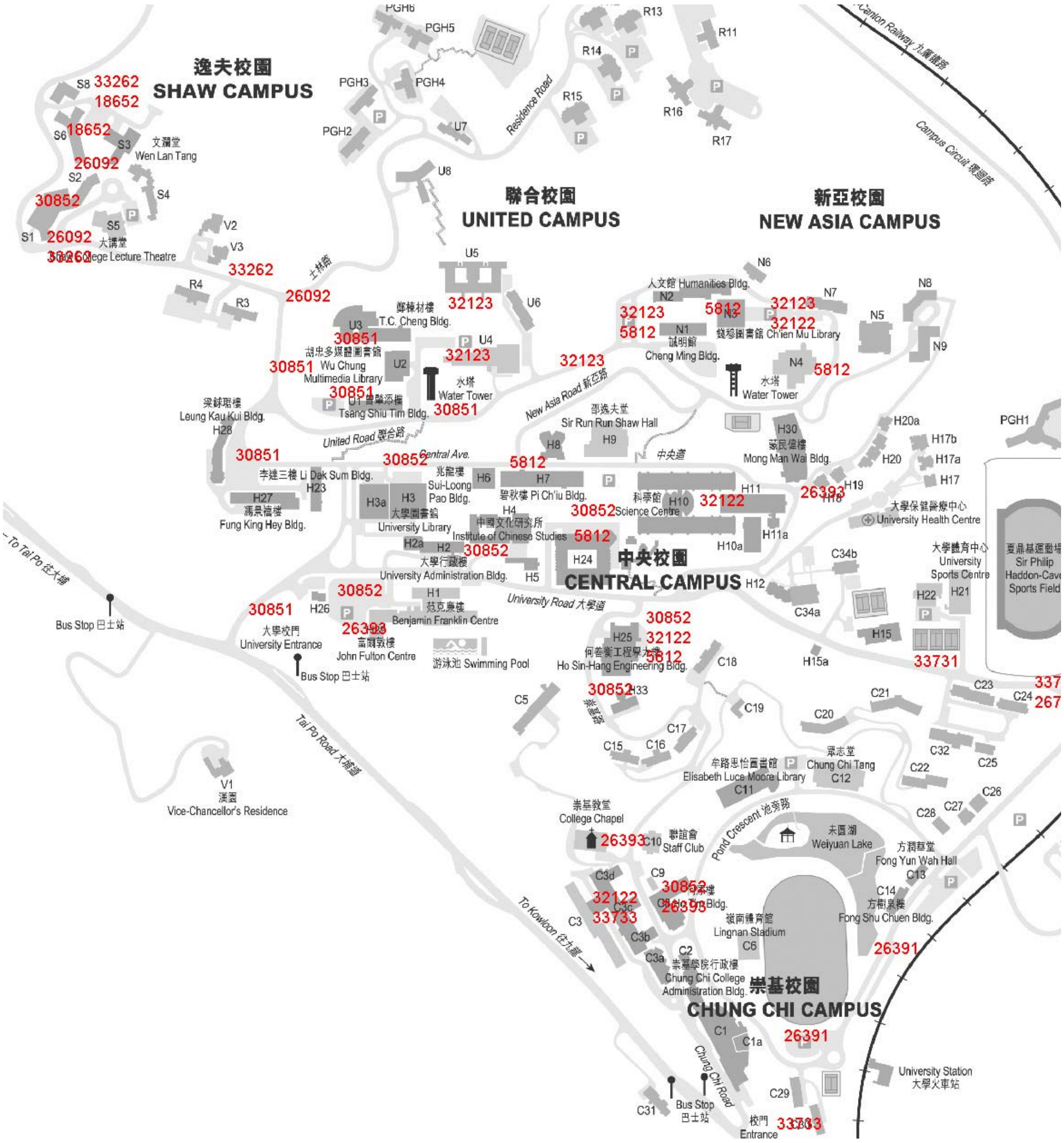


Figure 6.1: Experimental results for SmarTone

Outline of cell boundaries for SmarTone in the campus.
(The number represents the cell identifier)

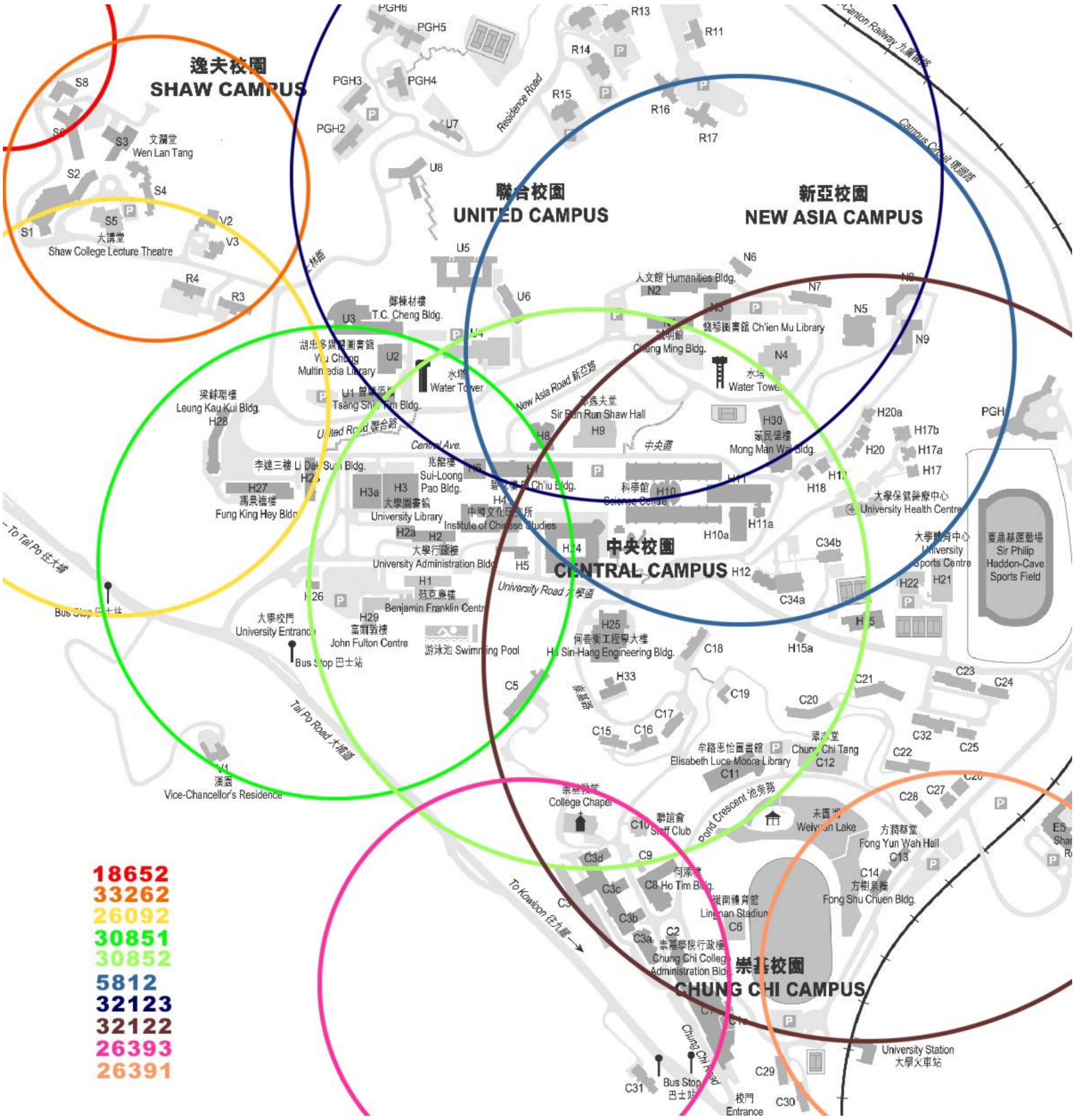


Figure 6.2: Approximated cells distribution in the CU campus for SmarTone

Distribution of cell identifiers in the CU campus for Peoples:
(LAI is the same throughout the CU campus, numbers represents CI)

Figure 6.3: Experiment result for Peoples

Outline of cell boundaries for SmarTone in the campus.
(The number represents the cell identifier)

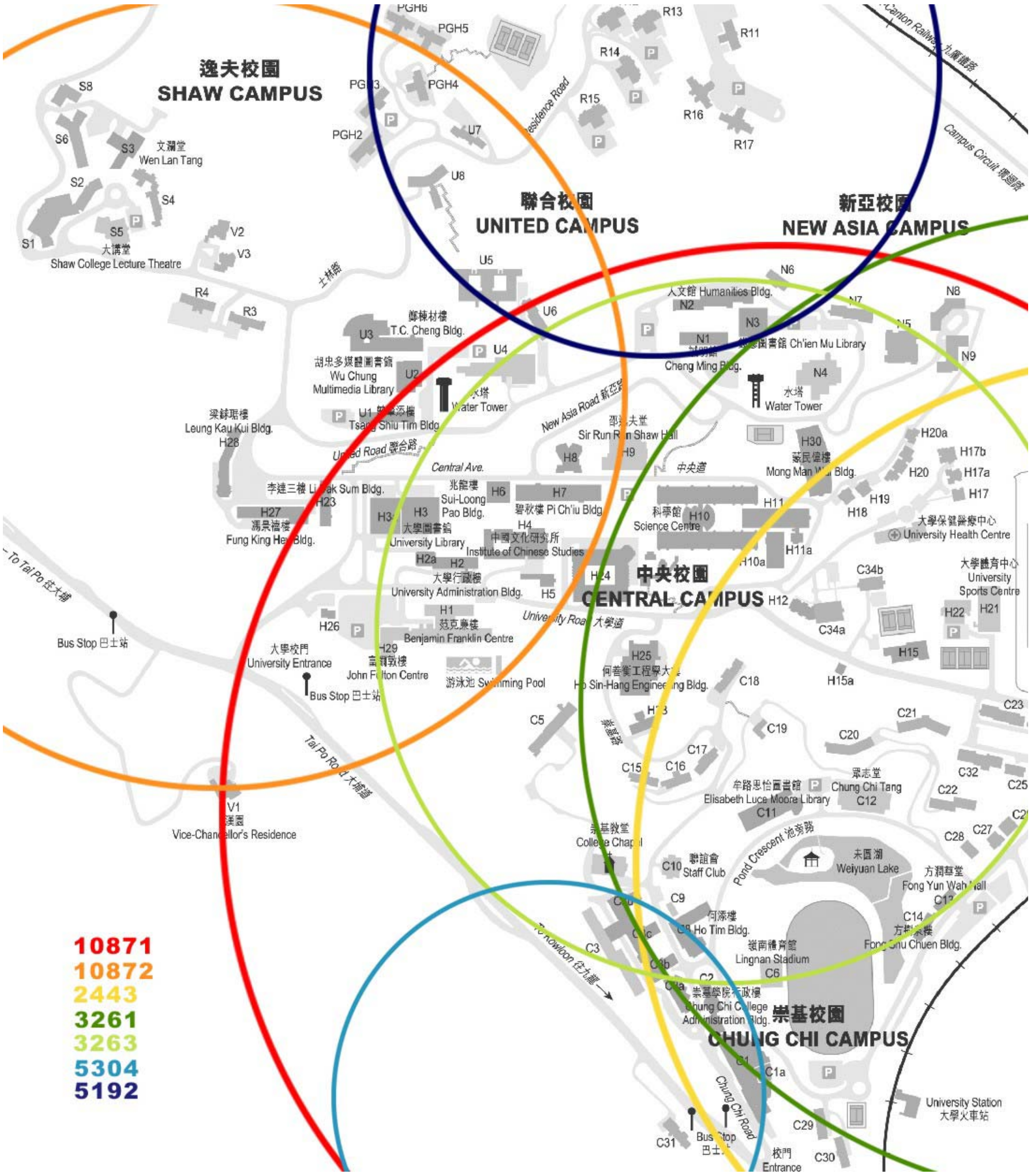


Figure 6.4: Approximated cells distribution in the CU campus for Peoples

6.5 Conclusion of the Experiment

After performing the experiment, we found there are some discrepancies with our expectations.

Firstly, cell varies greatly in size. For example, from the result of the experiment done on Peoples (figure 6.4), the cell in blue colour is very small compared with the orange one.

Secondly, the scale of cell overlap is quite large. From the result of SmarTone (figure 6.2), the light blue cell overlaps greatly with the light green one.

Furthermore, there are large macro-cell completely encapsulating a smaller micro-cell. From figure 6.4, a light green cell is totally within the red cell.

Also, some cell covers an area that is too large to be used in accurately determine the current location. For example, in figure 6.2, a brown cell covers a quite large area (including the Chung Chi campus and part of the main campus). This cell, without other cells' assistance, cannot give enough information for use to accurately determine our current location. An example is that: when we are at the CU bus station near the KCR station, we may receive the cell information of the brown cell (CI = 32122). However, the brown cell covers a very large area. With only this information we cannot say for sure that whether we are at the CU bus station or at the University sport centre.

Lastly, cell may change shape under different environmental condition. This may make the cell boundaries slightly different from the expected ones.

To conclude, there are several potential difficulties in using the LAI and CI to determine accurately the exact location of a mobile user. It is because the cell size is controlled by the telco, and those cells are usually too large to be used accurately in this way. Furthermore, mobile phones currently can only connect to one single mobile station. No other information can be retrieved other than the information of the base station we our currently connected to. All these make the use of LAI and CI not accurate to be used in the 2-dimensional space. Because of this inaccuracy, we turned to the 1-dimensional space LBS.

7. Location-based Service in 1-dimensional Space

7.1 The Principle

There are quite a number of technical problems we have to solve in the 2-dimensional space. Without the help of telcos it seems not very possible to provide LBS that are accurate enough to be used by the general public. Therefore we turned to the 1-dimensional space.

1-dimensional space here we mean a line. Since cell may vary greatly in size, the information of exactly which point we are currently at in a cell seems not very useful. We are more interested in the event of cell change in the 1-dimensional space.

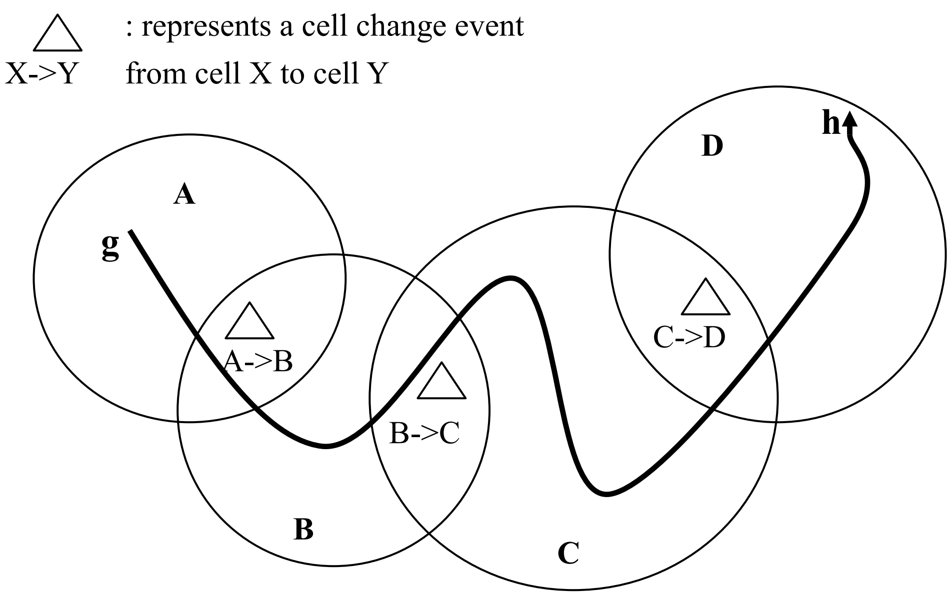


Figure 7.1: Cell change events happened in a 1D space

In the one dimensional space, we are interested in the “entrance” of a new cell. Other information becomes not important to us. In other words, we only want know when we transit from one cell to another cell.

In a 1-dimensional space, for a particular route, when it goes from one place to another place, it must pass through a set of cells that are located along the path. For example, in figure 7.1, a route from **g** to **h** passes through four cells: A, B, C and D. No matter how many times we go from **g** to **h**, if we take the same route, we must pass through these four cells. Thus the three events (A->B, B->C and C->D) must occur during the journey from **g** to **h**. With a route and a set of cell change events, we can say for sure that which region we are currently in with a very high confidence For example, when we detect a cell change event of P->Q, then we know that we are currently in the realm of base station Q, and we were in the realm of base station P..

Therefore, a pre-defined route can eliminate all the uncertainties and inaccuracies we have had in the 2-dimensional space.

With the principle of LBS in the 1-dimensional space, we turn to something along a path - rails and buses. Travelers, when touring around Hong Kong, must take public transports. However there are too little information can be retrieved from the public transport system. For example when a tourist want to go to Clear Water Bay, he takes an MTR, where should he change for other public transport, say, minibus? Another case is that when a tourist is traveling on an MTR, he is passing through Causeway Bay station, how can he know more about the sightseeing site there? Any shopping mall or restaurant there? Where and when to change train? With a view to this, we proposed to develop an LBS system for the rails (MTR and KCR) to give more information about the current location where they are in, and to prompt them to change for other trains. We will still use the LAI + CI information in Symbian OS to achieve this.

7.2 The Experiment

This time we do our experiment on two rail systems – MTR and KCR. We used our data collection kit to note down any cell change event happen during the journey. We travel to and fro different stations for more than five times so as to improve and verify the results.

To illustrate the experiment more clearly, refer to figure 7.2. In figure 7.2, we travel from station A to station B. Since there are three cells in between the two stations, we should experience three cell change events during the journey from A to B. We would then note down these cell change events between stations. These data would then be used to develop our system for the rail.

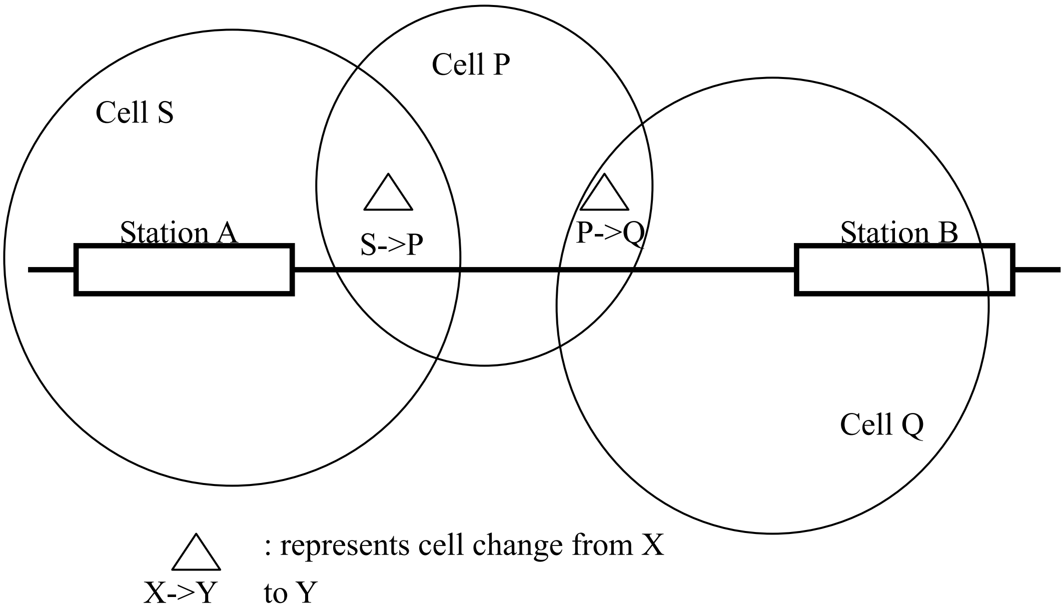


Figure 7.2: Principle of the experiment: records all the cell change event for the future use in developing our LBS system

7.3 Results and Statistics

This time, we also performed the experiment for both telcos, SmarTone and Peoples, on part of the MTR rail and KCR rail. The experimental data are tabulated as follows (left column represents the station transition, while the right represents the new cell entered during the journey) :

For Peoples:

Data for MTR:

Po Lam - Hang Hau	19061
Hang Hau - Tseung Kwan O	19051
Tseung Kwan O - Tiu Keng Leng	19041
Tiu Keng Leng - Yau Tong	19031
Yau Tong - Quarry Bay	9161

Quarry Bay - North Point	9231
Sheung Wan - Central	9031
Central - Admiralty	9011
Adimralty - Wan Chai	9371
Wan Chai - Causeway Bay	9021
Causeway Bay - Tin Hau	9321
Tin Hau - Fortress Hill	9081
Fortress Hill - North Point	9201
North Point - Quarry Bay	9211
Quarry Bay - Tai Koo	9231
Tai Koo - Sai Wan Ho	9291
Sai Wan Ho - Shau Kei Wan	9241
Shau Kei Wan - Heng Fa Chuen	9251
Heng Fa Chuen - Chai Wan	9131, {1381, 10713, 10712, 1381, 1401}
Yau Tong - Lam Tin	9161
Lam Tin - Kwun Tong	{842, 841, 1811
Kwun Tong - Ngau Tau Kok	1012, 931, 933, 3561
Ngau Tau Kok- Kowloon Bay	811, 3562, 11462, 11463, 2112
Kowloon Bay - Choi Hung	11451, 11452, 12131,} 9051
Choi Hung - Diamond Hill	9061
Diamond Hill - Wong Tai Sin	9381
Wong Tai Sin - Lok Fu	9171
Lok Fu - Kowloon Tong	9111
Kowloon Tong - Shek Kip Mei	9271
Shek Kip Mei - Prince Edward	9221
Prince Edward - Mong Kok	9192
Mong Kok - Yau Ma Tei	9391
Yau Mai Tei - Jordan	9101
Jordan - Tsim Sha Tsui	9341
Tsim Sha Tsui - Admiralty	9011

Note: CI in parenthesis represents cells in open area, other cells are in closed area. LAI in closed area is 120, while that in open area is 150

Table 7.1: Experimental results for MTR using Peoples

Data for KCR:

Kowloon Tong - Tai Wai	10203, 4293, 4091, {949, 9481, 661}
Tai Wai - Sha Tin	4511, 11571, 11572, 11573, 4101, 10202
Sha Tin - Fo Tan	4191, 11121, 11123
Fo Tan - University	2443, 10871, 3263, 4782, 10582, 10581, 5531, 4131, 4124

Note: CIs in parenthesis are in tunnel, others are in open environment. LAI in open area is 140, while it is 110 inside the tunnel.

Table 7.2: Experimental results for KCR using Peoples

For SmarTone:

Data for MTR:

Po Lam - Hang Hau	13051
Hang Hau - Tseung Kwan O	13041
Tseung Kwan O - Tiu Keng Leng	13031
Tiu Keng Leng - Yau Tong	13021
Yau Tong - Quarry Bay	13012
Quarry Bay - North Point	4921
Central - Admiralty	4942
Adimralty - Wan Chai	4941
Wan Chai - Causeway Bay	4934
Causeway Bay - Tin Hau	4935
Tin Hau - Fortress Hill	4936
Fortress Hill - North Point	4937
North Point - Quarry Bay	3438
Yau Tong - Lam Tin	42502
Lam Tin - Kwun Tong	12813, 42503, 12812, 42502
Kwun Tong - Ngau Tau Kok	4071, 43352, 40481, 20113, 43231, 43305, 12812
Ngau Tau Kok- Kowloon Bay	20382, 20001, 43352, 4071
Kowloon Bay - Choi Hung	41257, 43611, 9286, 43611, 41256, 4911
Choi Hung - Diamond Hill	4912

Diamond Hill - Wong Tai Sin	4913
Wong Tai Sin - Lok Fu	4914
Lok Fu - Kowloon Tong	81141
Kowloon Tong - Shek Kip Mei	81131
Prince Edward - Mong Kok	4961
Mong Kok - Yau Ma Tei	4962
Yau Mai Tei - Jordan	4963
Jordan - Tsim Sha Tsui	4964
Tsim Sha Tsui - Admiralty	4941
Admiralty - Central	4942

Table 7.3: Experimental results for MTR using SmarTone

For KCR:

Kowloon Tong - Tai Wai	61531, 682, 673, 5793, 30701, 31423, 4243, 4242, 8326
Tai Wai - Sha Tin	30766, 31422, 31983, 203, 30741, 33005, 31981, 8161
Sha Tin - Fo Tan	32611, 18431, 32611, 32353, 33232, 33231, 32042, 7482, 33135
Fo Tan - University	7353, 8341, 32041, 7481, 37513, 33553, 33731, 26391, 61511

Table 7.4: Experimental results for KCR using SmarTone

We discovered several observations associated with GSM cell change in MTR and KCR:

1. For underground stations, there is one particular GSM cell covering each station. Therefore, there is one and only one cell change detected in between any two nearby underground MTR stations. This facilitates the development of LBS since we can clearly identify the location (at which station) we are currently at.

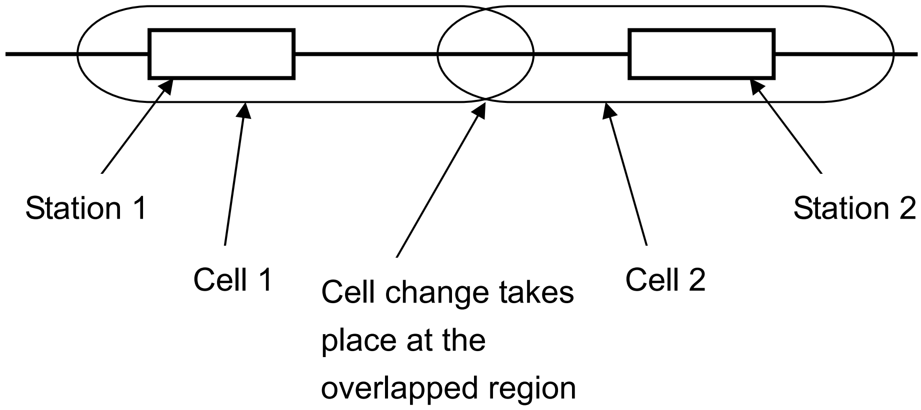


Figure 7.3: In tunnel or underground area, only one cell covers one station.

2. For station that is in open environment, for example Kwun Tong and Chai Wan, several cell changes are detected during a journey from one station to another nearby station. This is because in an open environment, GSM cells outside the MTR station are reachable from the train, so more that one cell can be reached by mobile phones when the train travels in open environment. For example, while traveling from Kwun Tong to Ngau Tau Kok, cell identifier changes from 1012 to 931, from 931 to 933 and finally reached 3561.

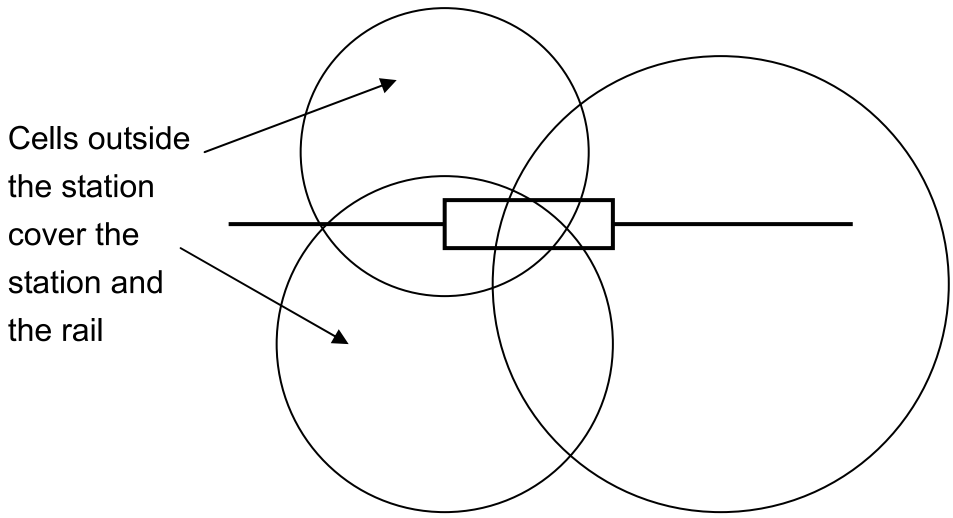


Figure 7.4: In open area, multiple cells may be detected.

3. When the train travels at constant speed, inside a GSM cell, received signal strength is about 100%. However, when the train starts to accelerate or decelerate, the received signal strength drops. It rises to 100% when the speed approaches to a constant. This could be resulted from Doppler effect. When the traveling speed changes, the apparent frequency emitted by GSM antenna varies, thus the mobile phone cannot receive the desired frequency well. Only when the traveling speed becomes steady will the apparent frequency become steady, and makes the signal strength return to 100%.

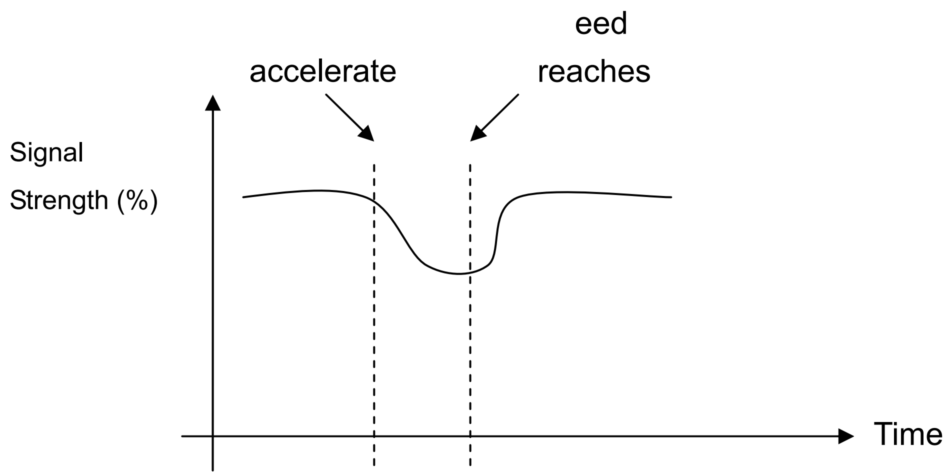


Figure 7.5: Signal strength drops when the train accelerates

4. Location area identity changes when entering or exiting a closed environment (tunnel or underground). This may indicate that GSM cells in closed environment are specially deployed, so as to let users within these places to access to the GSM network. For example, LAI change from 120 to 140 while traveling from Shau Kei Wan (underground) to Heng Fa Chuen (ground level).

7.4 Accuracy Measurement

After looking into the cell coverage in the 1-dimensional space (MTR and KCR in this project), we performed an experiment to investigate the accuracy of our LBS approach used in this space. In this experiment, we tried to measure the time needed for the train to get stopped at a particular station after our program detected a cell change event representing that we are entering that particular station.

Since the size of overlapping of cell is not negligible, a cell change event may happen at any point within the overlapping region. For example in figure 7.6, the route g to h passes through a cell overlap region. A cell change event may take place at any point inside the overlap region, for example point U or V. The place at which cell change take place is very dependent on the environmental condition, which affects the signal strength received by the mobile phone.

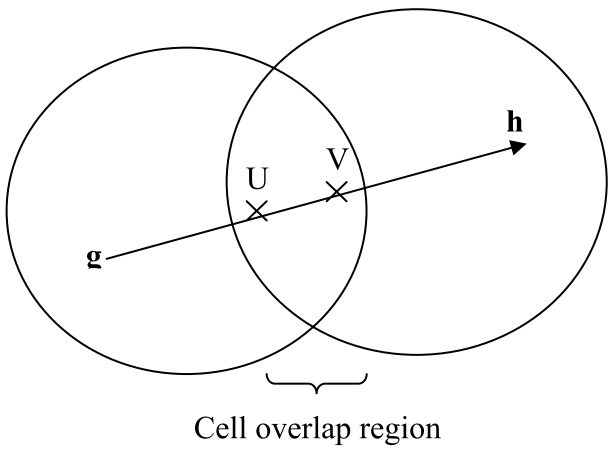
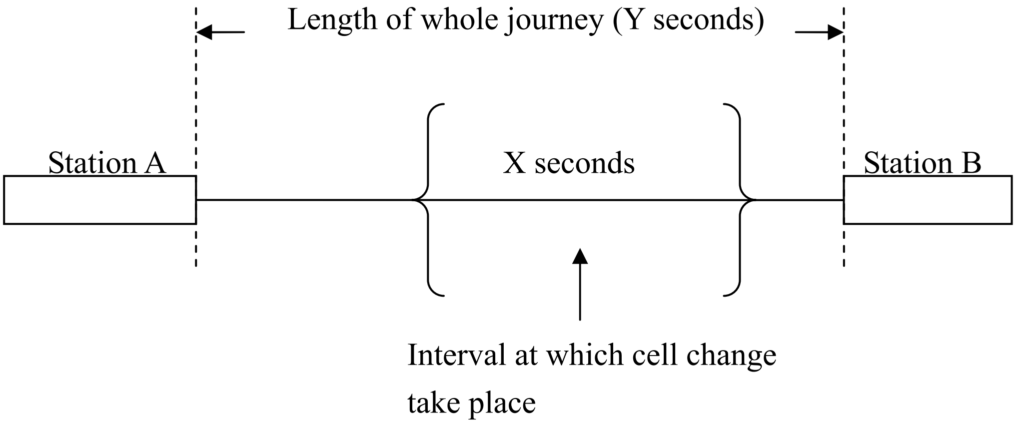


Figure 7.6: Cell change may take place at any point inside the overlap region

By finding out the portion of time in which cell change event may occur in the journey from one station to another nearby station, we can estimate the accuracy of using this LBS approach in the MTR.

We used time as our measurement metric since the speed of an MTR varies greatly in a journey, which make it hard to find out the exact length of the journey in term of distance.

We first estimate the average time needed for a train to go from one station to another station, and then we find out the interval of time at which cell change events occur by repeating the experiment several times..



If the interval at which cell change may take place is small compared with the length of whole journey, then the result is said to be quite accurate since all the cell change events happen nearly at the same point in the journey.

Table 7.6 shows the result of the experiment we done at some MTR station. All the data are taken in the metric “second”. We performed the experiment at different dates so as to get a more general result, since the environmental condition varies from day to day.

The column “Variation of time” shows the accuracy we obtained. The term “Variation of time” here we mean the interval of time at which cell change occur within the whole journey. We find out that most of the results we got are quite accurate. They are mostly around 10-15% of the length in the whole journey.

From	To	Average Total Time of Journey	11/17/2003	11/18/2003	11/20/2003	11/21/2003	11/22/2003	11/24/2003	11/25/2003	Variation of Time
Po Lam	Hang Hau	83.4	32.0	34.5	35.3	39.1	38.1	36.8	36.4	0.085
Hang Hau	Tseung Kwan O	106.3	45.0	56.4	71.5*	49.0	61.0	49.6	57.3	0.112
Tseung Kwan O	Tiu Keng Leng	74.0	21.0	29.2	30.5	16.8	31.1	18.4	X	0.136
Tiu Keng Leng	Yau Tong	136.8	42.1	62.7*	43.0	46.1	47.9	41.5	39.0	0.086
Yau Tong	Lam Tin	103.4	1.0	X	4.5	X	X	1.6	1.8	0.034
Choi Hung	Diamond Hill	79.8	34.3	31.4	38.6	X	X	27.1	31.5	0.144
Diamond Hill	Wong Tai Sin	74.5	16.8	22.2	23.3	X	X	21.4	21.8	0.087
Lok Fu	Kowloon Tong	97.2	25.0	60.0*	X	X	X	16.6	47.4	0.317
Kowloon Tong	Lok Fu	74.5	18.5	13.2	14.9	7.1	X	18.9	14.7	0.158
Wong Tai Sin	Diamond Hill	60.0	28.7	28.9	30.0	32.1	36.2	15.6	X	0.343
Diamond Hill	Choi Hung	83.8	36.8	36.7	32.4	33.9	33.3	32.1	35.4	0.056
Lam Tin	Yau Tong	80.2	59.5	40.0	44.1	44.0	45.3	41.7	43.1	0.243
Yau Tong	Tiu Keng Leng	150.7	78.0	81.3	70.6	91.6	88.3	91.3	84.3	0.139
Tiu Keng Leng	Tseung Kwan O	77.4	27.0	23.2	28.3	29.9	28.3	29.9	31.1	0.102
Tseung Kwan O	Hang Hau	58.5	46.3	31.8	37.7	39.2	46.7	37.5	X	0.254
Hang Hau	Po Lam	87.8	X	22.7	X	38.1	35.5	30.9	25.4	0.175

Note: X: Not recorded or MTR Traveller cannot detect the current station

*: The train travelled at an abnormal speed (e.g. stopping the train on the way between two stations)

Average speed of Hong Kong MTR should be 33km/h according to the webmaster of Hong Kong Rail Engineering Centre

Table 7.5: Statistics on the time needed to reach a station after our program prompts

7.5 Conclusion to the Experiments

The experiment shows that LBS in 1-dimensional space are feasible. No matter in closed environment or in open environment, we can easily determine which region or station we are at in the whole 1D route. For in close area, only one cell covering a station gives us a clear-cut of the realm covered by a particular base station, thus we are sure to be in a specific region without uncertainty. For in open area, several cells may be accessible to the route during a journey. This gives us higher accuracy of determining the location we are going to. For example, there are five cells (A to E) in between the journey from g to h. If we experience cell change events A->B, B->C and C->D, then we are quite sure that we are going to h before we exactly reach h. This extra information helps us to predict our destination.

LBS in 1-dimensional space can be used not only in MTR, KCR and buses. Everything that has a path can utilize this service. For example we can implement a system for passenger to use, so that the system will prompt the passenger when he should get off the bus, or prompt him when it is time to change for other means of transport. Furthermore, we can implement a system that warns the driver when he is entering a region where there would be speed inspection so that he could reduce his speed in time. Lastly, we could also use it to calculate a more accurate location in 2D space by the “history” of his path taken.

7.6 MTR Traveller

With the 1-dimensional data in MTR and KCR, we implemented a program named MTR Traveller. As its name implies, it is an electronic travel agent used in MTR (and part of KCR). It is still a prototype of the whole system.

7.6.1 The Function

In MTR Traveller, we have implemented function that tells the user his current location in the journey. It also displays the corresponding map of the railway for the user to refer to.

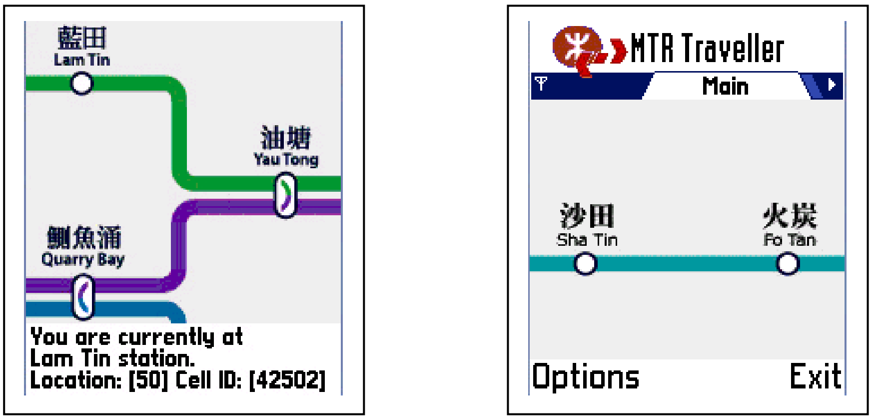


Figure 7.6: Screenshots for MTR Traveller, an electronic travel agent used in MTR and KCR

For example, when we are traveling from Causeway Bay to Wan Chai, the system will prompt the user that he is going from Causeway Bay to Wan Chai. It also displays the cellular information onto the screen. We could add more useful functions onto this program. For example, we could add a function that find the shortest path for going from one station to another station; we could make the program more informative by adding other public transport, like bus, that a user can change at that particular station; we could also add some information about the sightseeing sites nearby that station for tourist to go. All these can easily be implemented onto MTR Traveler. And we suppose we would do so in the next semester.

7.6.2 Program Architecture and Concept

This program is conceptually simple, though not easy to implement. In this program, there are three main components: the location sensor, the database controller and the graphical display unit.

The location sensor is in fact the extension of GSM Status. It detects the cell change event in every one second interval. If a cell change event is detected, it makes a query to the database controller for the current location using the current and the new LAI and CI pair. The database controller, upon receiving the query, returns with the names and GUI positions of the source station and destination station. These information are then displayed to the users.

We store all the information, including the LAI CI pairs, station name, and GUI positions into a database bundled with the Symbian OS. The DBMS inside supports simple data query only, no aggregate function is supported.

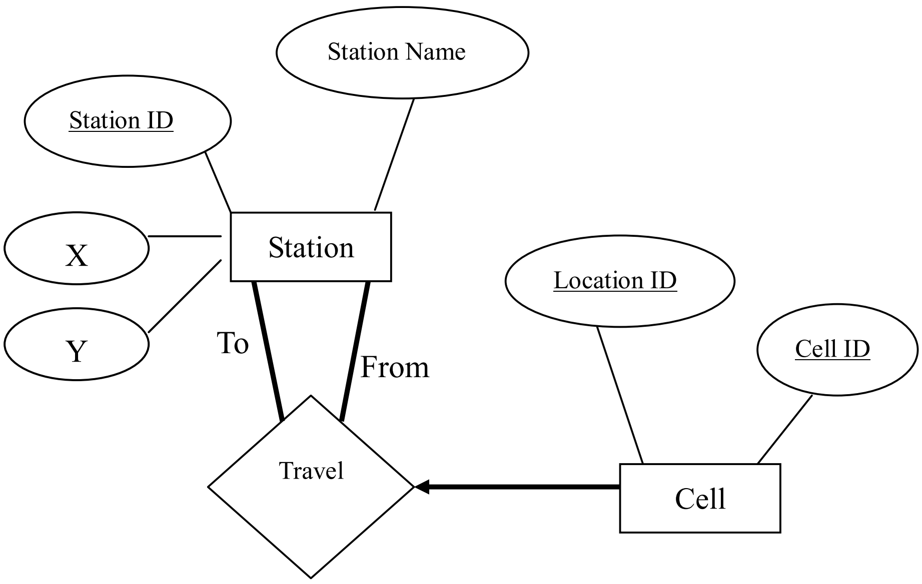


Figure 7.7: Entity-relationship diagram for the database we used

We used two tables to store the data for the transition from one cell to another cell during a journey from one station to another station. In the tables we record the data of stations, new cell entered and the transition information, which includes station IDs of the two stations, and the new cell discovered.

Schema of the tables we used:
Station (Station ID, Station Name)
Transition (Station ID 1, Station ID 2, Location ID, Cell ID, X, Y)

The GUI display in fact only displays one bitmap for the whole program. This bitmap includes all the necessary graphics for each station we did experiment. The GUI position required from the database indicates which portion of the bitmap is to be displayed. The GUI position is in fact an (X,Y) coordinate, which is used as the starting point (top-left corner) of the bitmap to be displayed.

8 Conclusion

In this project, we have done quite a number of things. We studied the characteristic and behaviour of the Symbian OS. We also learnt how to program applications used in Symbian OS. Furthermore, we studied some GSM topics and technologies that are useful to our project. They includes the system structure and addressing of GSM network, cellular information and cell reselection.

With the basic idea in programming for Symbian OS and GSM technologies, we then investigated current LBS solutions and try to figure out there relative advantages and disadvantages. Then we try to look into the capability of using cell information (location area identifier and cell identifier). The result turned out showed that the accuracy in 2-dimensional space is not accurate without any help from the telco. Then we started to look into the application in 1-dimensional space. The constraints in this space help to eliminate the uncertainties in 2-dimensional space.

Because of the accuracy and certainty in this space, we began to work in the 1-dimensional space. We came up with an idea of providing service and information to tourist in the transportation system. Thus we developed a system called MTR Traveller. This program will be our foundation of the future work we may carry out in the future.

9. Acknowledgement

We would like to thank our final year project supervisor, Professor Michael Lyu, who gave us unlimited support and invaluable advice. He even arranged meeting for us to discuss our project with people from ASTRI. We really want to thank him for his kindness and support.

We would also like to thank Mr. Edward Yau, for he gave a lot of useful ideas and technical help in our project. He enriched our project by providing us with a lot of information we need.

10. Future Work

There are quite a number of things that we can do. However, we will focus on the following field in the future:

- Implement an automatic cell information collection kit, which will be able to automatically collect data, input them into the DBMS
- Improve the MTR Traveller by adding the adding functions or enhancing its by
    - Allowing personalization
    - Making a more attractive user interface
    - Making the program more informative
    - Allow users to update information (eg: data stored in the database) using GPRS to keep information up-to-date
- Producing a generic middleware or library for developers to develop LBS using the LAI and CI approach.

11. Reference

[1] M. Tasker, Professional Symbian Programming, Wrox, Birmingham, 2002

[2] R. Harrison, Symbian OS C++ for Mobile Phones, Wiley, New York, April 2003

[4] Digia, Programming for the Series 60 Platform and Symbian OS, Wiley, New Your, November 2002

[5] M.J. Jipping, “Telephony,” Symbian OS Communications Programming, Wiley, New York, pp. 350-372, June 2002

[6] “Symbian OS – the Mobile Operating System,” Symbian Ltd., 2003 [Online]. Available: http://www.symbian.com

[7] “Nokia on the Web,” Nokia Ltd., 2003 [Online]. Available: http://www.nokia.com

[8] “Developer Discussion Boards – Symbian,” Nokia Ltd., 2003 [Online]. Available: http://discussion.forum.nokia.com/forum

[9] “Symbian OS Communicators and Smartphones Info Center,” 2002 [Online]. Available: http://my-symbian.com

[10] “All About Symbian,” 2003 [Online]. Available: http://allaboutsymbian.com

[11] J. Eberspacher, H. Vogel and C. Bettstetter, GSM Switching, Services and Protocols, 2nd ed., Wiley, Chichester, pp. 9-92, 2001

[12] M. Mouly and M.B. Pautet, “Current Evolution of the GSM Systems,” IEEE Personal Communications Magazine, October, pp. 9-19, 1995

[13] W.C.Y. Lee, Mobile Cellular Telecommunication Systems,

McGraw-Hill, New York, 1989

[14] “The Website of the GSM Association,” GSM Association, 2003 [Online]. Available: http://www.gsmworld.com
[15] J. Tisal, GSM Cellular Radio Telephony, Wiley, New York, 1997

[16] M. Mouly, M.B. Pautet, The GSM System for Mobile Communications, Palaiseau, France, 1992

[17] S.Y. Willassen, “Positioning a Mobile Station,” March 1998 [Online]. Available: http://www.willassen.no/msl/node6.html

[18] S. Buckingham, “Mobile Positioning – An Introduction,” December 1999 [Online]. Available: http://www.mobilepositioning.com

[19] J.H. Yap, S. Ghaheri-Niri and R. Tafazolli, “Accuracy and Hearability of Mobile Positioning in GSM and CDMA Networks,” 3rd International Conference on 3G Mobile Communication Technologies, pp. 350-354, 2002

[20] J. Costa-Requena, H. Tang and I. Espigares, “Consistent LBS Solution in Next Generations of Mobile Internet,” Parallel and Distributed Systems, Ninth International Conference, Proceedings, pp 637-642, December 2002

[21] “Garmin: What is GPS,” Garmin Ltd., 2003 [Online]. Available: http://www.garmin.com/aboutGPS

[22] “Global Positioning System Overview,” University of Colorado, 2000 [Online]. Available: http://colorado.edu/geograhpy/gcraft/notes/gps/gps.html

[23] K. Nakamura. “The GPS Resource Library,” 2001 [Online]. Available: http://gpsy.com/gpsinfo

[24] “Wireless World Forum,” W2F Ltd., 2003 [Online] Available: http://www.w2forum.com

[25] Bostock and Pollitt, “CPS Location Based Service”, Cambridge Positioning System Ltd., 2003 [Online]. Available: http://www.cursor-system.com

[26] N. Krishnamurthy, “Using SMS to Deliver Location-based Services,” Personal Wireless Communications, 2002 IEEE International Conference, pp. 177-181, December 2002

[27] K. Chadha, “The Global Positioning System: challenges in brining GPS to mainstream consumers,” Solid-State Circuit Conference, February 1998

[28] M. Goller, “Application of GSM in High Speed Trains: Measurements and Simulations,” Radiocommunicataions in Transportation, IEEE Colloquium, May 1995

[29] C. Drane, M. Macnaughtan and C. Scott, “Positioning GSM Telephones,” Communications Magazine, IEEE, vol.50, Issue 4, pp. 44-54,59, April 1998

[30] C.L.C. Wong, M.C. Lee and R.K.W. Chan, “GSM-based Mobile Positioning using WAP,” Wireless Communications and Networking Conference, September 2000

[31] M.P. Wylie-Green, P. Wang, “GSM Mobile Positioning Simulator,” Emerging Technologies Symposium: Broadband, Wireless Internet Access, April 2002

[32] E. Villier, L. Lopes and B. Ludden, “Performance of a Handset-assisted Positioning Method for GSM,” Vehicular Technology Conference, 1999 IEEE 49th, vol. 3, May 1999

[33] M. Silventoinen and T. Rantalainen, “Mobile Station Locating in GSM,” Wireless Communication System Symposium, IEEE, November 1995

[34] M.A. Spriito and M.P. Wylie-Green, “Mobile Stations Location in Future TDMA Mobile Communication Systems,” Vehicular Technology Conference, IEEE VTS 50th, vol. 2, September 1999

[36] S.B. Guthery, Mobile Application Development with SMS and the SIM Toolkit, McGraw-Hill, New York, 2002

[37] L.B. Gwenael, MobileMessaging Technologies and Service: SMS, EMS and MMS, Wiley, Chichester, 2003

[38] J.Y.B. Lin, Wireless and Mobile Network Architectures, Wiley, New York, pp. 1-12, 2001

[39] T. Gunstone, “SMS and Location-based Service in the Travel Industry,” October 2001 [Online]. Available: http://m-travel.com/11009a.shtml

[40] “Location Based Services: Heading in the Right Direction,” January 2001 [Online]. Available: http://www/geoplace.com/ge/2001/0101/0101lbs.asp