Collaboration of Mobile Commerce and Mobile GIS

A Dissertation Submitted To The Central Department Of Computer Science And Information Technology Tribhuvan University, Kathmandu, Nepal

For The Partial Fulfillment Of

Masters Degree Of Computer Science And Information Technology

By: Dipesh Man Joshi

[CDCSIT-TU, Nepal]

March, 2007

Tribhuvan University

Central Department of Computer Science and Information Technology

Date : _______________

LETTER OF RECOMMENDATION

The undersigned certify that I have read and recommended to the faculty of Central Department of Computer Science and Information technology for acceptance, a thesis entitled "**Collaboration of Mobile Commerce and Mobile GIS**" submitted by Dipesh Man Joshi in partial fulfillment of the degree of Master of Computer Science and Information technology. I, therefore recommend for further evaluation.

____

Mr. Lochan Lal Amatya

Member, Computer Science Subject Committee, CDCSIT-TU

Deputy Manager

Central Office, Bhadrakali Plaza

Nepal Telecom (NTC), Nepal

(Supervisor)

Tribhuvan University

Central Department of Computer Science and Information Technology

Date: _______________

LETTER OF APPROVAL

We certify that we have read this dissertation and in our opinion it is satisfactory in the scope and quality as a dissertation in the partial fulfillment for the requirement of Master's Degree In Computer Science And Information Technology.

Evaluation committee

______________________	______________________
Acting Head Mr. Min Bahadur Khati CDCSIT-TU, Nepal (Project Supervisor)	Mr. Lochan Lal Amatya Deputy Manager Central Office, Bhadrakali Plaza
______________________	______________________
(Internal Examiner)	(External Examiner)

Abstract

Advances in Internet and network technology and the rapidly growing number of mobile personal devices result in the fast growth of Mobile E-Commerce, M-Commerce. In this research we try to analyze the framework for the integration of Mobile Commerce and Mobile GIS System: A special case of integrating M-Commerce with Mobile GIS System where often the actual position of the terminal is used in the service provision. The whole study in concentrated on the requirements for the collaborated system. Because technology is an important facilitator and at the same time a limiting factor, were view shortly the technology aspects relevant for integrated system. We then present the most important user, system and infrastructure requirements. Finally we present a design and implementation of an integrated system prototype running on Java handsets and evaluate its properties and review. The integrated mobility system processes spatial cognition involving the knowledge of geographic facts and the ability to navigate around our environment.

Mobile GIS can add both geospatial information and global positional systems (GPS) coordinates from remotely located field-based personnel to spatial decision support systems (SDSS). By adopting broadband wireless telecommunication technology for connecting Internet GIS and mobile GIS devices, decision makers can gather near real-time information from field personnel and, equally quickly, distribute updated information back to the field.

Acknowledgement

I would like to thank and express my sincere appreciation to my thesis supervisor, Mr. Lochan Lal Amatya for his continuous support and genuine interest during my master studies. He has always shown faith in my abilities by sharing responsibilities and opportunities with me.

I would also like to thank the many professors, students, and staff in the Central Department of Computer Science and Information Technology who have made my time fruitful and enjoyable. I would thankful to Prof. Devi Dutt paudyal for his inspiration and encouragement throughout the whole span of Master degree.

I would like to express gratitude to thesis evaluation committee: Prof. Dr. Devi Dutta Paudyal and Prof. Dr. Subarna Shakya, also to all my professors, lecturers and teachers namely Prof. Dr. Laxmi Pd. Gawali, Prof. Dr. Shree Nath Shrivastava,mr. Sashidhar Ram Joshi, Mr. Sudarshan Karanjit , Mr. Arun Timilsina, Assist. Prof. Min Bahadur Khati who have taught in Master Degree and encourage in the field of Computer Science and Information technology.

Lastly I would like to thanks all my colleagues for their consistent effort and providing inspiration for me.

Dipesh Man Joshi

List Of Figures

List Of Tables

List of Abbreviations

3G	:	Third Generation
3GPP	:	Third Generation Partnership Program
AGPS	:	Assisted GPS
ANI	:	Automatic Number Identification
ALI	:	Automatic Location Information
AOA	:	Angle Of Arrival
CDMA	:	Code-Division Multiple Access
CGI	:	Cell Global Identity
CDPD	:	Cellular Digital Packet Data
DBMS	:	Database Management System
DGPS	:	Differential GPS
E-OTD	:	Enhanced Offset Time Division
FCC	:	Federal Communications Commission
GIS	:	Geographic Information System
GSM	:	Global System for Mobile Communication
GPRS	:	General Packet Radio Service
GPS	:	Global Positioning System
HLR	:	Home Location Register
IN	:	Intelligent Network
LBS	:	Location Based Service
LDT	:	Location Determine Technology
J2ME	:	Java 2 Micro Edition
J2EE	:	Java 2 Enterprise Edition
LBS	:	Location Based Services
MSC	:	Mobile Switching Center

MA	:	Mobile Agent
MIDP	:	Mobile Information Device Profile
OO	:	Object Oriented
OSG	:	Open GIS Consortium
OTDOA	:	Observed Timed Difference of Arrival
PDA	:	Portable Digital Assistant
PDC	:	Personal/Pacific Digital Cellular
PDE	:	Position Determination Equipment
SA	:	Selective Availability
SLoP	:	Spatial Location Protocol
STP	:	Signal Transfer Points
SMS	:	Short Message Services
SVG	:	Scalable Vector Format
TDMA	:	Time Division Multiple Access
TOA	:	Time Of Arrival
UMTS	:	Universal Mobile Telecommunications System
URL	:	Uniform Resource Locator
VLR	:	Visitor Location Register
W-CDMA	:	Wideband-CDMA
WWW	:	World Wide Web
XML	:	Extensible Markup Language

Table Of Contents

Chapter 1: Introduction

1.1 Background

Today, information technology is changing rapidly. Wireless services and Internet services are two successful businesses in communications today. The developments from the fixed line Internet to the wireless Internet have attracted a lot of attention. As the wireless Internet will be merged with the existing one through future technologies, common forecasts predict huge business opportunities, when the mobile telephone world will join with the IP world.

Mobile commerce applications are supposed to become an important area of electronic commerce generating substantial profits. Various mobile commerce providers have built their business plans for third generation networks based on this assumption. Therefore, it is necessary for them to participate in mobile commerce at an early stage to be able to acquire prospective customers and thus to reach a competitive starting position for their future businesses.

Emerging technologies are changing the future use of geographic information, moving geographic information systems (GISs) from the office desktops into the users' hands. The objective of this thesis is to investigate basic issues associated with information retrieval systems from GIS environments and integrate with Mobile commerce. In this scope, we explore mobile client-server techniques, an integrated prototype integrating mobile commerce with GIS environments.

Collaborative GIS are required to support interactive spatial query and visualization of geospatial problems from its participants. By combining Mobile GIS and Mobile Commerce, the new framework of collaborative GIS can provide more interactive GIS functions and real-time data update from distributed participants and decision makers. However, we need to establish a seamless and robust linkage between Mobile GIS and Mobile Commerce in order to provide such high-level services. An integrated spatial decision support system will rely on three major components:

Mobile GIS, Mobile Commerce, and broadband wireless communication networks[23]. Each component will need to be customized in order to provide real-time or near real-time GIS functions. Interoperability and upgradeability are two key issues for successful system integration and long-term operation because these GIS technologies change rapidly.

1.2 Objective

The main objective of this thesis work is to collaborate the Mobile Commerce technology with Mobile GIS System. By enhancing M-Commerce application that uses Mobiles GIS service for decision making and locating location effectively increase M-Business opportunities. Integration of M-Commerce and Mobile GIS technology results robust M-Commerce application that user GIS feature to provide efficient service to the mobile user and the enterprise.

The other main works to be performed in this research work is to design the framework for Mobile Commerce and Mobile GIS integration by studying individual framework. Then implement shortest path algorithm to the GIS data provided via Mobile GIS system which is in XML format and utilize it in M-commerce Application.

1.3 Thesis Outline

The Study is divided into 9 chapters. Chapter 2 deals with "Mobile Commerce and Mobile System". It describes the Mobile Commerce system, its requirements and various Mobile Environments. Moreover, it briefly describes the Mobile Commerce functionality and Mobile System.

Chapter 3, describes the GIS functionality, Geographic data models and its topography. It describes what is Mobile GIS and Location Based Service.

Chapter 4, describes about Mobile E-Commerce Mobile GIS System :Technology and Requirements. It raises the issues about the requirement and implementation framework of Mobile GIS.

Chapter 5, describes about Mobile GIS and Mobile Commerce integration framework. This chapter clarifies the architectural solutions for the collaborated system. It explains the distributed architecture as a prototype system where Application, Data, Presentation and Client tier communicates.

Chapter 6, describes the methodology used in this research work, where we first demonstrate the xml format of GIS data and applying the shortest path in xml formatted GIS data. The sample of M-Commerce application is developed and then the research enclosed the method for communication of M-Commerce Application with Mobile GIS System. To Simulate Mobile GIS System we have incorporate Microsoft MapPoint web services.

Chapter 7, Discusses about the issues in implementation, challenges that face during the research works. It also explains the limitation of current mobile system and technologies and securities issues along with the market for collaborative system.

Chapter 8 deals with future trends and challenges with new and advanced services that are upcoming. It also discussed cross-disciplinary research in both system.

Chapter 9, discusses about the experiment that we have conducted on Chapter 6 : Methodology part and its result and its impact and summarized the result as conclusion of the study.

Chapter 2: Mobile Commerce and Mobile System

2.1 Introduction

With the introduction of the World Wide Web, electronic commerce has revolutionized traditional commerce and boosted sales and exchanges of merchandise and information. Recently, the emergence of wireless and mobile networks has made possible the admission of electronic commerce to a new application and research subject: mobile commerce, which is defined as the exchange or buying and selling of commodities, services, or information on the Internet through the use of mobile handheld devices. In just a few years, mobile commerce has emerged from nowhere to become the hottest new trend in business transactions.

Mobile commerce is an effective and convenient way to deliver electronic commerce to consumers from anywhere and at anytime. Realizing the advantages to be gained from mobile commerce, many major companies have begun to offer mobile commerce options for their customers in addition to the electronic commerce they already provide (Yankee Group, 2002). However, it requires a tremendous effort to understand or construct a mobile commerce system because it involves such a wide range of disciplines and technologies. A mobile commerce system has six components[22]:

(i) mobile commerce applications,

(ii) mobile stations,

(iii) mobile middleware,

(iv) wireless networks,

(v) wired networks, and

(vi) host computers.

2.2 Requirements of a Mobile Commerce System

It is first necessary to examine what kind of features a mobile commerce system is expected to have in order to conduct effective and efficient mobile commerce transactions and what kind of challenges may be faced in the process of developing new mobile commerce systems. The requirements for a mobile commerce system are[24]:

It should allow end users to perform mobile commerce transactions easily, in a timely manner, and ubiquitously.

It should allow products to be personalized or customized upon request.

It should fully support a wide range of mobile commerce applications for content providers.

Maximum interoperability is desirable because so many technologies are now available and new techniques are constantly being invented for the use of mobile commerce systems.

Program/data independence is held, that is, changing the system components will not affect the existing programs/data.

End-to-end security is rigorously enforced.

2.3 Mobile Commerce Applications

The applications of electronic commerce are already widespread; mobile commerce applications not only cover these but also include new ones. For example, some tasks that are not feasible for electronic commerce, such as mobile inventory tracking and dispatching, are possible for mobile commerce. Table 2.1 lists some of the major mobile commerce applications [20] along with details of each.

Mobile Category	**Major Applications**	**Clients**
Commerce	Mobile transactions and payments	Businesses
Education	Mobile classrooms and labs	Schools and training centers
Enterprise resource planning	Resource management	All
Entertainment	Games/images/music/video downloads and online gaming	Entertainment industry
Health care	Accessing and updating patient records	Hospitals and nursing homes

Mobile Category	Major Applications	Clients
Inventory tracking and dispatching	Product tracking and dispatching	Delivery services and transportation
Traffic	Global positioning, directions, and traffic advisories	Transportation and auto industries
Travel and ticketing	Travel management	Travel industry and ticket sales

Table 2.1: Major mobile commerce applications

2.4 M-Commerce: Distinction from E-Commerce

There are two user-oriented core dimensions on which m-commerce has an advantage over e-commerce: "mobility" and "locatability". Simple CIP devices offer mobility. Mobile data networks that employ geographical positioning systems or use network elements to pinpoint the "cell" where the user is offer locatability. With locatability, CIP devices are transformed into CLIP (communication, location, information, payment) devices. Taken together, these two dimensions create a range of distinctions between m-commerce and conventional e-commerce [25](see Table 2.2).

Dimension	E-Commerce	M-Commerce
Core Dimensions		
Mobility	Limited: User can transact from locations with Internet access.	Ubiquitous: User can transact from anywhere in mobile network area.
Locatability	Client-specific: Client computer locatable via IP address.	User-specific: User, device, and geographic location can be identified.
Behavioral, Strategic, and Leadership Dimensions		
Key customer concern	Money: Because of "free Internet" culture, e-commerce users are money conscious.	Time: Evolving from mobile telephony culture, m-commerce users are time conscious. They are used to paying for mobile services.
Customer location and market served	Fixed Locations: Customers can be served at locations where they have Internet-linked computer access.	Ubiquitous and Global: Customers can be served anywhere within the mobile network coverage area.
National and regional leadership	USA-centric: E-commerce evolved in USA.	Europe and Asia-centric: Scandinavia and Asian nations (Japan) lead in m-commerce.
Industry leadership	New Players: Newer companies such as Yahoo, AOL, Amazon.com, Dell, Cisco, and FreeMarkets have emerged as E-commerce leaders.	Transformed Old Players: Older players such as FSPs, Mobile Device Makers, WSPs, and Banks have chance to grab leadership.1

Table 2.2: Distinction between m-commerce and e-commerce

Do the differences between e-commerce and m-commerce outlined in Table 2 .2 translate into differences in terms of business strategies for these two types of

Internet-based commerce? We believe that m-commerce strategies will be different from e-commerce strategies in significant ways. In particular, we focus on three aspects of the customer relationship—personalization, permission, and specification—where m-commerce is likely to differ from e-commerce.

2.5 Mobile Environment

The current market provides wide range of mobile terminals from usual cellular mobile phones to laptop computers. All these devices can be distinguished from each other by functionality, physics characteristics, and destination. It is very difficult to categorise mobile devices because their features vary a lot. We can propose the following basic groups, but note that the borders between them are blurring:

Normal cellular mobile phones have very small display, they are able to send and receive text messages with SMS (Short Messages Service);

WAP-phones have built-in micro-browser, which can interpret WML and WMLScript, the display is small (e.g. Ericsson R380, Nokia 7110);

3G –Phone that has own operating system with wi-fi tech and able to browse internet directly.

Communicators, a PDA-type equipment integrated with or attached to a mobile phone for data and voice transfer, usually have a small text keyboard, can run simple applications and connect to the Internet (e.g. Nokia Communicator 9110, Nokia Communicator 9110i, Ericsson MC218).

Note that the last two devices have also WAP stack and can run a WAP application ;

Laptop PC includes all the sub-notebook sized equipment.

2.6 Mobile Cellular Systems

First Generation

The first generation of mobile cellular telecommunications systems appeared in the 1980s. The first generation was not the beginning of mobile communications, as there were several mobile radio networks in existence before then, but they were not cellular systems either. The capacity of these early networks was much lower than that of cellular networks, and the support for mobility was weaker. In mobile cellular networks the coverage area is divided into small cells, and thus the same frequencies can be used several times in the network without disruptive interference. This increases the system capacity. The first generation used analog transmission techniques for traffic, which was almost entirely voice. There was no dominant standard but several competing ones.

Second Generation

The second-generation (2G) mobile cellular systems use digital radio transmission for traffic. Thus, the boundary line between first- and second generation systems is obvious: It is the analog/digital split. The 2G networks have much higher capacity than the first-generation systems. One frequency channel is simultaneously divided among several users (either by code or time division). Hierarchical cell structures—in which the service area is covered by macro cells, micro cells, and pico cells—enhance the system capacity even further. There are four main standards for 2G systems: Global System for Mobile(GSM) communications and its derivatives; digital AMPS (D-AMPS); code division multiple access (CDMA) IS-95; and personal digital cellular (PDC).GSM is by far the most successful and widely used 2G system[4].

Generation 2.5

"Generation 2.5" is a designation that broadly includes all advanced upgrades for the 2G networks. These upgrades may in fact sometimes provide almost the same capabilities as the planned 3G systems. The boundary line between 2G and 2.5G is a hazy one. It is difficult to say when a 2G becomes a 2.5G system in a technical sense. Generally, a 2.5G GSM system includes at least one of the following technologies: high-speed circuit-switched data (HSCSD), General Packet Radio Services (GPRS), and Enhanced Data Rates for Global Evolution (EDGE). An IS-136 system becomes

2.5G with the introduction of GPRS and EDGE, and an IS-95 system is called 2.5G when it implements IS-95B, orCDMA2000 1xRTT upgrades. The biggest problem with plain GSM is its low air interface data rates. The basic GSM could originally provide only a 9.6-Kbps user data rate. Later, 14.4-Kbps data rate was specified, although it is not commonly used. With GPRS technology, the data rates can be pushed up to 115 Kbps, or even higher if one can forget error correction. However, with adequate data protection, the widely quoted 115 Kbps is the theoretical maximum in optimal radio conditions with eight downlink timeslots. A good approximation for throughput in "average" conditions is 10Kbps per time slot. What is even more important than the increased throughput is that GPRS is packet switched, and thus it does not allocate the radio resources continuously but only when there is something to be sent. The maximum theoretical data rate is achieved when eight time slots are used continuously[4]. The first commercial launches for GPRS took place in2001. GPRS is especially suitable for non-real-time applications, such as e-mail and Web surfing. Also, bursty data is well handled with GPRS, as it can adjust the assigned resources according to current needs.

Overview of 3G System

There have been (and still are) several competing proposals for a global 3Gstandard. Below, these are grouped based on their basic technology, WCDMA, advanced TDMA, hybrid CDMA/TDMA, and orthogonal frequency division multiplexing (OFDM). WCDMA By definition, the bandwidth of a WCDMA system is 5 MHz or more, and this 5 MHz is also the nominal bandwidth of all 3G WCDMA proposals[4]. This bandwidth was chosen because:

- It is enough to provide data rates of 144 and 384 Kbps (these were 3Gtargets), and even 2 Mbps in good conditions.
- Bandwidth is always scarce, and the smallest possible allocation should be used, especially if the system must use frequency bands already occupied by existing 2G systems.
- Advanced TDMA Serious research was conducted around advanced TDMA systems in the1990s. For some time, the European 3G research

was concentrated around TDMA systems, and CDMA was seen only as a secondary alternative.

Hybrid CDMA/TDMA This solution was examined in the European FRAMES project. It was also the original ETSI UMTS radio interface scheme. Each TDMA frame is divided into eight time slots and within each time slot the different channels are multiplexed using CDMA. This frame structure would have been backward compatible with GSM. This particular ETSI proposal is no longer supported. OFDMOFDM is based on a principle of multi carrier modulation, which means dividing a data stream into several bit streams (sub channels), each of which has a much lower bit rate than the parent data stream. These sub-streams are then modulated using codes that are orthogonal to each other. Because of their orthogonality, the sub carriers can be very close to each other (or even partly overlapping) in the frequency spectrum without interfering each other.

Chapter 3 : Geographic Information

3.1 GIS Overview

There are very many definitions for Geographic Information System (GIS) and they are changing all the time along with enhancement of computer technologies and approaches in the field of GIS applications. According to the AGI (Association for Geographic Information) GIS Dictionary , GIS can be defined as: a computer system for capturing, storing, checking, integrating, manipulating, analyzing and displaying data related to positions on the Earth's surface. Another variant by ESRI: GIS is an organized collection of computer hardware, software, geographic data, and personnel design to efficiently capture, store, update, manipulate, analyze, and display all forms of geographically referenced information. Mobile computing has brought new opportunities for GIS applications and opened up a new class of services for handheld and palmtop computers. We will deal with a specialized GIS, which will work in mobile environment. However, the environment limits the features of the geographical system.

For objects description GIS uses information that consists of two components: location information, which reflects geographic features of the object, and descriptive information, which describe any other object features. Thus, a hybrid data model, which combines location data and descriptive attribute data, is used for managing such features. The spatial and attribute data are linked in such a way that both of the components are available in a combination.

There are two widely used basic spatial data models: vector data model and raster data model[11]. They will be described in the next section. These spatial data models are then implemented into a data management system with associated data structures. The applied data structures may vary among different GIS software packages. In addition, the models are augmented with the attribute data, stored in a set of tables.

3.2 Geographic Data Models

As was noted above, the two basic GIS data models are vector and raster models. In addition, there are also other data models: TIN (Triangulated Irregular Network), which is used primarily for representing three dimensional continuous surfaces, and lattice [15].

In the raster data model the surface of the earth is represented as a regular grid of cells (pixels) associated with a value describing some property of the defined area. A location in the raster data can be determined by using the information of the coordinates of the origin, the orientation of the coordinate system and the size of the cell. In raster data model, the continuous surface with attributes is divided into regular discrete units. The data involves no distinctive geographic features and thus have no implicit topological relationships like vector data. This limits the use of the data for spatial queries and certain spatial analyses. Geographic raster data corresponds to ordinary bitmap images with colours representing the attribute values. It only contains the corner coordinates as additional information.

The vector data model represents geographic features similar to the way maps do. In the vector data model, the basic units are points, lines and polygons. A coordinate system references real-world locations. The used coordinate system is usually fixed two-dimensional Cartesian system, where a location is recorded as an (x, y) coordinate. Points are abstract zero-dimensional representations of geographic features or objects that are taken as possessing no relevant dimensions in the data model in relation to the information collection, intended application or presentation of the data. In other words, points represent geographic features too small to be depicted as lines or areas. Points are recorded as a single (x, y) coordinate.

Lines are used to represent linear one-dimensional geographic features with length but not width. In short, lines represent geographic features too narrow to depict as areas. The width can be neglected because of its non-existence (e.g. abstract boundaries of countries), irrelevance to the application (e.g. sewer system network) or negligibility to the intended scale of presentation (e.g. street centre lines or streams). Lines are recorded as an ordered series of (x, y) coordinates, i.e. lines consist of points defining line segments. Polygons are used to represent two-dimensional homogenous area features. They are recorded as an ordered series of (x, y) coordinates defining line segments that enclose an area.

With x, y coordinates, we can represent points, lines, and polygons as a list of coordinates instead of as a picture or graph. Geographic vector data resembles ordinary vector images but it contains some additional information like absolute coordinates and topology. Advantages and disadvantages of the basic models are shown in Table 3.1. Note that the term "overlay" means an analysis procedure for determining the spatial coincidence of geographic features[15].

Rater Model	**Vector Model**
Simple data structure Rows/Columns of equal-size grid cells, each of which has real X, Y co-ordinates Resolution depends on cell size Each cell has single attribute (may link to separate attribute table) Large storage requirements Implicit topology Overlays easily Surface-oriented spatial analysis Best for capturing continuous features: Elevation Soil Temperature Vegetation Characteristics	Points, lines, polygons with topological relationships and real X, Y co-ordinates for all features Resolution depends on source data Objects may have a number of attributes Each feature has unique identifier linking to descriptive attributes Data files compact Explicit topology Overlays complicated Object-oriented spatial analysis easy Best for capturing features with discrete boundaries: Property Parcels Political Boundaries Utility Poles and Lines Street Network

Table 3.1 Raster vs. Vector

In this work, we need to use geographical information in order to represent the map of a city to a user and to perform simple search analysis of the street network. From this point of view, the raster GIS data is not very practical. But with the help of the vector model, we can effectively represent street network and points of interest. Moreover, in the relationships of the geographic features, the vector data contains an implicit topology. It enables one to perform analysis of the street network. Below we consider this topology and its features, since the vector model is more attractive in our case.

3.3 Topology

In this section, we discuss the topology we will use later on as a basis for the future information system. According to the AGI GIS dictionary, a topology is the relative location of geographic phenomena independent of their exact position. Alternatively and more specifically by the ESRI glossary of GIS terms, it is the spatial relationships between connecting or adjacent coverage features (e.g., arcs, nodes, polygons, and points). For example, the topology of an arc includes its from- and to-nodes, and its left and right polygons. Topological relationships are built from simple elements into complex elements: points (simplest elements), arcs (sets of connected points), areas (sets of connected arcs), and routes (sets of sections, which are arcs or portions of arcs) [14]. Redundant data (coordinates) are eliminated because an arc may represent a linear feature, part of the boundary of an area feature, or both. Topology is useful in GIS because many spatial modeling operations do not require coordinates, but only topological information. For example, to find an optimal path between two points requires a list of the arcs that connect to each other and the cost to traverse each arc in each direction. Coordinates are only needed for drawing the path after it is calculated.

Creating and storing topological relationships has a number of advantages. Data is stored efficiently, so large data sets can be processed quickly. For examples, we will store the network of city streets, which have many intersections. If we represent each street as a polyline, we need to describe some points for several streets many times. In that case, the information about points will be duplicated. It is an inefficient way. Consequently, more optimal way is to describe each point one time and refer to it further. The arc-node topology allows us to represent geographical information in such way.

Topology facilities analytical functions, such as modeling flow through the connecting lines in a network, combining adjacent polygons with similar characteristics, identifying adjacent features and overlaying geographic features.

The arc-node data structure supports three major topological concepts.

Connectivity: arcs connect to each other at nodes.

Connectivity allows to identify a route to the airport or connect streams to rivers or follow a path from the water treatment plant to a house. Here is how it works.

Recall the arc-node structure. An arc is defined by two endpoints, the from-node indicating where the arc begins and a to-node indicating where it ends. This is called arc-node topology. Arc-node topology is supported through an arc-node list. The list identifies the from- and to nodes for each arc. Connected arcs are determined by searching through the list for common node numbers.

Area definition: arcs that connect to surround an area define a polygon.

Many of the geographic features we wish to represent cover a distinguishable area on the surface of the Earth, such as lakes, parcels of land and census tracts. An area is represented in the vector model by one or more boundaries defining a polygon. While this sounds counterintuitive, consider a lake with an island in the middle. The lake actually has two boundaries, that which defines its outer edge and the island which defines its inner edge. In the terminology of the vector model, an island defines an inner boundary of a polygon. Here is how topology is used to define areas.

Recall that the arc-node structure represents polygons as an ordered list of arcs rather than a closed loop of x, y co-ordinates. That is called polygon-arc topology.

Contiguity: arcs have direction and left and right sides.

Two geographic features which share a boundary are called adjacent. Contiguity is the topological concept which allows the vector data model to determine adjacency. Recall that the from-node and to-node define an arc.

3.4 The Visualisation of the Geographic Information

Firstly, let us define one important element, annotations that contribute to visualisation. Annotation is descriptive text used to label coverage features. Annotation is used only for display purposes; it is not used in analysis. Annotation is used for cartographic display to label coverage features and geographic points of interest. Annotation in coverage can be organized into subclasses, each having similar properties.

There are many ways for geographic information representation and the most basic of this is a map. The traditional medium of the map was paper or some corresponding surface. However, GIS enables an effective work with maps through a computer display screen. It is suitable for situations, where we do not need to use maps outdoors. But the present range of diverse mobile devices open up opportunities to use digital maps anywhere we need. Unfortunately, these mobile devices have limited resources and a quite small screen, which makes work with maps more difficult [14]. Note that along with a map, other alternative ways exist to represent geographic information on computer devices. We can represent it verbally (e.g. with speech synthesis) or display in the form of text. In some situations, these may be more preferable for mobile devices.

A digital map in geographical information systems is a view to the geographic data. In this form, it includes potential access to all analysis and modeling capabilities of GIS system; the map view is actually used for many geographic analysis operations. Behind it, there are the GIS data formats with rich spatial information contents. However, a rich spatial information is not required when the digital map is used only for viewing. In that case, the digital map can be represented with a graphical image.

As with the vector and raster types of GIS data, there are two types of graphic image format: the raster or bitmap type and the vector map. We have already discussed these two types and the differences between them. Computer graphics do not allow analysis and modeling, and can be used primarily only for displaying and visual inspection, the graphics is only a picture. In general, the distinction between GIS formats and computer graphics is very clear. The question is about the richness of information content of the format with respect to spatial properties.

The graphics have two different applications in GIS. At first, graphics can be used as an attribute of a geographic feature, like any other "multimedia" data. For example, by pointing a point representing building on the digital map, we will see a picture of the referenced house. The other way to use graphics is to use it as a map (or part of a map). A graphics image can picture a geographic area, and its source may be a scanned paper map, satellite or remote sensed image, or aerial photograph. If the image has a geographic dimension, it could be geo-referenced, i.e. registered to a common geographic coordinate system [2]. Then, it could be used in conjunction with geographic data sets of the same area. A geo-referenced graphic image or drawing can be presented, for example, as a background or an overlay of a geographic data in a view. However, we can not identify a geographic feature or query a location from it; though a visual determination of coordinates can be done through the common coordinate system.

Both raster and vector graphics can be used for displaying geographic information. The preference of the format depends primarily on requirements for representation of graphics. The raster graphics suit better for representing continuous tonal surfaces, while the vector graphics are more appropriate for "object" type of graphics. Considering guide map types of applications in a city environment, the vector format should be more usable. When thinking about mobile applications, the vector graphics have also certain other favourable properties: the file sizes tend to be smaller (depending on the information contents) and they are scalable.

Viewing a digital map on a (limited sized) display would benefit of some possible operations. The most basic one is panning. This would allow maps larger than the actual display to be used. In addition to panning, a zooming facility is essential for useful digital map inspection. Actually, zooming means changing the scale of the map. This is a problem for a raster image, because the raster cell size is fixed to scale; thus, the vector graphics are more usable for zooming. The vector format introduces another desired feature from the digital map: a layered structure could be supported. Because the presentation of the maps content and form is highly dependent on the scale of the map, which is partially in relation to the resolution of the displaying device, the map "simplicity", i.e. amount of graphical elements, should be able to be modified with respect to the scale. The richness of the map can be controlled by displaying or hiding the layers, which contain different classes of geographic features, in the view. In addition of scale dependent viewing, the layered controlling of elements in the map view can also be used for specifying, or simplifying, the map for a particular purpose. The visualisation of text elements, annotations, can be controlled by layering. Of course, the presentation of annotations is also scale dependent. While thinking of scale dependent presentation of the maps, it must also be remembered that simple scale change and modification of the map by changing the elements do not necessarily produce informative and useful map in another scale. An appropriate generalisation of the geographic features is essential.

The graphics formats used for viewing digital maps could also be augmented with some spatial information, similar to GIS formats. Some existing graphics formats already support this kinds of properties. Most important is the definition of coordinate system that allows georeferencing. Another property would be the definition and identification of the graphic elements, especially in vector images. This identification would give a possibility to select geographic features in the map and further use them, for example, for displaying some additional information (like name of a street or street address of a building) or as a selector for query. Some added topological properties could also make it possible to do simple spatial analysis, like route definition and route distance calculations[7].

The desired operations of the digital map viewing and the associated features required from the graphics formats create some expectations for the used devices. For example, they must posses an interface and tools for panning and selection, i.e. a scrolling and pointing device with appropriate functionality. In the context of this work, by GIS format we mean a format with the vector model which contains topological information and attributes.

3.5 Mobile GIS System

Mobile GIS is a term that’s being used more frequently within the geospatial industry. Geographic information systems (GIS), and their application, have been around for long enough now that they are well understood. The concept of mobility is also something we are all familiar with, but our understanding of mobile GIS - its application, structure and operation – is still very much in its infancy.

While our understanding of mobile GIS may be in its infancy, the components that make up an operable mobile GIS are becoming more apparent and available. The industry is gaining a better understanding of how mobile GIS will be used for everyday applications, but the interaction of the components is still developing.

The technologies that are utilised by mobile GIS applications are generally well defined and consistent with an enterprise view of GIS, as depicted in Figure 3.1. The GIS engine, which still resides on a powerful, centralised server, carries out the main data management role and provides the underlying application development software for spatial analysis and query-based operations.

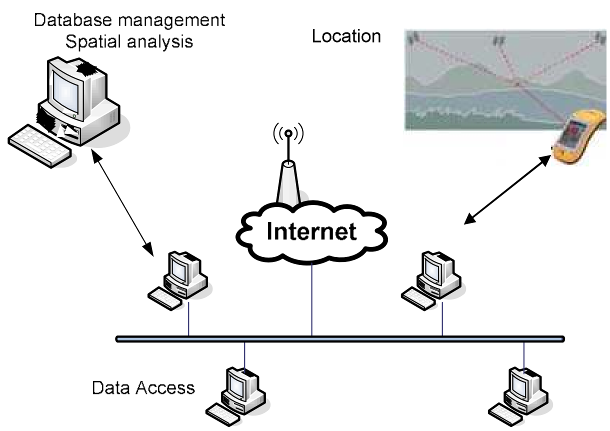

Figure 3.1: Underlying technologies associated with mobile GIS

This GIS engine feeds the data into the organisational network for all who have access to the enterprise. The mobile field application is simply an additional connection into the enterprise. Accessing the exact same data as the rest of the enterprise, as and when needed. This connection is usually a physical connection, either directly into an office-based computer or through the Internet. The Internet is also used by the entire enterprise for the distribution of data throughout the organisation and, ultimately, from business to business.

The present situation sees the mobile field device operating its own resident application and working only with a copy of the data that everyone else accesses within the traditional enterprise GIS. As such, mobile GIS do not interact seamlessly within the structure of the network.

Moving forward, however, the seamless interaction of the mobile field device as part of the enterprise network itself will eventuate through the utilisation of wireless technology. Ultimately, we could see the field application operate as a web-browser, where its interface and analysis is carried out entirely on the GIS engine. But this scenario is still some way off. The application structure most likely is a small operational application and some resident base data being held on the field device, with the field device only interacting with the enterprise when data is updated or new data is viewed or queried. Even this scenario is some way away, with wireless technology not offering the bandwidth required just yet to make the transfer of data transparent enough for efficient operation.

Although wireless technology is useful in the operation of mobile GIS, it is not an absolute requirement. Mobile GIS operate today; it is just that their interaction within the enterprise utilises landline communication technology rather than wireless [11].

Whether wireless technology is used or not, the ability for the field application and device to be aware of its location is a significant factor in the ultimate success and acceptance of mobile GIS – for both spatial awareness and operational efficiency. This is considered in more detail later.

It is easy to incorrectly think of mobile GIS in terms of the mobile application component alone. Mobile GIS is a system. It is dependent on the underlying GIS engine – utilising its power in the same way as any of the other components within the broader enterprise [12]. The field devices used within a mobile GIS are restricted, however, in terms of processing power and storage capacity, which means the mobile

GIS operates in a different way to the rest of the access points within the traditional enterprise GIS.

Mobile GIS removes the need for intermediary paper-based recording. It is not unknown within organisations for modifications made on paper maps to never make it into a digital format, which means many changes to the real world are never reflected correctly within the enterprise.

With mobile GIS, data is always in a digital format, making it easy and efficient to transfer from the field to the office without introducing interpretive errors. Checks and balances are still required, of course, but many of these can be automated so those carrying out the checks can focus on the real errors without the drudgery of filtering through large quantities of repetitive data.

Mobile geographic services are a fast growing field of GIS. Although there is much hype associated with the technology, it is clear that even today technology are finding real business benefits in using geographic information in a mobile context. From yellow pages to routing, to data collection there are many success stories. As the technology improves especially telephone data transfer speeds with the introduction of 3G systems, things can only get better.

3.6 Location Based Service

GSM Location Based Services provide personalised services to the subscriber based on their current position. Two categories of methods can be used to find the location of the subscriber: basic and Advanced. Basic positioning methods are based on the knowledge of the cell the subscriber is using (cell ID). This can be used alone, or together with other information available in the network such as timing advance (TA) and network measurement reports (NMR). This information is available for all handsets [8].

Advanced techniques will be available in new handsets such as Enhanced Observed Time Difference (E-OTD) and Assisted GPS (A-GPS) which uses the GSM network to help the reception of the freely available GPS system.

The division of position technologies above is based on accuracy of the positioning method. Other factors, also very important are for example complexity of the system, availability of the positioning technology in the network (e.g. A-GPS may not be available everywhere) and the investment needed on the network side and in handsets

LBS has the capability to find the geographical location of the mobile device and then provide services based on this location information. This would mean a person asking for information about the nearest ATM outlet based on his location in the city, which is within his reach. Thus we could say that Location based services deliver geographic information between mobile and/or static users via the Internet and/or wireless network.

This seems rather an exciting prospect, which could generate mass enthusiasm and could appeal to many prospective users and developers. LBS promises to facilitate the delivery of spatial data along with providing routing and direction-finding information. According to a poll by Integrated Data Communications, Inc. (IDC), two-thirds of Americans want wireless location-based services, with safety and security issues (emergency call, roadside assistance, and driving directions) as the primary reasons for wanting location-based services [9]. Prospective users are even ready to pay premium or even accept advertising on their devices for the services.

The applications of LBS could be varied and innovating. Services that exploit the users location could be wide spread in the years to come. The service providers could move or have moved into areas like providing services incase of emergencies, route finding and many more limited only by mans imagination. There are definitely challenges for this market, which include figuring out what applications to develop, how to generate revenue from the applications, overcoming consumer acceptance hurdles and how to protect user privacy. Nonetheless the concern of this paper is not to indulge in the future of these applications but discover the technical aspects to LBS.

Chapter 4: Mobile E-Commerce and Mobile GIS System

4.1 Technology and Requirements

Nowadays, the development and adoption of mobile technologies have made new services and related commerce more and more available. Some of the factors that contributed to this development are the tremendous development of the Internet and related technologies, the understanding and exploitation of the business potentials that rest behind this development, the boost of M-Commerce frameworks and technologies, and the impressive growth of wireless mobile networks. The number of mobile customers that use digital mobile networks for voice and data transfer has already exceeded one billion. The market is saturated in most of developed countries and it is still expanding rapidly in developing countries. This is a huge customer potential for commercial activities, if the environment supports it properly M-Commerce is said to "rapidly approach the business forefront" and the general feeling is that it "will soon be a dominant force in business and society"[21].

The high-end mobile terminals have already reached the complexity and performance of the personal computers of mid 1990's, and the development continues at fast pace. The new smart-phones are programmable and can run, in principle, any conceivable application. Wireless communication technologies combined with Internet-enabled terminals constitute an ideal platform for the realization of much new kind of business transactions. The small and light, but powerful, mobile terminals are almost always carried by their owners. Therefore, they are like a wallet, among other things. They can indeed also store electronic cash, credit card information, tickets, etc., and can thus also take the role of an electronic wallet. In addition, the terminals can be located by using satellite navigation systems, within mobile networks, or by some other available means. Recent developments in these areas seamlessly extend the positioning of wireless devices into all the environments where they can be used for voice and data communication. Thus, location-based services become possible. In the following, we will review the technological infrastructure and its development, which has made M-Commerce gradually available. We concentrate to Mobile GIS System, as the related M-Commerce is probably the best example of the new emerging applications [24]. From this angle, the technological infrastructure consists of mobile networks, mobile terminals and positioning technologies. Requirements considered for the Mobile GIS domain refer to all the involved actors i.e., the mobile users, the telecom operators, the service providers, the content providers, etc. In this paper, we mainly focus on the requirements that refer to end users. For supporting the study, we present a pilot system that fulfils a considerable part of the identified requirements.

4.2 Technological infrastructure

In this section we will briefly describe the technology leading to 3rd generation (3G) mobile networks, the technology used in Mobile E-Commerce and the technology for mobile terminals.

4.3 Mobile networks and their development

Our current mobile services infrastructure is mostly based on 2nd generation (2G) digital public mobile networks. The families of standards created to specify these systems include e.g. Global System for Mobile communications (GSM), CDMA standard (Qualcomm, USA) and PDC (Personal Digital Cellular, NTT DoCoMo, Japan). From M-Commerce point of view, Circuit Switched Data (CSD), High Speed Circuit Switched Data (HSCSD), Global Packet Radio Service (GPRS), and Short Message Service (SMS) are important in the GSM standards. They make the wireless data transfer services and applications possible in a wireless telecom network. The standardization for the third generation of mobile telecom systems (3G) was started at the beginning 1990's. The International Telecommunication Union (ITU) specified a framework standard for the global 3G system, called International Mobile Telecommunications 2000 (IMT-2000). The central idea of IMT-2000 standard is to enable advanced multimedia services on wireless telecom networks (video streams, animations, pictures, etc.), offer a truly global roaming, and the global coverage of the services. Universal Mobile Telecommunications System (UMTS) is the European view on the 3G systems envisioned in IMT-2000. The UMTS standard family was developed by 3G Partnership Project hosted by the European Telecommunications

Standard Institute (ETSI). It offers global coverage at different service levels (i.e., satellite, macro, micro and pico-cell environments) and a smooth transition path from the current GSM to 3G networks. In this evolution, there is an intermediate step, often called 2G+ or 2.5G. This covers (CSD) at 14.4 kbps, High Speed Circuit Switched Data (HSCSD), General Packet Radio Service (GPRS) and Enhanced Data Rates for Global Evolution (EDGE). The so-called core networks are common to TDMA-based access network (GSM) and a truly new access network type based on the W-CDMA technology. All the above network generations can support Mobile GIS. The work towards 4G networks has also already begun. They should be in operation around 2012. It is no yet clear what 4G systems should and could offer to the users. The hopes for the offerings are: seamless roaming and handoff between various types of technologies such as UMTS/W-CDMA, Bluetooth and Wireless LANs; truly seamless wireless Internet access and; and more bandwidth for wireless data transmission. The main objective as concerns roaming and handovers is to materialize the Always Best Connected (ABC) idea [22].

One of the main differences between these different network generations is the transmission capacity available to the user (see figure 4.1). In the current 2G networks the capacity ranges from 9.6 kbps to 14.4 kbps on CSD connections. The 2G+ HSCSD offers in practice 57.6 kbps and GPRS ca. 112 kbps transmission rates. The EDGE system delivers 384 kbps. 3G networks will enable speeds up to 2 Mbps depending on radio conditions as well as the network environment (e.g., satellite, indoor, outdoors). In worse circumstances (e.g., weak signal, or in the move) the wireless link capacity will be only a few hundred kbps.

The Japanese 3G mobile network FOMA, operated by NTT DoCoMo, offers currently in practice max. 384 kbps downlink capacity and 64kbps uplink capacity for packet data. In addition, it offers a connection-oriented full-duplex data transmission service with 64 kbps guaranteed capacity in both directions. Charging for the former service is based on the data volume and the latter on connection time.

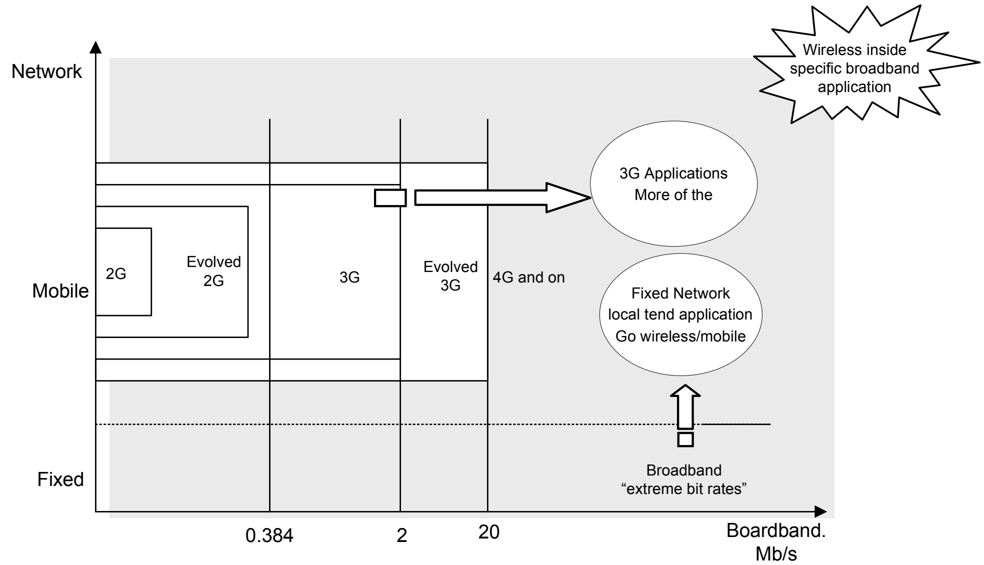

Figure 4.1 Network Technology Evolution

Another dimension of the access networks is the service capability. In the basic 2G network the services are voice, circuits switched data (CSD) transfer, and short messages (SMS). Especially short messages have been used to support financial services like banking and stock services. CSD can be used as the carrier in a TCP/IP network and it is possible to do Internet banking using e.g., hand-held Communicators over CSD.

At the service level in 2G+ networks the de-facto standard Wireless Application Protocol (WAP) provides an access to higher-level services. These include support for E-commerce. WAP technology brings Internet and wireless technologies together so that contents encoded in a mark-up language at WWW servers can be moved to handsets. In Japan, NTT DoCoMO developed a functionally similar system as WAP environment, but based it on cHTML, instead of WML WAP 2.0 specification adopted in 2001 the iMode end-to-end HTTP connection model as an alternative to WAP 1.x protocol stack and there is hope that Mobile Internet will converge into XHTML encoded contents and Java as programming language. As such, WAP specifications are independent of the underlying wireless network technology and can be used over GSM, CDMA, and W-CDMA air interfaces.

The popularity of i-Mode largely depends on the business model. i-Mode terminal can be all the time connected to the network, but the user is charged only for the data packets transferred, not for the connection time. The total cost of usage consists of i-Mode usage charge (150 Y/month)+ packet transmission charges + charges for the accessed contents. The packet transmission charge consists of a fixed monthly charge and the charge for the transmitted packets (0.02 Y/packet – 0.3 Y/packet). Sun Microsystems and DoCoMo have signed an agreement to incorporate Sun's Java, Jini and Java Card technologies into i-Mode cellular phones. The new i-Mode terminals support Java (Micro Edition). The current development phase in Japan is to offer downloadable Java applications and run them at terminals. Using this technology it will be possible to conduct MCommerce transactions without a heavy network infrastructure. Thus, handheld devices could talk directly e.g. with cash registers. Currently integration of Bluetooth and WAP are under way . Bluetooth can be also used for small LANs, in addition to ad hoc point-to-point links. The maximum rate achieved by Bluetooth1 is 722 kbps, and Blutooth2 2Mbps. Bluetooth advances are that the infrastructure is simple, relatively cheap, and operated in unlicensed frequency bands. From Mobile GIS point of view, Bluetooth is able to offer a local positioning method if some of the transceivers is in fixed position and this information can be used by Mobile GIS applications. The UMTS Forum has listed a broad range of services, which should be made available with the advent of UMTS. These services are grouped in six main categories. The required bandwidth, as well as some examples for these categories, are shown in table 4.1. [22]

Category	Service Bandwidth (uplink/downlink) [in kbps]	Sample Services
Voice Services	28.8/28/8	Voice, IVR, Voice Mail
Messaging Services	28.8/28/8	SMS, E-Mail
Switched Data Services	43.2/43/2	Low speed internet/intranet access, fax
Medium Multimedia	20/768	Interactive games, lottery, betting, simple e-commerce
High Multimedia	20/1000	m-commerce, m-portals, LAN access, Audio/video on demand
High Interactive MM	256/256	Video telephony, videoconference

Table 4.1. Categorisation of 3G Services

4.4 Mobile Terminals

The mobile terminals are developing, getting new features and enhanced hardware, at fast pace. The communicators and PDAs have advanced computing capabilities, and the mobile phones are all the time closing them. The new mobile phones are called smart phones, as their features can be added and modified. Most of the new mobile phones have already support for GPRS, Bluetooth and MMS, inbuilt or attachable camera, as well as sophisticated applications. The smart phones have high-resolution color display (e.g. 176x208 pixels with 4096 colours) and several megabytes of internal memory, which can be supplemented with memory cards.

Currently Symbian OS appears to be the preferred operation system for mobile terminals. However, there are also Microsoft's Smartphone and Linux based solutions available and in development. In addition to operating systems, common platforms platform has already been licensed by majority of the key players within the mobile phone industry. The common platform over the operation systems provides some common applications and they make application development easier and faster, still allowing required differentiation for markets.

The need for unifying and standardization of application development environments has been seen by the industry. The 3GPP (1) developed the Mobile Execution Environment specification that defines a framework for capabilities of devices. The technologies that are considered by MexE as fundamental for the future, and are covered by the developed specifications, are Java and WAP. Most of the new mobile phones support already Java, which gives a convenient way to download and add new applications to the terminals. MexE also defines security features: security certificates indicating that an application is secure and also where it can be executed. The Open Mobile Alliance (OMA) has now a central role in standardization with respect to mobile services and applications, and their interoperability. OMA was formed by consolidating several earlier industry fora, among them Open Mobile Architecture initiative, WAP forum and Location Interoperability Forum (LIF). The most important technologies considered by OMA are MMS, Java and XHTML. The mobile devices are becoming location-independent, widely distributed personal trusted devices (PTDs) that strongly support the realization of financial transactions and other activities related to M-Commerce. To address the need for extra security in the mobile environment, the Wireless Application Protocol Public Key Infrastructure (WPKI) was specified. The WPKI encompasses the necessary cryptographic technology and a set of security management standards that are widely recognized and accepted for meeting the needs of M-Commerce. It is also worth mentioning the Mobile Electronic Transactions (MeT) forum that refers to the application side of mobile commerce. MeT was established to further strengthen the framework for secure mobile transactions. It concentrates on defining a consistent and coherent framework that is based on existing standards and specifications.

4.5 Positioning systems of Mobile GIS

The Positioning system of Mobile GIS is dealt by Location based services, apart from the already described technology, require specific infrastructure for positioning the mobile terminal. Positioning means determination of the location of the object in a reference system. The reference system can be a coordinate or address system, aerial division or route system . Geo-coding is a process used for associating the object to general coordinate system, if some other system, like addresses, is used as reference.

The systems offering positioning for mobile terminals in LBS are divided to three main classes : satellite positioning, network-based positioning and, local positioning. Different positioning systems and techniques varies with their features, such as accuracy, reliability and time-to-fix. Each system has its own place, and they complement each other in certain cases. Satellite systems do not work well in deep canyons and indoors where cellular coverage may be denser. Network-based positioning may be more imprecise in rural environments with fewer basestations, where satellite visibility is better.

Satellite Positioning

Satellite positioning systems use an infrastructure of earth-orbiting satellites and receiver terminals. The terminals calculate the position based on the information received through the radio signals from three or more satellites. The terminal-based method provides 10-40 meter accuracy. It renders the user totally independent of the mobile network with respect to positioning, and, in principle, allows access to any location-based services from third-party service providers.

Most well know and widely used satellite positioning system is GPS, operated by US Ministry of Defense. Other currently existing system is Russian GLONASS. European Union has also plans to build a corresponding system called Galileo. Galileo is the European global navigation satellite system providing accurate and guaranteed global positioning service under civilian control. Commercial services availability will be guaranteed through several mechanisms, thus rendering the system ideal for critical applications. According the current schedule, Galileo should be operating in 2008. The intention is to make Galileo interoperable with GPS. For professional use a Differential GPS systems are used, which can potentially provide location accuracy at the meter and sub-meter level. It uses infrastructure-based assistance and different terminals. This method requires that there must be a terrestrial reference station that sends an additional correction signal to the terminal.

In mobile network environment, there exist also a solution called Assisted-GPS solution, where additional data for the GPS receiver is send through mobile network. Assisted-GPS makes the positioning of a mobile terminal including GPS receiver to be positioned faster and more accurately. It even can make possible to use GPS positioning also indoors, in a limited scale. Assisted-GPS is a hybrid solution, using both, satellite and network-based, features. As a result of a positioning request, the user gets a relevant map delivered to the terminal .

Network-based positioning

The network-based positioning systems refer to positioning methods where the mobile telecommunication networks are used for providing or supporting the positioning of mobile terminals. They include several different methods that are standardized in mobile network specifications .

The basis for the diverse network-based methods is that the coordinates of the base stations are known and that the distance of the terminal from the base stations can be measured or at least approximated. The approximation without any measurements is called Cell Identity (CI). In CI method, the terminal position is approximated by specified coordinates on the cell area, defined based on the cell coverage. The accuracy of the method varies from a few tens of meters (densely populated areas, indoor cells) to several kilometers (in rural areas). The above method can be improved by using Timing Advance (TA) parameter that is readily available in GSM networks at low cost. Using it, methods can be used to increase the accuracy estimate of the CI-method.

The more accurate network-based positioning methods use explicit measurements. In measuring the terminal receives signals from at least three different base stations, or three or more base stations receive signals from the terminal. Based on the measurement the position of the terminal can be calculated at the terminal or in the network. There are different network signal measurement positioning methods, including: Angle of Arrival (AOA), Time of Arrival (TOA), Enhanced Observed Time Difference (E-OTD) and Observed Time Difference of Arrival – Idle Period Down Link (OTDOA- IPDL). The last method is standardized in 3G networks, but all others are present already in 2G GSM network specifications.

The network-based positioning methods differ in many some aspects. The positioning accuracy is method dependent, between 50m to several kilometers; In CI method even up to tens of kilometers. The deployment by the network operators and requirements from users perspective have also significance. Some of the methods require new components and investments to network infrastructure, some other needs mobile terminal software or hardware updates. From the privacy point of view, notable is also where the actual position calculating is done; in the terminal or in the network.

Local positioning

The third main class of positioning systems is local positioning. It refers to positioning that operates only in restricted area and based on short distance signal transmission . It covers specifically location-based services in indoor environments like large buildings, shopping centers, etc., where satellite and mobile network positioning methods are not well applicable or precise enough. Local positioning methods include positioning methods where wireless local area networks (WLAN), Bluetooth technology, radio frequency identification (RFID) or Infrared (IrDA) technology (Active Badges etc.) are utilized. A number of systems/technologies have been proposed for the underlying supporting platforms

4.6 Location-Based Service Requirements

Besides the general requirements for collaboration of M-Commerce and Mobile GIS, location-based services are subject to a set of specific requirements. We classify them in the following requirements categories: user (functional), usability, reliability, privacy, location infrastructure and, interoperability. These requirements cover some basic issues in location-based services.

4.7 User Requirements – Functional Requirements

User requirements are formed on mobile user exceptions of the Mobile GIS. These requirements are the source for the functional requirements of an Mobile GIS application. Hermann and Heidmann analyzed a specific Mobile GIS application, namely location based fair guide. They conducted interviews with a set of potential users of such a system and a set of visitors of a book fair in order to investigate the user requirements empirically. The list of features the interviewees considered indispensable includes browsing of spatial information (locations of facilities and exhibitors) connected with catalogue data, activity planning and way finding [23].

The same requirements apply of course to any mobile application serving a visitor in unfamiliar environment, whether it is a fair, museum, city or country. Therefore, the main Mobile GIS functional requirements are the following:

Browsing of spatial information, e.g. in the form of a city map.

Navigation

The user must be able to acquire directions and guidance for reaching a specific point of interest.

Access to catalogue data, e.g. names of restaurants and their description like menu, range of prices, opening hours, etc. Catalogue data must be spatially referenced, so the facilities in the catalogue can be represented on the map and directions for reaching them can be acquired.

Location-based access

The user must be able to access map and catalogue data based on his/her present location. If locating the user cannot be resolved automatically by the system, the user must be provided with the option of manually entering his/her location. The system should be able to handle such requests through geo-coding procedures.

Personalized access: the user must be able to specify the type of information s/he needs, e.g. a map depicting just the street network or a map including various points of interest. A profile mechanism should also be provided so that the user does not have to input his preferences every time s/he uses the service.

Fast access: the user must be able to run various queries, e.g. for nearby restaurants or even for the nearby vegetarian restaurants that are opened after 9pm.

As can be seen, the Mobile GIS user interface must deliver to the user various information including a static map image of the desired area, the location of the user and possibly locations of other mobile objects, descriptive information on points of interest, route information, etc. Clearly, all this information cannot be presented at once, so an implied requirement is that Mobile GIS must provide a set of alternative views, some or even all from the list presented:

Geographical View

Graphical representation of a geographical area, usually in the form of a map.

View corresponding visual perceptions

The view as the user could see it if s/he went to the place of interest. This view can be implemented with photographs or 3D modeling.

Geographical information view: textual description of points of interest enabling the user to browse through it.

Status View

Description of the current state of the user in the form of picture, text of voice.

Context View

Tied to information representation about close objects and the user possibilities in the current context.

Route View

Graphical representation of a route from one point to another.

Logistic View: abstract route model, showing only intermediate and final points, but no specific distances and directions

Guidance View:

Turn-by-turn instruction for the user, usually in a text form and also possibly as voice.

4.8 Usability requirements

Mobile computing environment has certain features that impose restrictions. The properties of mobile networks are: (relatively) low bandwidth, strong bandwidth variability, long latency, unpredictable disconnections and communication autonomy. The properties of mobile terminals are: small and low-resolution displays, limited input capabilities, limited computing power, limited power and small memory size. The practical conditions, when and where the mobile devices are used, bring also additional restrictions. The using conditions cannot be expected to be constant, as usually is the case in "desktop" conditions. The mobile users are typically in very unstable environment in varying conditions, where their cognitive capacity is demanded for other tasks as well[23].

All these restrictions have to be taken very carefully into account when designing Mobile GIS. Some of the implied requirements are:

- Not very intensive use of mobile network and minimal volume of transmitted data.
- Possibility to offline operation.
- User interface should be very simple and user friendly and the amount of presented information content limited and well specified.

4.9 Reliability requirements

Mobile GIS are intended mainly for traveling people as a tool providing support in making decisions about where to go. Therefore, wrong information may mean wrong decisions, lost time and, as a result, anger of the client in the best case and a court examination in a worse case when. We can think, for example, a case where Mobile GIS misled the user when he needed an urgent medical assistance.

We split reliability requirements into a few sub-requirements:

Data reliability: the most important since Mobile GIS are built around spatial and catalogue data and incorrect or missed data can easily lead to problems.

Software reliability: applies to both server and client sides.

Appropriateness and precision of exploited algorithms and methods: depending on the task in hand, e.g., user positioning precision may be sufficient or insufficient, calculating distances by straight line may be sufficient for some tasks but insufficient for others (the road network thus must be taken into account).

4.10 Privacy Requirements

Privacy handling is a major issue in Mobile GIS deployment and provision and a critical success factor to the wide acceptance of this technology framework. The terms privacy handling consolidate issues like ownership of location information, use of location information, disclosure to service providers, etc. Skepticism arises as to where and how privacy handling should take place within the LBS provision chain. Existing proposals from operators and standardization bodies (e.g. OMA, 3GPP) specify a priority scheme whereby the core network elements (e.g., Home Location

Registers) have master control on location information. The provision/disclosure of such information to other entities (e.g., location servers, LBS serving nodes, ASPs) is subject to subscriber needs (e.g., registration information) and regulatory frameworks.

4.11 Mobile GIS infrastructure requirements

Mobile GIS service consists of roughly two phases, determining the position of the customer and providing service or contents based on the position. For the location method at least the following requirements can be listed [16]:

The method should provide good accuracy subject to the requirements of the application and the respective cost.

The area where the mobile device can be located should be as large as possible; it should be possible to determine in advance where the device can be located; ideally, this should happen within the whole coverage area of mobile networks.

The method should be fast: also in this respect the applications have different demands, e.g. the emergency applications and car navigation systems have higher demands than “FIND Restaurant” – type applications.

The location method should not generate too much signaling load within the mobile network.

The effects of adding location method support to a terminal and using it should be minimal, i.e. it should not increase its size, weight, power consumption, or price.

The location method should have a minimum impact on the mobile network in terms of complexity and cost.

It should be possible to locate all mobile devices irrespective of their type and whether they are roaming or not.

It should be possible to locate a large group of mobile devices at the same time.

Consumer privacy must be ensured, by, e.g., providing means for the consumer to turn off the locating feature of the terminal.

4.12 Service interoperability requirements

The interoperability should be assured on all levels of the system architecture. The Mobile GIS platform should be interoperable with several types of terminals (e.g., PDAs, GSM terminals) and positioning infrastructures (e.g., indoor, GPS). The platform should be able to handle different coordinate reference systems (e.g., WGS-84 and local systems) in order be able to utilize geographic data available in existing GIS databases. The seamless service provision when different infrastructures are visited by the user is also a requirement in the highly diversified modern telecomm landscape [16].

4.13 Implementation framework for Mobile GIS Services

When studying the aforementioned requirements for Mobile GIS, and for M-Commerce in some extent, and standard system was studied and tested.

The general Mobile GIS architecture is presented in figure 4.2. This is a fusion of the visions presented by OpenGIS/OpenLS Initiative and Open Mobile Alliance.

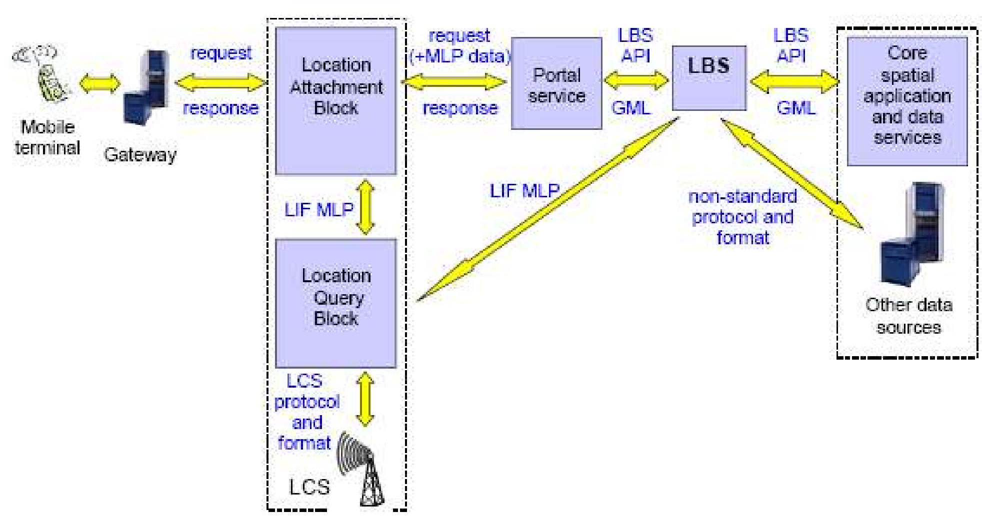

Figure 4.2. The Standard Mobile GIS architecture

In the system, a mobile terminal is used for accessing Mobile GIS and a gateway provides the conversion between data protocols in mobile network and TCP/IP network, if needed. LCS is a service providing information on the locations of mobile users. Location Query Block is a wrapper of LCS into (formerly LIF) Mobile Location Protocol (MLP). Location Attachment Block, if ordered, queries Location Query Block for the location of the mobile terminal originated the request and attaches this information to this request; transparent for responses. Note that in the case of terminal-based positioning (e.g. GPS) the entire LCS tier would be located inside the mobile device. Portal service provides transformation between data protocols and formats used by in the Mobile GIS and those appropriate for specific mobile terminals. Core spatial application and data services are building blocks for Mobile GIS, provide geographic data services and standard applications, such as geo-coding, geographic queries, route finding and other. The other data sources are sources using some non-standard data protocols and formats, for example, business databases.

Important features of this architecture are utilization of GIS data and algorithms and using of XML for data encoding and all the protocols. Such architecture provides high flexibility in fulfilling of the Mobile GIS functional requirements, as well as interoperability requirements. The client application was developed in Java and geographic data is delivered in (compressed) XML-encoded geographical vector form to the client. The client side is responsible for generating alternative views to the data and running simple queries. This allows reducing network load and working offline. The pilot system includes some additional special features. One special feature is user-centered data selection. The system server is capable of selecting geographic data that is relevant for a specific user at a specific time. This again further reduces network load and increases the usability of the service. An other special feature is a transaction monitor. In order to protect itself from the unexpected disconnections and crashes, the client application includes the transaction monitor. It keeps track of the history of interaction with the server (what data was sent and received) and stores all the data needed for recovery. This saves the application from information losses and guarantees basic transactional properties.

Chapter 5: Mobile GIS FrameWork

5.1 Introduction

Since 1990, geospatial information technologies and mobile wireless Internet have been rapidly developed. It is easy to see that the integration of geospatial information and mobile Internet is inevitable, which is simultaneity driven by market demands and technologies (Li et al., 2002b). The integrated system is designed to work on mobile intelligent terminals, and brings new dimension - at any time, any place - to access geospatial and attribute information in GIS. It is called Mobile Geographic Information System (Mobile GIS). Mobile GIS offers another new perspective for the use of GIS and further extends the “office” GIS works in mobile environment (Shi and Kwan, 2002). Mobile GIS was early applied to assist office and collect data in the field (Lee, 1993; North, 1997). It is not a conventional GIS modified to operate on a smaller computer, but a system builds using a fundamentally new paradigm (Maguire, 2001). It extends unlimited information on the Internet and powerful service functions of GIS to mobile devices, and can provide mobile users with geospatial information services. Mobile GIS creates a new channel of business practice, and thousands of potential applications and services can also be developed. Hence, it is exploring a new era of mobile geographic information services. The rest this paper is arranged as follow: Mobile-based Mobile GIS and its superiority to SMS/MMS-based Mobile GIS are briefly introduced in the next section. Three mainstream architecture solutions for distributed systems are presented thereafter. A distributed architecture for JAVA - based Mobile GIS based on J2EE and JAVA is proposed in the section “A distributed architecture”, and a prototype system is presented in the section “A prototype system”. Finally, the conclusion is drawn and outlook of JAVA -based Mobile GIS are discussed.

5.2 Mobile GIS Based on JAVA Architecture

Mobile GIS can be simply divided into two categories, according to currently popular solutions accessing to the Mobile Internet. The one is based on Short Message Service (SMS)/Multimedia Message Service (MMS), the other is based on Wireless Application Protocol (WAP).

SMS/MMS-based Mobile GIS (Ma et al., 2002; Wei et al., 2002; Zhang and Liang, 2003) can only be suitable for mobile phones with sample system functions, unfriendly graphical user interface (GUI), poor information presentation and the restricted application fields, because of the limitations of SMS/MMS such as restricted carrying information, time lag, unfriendly interactive mode, and so on.

The mobile devices and the communication system are developing very fast, on the other hand, the related software technologies including XML and J2ME accelerate the development of the wireless GIS. Arriving Java-enabled handheld devices, and feeding newly introduced formats such as XML into Java programs, make new opportunities to overcome the restrictions. In fact, Java acts as portable code and XML is portable data. Benefits of using XML-based data format, is that, it provides an object-based data format, in which each feature can be analyzed as an object or set of objects, independent from other features. Object-based formats make less time taking for communication and less transferred data amount. Developing in wireless Java (J2ME) provides flexibility to adapt many different platforms. Object-based data formats supply a spatially-suited data format. Combination of the two technologies introduces a powerful tool for serving spatial data through wireless network.

On the contrary, WAP is a bear-independent international standard protocol that has optimized for mobile devices with limited display and small keyboards of mobile handsets and low bandwidths of wireless networks, and permits applications and services to operate over all existing and foreseeable wireless networks such as GSM, CDMA, PHS, TDMA and WCDMA (WAP Forum, 2002). The WAP specification encompasses a relatively simple and compact version of XML (eXtendable Markup Language) called WML (Wireless Markup Language), which makes it possible to make requests to a mobile service from a mobile terminal and return a map in the form of an embedded bitmap (e.g. WBMP). So JAVA-based Mobile GIS has richer information presentation, friendlier GUI, more system functions and more application fields than the former. Moreover, it can work on a wide range of mobile devices with a J2ME support only, from Personal Digital Assistants (PDAs), mobile phones, and in-car computers to other small mobile devices, and keeps the standard browsing style. JAVA-based Mobile GIS can be described as mobile users (with a J2ME mobile terminal only) can perform the almost same functionality as of Internet GIS but in a mobile environment at any time, any place and without the limitation of operating system and wired link. Altogether, it is because of the advantages of JAVA that we believe JAVA -based Mobile GIS will play a leading role in our mobile information services markets.

The framework and soul of a distributed system is the system architecture deciding the distribution of system functions and data. JAVA -based Mobile GIS, as a new development to Internet GIS, is still in a preliminary stage, whose research issues need be deal with. However, the architecture solution to JAVA -based Mobile GIS is the key and core of its whole researches. Although Li et al. (2002b) and Li et al. (2002a) used to present an architecture for Mobile GIS respectively, the architectures were conceptual ones and had difficulty playing a substantially directive role to building a Mobile GIS system, let alone JAVA -based Mobile GIS.

5.3 Architectural Solutions

JAVA -based Mobile GIS is a “thin client” distributed system via mobile Internet. It must be an open, extendable, stable and cross-platform distributed system, because of the increasing application demands and the diversity of mobile terminals.

Currently, popular system architecture solutions for distributed systems are all based on distributed object technologies. There are three mainstream industry standards: Microsoft's Windows Distributed Network Architecture (Windows DNA)/.NET Architecture, Object Management Group's Common Object Request Broker

Architecture (CORBA), Sun's Java 2 Enterprise Edition (J2EE) Architecture. A system based on Windows DNA/.NET Architecture solution can only use Microsoft's platforms from development, deployment to running, including developing platforms and operating system. It can not cross operating system platform especially, which is the fatal weakness of Windows DNA/.NET Architecture. CORBA is too huge and complicated. And its technologies and standards are updated relatively slowly. J2EE is a specification and standard created by Sun and her industry partners, and it evolves quickly. J2EE provides support for the technologies such as Enterprise JavaBeans (EJB), Java Servlets API and Java Server Pagers (JSP), and so on. J2EE solution reduces the cost and complexity of developing a multi-tier distributed system which can be rapidly developed and deployed, and can enhance the portability, security, load balancing and extensibility of a distributed system. There are the following advantages to build distributed GIS based on J2EE (Mao, 2004): (1) cross-platform; (2) multi-tier separating to complicated tasks; (3) component reusing and (4) module developing. Thus, J2EE solution can satisfy the system requirements of JAVA -based Mobile GIS. Based on J2EE, this paper proposes a distributed architecture for JAVA - based Mobile GIS.

5.4 A Distributed Architecture

As showed on Figure 5.1, the J2EE-based distributed architecture is composed of four logic tiers from the client side to the server side: presentation tier, Mobile service tier, application tier and data service tier. At the fat server side, Geospatial information Web caching mechanism is used to optimize the performance capability and reducing the access delay of the client. Functions and components of each logic tier and the statement about the distribution of presentation logic, business logic and geospatial data in the architecture are detail stated as follow.

Presentation tier

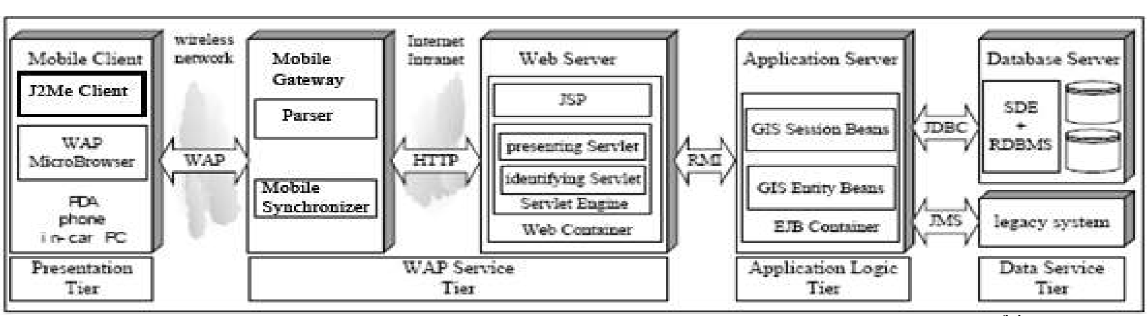

Figure 5.1 A distributed architecture of JAVA-based Mobile GIS

Presentation tier is a carrier of the client of JAVA -based Mobile GIS, and is mainly responsible for implementing the presentation logic of GIS data. Generally, the client without local-storage data is a J2ME Midlet that controls the GUI and is analogous to a standard Web browser. The J2ME Midlet needn't perform any GIS business logic, don't directly connect to a back-end database server, and don't store itself state information, so it is a really "thin client".

Mobile service tier

JAVA service tier includes a JAVA Proxy (often referred to as a Mobile Gateway) and a Web Server. The JAVA Gateway was required to handle the protocol interworking between the client and the Web Server. As depicted in Figure 1, the JAVA Gateway consists of parsers and mobile device synchronizers. The JAVA Gateway can optimize the communication process and may offer mobile service enhancements, such as location, privacy, and presence based services. The JAVA Gateway communicates with the client (j2ME) using the TCP/IP protocols and it communicates with the Web Server using the standard Internet protocols such as HTTP/HTTPS. Once the JAVA Gateway receives requests from the client, it translates the requests to HTTP requests, and then sends them to the Web Server. Once the JAVA Gateway receives HTTP responses (web contents) from the Web Server, it translates the web contents to compact encoded binary formats for reducing the size and number of packets traveling over the wireless network to the client for displaying and/or processing.

The Web Server includes a Web Container and Web protocols support, security support, and so on. Web caching mechanism for Geospatial Information The Web Container is responsible for managing a Java Servlet Engine and Java Server Pagers (JSP). An Internet GIS system based on Java often uses Java Applets and/or Servlets to extend the dynamically displaying functionality for the Web browser. This paper uses a Servlet Engine instead of running Applets and/or Servlets inside the Web Server, which has several advantages as follow:

(1) The Servlet Engine runs itself inside the Web Server. But applets inside the Web Server must be dynamically downloaded to the client for processing, which would increase the load of wireless and mobile devices (Especially, a WAP phone has not the capability of running applets.).

(2) The Servlet Engine can host Servlets and provide them with standard Java Servlet APIs.

(3) The Servlet Engine decouples Servlets processing from specific implementation details of the Web Server. This increases the flexibility of the system architecture, as it allows the Web Server to change without impacting on the overall system. The Servlet Engine can also provide various management features that help to shift the load of Servlets processing away from the Web Server.

The client devices of JAVA -based Mobile GIS have several types (e.g. PDA and J2ME supported phone) whose displaying and presenting capabilities are different. So there must has a mechanism at the server side, that is, the Web Server may determine the type of the client device and generate corresponding presentation logic for the client. Fortunately, the Servlet Engine can solve the problem effectively. The Servlet Engine may provide two kinds of Servlets: the one is presenting Servlet which is responsible for generating corresponding presentation logic according to the type of the client device for the client; the orther is identifying Servlet which is responsible for determining the type of the client device (through the only ID of the client device accessing mobile networks) and then notify presenting Servlet.

The Servlet Engine is responsible for manage Servlets and provide support for JSP. Since Servlets run inside the Web Server, everyone does not need a GUI. Servlets in the same servlet engine can share resources and chain together. So the Web Server can generate dynamical Web contents by running Servlets (i.e. presenting Servlets) to response for client requests like CGI. And Servlets are more efficient than the CGI approach and the Fast-CGI approach. Through Servlets and/or JSP, the client side may indirectly perform EJB components inside the Application Server to implement GIS business logic, such as spatial analysis, spatial and attribute querying, route planning, geo-coding and gazetteer, and so on.

Application tier

Application tier is the core of the architecture. It corresponds GIS Application Servers that communicate with the Web Server in JAVA service tier through Remote Method Invocation (RMI). An EJB container at an Application Server is the runtime environment of EJB components including GIS Session Beans and GIS Entity Beans, and controls these components to be performed and transferred. At the same time, the container also provides these components with all required services for distributed computing environments. Thus, these EJB components could more efficiently execute in the Application Server. The EJB components can use JDBC (Java Database Connection) technology to access to database servers, and use JMS (Java Message Service) technology to connect to back-end legacy systems.

Data service tier

Data service tier corresponds Database Servers that are used to manage and store geospatial and attribute data of the whole system. Object oriented database management system (OODBMS) is the most desired database server for a GIS system, but OODBMS is immature and very costly currently, so it is not popular and to be used commonly (Gong, 2001). At present, the mainstream solution is that large object-relation database systems such as DB2, Oracle, Sybase, SQL Server, and so on are used to manage and store GIS data, at the same time, spatial data engine (SDE) also can be developed to build the communication between data service tier and application tier. SDE is an open standardsbased middleware such as ArcSDE, Spatial Ware and Oracle Spatial.

5.5 A Prototype System

Using the 100% Pure Java J2EE solution, the above architecture is open and crossplatform, and has the characteristics such as extensibility, stability and reliability, which could satisfy system requirements of JAVA -based Mobile GIS. Based on the architecture, a prototype system of JAVA -based Mobile GIS has been developed. In the prototype system, the server side uses Apache Tomcat 5 as the web server, BEA WebLogic 8.1 as the application server, Oracle 8i and ESRI ArcSDE as the database server, and it provides a range of basic GIS functions (e.g. spatial and attribute querying, route planning, geo-coding, mapping, etc.) to a Mobile GIS user using the Servlet Engine and EJB components to gain and process spatial geometry object in back-end spatial databases. The client side uses J2ME supported phone with a WAP microbrowser and may obtain maps in WBMP format from the server side, and it has the basic map browsing functions such as zooming out, zooming in, moving (up, down, left and right). The running results are showed in Figure 5.2.

5.6 Prospects

JAVA -based Mobile GIS is a very new study field of Geo-informatics. Above all, the architecture is solved, because it is the key and core of the distributed system. In this paper, based on J2EE, an open distributed architecture is proposed for JAVA -based Mobile GIS. And a prototype system is designed and developed based on the architecture which is composed of four logic tiers. The prototype system can successfully work on JAVA phones, which proves that the architecture is feasible.

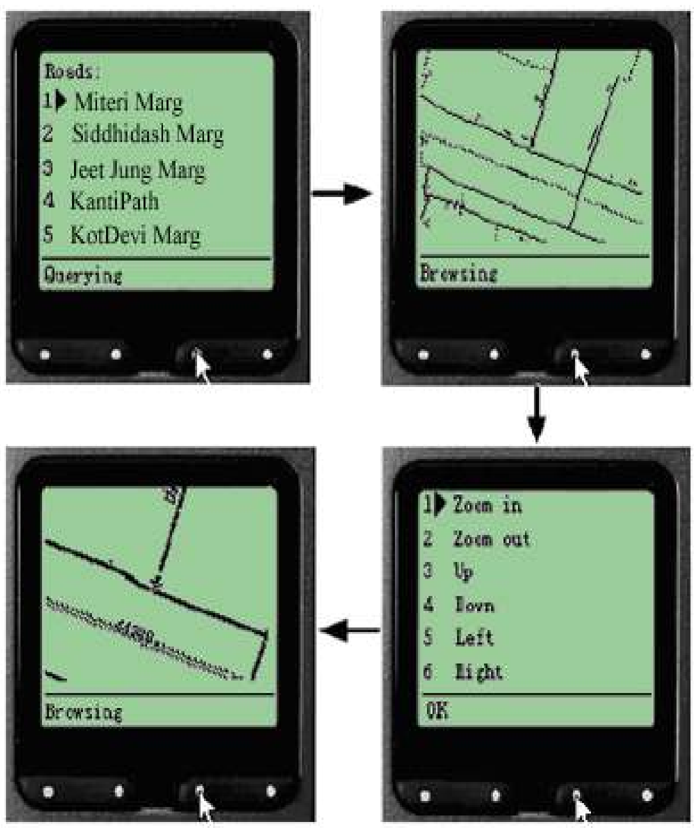

Figure 5.2 A prototype system Mobile GIS

At present, low bandwidths are still the main bottleneck of all mobile applications (Wei et al., 2003). So researches on the organization of geospatial data at the fat server side and on the presentation of geospatial information at the thin client side should be done farther. Fortunately, the mobile network is towards the development of 3G. In 3G age, when the mobile terminal moves at the same speed as vehicle, the transmission speed is 144 kbps, when the mobile terminal moves at the walking speed or un-moves in outdoor, the transmission speed is 384 kbps, when the mobile terminal is in the room, the transmission speed is up to 2 Mbps. The 3G bandwidths will satisfy with the requirements of geospatial information wireless transmission. Therefore, it is not hard to understand that JAVA -based Mobile GIS will have better development and application perspectives and considerable business value.

Chapter 6: Methodology

6.1 Mapping to Graph Structures

Trees, tables, and hash tables are optimized for different access patterns. Here, we consider another frequently used data structure: graphs. Graphs are convenient when the access pattern of your program involves following links between shared data.

Let us take an application that computes the shortest path between two cities as our example. Figure 6.1 shows our problem. Circles represent cities, and arrows between them represent connections between cities. The numbers associated with the arrows are the cost of moving to one city from another. What is the shortest path from Fukuoka to Yamato? What is the cost of the shortest path? There is a well-known, simple but efficient algorithm known as Dijkstra's algorithm to solve this problem.

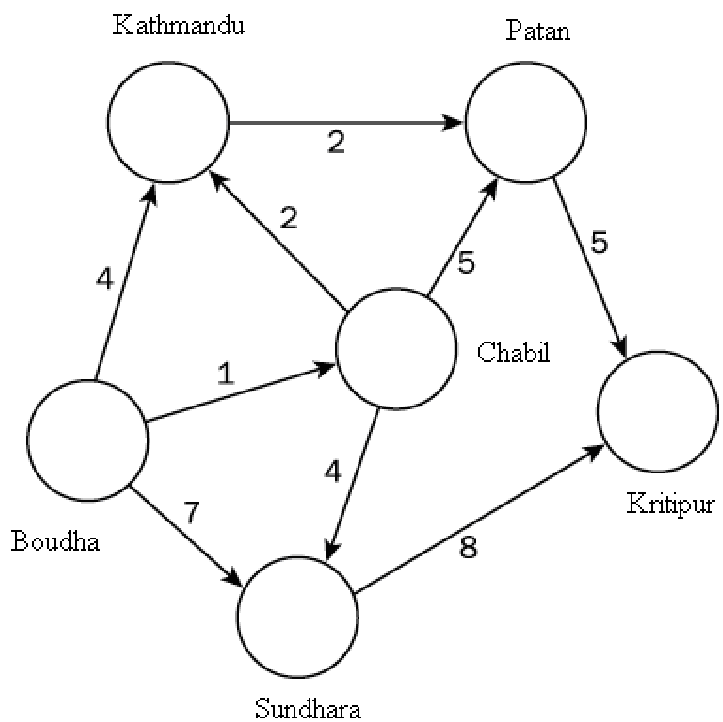

Figure 6.1. Graph data

We want to express this problem as an XML document and map it into a graph data structure, as shown in Figure 6.1, that is needed for implementing Dijkstra's algorithm efficiently. One natural XML encoding of the problem is something like cities.xml, shown in Listing 6.1.

The developed code will convert above XML document into a graph structure, whose vertexes are represented by Vertex objects and whose edges are represented by Edge objects.

Part of the Vertex class is shown in Listing 6.2. A Vertex instance has a string, label, for the city's name and a list, outgoingEdges, that holds all the edges pointing to other cities. In addition, it has two variables, bestScore and bestPath, to keep track of working values during the computation. All the vertexes are kept in a single hash table named vertexList. When a vertex is needed, we call the static method register() instead of creating a new instance. This method checks whether there is already a vertex for the city and returns it if there is.

The Edge class has fromVertex and toVertex to keep both ends of the connection (see Listing 6.3). When a new edge is created, it registers itself in outgoingEdges of fromVertex.

Now that we have both our application data structure and the input XML structure, we are ready to consider the mapping between them. Our strategy is this. We will scan an input XML document. Whenever we encounter a city, whether it appears in a Path element or in TargetCity or StartCity, we register it using Vertex.register(). When we see a Path element, we also need to create an Edge instance. Because the constructor of the Edge class takes care of updating the appropriate variables in the structure being built, we do not need to do anything other than create a new instance of Edge.

We implement our strategy in a SAX handler, ShortestPath.java, shown in Listing 6.4.

For the StartCity and TargetCity elements, we need to accumulate possibly fragmented characters in text content using a StringBuffer. These city names are registered when end tags are processed . When a path element is found, its three attributes are extracted, the from city and the to city are registered, and an Edge object is created. In this way, we can map an input XML document into our application data structure.

Dijkstra's algorithm computes the shortest paths from a given vertex to all vertexes simultaneously. Therefore, the value of TargetCity is not used in this program.

For completeness, we show the rest of our main class, ShortestPath, in Listing 6.5.

Above listing contain implementation Dijkstra's algorithm. It uses a helper class, VertexComparator, to compare the current cost values between two Vertex objects.

As we have done in this example, if we need to map XML documents into a graph structure, we need to give labels to each node and refer to these labels to express the links between nodes. Having a hash table to record the mapping between node labels and node objects is essential in this process.

6.2 Prototype of M-Commerce Application

The managed GUI bundle uses only a fraction of the power provided by the OSGi container. Through its services, the OSGi container supports external applications and devices over the network. To use a separate program to render the UI, we can more effectively separate the business and presentation layers.

The "pizza order" example application distributed with the SMF illustrates the use of HTTP services in the OSGi framework. After starting the bundle from the SMF console, we can launch the device's built-in HTML browser (Internet Explorer for PocketPC or the Opera browser for Embedded Linux) and point the URL to `http://localhost/pizza`. An HTML page of a dummy pizza store appears. We can fill out the pizza order form, submit the form, and get response from a servlet running inside the bundle. The screen flow is shown in Figure 6.2

Figure 6.2. The Mobile application in action.

This design allows us to build a simple UI very quickly, using HTML without messaging with complex event handlers in AWT code. It also allows the vast majority of server side Java developers to transfer their skills and make use of their familiar patterns, such as the Model-Viewer-Controller pattern. The overall architecture of the smart client with HTTP front end is illustrated in Figure 6.3.

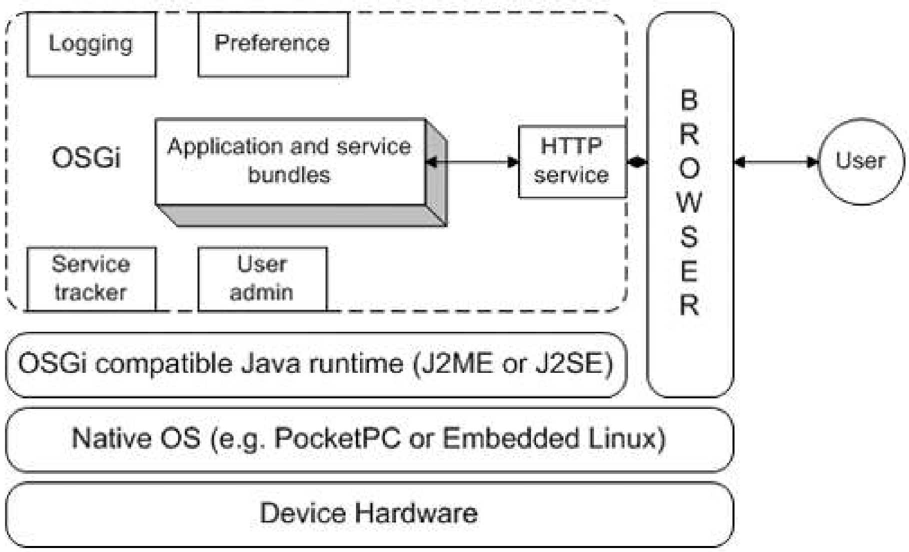

Figure 6.3. The smart client with HTTP front end.

6.4 The Order System Bundle

The PizzaBundle class (Listing 6.6) implements the BundleActivator interface. This bundle does not register or provide any new services. It customizes the container HTTP service to serve pizza order HTML content at a specified URL. It also uses the container logging services to record activities inside the bundle. If no logging service has been registered for this bundle, it logs to the standard output.

The bundle start() method instantiates trackers for LogService and HttpService. The bundle then invokes their open() methods.

When the ServiceTracker for the HttpService is opened, it obtains all registered HttpService references from the container, adds them into the tracker, and invokes the corresponding ServiceTrackerCustomizer's addingService() method for each added reference.

The addingService() method obtains the HTTP service object from the container and customizes it with the pizza order servlet and other resources.

When the bundle stops, its stop() method calls the close() methods of the two ServiceTrackers. The HttpService tracker in turn calls the removedService() method, which unplugs the servlet from the HTTP service.

6.5 The Order Servlet

In the addingService() method, we use a servlet Pizza (Listing 6.7) to provide the custom HTTP service (the application logic). This is just a standard Java servlet that reads from HttpRequest and writes HTML data to HttpResponse objects. Those HTTP context objects are provided by the container.

6.6 The Logging Service

A container can have multiple logging services. For example, one service implementation could log messages to a disk file while another could send the critical message as Instant Message alerts to administrators. The OSGi framework provides a common LogService interface for all logging services. Implementations of the LogService interface are provided and shared by individual bundles. In our pizza example, the LogTracker (Listing 6.8) object is a ServiceTracker object that tracks all available logging services from the container registry and makes sure each message is logged by all services.

The LogTracker.open() method is invoked in the bundle start() method to initiate the tracker.

LogTracker.open() calls its base class's open() method, which obtains all LogService references in the container.

When the bundle needs to log a message, it calls the LogTracker.log() method, which iterates through the current list of tracked LogService references. It obtains the service object for each reference and pushes the message to all available logging services.

If the container does not have any registered logging service (the reference list size is zero), the LogTracker object will call its noLogService() method to log to the standard output.

The LogTracker class does not register any new LogService to the container.

6.7 Rich UI Clients for the HTTP Service

Although the pizza order example supports clearly separated application layers, the drawback is that it does not really take advantage of the rich UI capability of smart clients. There are several ways to create rich UI clients for the HTTP service.

Rich browsers: Instead of plain HTML content, the servlet can provide rich content such as Java Applet and Flash for capable browsers.

Standalone GUI: We can also replace the browser completely with a standalone GUI application. The OSGi HTTP service can serve binary or XUL (XML User Interface) content and allow the standalone GUI front end to decide how to render it.

6.8 Mobile Clients that uses Mobile GIS Services

6.8.1 Mobile GIS Services

Mobile GIS include all applications that make use of location and geographic data. It is touted as the killer app for mobile commerce because it enables brand new applications not possible in the desktop world. There are several types of Mobile GIS applications:

Pull-based applications: The user actively sends out her location information to a server and pulls in location information such as driving maps and nearby stores.

Push-based applications: Service providers detect users in their proximity and push out services such as coupons and advertisements. A central issue in push-based applications is how to protect user privacy. Location information can be highly sensitive. It is extremely important vendors get information from and push services to only opt-in users.

A combination of pull and push: For example, a user can first pull nearby coffee shop locations and make a mobile reservation. Then he can send meeting alerts to peers within certain distances from the shop. The peers will be able to accept or reject the invitation.

Although Mobile GIS offers appealing business values, its implementation technologies are sophisticated. The core technology for any Mobile GIS solution is Geographic Information System (GIS), which performs important functions such as determining street addresses from coordinates and vice versa; look up yellow pages and landmarks; calculate optimal routes and render custom maps. It is the integration point of many heterogeneous components in the Mobile GIS application. This makes XML Web Services a perfect fit here. In the next section, we discuss a leading commercial GIS Web service: Microsoft MapPoint.

6.8.2 Microsoft MapPoint Web Services

Standalone GIS servers (e.g., Oracle GIS server) that are capable of geo-coding, map rendering, and other complex geographic algorithms have been commercially available for a long time. However, running a GIS server by yourself is probably too expensive for most small to mid-sized businesses. The costs include:

Hiring GIS experts to set up and maintain the system.

Licensing high-resolution digital maps and business yellow page listings. This could be very expensive if your users roam across many metropolitan areas.

Aggregating geographic information updates from many sources (e.g., road construction and business address changes) and updating them frequently to the database.

Ongoing server administration for secure and high availability services.

Licensing cost of the GIS server software itself.

For most companies, it makes sense to outsource the entire GIS operation to specialized providers. Microsoft's MapPoint Web Service is a managed GIS solution accessible via a set of platform-independent SOAP Web Services APIs. MapPoint also allows users to import their own geo-coding data and points-of-interest listings for customized services.

6.8.3 The MapPoint v3.0 SOAP API

MapPoint Web Services expose a very rich set of SOAP APIs. Important remote methods in MapPoint v3.0 are divided into four categories.

Common Service contains utility functions that are common to other services.

Find Service allows the user to do generic geo-coding tasks, including finding addresses from latitudes and longitudes, and vice versa. It also allows searching for nearby points of interests.

Route Service calculates routes and directions based on locations and waypoints. It also generates map view representations of the calculated routes. The map view can be used to highlight routes on a map.

Render Service renders the map for the locations or routes. It highlights routes, places pushpins and icons, and supports zoom.

Table 6.1 lists the methods from all categories. The first 5 methods in that table belong to the common service; the next 5 methods belong to the find service; the next 2 methods belong to the route service; and the last 4 methods belong to the render service. The API also defines many composite data types (e.g., Address, MapView, and RouteSpecification) to pass data to or from those service methods.. For interested readers, the Axis gateway example demonstrates the use of many basic API functions from a Java client.

Method	Description
GetCountryRegionInfo	Look up country or region information.
GetDataSourceInfo	Return information of a data source (e.g. North America and Europe).
GetEntityTypes	Look up the data source to find information about entities.
GetGreatCircleDistances	Calculate an array of great circle distances for a set of location points.
GetVersionInfo	Return version information of the current MapPoint service.
Find	Find geographic entities based on search options.
FindAddress	Return a list of addresses based on search options.
FindNearby	Find points of interests close to a specified location.
GetLocationInfo	Return address or other information of a specified pair of latitude and longitude.
ParseAddress	Return an Address object from a string representation of an address.
CalculateSimpleRoute	Calculate a route from start/end points specified by their latitude and longitude coordinates using default options.
CalculateRoute	Calculate routes or route segments from route specifications. It allows more options than the CalculateSimpleRoute method.
GetMap	Retrieve the rendered map according to the rendering options and map specifications pass to this method.
GetBestMapView	Get an optimal map view that contains all locations and routes. We can pass this map view as part of the map specifications to the GetMap method to get the actual map.
ConvertToLatLong	Convert pixel coordinates on a map to latitude and longitude coordinates.
ConvertToPoint	Convert latitude and longitude coordinates to pixel coordinates.

Table 6.1. The MapPoint v3.0 SOAP RPC API

6.8.4 The Aggregated API

Through the SOAP API, MapPoint Web Service offers fine-grained access to its GIS back end, allowing us to design flexible applications without arbitrary limitations imposed by preset scripts. However, the trade-off is that we have to make separate method calls for each small step to complete a task. For example, in order to retrieve driving directions between two addresses, we have to go through the following steps.

Convert the addresses to latitude and longitude coordinates.

Calculate the route according to options.

Render the overview map with the route and start and end locations highlighted.

Retrieve turn-by-turn instructions for each route segment.

Render highlighted turn-by-turn maps for each route segment.

The last two steps need to be performed repeatedly for a long route with many segments. Since all those method calls are remote SOAP calls, those excessive round trips cause long delays over slow wireless networks. To minimize the data transfer over wireless networks, we can place a remote facade between the mobile client and MapPoint. The facade takes in two human-readable addresses in a single SOAP call, queries MapPoint via many API calls over the wired Internet, constructs the resultant driving directions, and returns the results. It provides a coarse-grained, simple interface that aggregates functions of the fine-grained MapPoint API methods for specific applications. Figure 6.5 illustrates the facade architecture.

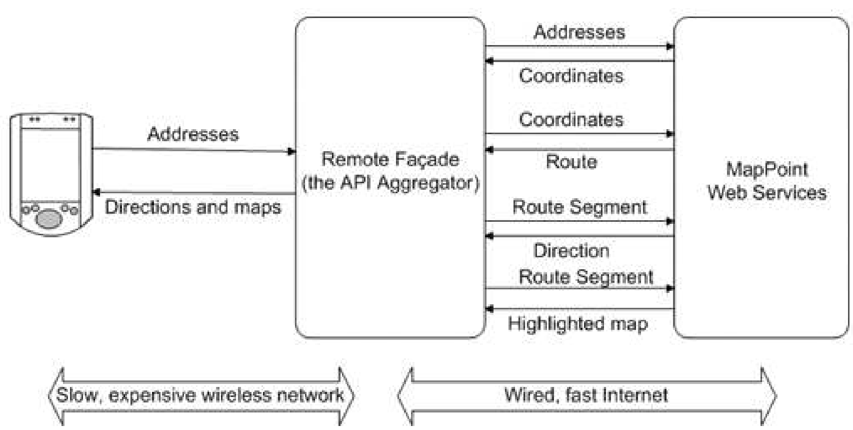

Figure 6.5. The remote facade architecture.

6.8.5 MapPoint Authentication

MapPoint Web Services use HTTP Digest Authentication to authenticate users and keep track of billing. So, any client application must support HTTP Digest headers in the network transport layer.

As a Microsoft platform, MapPoint was never designed for Java clients. But the use of SOAP Web Services makes it platform-independent. This highlights the power of Web Services.

6.8.6 The Axis Facade

In this section, we introduce a Java-based SOAP facade (gateway) that aggregates generic MapPoint APIs into convenient remote methods for specific tasks to minimize round trips for mobile clients. The gateway is built using Apache Axis (v1.0). It is developed as follows:

Generate Java SOAP stub classes for the MapPoint SOAP API using the WSDL2Java tool in Axis.

Use those Java stub classes to develop a Java object, MPClient, that contains convenience aggregated methods.

Publish the MPClient object as a SOAP Web Service via Axis's server interface. This is the RPC interface for mobile devices. The steps are as follows.

Configure and run Axis server inside a Tomcat server.

Copy the JAR file containing the MPClient class to Axis library directory.

Specify the scope of the service object, methods to expose, and other options via a .wsdd file.

Deploy the .wsdd file to Axis through its server administration utility.

Due to the limited space, I show only the public interfaces of remote methods in the MPClient class here (Listing 6.9). The API usage is embedded in the code comments. The complete code is available in the AxisFacade project, downloadable from Web site. The project archive also contains generated stub classes, Axis libraries, sample .wsdd files, and ANT script to build and deploy the facade gateway.

We have to call the getDirections() method to obtain the route before we can retrieve any map.

The MPClient facade class aggregates driving direction services for illustration purposes only. In real-world, location-based applications that require more functionalities, we should design and implement facade gateways that fit your own application needs.

6.8.7 MapPoint J2ME Clients

With the Axis facade gateway, we have transformed the MapPoint Web Service to a wireless-friendly service while still preserving the benefits of SOAP Web Services. In fact, we can use any Web Services-compatible client to access MapPoint services through the Axis gateway. Those clients can be developed on almost all Microsoft platforms, all Java editions, script languages such as Perl and AppleScript, as well as many native C/C++ SOAP frameworks. In this section, we demonstrate two simple J2ME clients for driving directions.

6.8.8 CDC/PP and Personal Java Clients

Figure 6.6 demonstrates the driving directions client in action on a PocketPC device running Jeode PersonalJava VM. Thanks to the standard AWT library, the same client runs on CDC/FP/PP devices (e.g., IBM WebSphere Micro Edition's Personal Profile runtime) and standard J2SE desktops (e.g., Windows, Linux, and Mac OS) without modification.

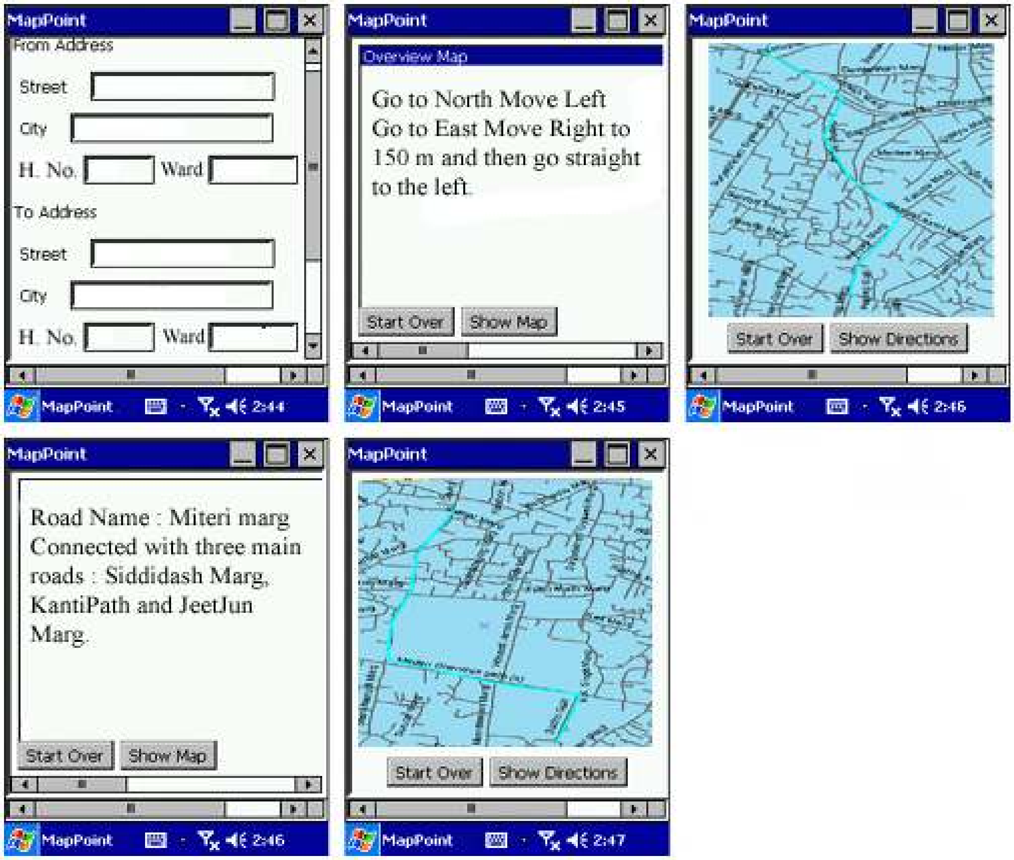

Figure 6.6. The PDA-based driving directions client in action.

Key code snippet that queries the Axis gateway service for directions and maps is shown in Listing 6.10.

6.9 MIDP Clients

Figure 6.7 shows the MIDP client in action in a MIDP v2.0 emulator. Since MapPoint supports only GIF image format rendering at this stage, but MIDP requires PNG format, driving maps are not yet available in the MIDP client. A quick fix would be to convert GIF images to PNG images inside the facade using the Java imaging API.

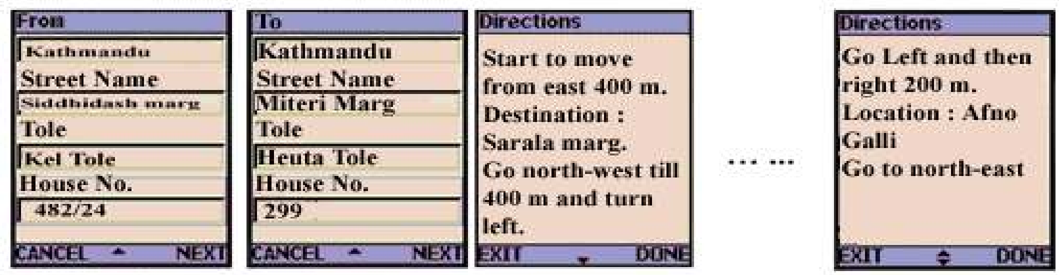

Figure 6.7. The MIDP phone-based driving directions client in action.

Enhancing the Driving Directions Application

The driving directions application we demonstrated above is not particularly easy to use. For example, if we are lost while driving around, we need to find the route from the current location to the destination. To find out the current street address and manually enter it into the device could be a major nuisance. A driving directions application that is aware of the current location context and automatically fills out the From field can be much more valuable to users. In fact, most LBS business models require access to the mobile client's current location.

6.10 Location Determination Techniques

There are several ways to obtain the current locations of mobile devices.

Terminal-based: A GPS-equipped device can calculate its coordinates using GPS satellite signals. This method is accurate and straightforward, and it works everywhere in the world. However, GPS modules are expensive, slow, and a drain on batteries.

Network-based: Cellular network operators can determine location of any phone in the network using its signal strength received by three nearby access stations (triangling). On-device smart applications (e.g., the driving directions application) can access the location data via the data network. With user authorization, the location data can also be made available to third parties from the carrier's Web site. This is an excellent approach if we need to track a large number of users from the back end (e.g., push services). However, the drawbacks are that the user must be within the cellular network coverage; the service is not yet available nationally, and the location accuracy is not as good as the GPS approach.

Network-assisted GPS: Of course, we can combine the above two approaches: GPS devices can use network data to determine an approximate position first and then use the GPS module to get accurate corrections. This speeds up the GPS look-up process considerably.

Local wireless network-based: Advances in new technologies will also enable location determination in local wireless networks, such as WiFi and Bluetooth networks, in the future.

User-assisted: As the last resort, we can also ask the user to identify the closest landmark and calculate an approximate coordinate based on the known coordinate of the landmark. This works well in controlled environments like a college or company campus.

The Oracle wireless application server works with wireless carrier location servers to provide SOAP interfaces for network-based location data of all devices in the network.

6.11 The Location API for J2ME

The J2ME Location API (JSR 179) is a standard Optional Package for both CDC-based and CLDC-based devices. It allows J2ME applications to access location information on any device through standard APIs regardless of the underlying technology and service provider. Table 6.2 shows the entire J2ME Location API defined in the javax.microedition.location package. It is based on the v1.0 final release (September 2003).

Class	Description
AddressInfo	Contains information about an address, such as building number, street number, city, and state.
Coordinates	Represent three-dimensional geographic coordinate information (i.e., longitude, latitude, and altitude) from GPS devices or network operators.
QualifiedCoordinates	Extend the Coordinates with 1-sigma error bars qualifying the accuracy of the coordinates data.
Orientation	Contains data about the device's three-dimensional orientation. It is up to the device to define its own axes.
Location	Holds information about the device's current location, including the current QualifiedCoordinates, speed, course, coordinates retrieval method, and an optional AddressInfo.
LocationProvider	It is the factory class to retrieve Location from underlying providers (e.g., GPS device interface or network location server).
Criteria	Specifies criteria to select location providers. Those criteria include cost, accuracy, response time, power consumption, and supported features.
LocationListener	It is a listener interface that can be registered with a LocationProvider. It handles events of location and provider state changes.
Landmark	Represents a location with a known address.
LandmarkStore	Represents a collection of Landmark objects.
ProximityListener	This is a listener interface that can be registered with a LocationProvider. It handles events when the device enters the proximity radius of the specified coordinates.

Table 6.2. The javax.microedition.location Package

CLDC v1.1 is required for the J2ME Location API because coordinates are expressed in float or double numbers.

Listing 6.11 illustrates how to construct a simple location-based application using the JSR 179 API. For more information about the J2ME Location API and usage examples, please refer to JSR 179 documentation.

Chapter 7: Issues in Implementation

7.1 Challenges in M-Commerce

NTT DoCoMo released 3G in May 2001. The response is unexpectedly low. Only 114,500 users signed up for the 3G services. This failure is due to the high cost of 3G services. Also, the users have to buy a new mobile phone. Furthermore, the networks have limited coverage. M-commerce, which is more complex than e-commerce, faces a number of challenges (see Figure 7.1).

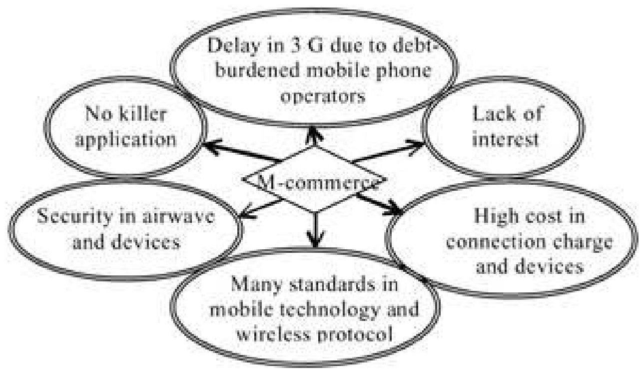

Figure 7.1: Challenges in M-Commerce

7.2 Delay in 3G

Mobile network operators (MNOs) are pushing m-commerce. The success of m-commerce in Japan changes the concept of "free" Internet to "paid" Internet. Users are willing to pay for the service. MNOs have foreseen a huge profit in taking control of the backbone of m-commerce—the wireless infrastructure. In addition, MNOs also play a dominant position in providing m-commerce applications. This created unreasonably high expectations of 3G services. Big companies in Europe, such as Deutsche Telecom, France Télécom, Spain's Telefónica, and the U.K.'s Vodafone spent an estimated US$125 billion to US$150 billion on 3G licenses [20]. Many of them are burdened with high debts. Banks downgraded their credit ratings. At the same time, the global economy is in a downturn after September 11, 2002, and the accounting scandal at Enron, the energy firm, which was followed by similar problems at WorldCom, the telecommunication giant in the United States. All of this bad news destroyed the confidence of investors in technology stock. It is extremely difficult for any of the MNOs to raise funds through the capital markets. Hutchison postponed the plan of raising money through the issue of bonds. With their huge debts, network operators find it hard to cut costs. Spectrum trading and network sharing are the ways to cut costs. These enable network operators to save money when deploying 3G. However, there are many unknowns in sharing and spectrum trading: competition, ownership of the license obligations, and responsibility for the license debts [21]. Government support and assistance are important in resolving difficulties that arise.

The 3G requires more cells, and thus, more new radio stations have to be built. This requires a lot of investment. With huge debts, MNOs have significant difficulties in meeting the requirements to provide the new 3G services. MobilCom, the German mobile operator, is cutting more than half of its workforce. The preparation for 3G is on the shelf. In Singapore, mobile phone operators held discussions with governments to delay the rollout deadline for 3G. The debut of integrated 3G IC solutions was delayed; the "life-cycle" of GSM, CDMA, TDMA, AMPS, and particularly, 2.5GPRS systems will be extended [21]. In the near term, the future of 3G looks gloomy. NTT DoCoMo wrote off a 126 billion yen (about US$1 billion) investment in Hutchison 3G (U.K.). Also, Orange, which is billions of euros in debt, shelved its plan to run a 3G mobile service in Sweden.

7.3Lack of Interest and High Cost

On the other hand, MNOs in Europe have the problem of falling revenues. According to Gartner's report, the connection growth for last year declined from 57.4% to a miserable 19.5% in the United Kingdom (NUA analysis, 2002). The Western European market reached the saturation point, where the mobile possession rate is close to 100% in some countries. In addition, mobile users have "upgrade fatigue," i.e., they are reluctant to upgrade their mobile phones. A report from the World Market Research Center, released in May 2002, predicted that 3G sales are likely to be sluggish because of a lack of consumer interest in new services. The research done by the mobile consultancy group Detica also warned that there is no single "3G killer application" that will entice large numbers of people to embrace 3G. In 2002, the mobile phone business pushed hard on picture messaging, which requires new expensive handsets. The response has been poor. The mobile revenue mainly comes from voice calls and SMS messaging.

7.4 Young Market with Many Standards

This is a young market with many existing standards. 3G standards have some commonalities, but they are not fully compatible with each other at the air interface (radio-transmission) level. This imposes a limitation on global roaming and the creation of mass-market volumes for terminals and infrastructures. The third-generation partnership project (3GPP) was formed to produce technical specifications for 3G mobile systems and works with different 3G systems. In addition, the spectrum problem in the United States casts a shadow on global roaming. The Department of Defense occupies the spectrum band needed for 3G services. After September 11, 2002, the U.S. military is unlikely to give up the spectrum. This introduces further technical difficulties. In addition, Japan is going it alone to develop a fourth-generation standard (4G) that could offer data rates of up to 100 Mbits/s. However, during an International Telecommunication Union meeting on 4G, Japanese representatives limited the debate in order to enshrine Japan's vision as the de facto standard for 4G (Wireless Web Analysis, 2002).

WAP is disappointing. Though i-mode is popular, it is proprietary. However, both formats have the quality that they benefit from each other. There is a need for an open standard for wireless protocol. Industry cooperation is a must if a solution is to be found. At present, industry is working toward limiting the number of markup languages available for mobile devices. Both WML and cHTML are now migrating into XHTML (extensible HTML), which are XML based. There are still many multiple proprietary browsers and operating systems. Developers find it hard to develop applications with the myriad of handheld devices. In the future, it all depends on the role of XML and the evolution of the WAP standard.

The advances in technology resulted in a convergence of the functions of mobile phone and PDA devices. Mobile phones are commonly equipped with PDA functionality, such as a large screen and easier methods of input, e.g., pen with touch screen. PDAs have a phone function. There are different kinds of handheld devices from phone and computer manufacturers. The market for handheld devices is different from the personal computer market. For instance, Nokia not only produces the phone (hardware) but also develops Symbian (the software), together with other phone manufacturers such as Motorola. This is also the problem with PDAs. Each device comes with its own proprietary operating system. Many standards are competing with each other. Microsoft joined the competition by rolling out two operating systems: Pocket PC2002 and the smartphone system for PDAs and mobile phones, respectively, in 2002. Both systems can be synchronized to a desktop PC, a convenience for business users. Microsoft, together with Orange, launched an SPV (Sound, Picture, and Video) phone, the first mobile phone with the Microsoft smartphone system, in October 2002. The participation of Microsoft in the handheld device market will have a major impact on market development.

7.5 Security

Mobile communications offer users many benefits, such as portability, flexibility, and increased productivity. Handheld devices allow remote users to synchronize personal databases, and they provide access to e-mail, Web browsing, and Internet access. The most significant difference between wired networks and mobile communication is that the airwave is openly exposed to intruders. This is the main source of risk—some existing problems have been exacerbated and some are new. Analog phones using 1G technologies are more vulnerable to eavesdropping than digital cell phones. The unencrypted analog signal can be intercepted by using simple radio scanners, whereas digital phones are protected using encryption. Nevertheless, digital systems are also subject to attack. For instance, there is the "man-in-the-middle" problem, where the intruder sets up a base station transceiver with a modified phone. The intruders put themselves in between the target user and a genuine network and have the ability to eavesdrop, modify, delete, reorder, replay, and send spoof signaling. The intruders eavesdrop on signaling and data connections associated with other users by using a modified mobile phone. 3G imposes more security measures to defeat false base-station attacks. A security mechanism including a number sequence is implemented to ensure that the mobile can identify the network. Overall, 3G technology has far greater security than 2G and 2.5G.

The wireless handheld device provides the advantage of mobility for mobile workers but also creates new types of security risks. The types of risks are summarized as follows [25]:

Their small size, relatively low cost, and constant mobility make them likely to be lost or stolen. Sensitive information such as the "private-key" is thus vulnerable. The mobile phone tops the list of property left behind in the U.K. railway system.

PDAs can beam information from an IR (Infrared) port to another IR port to easily exchange contact information such as telephone numbers. The data is unencrypted, and any user who is in close proximity to the handheld device and has an appropriate device pointed in the right direction can intercept and read the data. This is known as data leakage.

Smart phones may lose network connectively not only when they are outside the coverage area but also from the use of cell phone jammers, which block cell phone communication. Many restaurant and movie theaters installed such devices without warning.

A 3G mobile device, when connected to an IP network, is in the "always-on" mode. This mode minimizes the need for the device to go through an authentication procedure each time a network request is made. Thus, this "always-on" mode makes the device susceptible to attack. Moreover, it also provides the opportunity to track a user's activity, which may be a violation of privacy.

Viruses have not been widely considered a security threat in PDAs or mobile phones because of their limited memory and processing power. Moreover, users typically synchronize their data with their PCs, so they can recover any lost or corrupted data simply by synchronizing with their PCs.

7.6Limitations of Handheld Devices

Technological developments will increase the computing power and storage in handheld devices. However, insufficient battery life and power consumption will impede the potential growth of m-commerce, even when 3G is widely available. At present, the battery life is very short. It can last for two or three hours for surfing the Internet or talking. The battery has to be recharged from time to time, or a second battery must be used. Fuel cells will be a solution to this problem, but they are in the early stages of introduction and will be expensive. In the near future, short battery life is a major barrier in handheld devices. Besides this, the small screen is another limitation. PDAs have a larger screen (4.9×3.1 inches). The color touch screen makes it easier for users to navigate the Web, however, they are expensive, normally around US$600 compared to a black and white one priced at US$100. Furthermore, mobile phone users still prefer the small-sized phones. The screen of a mobile phone is 2×2 inches, which causes difficulty when surfing the Web. Many mobile users are aware of 3G but do not understand the "user experience." A low-power, inexpensive, high-resolution color display would seriously increase the growth of m-commerce.

Chapter 8: Future Trends

8.1 Introduction

Some important issues and concerns must be addressed and solved in order to embrace and deploy mobile commerce. Future and emerging trends include three main challenges: security [24], service discovery and transaction management [25].

Mobile commerce offers an exciting new set of capabilities that service providers can leverage to increment their revenue base while attracting new services that enhance the end-user's experience. With these new opportunities, the risk of new security threats also arises [24]. New mobile devices such as PDAs and GSM/UMTS terminals enable easy access to the Internet and strongly contribute to the development of e-commerce and m-commerce services, whereas Smartcard platforms will enable operators and service providers to design and deploy new m-commerce services. This development can only be achieved if a customer's information and transactions are guaranteed to be protected by a high level of security (22). Thus, establishing security mechanisms which allow diverse mobile devices to support a secure m-commerce environment in wireless Internet is critical [24]

Providing security provisions for the m-commerce community is also challenging due to the insecure air interface of wireless access networks, limited computational capability of mobile devices, and users' mobility [22]. The limited equipment resources in terms of equipment require the e-payment protocol in the wireless Internet environment to be designed in consideration of the efficiency of the computing functions and the storage device.

Until now, much of the research on m-commerce has focused on the problem of service discovery. However, once a service is discovered, it needs to be provisioned according to the goals and constraints of the service provider and consumer. In this context, automated negotiation protocols and strategies that are applicable in m-commerce environments must be proposed [21]. Specifically, time-constrained bilateral negotiation algorithms that allow software agents to adapt to the quality of the network and/or their experience with similar interactions must be developed and evaluated.

Finally, transaction management is a major issue in m-commerce. It enables people to order goods and access information anywhere, anytime. Given the nature of mobile computing, there is a need for a generic approach that adapts to the needs of m-commerce applications..

8.2 Challenges for Technical Research

Our approach is to create broad scenarios shaped by "mega-trends" within technology, society, industry, and among users, and then use them to identify important research areas. We believe it is a fruitful approach, but we do not believe we have identified all, or even most, important fields where a lot of work has to be done. The scenarios represent our visions of the future and are open to further interpretations. Hopefully, the reader of our scenarios can draw other conclusions about technical implications, thus identifying other important research topics. Many of the areas identified here are well in line with other future-oriented studies, for example WWRF (2001) and Bria et al. (2001), and are well supported by the scenario descriptions.

The research issues we feel are the most important for creating a positive wireless future are

(1) low-cost infrastructure and services,

(2) seamless mobility,

(3) new and advanced services,

(4) usability and human-machine interface, and

(5) health and environment.

8.3 Low-Cost Infrastructure and Services

The cellular systems of today are primarily designed to provide cost-effective wide area coverage for a rather limited number of simultaneous users with moderate bandwidth demands. In order for new wireless services (e.g., multimedia services) to succeed on a large scale, they have to be widely available, simple to purchase and access, and affordable to large numbers of users. This is not only a question of high bandwidth and quality of service. The cost for the user must be comparable for services offered in the current generation of cellular systems.

One problem with conventional cellular systems (2G, 3G, etc.) is that they don't scale in bandwidth in the economic sense. A large part of the cost for building the infrastructure is related to network planning and site work. Economies of scale, and certainly Moore's law, are not applicable on site acquisition, roadworks, erecting towers, etc., in the sense that they do not follow the same price/performance evolution like electronic components. Also, the cost of the wireless networks depends rather weakly on the basic radio technology (e.g., the air interface) since current modulation and signal processing technologies are quite advanced and very close to the theoretical limits (Shannon's law) that not even a radical improvement in processing capabilities will significantly improve performance. If we maintain the area of coverage, capacity in terms of the number of users, and quality of service, the infrastructure cost will be directly proportional to the user data rate . In other words, the higher the data rate a service requires, the more it will cost to deliver. In the end this will translate into a higher price for the user, meaning that advanced multimedia services will be expensive. Today's users are accustomed to being connected anytime and anywhere (i.e. large coverage areas and high availability), so these parameters can hardly be compromised. If affordable multimedia services are to be possible, i.e. higher data rates at constant or lower cost per user, either some of the other QoS parameters have to be sacrificed or architectures with radically lower cost factors have to be developed.

There are two ways of approaching these cost limitations of broadband wireless systems. One way is to go, from the outset, for a larger number of small, cheap, and ad hoc deployed access points, similar to WLANs. That means compromising anytime/anywhere coverage or giving up on quality of service, but it will make wireless broadband communication available at least at hotspots. The second option, more suited to established cellular operators, is to slowly improve the capacity of their networks, thereby increasing the number of supported users by upgrading to more expensive, but more advanced base stations and smart antennas.

Due to the high cost of developing, building, maintaining, and operating wireless infrastructure in general and cellular infrastructure in particular, issues relating to efficient resource management will become increasingly important. By "resources" we mean spectrum, power, and available infrastructures. The large number of services and the multitude of service types expected in the future are the main challenges for resource management mechanisms. The necessary data rates, for example, may vary substantially, from perhaps 10 kbps for a certain user under certain conditions, to 100 Mbps for another user under more demanding conditions. We will here discuss a few issues that can lead to cost reductions in the future wireless systems, especially in the case of cellular operators.

Even today, techniques like RoF (radio-over-fiber) allow new network configurations and protocols, leading to cost savings compared to traditional cellular configuration if the optical fiber is already available or is cheap to deploy. High-altitude platforms (HAPs) or low earth orbit satellites (LEOs) may provide coverage and high capacity, sometimes in a cheaper manner than ground infrastructure. These techniques may well be helpful for future 3G systems in a long-term perspective. A promising development is the use of smart antennas to improve the link quality and achieve higher capacity in the networks. Multiple-input multiple-output (MIMO) channels are a good example, offering great potential, at least in some environments. The air interface protocols of the future have to provide several times better resource utilization and spectrum efficiency than today.

Better utilization of resources and lower costs can be achieved if several operators share the same infrastructure (infrastructure sharing) or the same spectrum (spectrum sharing and co-farming). In the case of infrastructure, both radio access network (RAN) sharing and transport network (TN) sharing should be considered. One example is implementing the connections between the base stations of cellular systems with the same cheap and shared Ethernet connections as for WLAN. In the future, the optical fiber infrastructure will provide immense bit capacity, and sharing it among different service carriers will be the key to low cost. In the case of spectrum, we will need efficient algorithms for avoiding collisions between transmissions from different systems coexisting in the same spectrum.

Since the demand for different services typically varies over time, the available spectrum resources can be used more efficiently by advanced re-allocation procedures (dynamic spectrum allocation). Peer-to-peer communication demands high capacity during working hours (from 9 A.M. to 5 P.M.), when spectrum could perhaps be borrowed from broadcasting systems. During evenings, when people consume more television, less capacity can be allocated for telecommunication and more for broadcasting. During late-night hours, the spectrum can be used for broadcasting or multicasting information and media content to caching-enabled terminals. The integration of personal communication and broadcasting systems is a hot issue for the future.

It is a considerable challenge to provide high capacity wherever it is needed, instantly and temporarily. Examples of this could be for an event company to provide voice, messaging services, and web surfing to a large audience at a music festival or for a news company to broadcast images of a crashed airliner in a thinly populated area. In both cases, consumers or media companies might be willing to pay for this type of temporary broadband service. Research has to be done on techniques for dynamic allocation of resources in time and space in a very flexible manner.

A consumer of wireless services typically retrieves more information from the network than he or she sends, making the traffic pattern asymmetric. This could lead to cheaper infrastructures and terminals, as well as longer usage time before recharging the terminal battery, and increased spectrum efficiency. How to make better use of this traffic asymmetry is an interesting topic.

Ad hoc deployment of the access points and minimal work on planning and maintenance are the strengths of today's WLANs. Technologies like WLAN and Bluetooth will be embedded in all future communication devices and personal computers. This will require the development of policies for routing signals through many devices organized in ad hoc networks (e.g., M2M communication and multi-hopping). Decentralized resource management will be an option when network topologies become large. In a very diverse and heterogeneous environment, complexity means that a decentralized control scheme is preferred to a centralized one. Future research should focus on decentralized resource allocation, possibly involving software agents able to trade resources in real time according to supply and demand.

The increasing demand for new wireless services will lead to the deployment of new wireless systems of different types: new cellular systems, wireless LANs, satellite communication systems, etc. In a few years there will be a need for spectrum reallocation. More available spectrum for an operator makes the network cheaper. It is very important to identify and release new frequency bands for personal communication, for licensed and unlicensed operations.

8.4Seamless Mobility

As the wireless world becomes increasingly heterogeneous, with many types of access network, a potential problem is the difficulty of controlling and managing this complex environment. When the complexity increases, it becomes more and more difficult to design centralized systems for controlling the networks, such as systems providing seamless roaming and billing solutions. Moreover, as the complexity increases, the modular design of the cellular networks becomes increasingly wasteful of bandwidth and energy. At the same time, it is very hard to predict and manage complex interactions in decentralized (distributed) systems as well, making this a difficult problem.

One reason for the importance of complexity management is that most users have come to expect a rather high quality of service, for example seamless roaming between access points within a single network and between networks operated by different operators. When the number of systems operated by different actors increases, this problem will be accentuated. This is not only a technical problem, but also an administrative and business problem. From the user's perspective, the important thing is the services offered and their quality, not how the systems look or who operates them.

If seamless services, always best connected, and infrastructure-independent services are to be possible in the future, different systems and access networks have to be integrated into a "single" network, transparent for data traffic. There are several ways to achieve this integration. One way involves standardization of interfaces at different system levels. It is therefore necessary to develop networks and terminals that can switch between different air interfaces and protocols. Multimode and adaptive radios will enable this feature by employing software implementations on a common hardware platform or system-on-a-chip technologies. Research should be carried out on how to implement low-cost and efficient multimode and adaptive radios for terminals and infrastructure.

User access to information will be provided in real time and non-real time. If there is no delay constraint, then the non-real-time choice is cheaper to deliver, but still well suited to a large number of services based on downloading and caching data for later use (multimedia messaging, news, music, movies, etc.). All this traffic can happen in off-peak hours. The use of information gathered in this fashion can indirectly influence performance in a mobile situation. Research should be carried out into the implementation of services based on collecting, memorizing, and organizing information in the terminal in a smart and personalized manner.

Due to the wide range of terminal capabilities, seamless services need to be provided in a scalable manner. The same application should be adapted for low and high data rates, for small and large screens, and for low and high price ranges. As the number of customers increases, the resources have to be scaled up to the new requirements in a linear manner. Research on scalability of services as well as techniques for supporting different QoS levels will be very important in enabling viable business models in the future wireless systems. Advanced signal processing in the areas of voice and image coding has a large potential to contribute. Multi description coding, for example, divides video data into equally important streams so that the decoding quality using any subset of them becomes acceptable. Higher quality is obtained by increasing the number of descriptions, so the probability of losing all the descriptions is small. There are already a few implementations that look like very promising ways to provide scalable services. More research is needed on this kind of signal processing.

Future consumers of wireless communication will be increasingly mobile, moving from place to place to meet friends, do business, watch movies, etc. As the requirements on bandwidth and infrastructure density grow, mobility management could become a problem. User identification, handover procedures (when moving from one cell to another or from one system to another), roaming, billing, etc., will consume an increasing amount of resources and time, contributing to the perceived delay of the services. Research is required on how to efficiently handle these aspects of mobility.

8.5 New and Advanced Services

In the future a multitude of new service types will be introduced. In several of these services the typical content will be location dependent, for example enabling services to provide us with a city map when we arrive at an airport, giving special offers when we pass a shop, or providing online games when we wait at a railway station. Some content varies from place to place or environment to environment, and the personal terminal is able to memorize and collect information while moving in the network. Future investigations are needed into developing context- and location-aware services plus investigations into defining the requirements for the terminals and the infrastructure.

Multiple-party meetings in real time are an important and demanding future service, which can be further extended to so-called telepresence. Telepresence is a virtual meeting that provides the illusion of actually being somewhere else, an illusion very close to the real thing. The bandwidth required for telepresence is expected to be about 100 Mbps (with efficient data compression techniques and fast sensory feedback). Techniques are needed to enhance telepresence over future wireless channels and sensor technology is needed to enable it. Multicasting can be used to increase spectrum efficiency when offering multiparty telepresence.

Moving computers into the network and making them invisible to the end user are important features of the future wireless world. Being always connected and having access to computational and communication resources will lead to ubiquitous and seamless services, enhanced by smart spaces for example, with a multitude of displays and other resources surrounding the user. This will require the development of suitable services with advanced user interfaces having appropriate sensors, good and cheap output mechanisms, and smart devices.

Security will be one of the most important service features in the future. Better techniques to ensure protection of data and user privacy (position, traffic patterns, private information, etc.) have to be developed. Ways to provide very secure connections and secure data transfers, at least in services like electronic payments or banking, have to be studied. Unauthorized tracking of users or access to personal and private information has to be avoided. The whole area of security, privacy, and integrity has to be approached seriously.

With the increasing use of new and demanding services (e.g., multimedia services) there will be a need for efficient and very high rate air interfaces. Air interfaces of at least 100 Mbps for wide area coverage and up to 1 Gbps for very short-range personal communication seem to be needed. These data rates are required to support services like media downloading, fast synchronization, and advanced videoconferencing. OFDM (orthogonal frequency division multiplexing) and UWB (ultra wide band) are two examples of candidate technologies.

8.6 Usability and Human-Machine Interface

Usability refers to how easy it is to handle a product or service. Ways to improve usability include shortening the time to accomplish tasks, reducing the number of mistakes made, reducing learning time, and improving people's satisfaction with a system.

The user interfaces of wireless terminals are currently based mainly on physical or virtual buttons (existing on a small touch-sensitive screen), input devices like micro joysticks or roller wheels, and even simple voice recognition. The output capabilities of wireless devices are also hampered by the small size of the devices, battery constraints, and the cost of displays and sound cards. This has important limitations for usability, especially when we consider senior citizens, disabled persons, and other people with special needs or requirements. The user interface of a physically small device has to have a small scale, at least today. Furthermore, the user interfaces of wireless devices follow proprietary models and are neither standardized nor very intuitive for the user. If the user has learned to be efficient and enjoys the interface on one device from a certain vendor, they will be very frustrated when having to learn the interface on a device from another manufacturer.

Research needs to focus on ways to handle these problems, for instance through splitting the single wireless device into simpler information appliances, each taking care of a specific task, thus providing better opportunities for user interfaces that are easy to understand and use. If the radio part of a mobile phone were embedded in a device of its own, no direct manipulation would be needed. This micro base station could be placed anywhere on a user's body or in their vicinity. A specialized device for user input could be connected. This device could have a larger display than battery constraints would allow on an integrated device. In addition, control through new modalities, such as voice, could be used to control this personal communication system, thus completely removing the need for buttons on a device or virtual buttons on a display. Augmented reality using goggles and fold-out screens are promising topics that deserve investigation.

For users with special needs, such as people with failing eyesight, hand tremors, or rheumatism (all of which make it hard to manipulate small objects), certain parts of a personal network could be exchanged for specialized devices better suited to these users' needs. Networks like these could also interact with fixed devices surrounding the user. A user could seamlessly move information from their personal display to a larger display with high resolution somewhere in their vicinity.

Research should therefore be focused on developing a more user- and human-centered system of appliances where the technology of the computer disappears behind the scenes into task-specific devices (Norman 1998), perhaps integrated into everyday things. But fragmentation of the user interface technology does not solve the whole problem. Improved design is still required to ensure uniform and intuitive interfaces.

8.7 Health and Environment

Use of communication systems, especially wireless systems, raises several important issues of health and environmental impact.

The effect of electromagnetic radiation on the human body is an area of crucial importance for the whole wireless community. As yet, no generally accepted scientific research has proved that use of wireless terminals is dangerous, at least not with the radiation levels allowed today. However, there is a need for more research in this area, especially studies to examine the long-term effects of radiation on the human body. Current standards and limitations on mobile phone transmitted power are based only on the thermal effects of electromagnetic radiation. There are some indications that there might also be radiation effects on DNA structure and cell reproduction. The protection standards and maximum limits will probably have to be modified according to all possible negative effects of electromagnetic radiation.

The impact of communication infrastructure on the environment is another important issue. One important area is the power consumption in the systems to drive computers, servers, base stations, etc. The environmental impact of producing the required electric power is not negligible. Towers, underground cables, high-power transmitters, and access roads to base station sites can also have a negative impact on the environment and on people's esthetic values. The likely development is towards an increased density of ground infrastructure for communication, which means future systems will have to be integrated into the environment in a sustainable way. Another way to tackle this issue is to move the infrastructure into the sky by using HAPs (high-altitude platforms) or different types of satellite.

Increased use of wireless services also has a big potential to affect the environment in a positive way. New and advanced services might decrease the growth in travel. The transportation sector might benefit by being able to rationalize the distribution channels (fleet management), saving fuel and decreasing congestion. There are several other benefits too.

In conclusion the impact on health and environment is a cross-disciplinary area that needs to be investigated by examining radio communications technology, medicine, biology, environmental engineering, and business.

8.8 A Need for Cross-Disciplinary Research

One important conclusion from this work is that the academic research tradition with well-established and very specialized areas of research will not suffice in the future. The need for parallel cross-disciplinary research is becoming increasingly important.

The reason is that research needs to widen its focus beyond creating new technology and knowledge within specific areas, otherwise we run the risk of developing new technologies with little or no connection to the needs of users or the industry. Problems outside the computer simulation program or the laboratory very rarely lend themselves to formulation as well-defined entities in the same way that academic research groups are organized. The world is cross-disciplinary. It is crucial that academic research also takes real-life problems as a starting point, not only problems from the academic tradition. Instead of focusing on developing a new high-bandwidth air interface, for example, research can begin by asking whether this has the potential to solve a problem for users or the industry. What if increased bandwidth is not a solution to a relevant problem? It might be that wireless systems are too expensive and research should focus on developing extremely cheap base stations which can be deployed very easily. Both research approaches have their merits and both are needed for us to create a good future.

8.9 Issues in Implementing M-Commerce and Mobile GIS

The mobile Internet is ideal for particular applications and has useful characteristics that offer a range of services and content. As the connection charge and the handset are still costly, this caters for a niche market of mobile users who are young and affluent and mobile business users. For example, Tesco's mobile service is targeted toward business executives. It offers convenience for those who use trains or public transport to get home. They can place orders on their way home. When they get off the train, they just go to the nearby Tesco store to pick up the order; this saves a lot of shopping time. In the United Kingdom, the HSBC bank notifies customers about their current balance by SMS messaging. It is a free service and is welcomed by many mobile users. Utility services such as British Gas use mobiles extensively. The off-site engineers access back-office systems to order parts, download schematics for fixing appliances, and so on, through handheld devices. So when implementing m-commerce, businesses should consider the following issues:

- Understand the position of the business in the industry.
- Evaluate the overall usefulness and convenience to the customers—both internal, e.g., off-site employees, and external.
- Develop applications with a fun and hassle-free user interface.
- Offer fast and secure ordering facilities.
- Allow users to recover from an aborted session.

The advent of m-commerce is fast approaching. In business, the constant word is "change." Winning customers in today's highly competitive, demanding world is the key to survival. The mobile world is changing the logic of business, businesses have to implement effective strategies to capture and retain increasingly demanding and sophisticated customers. Furthermore, it is timely for the business community to prepare for the approaching economic boom, which always follows an economic downturn. Business needs to think critically about how to integrate the mobile Web to the wired Web, which requires careful strategic thought, as not all applications are appropriate. Those that make the correct decisions will control the future.

Chapter 9: Results and Conclusion

9.1 Experiment and Results

Collaborative GIS is an integrated GIS framework that can facilitate its participants to work on geospatial related tasks with a shared understanding of geospatial information. The development of collaborative GIS is related to collaborative spatial decision making (CSDM), computer-supported cooperative work (CSCW), and group decision support systems (GDSS). The term "collaborative" indicates a higher level of operation agreement among the participants comparing to other terms such as "communication," "cooperation," and "coordination." Collaborative GIS, or spatial understanding (and decision) support systems can be used to "facilitate geographical problem understanding and decision making for groups, including groups embroiled in locational conflict.

One major research gap in collaborative GIS is the connection between system design and actual GIS applications and usage. Most collaborative GIS projects emphasize the location/relocation problem tasks, rather than the spatial dispatch coordination or emergency response plan. With the availability of the Internet, many collaborative GIS started to adopt the Web-based environment for their development frameworks. However, very few collaborative GIS extend the system framework from Web-based GIS to wireless GPS-enabled devices. This thesis will try to fulfill these gaps to combine both the application-oriented system design strategy and the integration of Web-based GIS and wireless mobile GIS devices. The research highlight recent developments in Internet GIS, mobile GIS, and wireless networks used to establish collaborative GIS architectures.

The driving force behind this GIS architecture transformation is the availability of new technology in network communications and programming. New languages such as Java, Python, and C# (C-sharp) support platform-independent applications . Advanced network technologies such as Microsoft .NET framework, J2EE platform, and Common Object Request Broker Architecture (CORBA) provide a comprehensive scheme for distributed component technology essential to the development of Mobile GIS. Distributed component technology allows clients to access heterogeneous servers dynamically, which is an essential feature of distributed geographic information services (GIServices). In the future, traditional geographic information systems (GISystems) designed as isolated islands, will become increasingly less attractive, possibly disappearing altogether. The cost efficiencies and flexibility of reusable and interoperable open and distributed-services interfaces provide much greater economies. GIServices focus on open, distributed, task-centered services that broaden geographic information uses into an increasingly wider range of online geospatial applications These include digital libraries, digital governments, online mapping, data clearinghouses, real-time spatial decision support tools, distance learning modules, and so on.

9.2 Integration of Mobile GIS and Mobile Commerce

The first issue of the linkage between mobile GIS and mobile commerce is the communication type. There are two types of communication: real-time synchronized communication and cable-based synchronous. Both mechanisms can provide two-way communication between Mobile GIS and mobile Commerce. For cable-based connections, the Mobile GIS servers can send geodata to the mobile Commerce receivers whenever the connection is available. Then the mobile Commerce receiver can query geodata or new information back to the Mobile GIS server later on. For real-time wireless communication, the mobile Commerce receivers can create a continuous link to the Mobile GIS server, and the server responds to the request in real time by sending the new spatial query to the receiver. To facilitate two-way communications, several middleware or data synchronization software packages (such as Microsoft ActiveSync or Web Services) are required for bridging the mobile Commerce applications and Mobile GIS. For collaborative GIS, Internet-based protocols, such as TCP/IP and HTTP, can provide an open and standardized communication channels for the linkage between Mobile GIS and mobile Commerce applications.

The second issue of the linkage is the data exchange methods between Mobile GIS and mobile Commerce application. There are two major methods: data transmission and database connection. The first method, data transmission, is to utilize network data transfer tools like HTTP to upload or download the whole data items. The second method is to create a database connection between the Mobile GIS and mobile Commerce application that can provide more advanced database query and search functions. Most collaborative GIS will prefer the second method because the first method (data transmission) requires a higher network bandwidth when the size of GIS data becomes large. The format of data/databases is another consideration for data exchange methods. For example, mobile Application units can send text-based GPS signals to the Mobile GIS via a database connection. An alternative method is that a segment of GPS signals can be converted into a compressed binary file, then use http to send the file back to the Mobile GIS server. Both approaches can fulfill the need of data exchanges between Mobile GIS and mobile Commerce Application.

9.3 Conclusion

Mobile Internet offers very promising possibilities for a broad range of business sectors. It can be seen as complementary with other upcoming trends and lifestyles and hence the increasing demands for communication that is multifunctional and location independent.

When the Internet hype reached its peak, mobile commerce was postulated to be the next logical step. Mobile commerce was expected to provide both the freedom of information brought along with the Internet and the freedom of mobility introduced by the cellular networks. The combination of these technologies was expected to be a true break-through for both Internet and communication technology. The trust in this promising vision and its related profit prospects were good enough to justify nearly every investment.

In order to achieve the trade-off between functionality on the Mobile GIS and Moblie Commerce application simplicity, we have proposed the use of the ad hoc XML-based GIS format. The XML technology provides interpretability, flexibility with content manipulation, and expandability for particular problems. The format encodes basic information about street network with the possibility to perform search queries, and descriptions of city objects. The logical structure allows us to easily compose requested information on a server side. The server process is performed through J2EE Application server. The client software can represent contents in more appropriate way on a particular device. In this work J2ME is used to simulate client. In this paper, based on J2EE, an open distributed architecture is proposed for Mobile Commerce and Mobile GIS integration. And a prototype system is designed and developed based on the architecture which is composed of four logic tiers. The prototype system can successfully work on J2ME supported phones, which proves that the architecture is feasible.

In this paper, we briefly reviewed some of the most important technologies for Mobile E-Commerce and GIS and deployment of integrated system. We discussed the emerging telecommunication platforms and positioning systems along with issues related to contemporary mobile computing. We elaborated on this issue of integrated system by presenting the requirements posed by users for this evolving family of modern mobile services. Lastly, we presented a built pilot system, which exemplifies how such requirements are reflected in the integrated implementation. The integrated system is surely an area of modern mobile services where considerable growth is observed. The developments in the Internet domain, wireless/mobile networking as well as the proliferation of positioning technologies expedited such evolution. The impact on nomadic users is tremendous. It is evident that such progress needs to be addressed in a methodological manner and supported, where appropriate by coordinated standardization efforts. Requirements need to be reviewed and studied very carefully by all the involved actors. Their coverage – fulfillment by modern technological platforms is also a very important issue. Our work is mainly moving along these lines. Our analysis, as presented in this paper, shows that the technologies and issues involved in integrated system deployment and provision cover a very wide spectrum including operating system capabilities, user interface design, positioning techniques, terminal technologies, network capabilities, etc. The meticulous mapping of these technical aspects to the identified requirements is a critical success factor for the integrated system.

The future of Mobile commerce in integration with Mobile GIS is bright and brimming with opportunities. This statement has probably never before been as applicable as present. Wireless Technology have enabled GIS technology to open new paths for disseminating, sharing, displaying and processing spatial information on the mobile. Currently, one of the biggest limiting factors is man's imagination itself. Although the technologies will keep on developing, we are left with the challenge of trying to use these, in as many scenarios as possible.

This study introduces a prototype that demonstrates an interoperable framework for combining Web-based GIS technologies, wireless mobile GIS applications and Mobile Commerce Application. The integrated framework provides real-time or near real-time GIS data update functions (such as adding new spatially located map features or GPS tracking locations) between mobile GIS devices and Internet GIS servers. Although these real-time GIS functions can be very important during time-urgent emergencies, they can be equally beneficial and highly cost effective during routine field activities.

References

[1] **ZHE LIU**, *A Java-Based Wireless Framework for Location-Based Services Applications*, June 2002. http://www.geomatics.ucalgary.ca/links/GradTheses.html

[2] **JOHN E. HARMON AND STEVEN J. ANDERSON**. *The Design and Implementation of Geographic Information Systems, 2003 , John Wiley & Sons.*

[3] **WANG FANGXIONG A, JIANG ZHIYONG**, *Research On A Distributed Architecture Of Mobile Gis Based On Wap.*

[4] **Wireless Developer Network**, http://www.wirelessdevnet.com

[5] **MICHAEL JUNTAO YUAN**, *Enterprise J2ME: Developing Mobile Java Applications, Prentice Hall PTR , October 23, 2003.*

[6] **REENA SHUKLA**, *LBS, the ingredients and the alternatives, GIS Development,* http://www.gisdevelopement.net

[7] **JACEY-LYNN MINOI, DR. PETER GREEN, SYLVESTER ARNAB**, *Navigation Application with Mobile Telephony: Shortest-path, GIS Development,* http://www.gisdevelopement.net

[8] **ANAND RAMAMOORTHY**, *New approaches in GPS based location systems*

[9]. **JIANG BIN, ZIPF ALEXANDER**. *An Introduction to the Special Issue on LBS and GIS, Vol. 10, No. 2, December 2004.*

[10] **P. M. UDANI, R. K. GOEL**, *GPS enabled mobile GIS services*

[11] **PLEWE, B**., *GIS Online: Information, Mapping, and the Internet. Onward Press, 1997*

[12]**Narikka J.,**Spatial *Data and Application in a mobile client-server environment, technical report, 1999*

[13]*The Demand for Mobile Value-Added Services, market study, 1999,* http://www.nokia.com

[14]**David Caduff,** *Sketch-Based Queries in Mobile GIS-Environments,2002*

[15]**G. Artem,** *Management of Geographic Information in mobile Environments, 2000.MS, University of Jyvaskyla, Finland.*

[16] **Andrew Harrington,** *Mobile GIS: using your enterprise GIS in the field,2004.*

[17]*The Mobile Information Device Profile (MIDP),* http://java.sun.com/products/midp

[18] *GIS Development*, http://www.gisdevelopment.net

[19] *The J2ME Web Services API.* http://www.jcp.org/en/jsr/detail?id=172

[20] **Mehdi Khosrow-Pour** , *Encyclopedia of E-Commerce, E-Government, and Mobile Commerce, Volume I, A-J, 2006*

[21] **Mehdi Khosrow-Pour** *, Encyclopedia of E-Commerce, E-Government, and Mobile Commerce, Volume II, K-Z, 2006*

[22] **Norman Sadeh,** *M-Commerce: Technologies, Services, and Business Models,2002*

[23] **Dr. Zhong-Ren Peng and Dr. Ming-Hsiang Tsou**, *Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks 2005*

[24] **Ee-Peng Lim and Keng Siau***, Advances in Mobile Commerce Technologies, 2003*

[25] **Nansi Shi**, *Wireless Communications and Mobile Commerce ,2004*

Appendix

Listing 6.1 Graph represented as XML, cities.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<FindShortestPath xmlns="urn:shortest-path">
   <StartCity>Chabil</StartCity>
   <TargetCity>Kritipur</TargetCity>
   <Paths>
      <Path from="Boudha" to="Chabil" cost="1"/>
      <Path from="Boudha" to="Kathmandu" cost="4"/>
      <Path from="Boudha" to="Sundhara" cost="7"/>
      <Path from="Chabil" to="Sundhara" cost="4"/>
      <Path from="Chabil" to="Kathmandu" cost="2"/>
      <Path from="Chabil" to="Patan" cost="5"/>
      <Path from="Kathmandu" to="Patan" cost="2"/>
      <Path from="Patan" to="Kritipur" cost="5"/>
      <Path from="Sundhara" to="Kritipur" cost="8"/>
    </Paths>
</FindShortestPath>
```

Listing 6.2 Vertex class, Vertex.java

```
    class Vertex  {
     private String label;
     private List outgoingEdges;
       private int bestScore;
       private Edge bestPath;
```

```
        private static Hashtable vertexList = new
Hashtable();

        static Vertex register(String label) {
           Vertex v = (Vertex)vertexList.get(label);
           if (v==null) {
              v = new Vertex(label);
              vertexList.put(label,v);
           }
           return v;
        }

        Vertex(String label) {
           this.label = label;
           this.outgoingEdges = new LinkedList();
           this.bestScore = Integer.MAX_VALUE;
           this.bestPath = null;
        }
        void addEdge(Edge e) {
           this.outgoingEdges.add(e);
        } /* end excerpt1 */
        ...
    }
```

Listing 6.3 Edge class, Edge.java

```
class Edge {
   private Vertex fromVertex;
```

```
   private Vertex toVertex;
   private int cost;

   Edge(Vertex from, Vertex to, int c) {
      this.fromVertex = from;
      this.toVertex = to;
      this.cost = c;
      this.fromVertex.addEdge(this);
   }
   ...
}
```

Listing 6.4 ShortestPath class, ShortestPath.java

```
    package myGraph.graph;
    import java.io.IOException;
    import org.xml.sax.SAXException;
    import org.xml.sax.Attributes;
    import org.xml.sax.helpers.DefaultHandler;
    import javax.xml.parsers.SAXParser;
    import javax.xml.parsers.SAXParserFactory;
    import java.util.*;
    public class ShortestPath extends DefaultHandler {
       static final String SHORTEST_PATH_URI =
"urn:shortest-path";
       StringBuffer buffer = null;
       static Vertex startCity;
       static Vertex targetCity;
```

```
        public void startElement(String uri, String local,
String qname,
        Attributes atts) throws SAXException {
           if (!uri.equals(SHORTEST_PATH_URI)) return;
           if ( local.equals("StartCity") ||
local.equals("TargetCity")) {
              this.buffer = new StringBuffer();
           } else if (local.equals("Path")) {

            String fromString = atts.getValue("from");
           if (fromString==null) throw new
SAXException("Attribute
              'from' missing");
              Vertex from = Vertex.register(fromString);
              String toString = atts.getValue("to");
             if (toString==null) throw new
SAXException("Attribute 'to'
              missing");
            Vertex to = Vertex.register(toString);
              String costString = atts.getValue("cost");
              if (costString==null) throw new
SAXException("Attribute
              'cost' missing");
             new
Edge(from,to,Integer.parseInt(costString));
           }
        }
```

```
        public void endElement(String uri, String local,
String qname)
        throws SAXException {
           if (!uri.equals(SHORTEST_PATH_URI)) return;
           if (local.equals("StartCity")) {
           this.startCity = Vertex.register(new
String(this.buffer));
              buffer = null;
           } else if (local.equals("TargetCity")) {
           this.targetCity = Vertex.register(new
String(this.buffer));
              buffer = null;
           }
        }
        public void characters(char[] ch, int start, int
length) throws
        SAXException {
           if (this.buffer != null) {
              this.buffer.append(ch, start, length);
           }
        }
        /* end excerpt1 */
```

Listing 6.5 main() method for ShortestPath class, ShortestPath.java

```
 public static void main(String[] argv) throws Exception
{
           if (argv.length < 1) {
              System.err.println("Usage: javaShortestPath
```

```
            file");
            System.exit(1);
        }

        SAXParserFactory factory =
SAXParserFactory.newInstance();
        factory.setNamespaceAware(true);
        SAXParser parser = factory.newSAXParser();
        ShortestPath handler = new ShortestPath();

        parser.parse(argv[0],handler);

        SortedSet open = new TreeSet(new
VertexComparator());
        startCity.update(0,null,open);

        while (!open.isEmpty()) {
         Vertex v = (Vertex) open.first();
          open.remove(v);
          for (Iterator i = v.getEdges(); i.hasNext();
) {
                Edge e = (Edge)i.next();

e.getToVertex().update(v.getBestScore()+e.getCost(),e,
                open);
            }
      }
```

```
            Vertex.showAll();
        }
```

Listing 6.6. The PizzaBundle class

```
public class PizzaBundle implements
  BundleActivator, ServiceTrackerCustomizer {
  /** BundleContext for this bundle */
  protected BundleContext context;
 /** Log Service wrapper object */
  protected LogTracker log;
  /** Http Service tracker object */
  protected ServiceTracker tracker;
  /** HttpContext for HTTP registrations */
  protected HttpContext httpContext;
 // ... ...
  public PizzaBundle() { }
  // Methods in BundleActivator
  public void start(BundleContext context) throws
Exception {
    this.context = context;
    httpContext = new HttpContext() { ... ... };
    log = new LogTracker(context, System.err);
    tracker = new ServiceTracker(context,
        HttpService.class.getName(), this);
    tracker.open();
  }
  public void stop(BundleContext context) throws
Exception {
```

```
    tracker.close();
    log.close();
  }
  // Methods for ServiceTrackerCustomizer
  public Object addingService(ServiceReference reference)
{
    HttpService http =
(HttpService)context.getService(reference);
    if (http != null) {
      try {
        http.registerServlet(servletURI,
            new Pizza(), null, httpContext);
        http.registerResources(servletURI+imagesURI,
                               imagesURI, httpContext);
        log.log(log.LOG_INFO, "Pizza Servlet
registered");
      } catch (Exception e) {
        // handle the exception
      }
    }
    return http;
  }
  public void modifiedService(ServiceReference
                    reference, Object service) {
  }
  public void removedService(ServiceReference
                    reference, Object service) {
```

```
    HttpService http = (HttpService) service;
    http.unregister(servletURI);
    http.unregister(servletURI+imagesURI);
    context.ungetService(reference);
    log.log(log.LOG_INFO, "Pizza Servlet unregistered");
  }
}
```

Listing 6.7. The Pizza servlet

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
public class Pizza extends HttpServlet {
  protected void doGet(HttpServletRequest req,
                       HttpServletResponse res)
            throws ServletException, IOException {
    // Generate some output
    res.setContentType("text/html;" + "charset=iso-8859-
1");
    PrintWriter out = res.getWriter();
    out.print(" ... ... ");
    // Get query parameters
    String queryString = req.getQueryString();
    // any pizza order logic
    // ... ...
    out.println("</body></html>");
```

```
  }
}
```

Listing 6.8. The LogTracker class

```
public class LogTracker
      extends ServiceTracker implements LogService {
  protected final static String clazz =
         "org.osgi.service.log.LogService";
  protected PrintStream out;
  public LogTracker(BundleContext context, PrintStream
out) {
    super(context, clazz, null);
    this.out = out;
    open();
  }
  // Implements various log() messages with
  // different signatures.
  // ... ...
  public synchronized void log(ServiceReference
reference,
       int level, String message, Throwable exception) {
    ServiceReference[] references =
getServiceReferences();
    if (references != null) {
      int size = references.length;
      for (int i = 0; i < size; i++) {
```

```
        LogService service = (LogService)
getService(references[i]);
        if (service != null) {
        try {
          service.log(reference, level, message,
exception);
        } catch (Exception e) { }
      }
    }
    return;
  }
  noLogService(level, message, exception, reference);
}
protected void noLogService(int level, String message,
   Throwable throwable, ServiceReference reference) {
  if (out != null) {
    synchronized (out) {
      switch (level) {
        case LOG_DEBUG: {
          out.print("Debug: ");
          break;
        }
        case LOG_INFO: {
          out.print("Info: ");
          break;
        }
        case LOG_WARNING: {
```

```
}

          out.print("Warning: ");
          break;
        }
        case LOG_ERROR: {
          out.print("Error: ");
          break;
        }

        default: {
          out.print("Unknown Log level[");
          out.print(level);
          out.print("]: ");
          break;
        }
      }
      out.println(message);
      if (reference != null) {
        out.println(reference);
      }
      if (throwable != null) {
        throwable.printStackTrace(out);
      }
      }
    }
  }
}
```

Listing 6.9. The Axis facade MPClient class

```
public class MPClient {
  // Authentication credentials obtained from
  // MapPoint Web site
  private static String userName = "userid";
  private static String password = "password";
  // other variables
  // The cached route segments
  private Segment[] segments;
  // Get the driving directions between
  // two human-ready addresses. This method
  // also caches the route in the segments array
  //
  // We have to run this method before we can
  // retrieve maps for the entire route or for
  // each route segment.
  public String [] getDirections (
   String fromStreet, String fromCity,
   String fromState, String fromZip,
   String toStreet, String toCity,
   String toState, String toZip
                    ) throws Exception {
    // method body
  }
  // Return the number of segments of the current
```

```
  // cached route. The number is available after
  // we call the getDirections() method.
  public int getSegmentNum () throws Exception {
    // Method body
  }
  // Get a map from the current cached route.
  // The return value is a byte array for
  // the GIF image.
  //
  // index == 0 for the overview map
  // index <= segmentNum for a segment map
  public byte [] getMap (int index,
          int width, int height) throws Exception {
    // Method body
  }
}
```

Listing 6.10. The CDC/PP driving directions client

```
public class AWTMap extends Frame
    implements WindowListener, ActionListener {
  private String endPointURL;
  private TextField fromStreet;
  private TextField fromCity;
  private TextField fromState;
  private TextField fromZip;
  private TextField toStreet;
```

```
private TextField toCity;
private TextField toState;
private TextField toZip;
private java.awt.List directionsList;
private ClassMap cm;
private Marshal md;
// Other variables
// Lists the driving directions
private void listScreen (boolean newSearch) {
  try {
    if (newSearch) {
      SoapObject method =
          new SoapObject("", "getDirections");
      // Use the SE version for standard JDK
      // Http methods
      HttpTransportSE rpc =
        new HttpTransportSE(endPointURL, "\"\"");
      rpc.setClassMap(cm);
      method.addProperty("in0", fromStreet.getText());
      method.addProperty("in1", fromCity.getText());
      method.addProperty("in2", fromState.getText());
      method.addProperty("in3", fromZip.getText());
      method.addProperty("in4", toStreet.getText());
      method.addProperty("in5", toCity.getText());
      method.addProperty("in6", toState.getText());
      method.addProperty("in7", toZip.getText());
```

```
      Vector v = (Vector) rpc.call (method);
      directionsList = new java.awt.List(10, false);
      directionsList.add("Overview Map");
      for (int i = 0; i < v.size(); i++) {
        directionsList.add((String) v.elementAt(i));
      }
      directionsList.setSize(200, 200);
    }
    Panel top = new Panel ();
    top.setLayout(new FlowLayout(FlowLayout.LEFT));
    top.add(directionsList);
    Panel bottom = new Panel ();
    bottom.setLayout(new FlowLayout(FlowLayout.LEFT));
    bottom.add(startOver);
    bottom.add(showMap);
    scroll.remove(content);
    content = new Panel ();
    content.setLayout(new BorderLayout());
    content.add(top, BorderLayout.CENTER);
    content.add(bottom, BorderLayout.SOUTH);
    scroll.add(content);
    setVisible(true);
  } catch (Exception e) {
    e.printStackTrace();
  }
}
```

```
private void mapScreen (int i) {
  try {
    ImageItem img;
    byte [] imgarray;
    SoapObject method = new SoapObject("", "getMap");
    HttpTransportSE rpc =
      new HttpTransportSE(endPointURL, "\"\"");
    rpc.setClassMap(cm);
    method.addProperty("in0", new Integer(i));
    method.addProperty("in1", new Integer(200));
    method.addProperty("in2", new Integer(200));
    imgarray = (byte []) rpc.call (method);
    img = new ImageItem(imgarray, 200, 200);
    Panel top = new Panel ();
    top.add(img);
    Panel bottom = new Panel ();
    bottom.add(startOver);
    bottom.add(showDirections);
    scroll.remove(content);
    content = new Panel ();
    content.setLayout(new BorderLayout());
    content.add(top, BorderLayout.CENTER);
    content.add(bottom, BorderLayout.SOUTH);
    scroll.add(content);
    setVisible(true);
  } catch (Exception e) {
```

```
      e.printStackTrace();
    }
  }
  // Other UI and event handling methods
}
```

Listing 6.11. A simple J2ME Location API application

```
Criteria criteria = new Criteria ();
criteria.setPreferredResponseTime(20);
criteria.seVerticalAccuracy(10);
// set other criteria
LocationProvider provider =
    LocationProvider.getInstance(criteria);
// The StepTracker's locationUpdated() method
// will be called every 1 second until the 100th second.
// The passed location data cannot be more than 2 seconds
old.
StepTracker tracker = new StepTracker ();
provider.setLocationListener(tracker, 1, 100, 2);
// Add the collision handling logic. The
CollisionHandler's
// proximityEvent() method is called when the device
// enters the 0.5 meter radius of either coord1 or coord2
Coordinates coord1 = new Coordinates(lat1, long1, alt1);
Coordinates coord2 = new Coordinates(lat2, long2, alt2);
CollisionHandler collision = new CollisionHandler ();
provider.addProximityListener(collision, coord1, 0.5);
```

```
provider.addProximityListener(collision, coord2, 0.5);
public StepTracker implements LocationListener {
  public StepTracker () {
  }
  // Both threads below must return immediately.
  // So, put long processes in a separate thread.
  public void locationUpdated (LocationProvider provider,
                               Location location) {
    // Do something with the new location
    // For example, update the steps on a map.
  }
  public void providerStateChanged(LocationProvider
provider,
                                   int newState) {
    // Handle the state change. For example, if the
    // provider becomes unavailable, alert the user.
  }
}
public CollisionHandler implements ProximityListener {
  public CollisionHandler () {
  }
  public void proximityEvent(Coordinates coordinates,
                             Location location) {
    // Handle the collision here.
    // For example, alert the user and provide a
direction
    // to move away from the collision point.
```

```
  }
  public void monitoringStateChanged(boolean isActive) {

    // Handle the state change in the provider
  }
}
```