Garmash Artem

# MANAGEMENT OF GEOGRAPHIC INFORMATION IN MOBILE ENVIRONMENT

Pro graduate thesis in
Information Systems
19.09.00

University of Jyväskylä
Department of Computer Science and Information Systems
Finland

Department of Computer Science and Information Systems
University of Jyväskylä

Author: Artem Garmash
Adviser: Jari Veijalainen

Topic: Management of geographic information in mobile environment
Discipline: Information Systems
Date of completion: 19.09.00
Number of pages: text pages 97, references 7

KEYWORDS: Mobile computing, mobile GIS, XML format, WAP

Abstract:

Putting the Internet in the palm of a hand is the next technology step and it can change the style of our life dramatically. We will get a universal access with personal devices to information anytime, anywhere, and anyhow. New emerging telecommunication technologies offer an opportunity for providing new generation services and applications for mobile users. Even the term "mobile commerce" (M-commerce) has appeared: it is a subset of E-commerce that is conducted via a mobile telecommunications network. Wireless data networks represent a new application environment with strong restrictions because of limitations of mobile devices and wireless communication environment. WAP(Wireless Application Protocol) Forum defines standard architecture, programming model, and a set of protocols intended to implement wireless Internet access.

One of the service types is tied to mobile location technology. The ability to locate the position of a mobile device is the key to providing geographically specific information. By combining positioning mechanisms with location information, we can offer truly customised personal communication services through the mobile phone or other mobile devices. Analogues of similar application are navigation systems for vehicles, advanced GPS receivers which are able to display geographical maps. The range of functions that could provide geographical information service is rarely wide, from current location presentation on the map to dynamic tracking of other mobile users and performing search operations to find the shortest path or the nearest object.

In this thesis, I examine information services for mobile users, which allow city navigation and provide related additional information. In this case, one of the problems is geographical data transfer through a wireless network. Applications require careful development of the general architecture as well as a data format for information transfer to make user interaction more satisfactory and to optimise wireless network traffic. This is a serious problem, especially, for geographical data. Map images and descriptions of points of interest should be shown to the user. In some cases, we need to update a map image many times when the user is moving.

I examine the use of different format types for representation of geographical information in mobile information services: bitmap, vector, and GIS formats. Based on the results of the analysis, I describe the new XML-based geographical data model language designed for the use in the mobile environment.

**KEYWORDS:** *Mobile computing, mobile GIS, XML format, WAP*

TABLE OF CONTENTS

# 1 INTRODUCTION

In the present time, the most rapidly growing direction of information technology industry is mobile Internet. On the one hand, Internet technologies have become widely used and penetrate almost every possible area of our life. There are a lot of services for any requirements of common users. You can communicate with your friends and colleagues, videoconferencing with them even, receive latest news every hour, manage your bank accounts, do shopping, and so on. And the range of suggested services is extended all the time.

On the other hand, the number of wireless subscribers is increasing constantly. According to recent research, the total number of mobile phone users worldwide is over 300 millions, double the number of Internet users. In addition, it is estimated that in 2005 there will be about 1 billion mobile phone subscribers [WAPP99]. Information services provided by network operators are becoming more and more advanced.

Putting the Internet in the palm of a hand is the next big technology step and it can change the style of our life dramatically [Wies91]. We will get a universal access with personal devices to information anytime, anywhere, and anyhow. WAP (Wireless Application Protocol), smart phones, wireless network protocols and other new technologies offer an opportunity of providing new generation services and applications for mobile users. People will have uniform access to information whether they use the computer or mobile phone. Even the term "mobile commerce" (M-commerce) has appeared signify a subset of E-commerce that is conducted via a mobile telecommunications network [Mob99].

However, wireless data networks represent a new application environment with strong restrictions because of limitations of mobile devices and wireless communication networks. It requires additional efforts for information service developers to design the systems. Wireless Application Protocol Forum Ltd. defines standard architecture, programming model, and a set of protocols intended to implement wireless Internet access [WAP99]. WAP helps significantly by providing unified application environment but does not solve all the problems.

One of the service types is tied to mobile location technology. The ability to locate the position of a mobile device is the key to providing geographically specific information. Weiser wrote in his paper "The Computer for the 21st Century" in 1991, that the combination of networking and mobility will engender new applications and services, among them navigation software to guide users in unfamiliar places and on tours [Wies91]. As was noted by Katz in [Kat94], the situation-aware and location-aware nature of mobile systems makes possible new kinds of location-dependent information services. By combining positional mechanisms with location information, we can offer truly customised personal communication services through the mobile phone or other mobile devices [SveG99]. And market research shows that about 70% of participants report a need for city navigation services [Nok99]. Many forecasts point to the emergence of personal navigation services as the next big development in mobile telephony. Various consulting agencies have estimated the personal navigation services market to reach somewhere between 3 and 8 billion Euros by 2005 [Mob99, Nok99].

Such applications would include navigation systems for vehicles, advanced GPS (Geographical Positioning System) receivers which are able to display geographical maps. The range of functions that location-aware applications could provide is rarely wide, from current location presentation on the map to dynamic tracking of other mobile users and performing analysis operations to find the shortest path or nearest object. Applications using mobile location service technologies include fleet management, vehicle tracking for security, tracking for recovery in the event of theft, telemetry, emergency services, location identification, navigation, location based information services and location based advertising [Mob99].

In this work, I examine some topics related to the systems for city navigation and the related information service. Such systems provide geographic information, usually in the form of a map, along with respective information services for mobile users. You can view the map and perform operations like zooming. "Zooming in" means changing of map scale in order to see more details or "zooming out" to see the map outline. "Panning" is scrolling of the map on a screen in order to look at hidden parts of a map, which could not fit on a screen.

The services include providing information about points of interest, a range of search functions, which can use data obtained from location services, and other similar services. By the term "point of interest" we mean city objects which can be interesting for people; for instance: restaurants, hotels, airports, gas stations, theatres, schools, railway stations, and so on. The location service is used in order to automatically determine the current location of the user or locations of other users. Search functions can find the route to any point of interest or to other mobile user. Along with geographical information, such a system could make use of additional information about points of interests. It opens up opportunities to make location queries. For example, if we need to find the cheapest restaurants near the user, information about prices of restaurants should be used along with geographical coordinates of restaurants. Below, we give some examples of similar systems.

One of the first navigation systems implemented in a mobile phone was the Benefon Esc!, which combines GPS locationing and mobile maps with GSM (Global System for Mobile communications) phone. Benefon, a Finnish mobile phone manufacturer, first displayed it at Telecom 99. Along with buying a Benefon Esc!, users can subscribe to the Arbonaut Mobile Map Service by Arboreal Ltd. Arbonaut users will be able to download maps directly to their mobile phones, and have their own and their friends' locations displayed on the map on their phone. In addition to delivering maps onto a smart phone, Arbonaut is a Web service that manages all the geographic information to be sent to, or retrieved from, the mobile phone. Arbonaut features technology with which to handle the dialogue between the Web site and the mobile phone, using the WAP protocol or a cable connection, whichever is more convenient. Later on, Arbonaut will be available for WAP devices in general, and as a Java applet for Epoc smart phones [Ben99a]. Benefon Esc! has a twelve-channel GPS receiver and an integral high-gain flip antenna for positioning. The product enables graphical and numeric display of navigational information like location, speed, direction and world time. Additional functionality includes waypoints, routes, and trip odometer. Of course, it also includes functions of a personal organiser and communication centre [Ben99b]. Benefon Esc! only concentrates on navigation and there are no additional information services related to the point of interest.

In contrast, Citykey, a mobile platform for city information and services, emphasize the information service. It provides access to the rich description of points of interest, search among them and streets by many attributes, map viewing, information on public transport, a calendar of events and so on.

Figure 1.1 shows the screenshot of the Citikey system. But you need to download all this information on your device in order to use it: the current service does not allow you to choose the necessary part of data only.

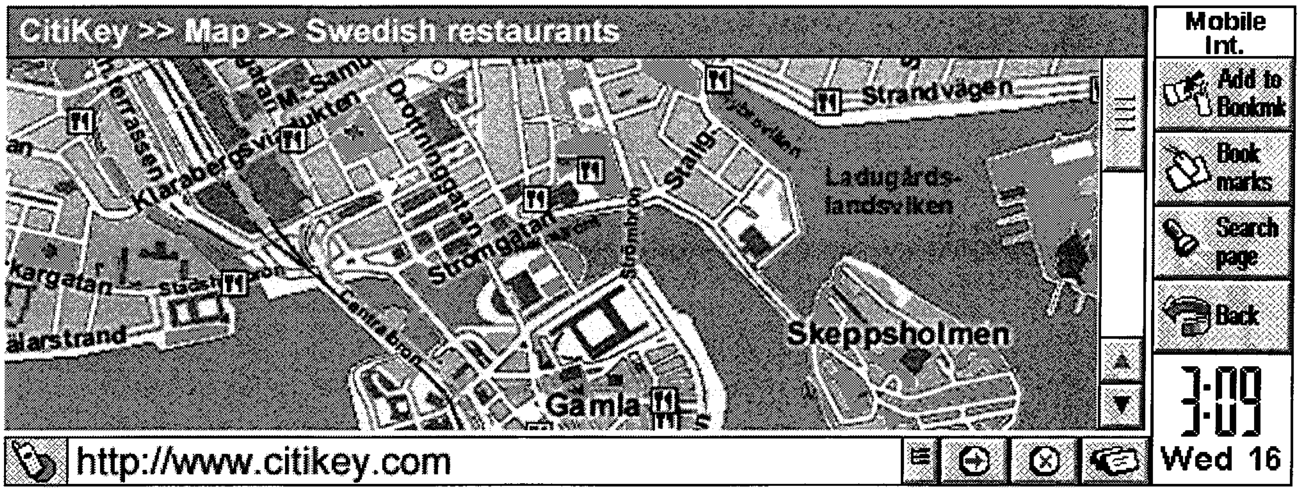



**Figure 1.1: Screenshot of Citykey system**

Currently, Citikey platform exist in three variants: for Palm devices, for Windows CE devices, and for WAP phones. In the case of WAP phones, you can not view maps, only textual information is provided, including the ability to perform search operations. Unfortunately, location services are not supported by Citikey and you can not use your current location for search operations. At present, this service is available only for London and Stockholm because of difficulties to build such a complete database on city information [CIT99].

The Car Navigation System (CNS) is example of the navigation system from the motorcar industry. The purpose of a car navigation system is to guide a driver to his or her destination safely and comfortably. The main functions of a car navigation system are as follows:

- vehicle positioning;
- route guidance;
- map displaying;
- indication of distance and direction to destination;
- calculation and display of optimum routes;
- driver guidance (voice & magnified intersection display);
- use of traffic information.

These functions are very similar to functions desirable of the navigation system for mobile phones. Moreover, the navigation system for mobile phones could replace CNS and become a universal personal navigation system.

The major provider of map database for car systems is Navigation Technologies (NAVTECH®), which makes navigable map databases for both Europe and North America. The NAVTECH database is a detailed, digital representation of the road network that provides the depth, accuracy and coverage needed to enable quality route guidance. Every road segment can have up to 150 attributes attached to it, including information such as street names, address ranges and turning restrictions. In addition, the database contains hundreds of thousands of points of interest information in many categories [NAV00]. The size of such a database requires the use of CD. Obviously, such a system is not a solution for a mobile user, but the database structure and other elements could be used.

As we see, in order to have full service many systems require keeping of all necessary information on mobile devices. However, in many situations the user does not know what kind of information he will need and when. The solution in that case is to organise a service on a server, which will generate requested files and send them to the user. One of the problems that arise is in the data transfer from the server to a mobile device. It is a serious problem for geographical data. As an example, a usual medium for data in car navigation systems is CD. Even a little part of a city on the map is still significant for a wireless link. Map images, descriptions of points of interest should be shown to the user. In some cases, we need to update a map image many times when the user is moving. Therefore, in this work I concentrate on the problem of geographical data transfer through a wireless network. Applications require careful development of the general architecture as well as a data format for information transfer to make user interaction more satisfactory and to optimise wireless network traffic.

I have conducted the following two major steps. The first one consists of research and analysis of formats, which can be used for geographic data transfer to the mobile client. I examine the following type of format types: bitmap, vector, and GIS formats. Each of them has its own particular features and distinctions for map representation in mobile environment. In addition, the overall schema of interaction between the server and clients is changed depending on the format. After that, I compare existent XML (Extensible Markup Language) vector formats and binary vector formats in order to define their ability to handle map images. This part is based mainly on intensive review of scientific papers and standard specifications from traditional published sources and Web-sites. The analysis of interaction of major system components was conducted with the help of UML (Unified Modeling Language) sequence diagrams. Also, additional experiments were used for formats comparison.

The second part is a design of a new XML-based geographic data model language, which is intended for describing geographical information and be efficient for the processing in mobile environment. Prerequisites for such a language are based on the results of the analysis in the first part. This part has mainly a constructive nature. We apply methods from geographic database design at the initial stage of format design. The results of this part are the detailed description of all tags and complete DTD (Document Type Definition), produced according to the XML specification.

The organisation of the thesis reflects the logical consecution of the research work. In Chapter 2, we give an overview of the mobile environment, and of the limitations arising in this environment. After that, we discuss the WAP specification, especially the WAP application environment, and its facilities for data management. Location services and major positioning methods which are used in location services, are analysed in the second part of the chapter. In Chapter 3, we give main concepts of the XML technology, and survey existing XML vector formats and binary ones. At the end of the chapter, we compare the file size of different vector formats for the same map images and compressibility of these formats. Also we discuss bitmap and vector formats in respect of their use for map representation. Chapter 4 contains basic concepts and ideas of GIS and topics related to the GIS on mobile devices.

In Chapter 5, we analyse the use of bitmap, vector, and GIS formats in location-based applications, and discuss information flow and function distribution for each case, with all their pros and cons. Based on conclusions of this chapter we develop a new XML data model language. Chapter 6 describes all steps of the design process. At the end of this chapter, we discuss the question of implementation of this new format. The DTD for our format and a simple example are included in the appendix. Section 7 concludes the work.

# 2 MOBILE ENVIRONMENT

The term "mobile environment"' is very closely associated with the term of "mobile computing". Mobile computing is a new emerging computing paradigm, which has become very popular lately, thanks to the rapid growth of telecommunication technologies. Cellular communication systems, wireless LAN, and wireless data network give to mobile users the capability of accessing information anytime, anywhere, and anyhow [Imi93]. As a result of that, people can manage their information, which is located at the servers in wired networks, with their mobile terminals. We can characterise mobile computing as follows: mobile computing consists of travelling people using wireless information devices connected to the computer network infrastructure supported global connectivity and remote computing [Vei00].

Such a paradigm creates a new application environment that has quite strong restrictions. Another side of such a rapid breakthrough in the telecommunication technologies is the expansion of the Internet to mobile users and a possibility for development of new class information services. Below, we discuss distinctions of mobile environment in comparison with conventional wired network environment and WAP standard, which offers optimised and a bearer independent platform for information services.

Firstly, wireless network provides more complex environment compared to wired networks because of the limitations of power, available spectrum, and mobility. These are some of the basic distinctive features of wireless network:

- Low bandwidth;
- Frequent disconnections;
- Long latency;
- High bandwidth variability;
- Unpredictable disconnections;
- The C-autonomy of terminals.

The last feature means that it depends on the terminal when to connect to, when to disconnect from the network. Therefore, a C-autonomous terminal is not reachable through the network during certain periods for other terminals [Vei00].

We can use different network technologies for data transfer: some of them are available at present, and some of them will be available in the near feature. The current GSM network provides bandwidth of 9.6 Kbits/s or 14.4 Kbit/s for data transfer. HSCSD (High Speed Circuit Switched Data), a circuit switched protocol, which is based on GSM and uses 4 radio channels simultaneously, provides 57.6 Kbits/s. HSCSD service was launched in 1999 by operators such as E-Plus and Orange [Mob99]. GPRS (General Packet Radio Service) is a packet switched wireless protocol that offers instant access to a data network. Although theoretical GPRS speed for data transmission is up to 171 Kbits/s, the actual speed will be 43.2 Kbits/s downstream and 14.4 Kbits/s upstream up to 56 Kbits/s bi-directional [Mob99]. GPRS is the first system which allows full instant mobile internet access. Pilot GPRS network is already functioning today, but a full service will be available in 2001. EDGE (Enhanced Data Rates for Global Evaluation) is a higher bandwidth version of GPRS permitting transmission speeds of up to 384 Kbits/s and will be launched in around 2002. After the EDGE, the next technology is UMTS (Universal Mobile Telephone System), third generation mobile phone system. The envisaged UMTS speed is 2 Mbits/s and will be available from 2003.

The second source of restrictions are mobile terminals because of limitations of power and form factor. By the form factor of a mobile terminal we mean its physical and constructive characteristics. For example, a mobile terminal should always have a small size in order to be mobile. Therefore, the small size leads towards the small screen, limited amount of batteries. Used input devices for mobile terminals are also differ from a conventional computer keyboard and mouse. A 12-buttons keypad or stylus is more usual for mobile devices. Developers have to keep in mind these factors as well as the environment in which people use mobiles. Consequently, the user interface will be different from one for conventional personal computers.

The following features characterise almost all handheld devices:

- Limited memory;
- Limited computational power;
- Small screen;
- Limited battery life;
- Relatively unreliable;
- Frequent location updates.

As was noted by George Forman [For94], certain data considered static for stationary computing becomes dynamic for mobile computing: geographical location, network addresses, network environment. Mobility means changing of the geographic location of a mobile terminal and the location becomes an important attribute of mobile devices. Ability to obtain its value, current geographical coordinates of the unit, is the basic factor for location-aware applications, which process data in respect to it. Because a mobile terminal often makes requests to a fixed network from different points, the other kind of mobility arises. This is the network mobility, changing of the network address. The new version of IP protocol, Ipv6, tries to solve this problem and make this mobility transparent for the application layer [MIP00].

Accordingly, the restrictions mentioned above that reflect hardware properties of mobile environment are passed to the application level. A software developer should take into account all of them during a development process. An application should respond to changes in network conditions and local resource availability when accessing remote data.

There are different strategies of application adaptation for a mobile environment. On the one hand, adaptation can be entirely the responsibility of individual applications. On the other hand, so-called application transparent adaptation can place the entire responsibility on the system. But typically, there are many possibilities between these two extremes. This approach supports collaborative adaptation between the application and the system. Proxies are used very often to perform adaptation on behalf of applications [Sat96a].

## 2.1 MOBILE DEVICES

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);

- 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.

## 2.2 WAP

WAP (Wireless Application Protocol) Forum is the industry association that has developed the WAP standard. The objectives of WAP Forum, as articulated in their published documents are as follows [WAPA99]:

- to bring Internet content and advanced data services to digital cellular phones and other wireless terminals.
- to create a global wireless protocol specification that will work across differing wireless network technologies.
- to enable the creation of content and applications that scale across a very wide range of bearer networks and device types.
- to embrace and extend existing standards and technology wherever possible.

The WAP specification defines standard architecture and a set of protocols intended to implement wireless Internet access. Roughly speaking, the WAP standard defines two things: an application environment and a protocol stack. The application environment consists of a markup language, WML, and a programming language, WMLScript [WAE99]. The protocol stack consists of a session protocol, transaction protocol, security protocol, and datagram protocol. The protocol set isolates the application from the bearer so that the application can be run regardless of the actual transport service being used.

Figure 2.1 illustrates WAP architecture and its major elements [WAPA99].

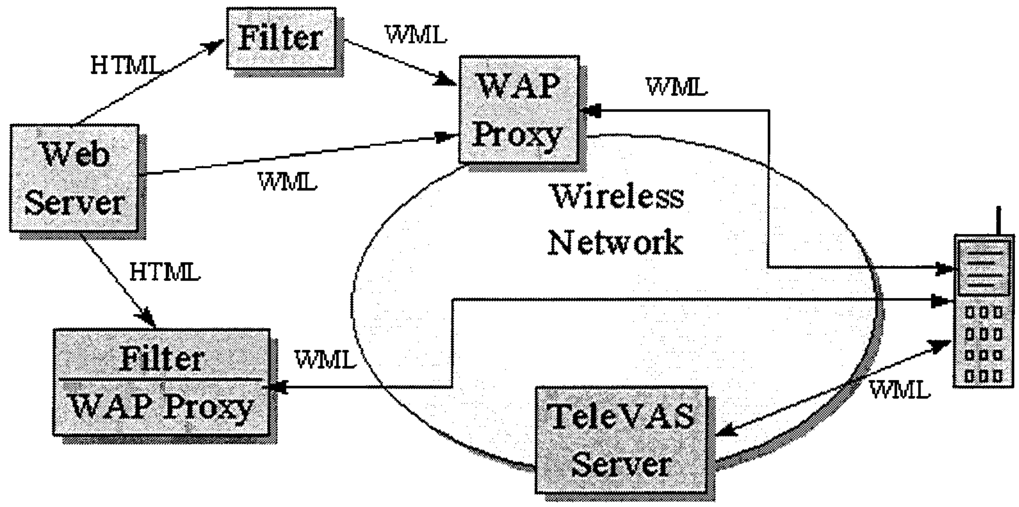



**Figure 2.1: WAP Architecture**

### 2.2.1 WAE

The WAE logical model is very similar to the WWW logical model and includes the following major elements: a mobile client, gateway, and the origin server. The WAE architecture includes all elements of the WAP architecture related to application specification and execution and it is predominantly focused on the client-side aspects of WAP's system architecture, namely items relating to user agents. Specifically, the WAE architecture is defined primarily in terms of networking schemes, content formats, programming languages and shared services. Interfaces are not standardised and are specific to a particular implementation. This approach allows WAE to be implemented in a variety of ways without compromising interoperability or portability. This approach has worked particularly well with a browser (a class of user agents) model such as that used in the WWW. To this end, one of the primary objectives of the WAE effort is to establish an interoperable environment that will allow operators and service providers to build applications and services that are equipped to reach a wide variety of wireless platforms in an efficient and useful manner.

The major components of WAE are:

- *WAE User Agents* are software that reside at the client, and provide specific functionality such as content display to the end-user. User agents, such as browsers, are integrated into the WAP architecture. They interpret network content that is referenced by URLs. WAE includes user agents for the two primary standard contents, which are encoded Wireless Markup Language (WML) and compiled Wireless Markup Language Script (WMLScript) [WAE99].

- *Content Generators* are applications or services, such as CGI scripts, on origin servers that produce standard content formats in response to requests from user agents in the mobile terminal. WAE does not specify any standard content generators, but expects that there will be many such generators running on typical HTTP origin servers commonly used in the WWW today [WAE99].

- *Standard Content Encoding* is a set of well-defined content encoding mechanisms and formats that allow a WAE user agent, such as a browser, to conveniently navigate web content. The standard content encoding includes compressed encoding for WML bytecode, WMLScript, standard image formats, a multi-part container format and adopted business and calendar data formats [WAE99].

- *Wireless Telephony Application (WTA)* is a collection of telephony specific extension for call and feature control mechanism that provide advanced Mobile Network Services.

[WAE99], which provides on overview of the WAE architecture, shows that the WAE is composed of two logical layers, which are the user agents layer, and the services and formats layer. The WAE, by default, separates services from user agents, and assumes an environment with multiple user agents. However, this logical view is not intended to prescribe an implementation model. Thus, for example, a designer may choose to combine all the services into a single agent. Alternatively, there is also the option to distribute all the services among several user agents [WAE99].

#### *2.2.1.1 WAE User Agents*

The Wireless Markup Language (WML) user agent is the only fundamental user agent of the WAE. However, the WAE is not limited to this single WML agent. The WAE allows the integration of domain specific user agents with varying architectures and environments. In particular, a Wireless Telephony Application (WTA) user agent has been specified as an extension to the WAE specification for the mobile telephony environments. The WTA extensions allow application developers to access and interact with mobile telephone features, such as call control, as well as applications on the telephone, such as phonebook and calendar applications.

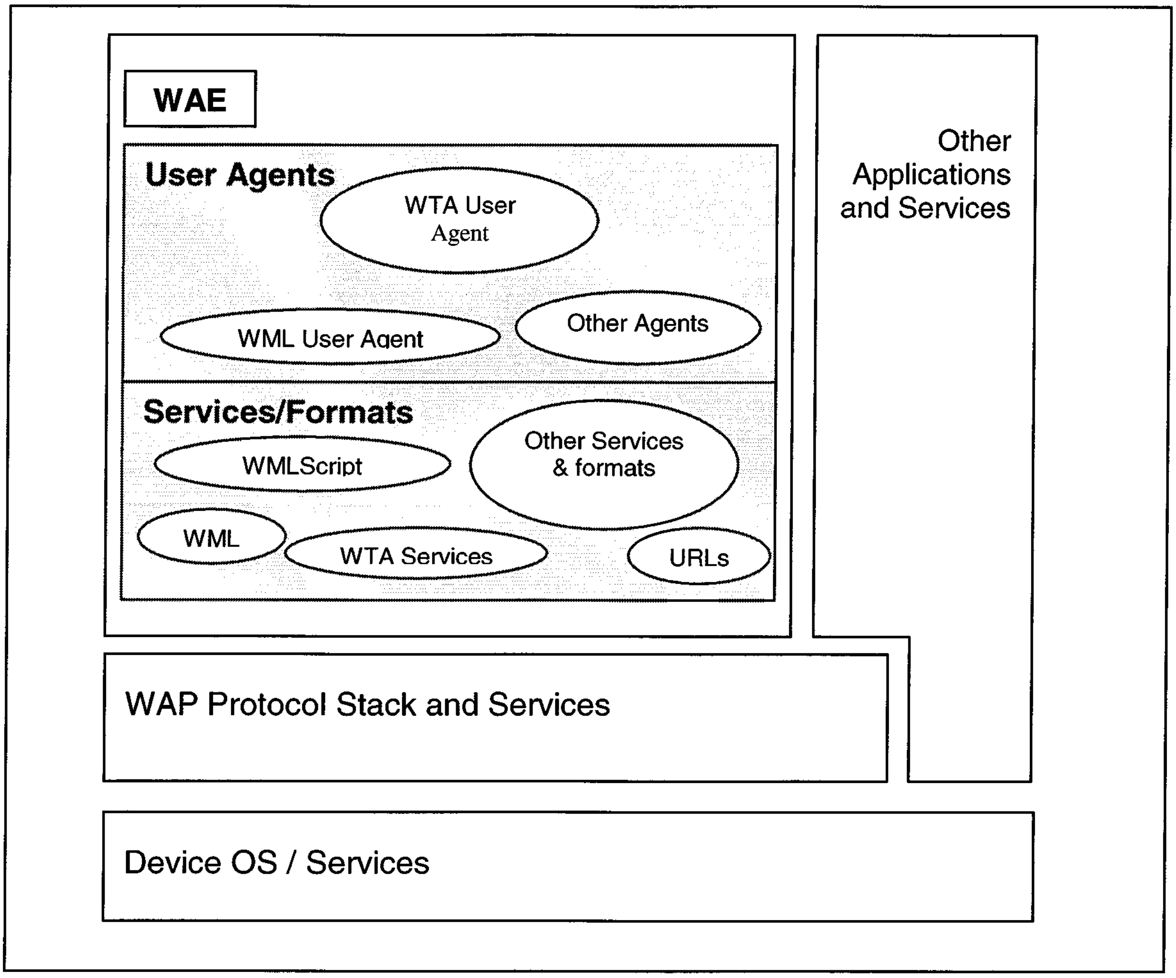



**Figure 2.2: WAE Clients Component**

The WAE does not specify any user agent. This is in keeping with the openness of the WAP philosophy. The features and capabilities of user agents are left entirely to the system designers. Instead, the WAE only defines fundamental services and formats that are needed to ensure interoperability among implementations. This services and formats layer is discussed in the next section [WAE99].

#### *2.2.1.2 WAE Services and Formats*

As is depicted in , WML and WMLScript form part of the services and formats layer. These two languages are the major elements through which knowledge and intelligence are expressed and represented in the WAP model. WML is a tag-based language that is optimised for specifying presentation and user interaction on limited capability devices such as hybrid communicators, and other mobile devices. Some of the major features of WML include support for text and images, various forms of user input, and a number of navigation mechanisms that allow roaming the history stack. In addition, there is the international support in the form of the document character set, Man Machine Interface (MMI) independence, narrow band optimisation, and state and context management (for a discussion of these features see [WAE99]).

WMLScript is the second major element of the services and formats layer. WMLScript is a lightweight procedural scripting language that is based on a subset of the JavaScript scripting language. It enhances the standard browsing and presentation facilities of WML with behavioural capabilities, supports more advanced user interaction behaviour, adds intelligence to the client, provides a convenient mechanism to access the device and its peripherals, and reduces the need for round-trips to the origin server (see [WAE99] for a fuller discussion).

Another important element of the services and format layer is the WAE content formats. The WAE includes a set of specified content formats that facilitate interoperable data exchange. The method of data exchange depends on the data and the targeted WAE user agent. The two most important formats defined in WAE are the encoded WML, and the WMLScript bytecode formats. These two encoding formats allow transmission of WML and WMLScript more efficiently, as well as minimise the computational efforts that are required of the client. WAE adopts two additional content formats specific to data exchange among user agents suitable for both client-server communication and peer-to-peer communication: electronic business card (vCard 2.1), and electronic calendar and scheduling exchange format (vCalendar 1.0) [WAE99]. There are a number of other formats for data types that can be adopted in WAE.

### 2.2.2 Graphical facilities

In this section, we describe what facilities are provided by WAP for dealing with graphical information. According to the current WAE specification, WAE supports following standard WSP/HTTP media content types for commonly used image formats: image/jpeg, image/gif, image/tiff, and image/png. Also an optimised bitmap format WBMP (content type vnd/wap/wbmp) is introduced in the specification. WBMP is an encapsulated format; in other words, a WBMP object is wrapper object that maps the verbose headers of the full image format to an identification of the contents. The actual image contents contain all other necessary information [WAE99].

The WBMP specification is thus divided into two parts. The first part is the generic header for all image formats, which contains the following information:

- Type;
- width and height;
- WBMP version number.

The type identifier denotes the format of the embedded image. Type 0 is currently specified for the black-and-white non-compressed format [WAE99].

The second part is the type-specific formats specification, indicating the data format for a particular WBMP type. The WBMP format supports the definition of compact image formats suitable for encoding a wide variety of image formats and provides the means for optimisation steps such as stripping of superfluous headers and special purpose compression schemes. This leads to efficient communication to and from the client and for efficient presentation in the client display. A WBMP image has the following characteristics:

- Compact binary encoding
- Scalability, i.e., future support for all image qualities and types (colour depths, animations, stream data, etc.)
- Extensibility (unlimited type definition space)
- Optimised for low computational costs in the client.

The WAE specification provides a universal medium for further implementation of new image bitmap formats, but as was noted above, at the present, only the simple black-and-white format is defined. As for vector formats, note that one of the candidates is SVG (Scalable Vector Graphics). We will discuss SVG in addition below.

### 2.2.3 WAP Binary XML Content Format

Binary XML content format specification defines a compact binary representation of the XML. The binary XML content format is designed to reduce the transmission size of XML documents, allowing more effective use of XML data on narrow communication channels. The WML specification includes an example use of the binary XML content format for WML documents encoding [BXML99].

The binary format was designed to allow compact transmission with no loss of functionality or semantic information. The format is designed to preserve the element structure of XML, allowing a browser to skip unknown elements or attributes. The binary format encodes the parsed physical form of an XML document, i.e., the structure and content of the document entities. Additional information, such as comments, processing directives, the document type definition and unnecessary conditional sections, is removed when the document is converted to the binary format. An XML document must accurately represent the logical structure and semantics of the source document, as defined by the XML specification. Consequently, the source document must be well-formed [XML98].

The WML specification defines a set of single-byte tokens. Each token corresponds either to the tags or to the attributes names and values defined in the DTD (Document Type Definition) [WML99]. In other words, tokens are assigned to elements from the DTD. For example, the WML specification includes the table for tag tokens, the table for attribute start tokens, and the table for attribute value tokens. According to the binary XML content format specification, the encoder uses the set of defined tokens to code XML document. All elements, which are defined in the DTD, are replaced by corresponding tokens.

The following is an example of a simple tokenised XML document. It demonstrates basic element, string and entity encoding. Source document:

```
<?xml version="1.0"?>
<!DOCTYPE XYZ [
<!ELEMENT XYZ (CARD)+>
<!ELEMENT CARD (#PCDATA | BR)*>
<!ELEMENT BR EMPTY>
<!ENTITY nbsp " ">
]>
<XYZ>
   <CARD>
      X &amp; Y<BR/>
      X = 1
   </CARD>
</XYZ>
```

The following tokens are defined for the tag code space: BR – 5; CARD – 6; XYZ – 7.

Tokenised form (numbers in hexadecimal) follows. This example uses only inline strings and assumes that the character encoding uses a NULL terminated string format. It also assumes that the transport character encoding is US-ASCII. This encoding is incapable of supporting some of the characters in the deck (e.g.,  ), forcing the use of the ENTITY token.

Thus, the binary format enables compressing of any XML document in a simple manner without use of complex compression methods, which would require additional resources.

## Table 2.1: Example tokenised deck

| Token Stream | Description |
|---|---|
| 02 | Version number - WBXML version 1.2. |
| 01 | Unknown public identifier |
| 03 | Charset=US-ASCII (MIBEnum is 3) |
| 00 | String table length |
| 47 | XYZ, with content |
| 46 | CARD, with content |
| 03 | Inline string follows |
| ’ ’, ’X’, ’ ’, ’&’, ’ ’, ’Y’, 00 | String |
| 05 | BR |
| 03 | Inline string follows |
| ‘ ‘, ‘X’, 00 | String |
| 02 | ENTITY |
| 81 20 | Entity value (160) |
| 03 | Inline string follows |
| ‘=’, 00 | String |
| 02 | ENTITY |
| 81 20 | Entity value (160) |
| 03 | Inline string follows |
| ‘1’,’ ‘,00 | String |
| 01 | END (of CARD element) |
| 01 | END (of XYZ element) |

## 2.3 LOCATION SERVICES

One of the important parts of personal navigation system is the block for location determination. It provides information about the user's geographical location on the Earth. Output of such a block is usually coordinates in one of the possible coordinate system. There are two possible ways to determinate user's coordinates:

- Use of satellite-based navigation systems such as GPS or GLONASS (Global Navigation Satellite System);
- Use of network infrastructure based location services (LCS).

GLONASS is Russian satellite-based navigation system that is very similar to GPS and not so widely used as GPS [GLONASS]. Therefore, we will deal only with GPS. Sub-committee T1P1 of the American T1 Standards Committee has standardised the first three positioning methods for GSM: GPS, TOA (Time Of Arrival), and E-OTD (Enhanced Observed Time Difference) [LCS00]. Note that according to this document, all methods require, at least, initial interaction with SMLC (Serving Mobile Location Centre). SMLC is a new element of the GSM network infrastructure that contains functionality required to support location services. Two types of SMLC are possible: either NSS based or BSS based.

Figure 2.3 shows logical architecture of a GSM network with location service support [LCS00].

The mobile station can obtain either its own geographical coordinates or coordinates of a target mobile station. As a result of this, the specification specifies all security and authorisation topics for location services. Figure 2.4 depicts general information flow between the major elements of positioning system [LCS00]. The client makes request for positioning of the target station. The location service server should check rights of the requesting client for such types of operation and ability of performing positioning of the target terminal. If all rights were in order, the location server would define the position of the target mobile terminal and return the result to the requesting station. Location service utilizes one or more positioning mechanisms in order to determine the location of a target mobile terminal.

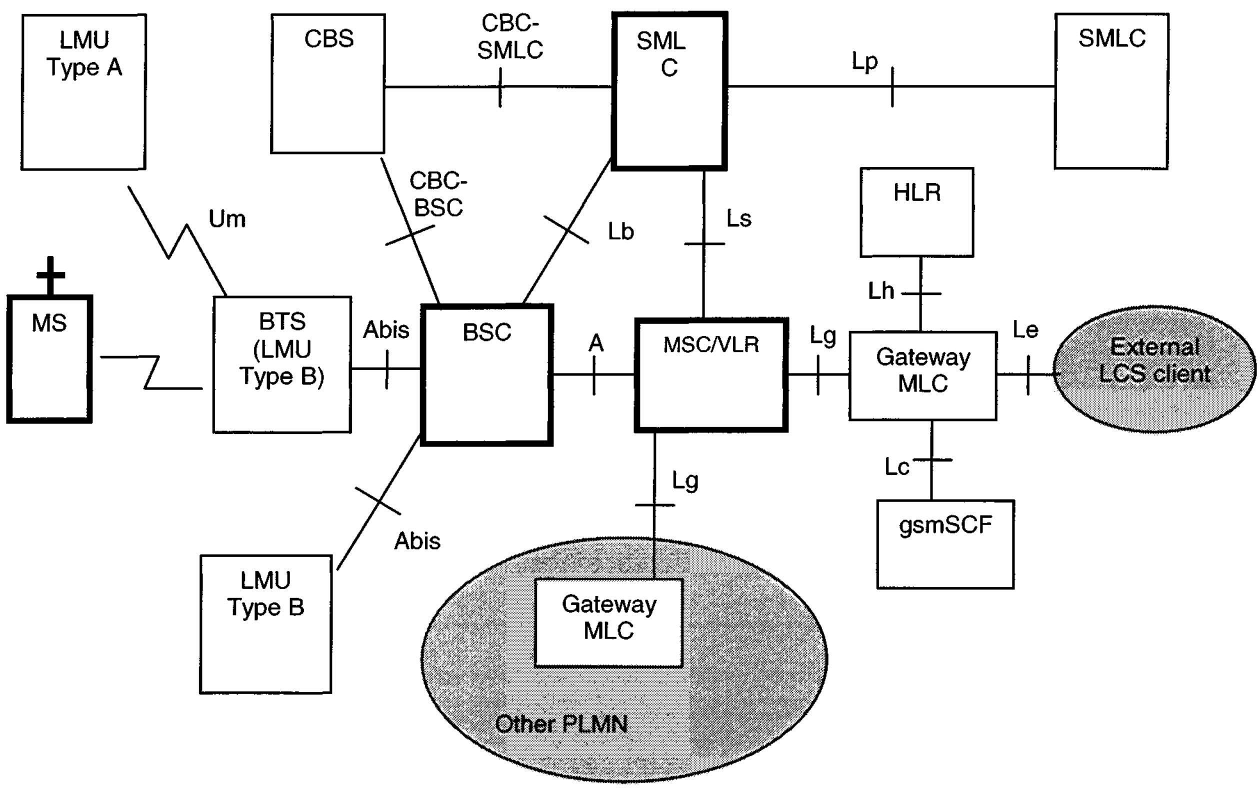



**Figure 2.3: Generic LCS Logical Architecture**

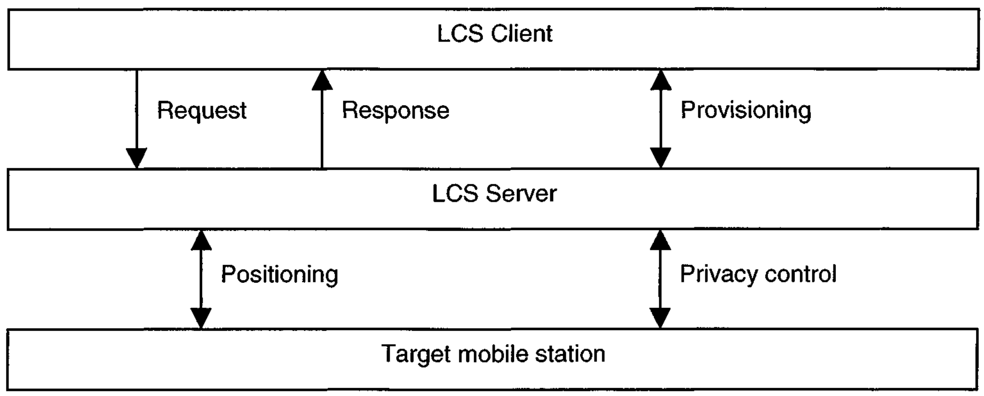



**Figure 2.4: LCS Logical Reference Model**

Below we overview facilities of conventional GPS systems and major network-based positioning methods.

### 2.3.1 GPS based positioning method

GPS is a satellite-based radio-navigation system established by the U.S. Department of Defense for military positioning applications and has been made available to the civilian community. GPS is a complex system, which can be used to achieve position accuracy ranging from 100 m to a few millimetres depending on the equipment used and procedures followed. In general, higher accuracy corresponds with higher costs and more complex observation and processing procedures. GPS provides two levels of service – a Standard Positioning Service (SPS) for general public use and an encoded Precise Positioning Service (PPS) primarily intended for use by the Department of Defense. There are two possible varieties of GPS:

- standard GPS;
- differential GPS (DGPS).

The accuracy of standard GPS is 58 meters according to standards [GPS96], which means that the accuracy is at least that 65 % of the time (1 sigma), under Selective Availability conditions. The policy is called "Selective Availability" or "SA". It is intentionally degrading GPS accuracy for SPS in order to make it impossible to use GPS for accurate weapons projection by terrorist groups [USP96]. Fortunately, even with all these inaccuracies, the error margin remains tolerable and DGPS can significantly reduce these problems.

President Clinton ordered that the intentional degrading of the civilian GPS signal be discontinued starting the 1 of May, 2000 [GPS00]. If the 20-meters accuracy is enough for some customers, the basic civil signal may be sufficient for them and they would not use DGPS augmentations any more. But if higher accuracy (1-3 meters) is necessary, the use some form of DGPS will still be needed.

DGPS is a way to make GPS much more accurate. It does this by comparing the GPS measurements in the mobile unit with GPS measurements taken at a reference station. Since the reference station is at a fixed location, it can find the difference between its known position and the information received from the satellites. It then uses this difference to calculate the errors in each satellite's signals – mostly the errors imposed by SA. This information can then be used to correct the satellite signals received by the mobile units to get much more accurate positions. Accuracy can be improved from 58 meters to 2 meters, at least 65 % of the time depending on how often corrections are applied.

Requirements of DGPS:

- Since DGPS works by correcting for the errors specific to each satellite, the mobile unit should be in the same general area as the reference station so that they are in view of the same satellites. This area can extend up to hundreds of kilometres in relatively flat regions, or it can be restricted to less than 100 kilometres in mountainous terrain.
- GPS errors may vary quite a bit within a minute, so to maintain accuracy GPS corrections should be received and applied every 1 to 20 seconds, depending on how much accuracy is needed [Trimb98].

GPS is available everywhere as a standard version and one does not need any contract with the local cellular network owner. The GPS solution does not work properly in some city areas, where high skyscrapers are located, and inside buildings because of very high frequency and the low field strength of the signal. The DGPS system only increases accuracy of positioning and does not avoid these problems.

Leading manufactures of GPS products offer both complete mobile and base GPS receivers and OEM boards for 3rd party products. For instance, Trimble Navigation Ltd., which is a leading manufacturer of GPS system, offers a miniature Lassen LP GPS receiver module [Lass99]. Its total weight (without antenna and power unit) is 12.5 grams and the dimensions are such that it could fit in a mobile phone. It is possible to make a built-in GPS for a mobile terminal like in the Benefon Esc! Device [Ben99b].

We can transfer assistance data from a reference station through a GSM network to the mobile terminal. In that case, a built-in GPS receiver has the ability to improve significantly the accuracy of positioning and the response time. Such a solution is known as Assisted GPS (A-GPS).

Usually, the GPS receiver employs ECEF (Earth Centered, Earth Fixed) Cartesian coordinates. ECEF X, Y, and Z, Cartesian coordinates (XYZ) define three-dimensional positions with respect to the centre of mass of the Earth. Further, latitude and longitude are usually provided in the geodetic datum on which GPS is based (WGS-84). Datum, which is the basis for a planar coordinate system, is a set of parameters and control points used to accurately define the three-dimensional shape of the Earth (e.g., as a spheroid). Receivers can often be set to convert to other user-required datums. Position offsets of hundreds of meters can result from using the wrong datum [Dana99]. The GPS position report varies somewhere from 34 to 101 bytes in ASCII or from 11 to 22 bytes binary. Depending on facilities of a GPS unit, additional data processing of GPS data could be required at the mobile terminal.

### 2.3.2 Cellular network based positioning method

In GSM there are several timing parameters which can be used to calculate the distance from a base station to a mobile phone, or vice versa. If we know coordinates of at least three base stations and distance from them to a mobile terminal, we can determine geographical coordinates of a mobile. Therefore, positioning a target mobile terminal involves two main steps: signal measurements and location estimate computation based on the measured signals.

The simplest way to obtain the coordinates of a mobile station is to use the cell global identity (CGI), where the user is located. Because we know the location of each cell, we can approximately assess the position of the user. A cell can be a circular or triangular sector. Of course, the accuracy will depend on the size of a cell and will vary greatly on different sites. It is higher in cities and lower in rural areas. The radius of a cell may vary from 100 metres to 35 kilometres. The width of an arc for a sector is 550 meters. The Time Advance (TA) parameter can be used in order to improve accuracy of positioning. The TA parameter is an estimate of the distance from the mobile terminal to the base station. The measurement is based on the access delay between the beginning of a time slot and the arrival of bursts from the mobile terminal. The access delay is proportional to the distance between BTS and the terminal [SveG99]. Further, we can use time parameters from several nearby base stations. There are two such types of positioning mechanisms.

*Time of Arrival* (TOA) positioning mechanism is based on measuring TOA of a known signal sent from the mobile terminal and received at three or more measurement units. The known signal is the access bursts generated by having the mobile perform an asynchronous handover. SMLC calculates Time Difference of Arrival values by pair-wise subtracting the TOA values. The mobile position is then calculated via hyperbolic trilateration assuming that the geographic coordinates of the measurement units are known, and the timing offset between the measurement units involved in the measurement are known. This method will work with existing mobiles; i.e. there is no modification to the handset required. However, one requires additional hardware at the listening base station to accurately measure the TOA of the bursts.

*Enhanced Observed Time Difference* (E-OTD) positioning mechanism was developed from the Observed Time Difference (OTD) feature. For synchronised networks, the mobile station measures the relative time of arrival of the signals from several BTSs. For unsynchronised networks, the signals are also received by a fixed measuring point known as the Location Measurement Unit (LMU) whose location is known. The position of the mobile station is determined by deducting the geometrical components of the time delays to a mobile from the BTSs. Measurements are performed by the mobile station without any additional hardware. For OTD measurement synchronisation, normal or dummy bursts can be used. When the transmission frames of BTSs are not synchronised, the network needs to measure the Real Time Differences between them. To obtain accurate triangulation, OTD measurements and, for non-synchronised BTSs, RTD measurements are needed for at least three geographically distinct BTSs. Based on the measured OTD values, the location of the mobile can be calculated either in the network or in the mobile terminal itself, if all the necessary information is available.

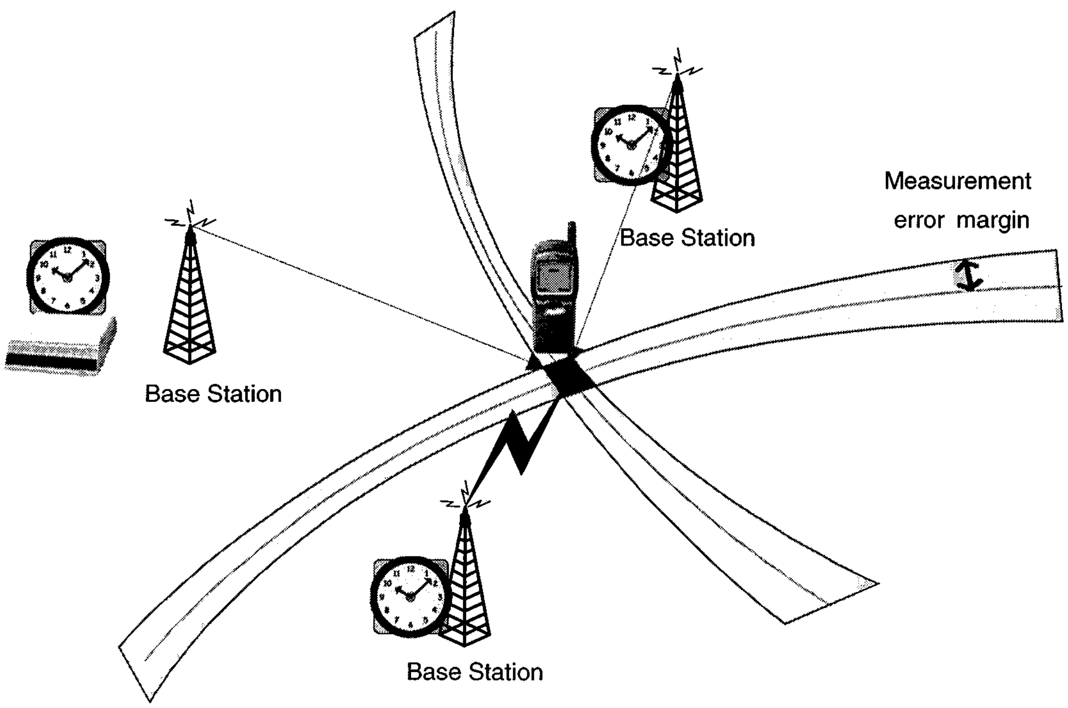



**Figure 2.5: E-OTD positioning**

One of the very important services for personal navigation systems is self-location. Self-location is the capability of the mobile station to obtain its own geographical location. There are two classes of mobile user's location determination [LCS00]:

- *Basic Self-Location* – the mobile device needs to interact with the network for each separate location request.
- *Autonomous Self-Location* – the mobile device does not need to interact with the network for each separate location request. One interaction with network enables the mobile device to obtain multiple positionings over a predetermined period.

### 2.3.3 Overview

There are many available location positioning methods, each of them having their own advantages and disadvantages. Table 2.2 compares described methods. Note, that an accuracy is distinguished in different sources; therefore, these are average values.

**Table 2.2: Positioning methods**

| Positioning method | Handset modification | Network modification | Accuracy | Place of calculations |
|---|---|---|---|---|
| CGI+TA | No | Yes | 300-1000 metres | Network |
| TOA | No | Yes | 50-150 metres | Network |
| E-OTD | Yes | Yes | 60-200 metres | Network or handset |
| A-GPS | Yes | Yes | 10-20 metres | Network or handset |

The major international benchmark for location system performance is the US FCC's E911 requirement that fixes following location accuracy.

- For network-based solutions: 100 meters for 67 percent of calls, 300 meters for 95 percent of calls;
- For handset-based solutions: 50 meters for 67 percent of calls, 150 meters for 95 percent of calls [FCC99].

### 2.3.4 Market of a location service

There are several companies working towards commercial implementation of a location service system. One commercial product in GSM environment based on the E-OTD method was released by Cambridge Positioning Systems Limited (CPS) during the summer of 1999 in England. The product is to be marketed outside England in the year 2000. The results of CPS's trials on its commercial E-OTD network consist of more than 9,000 measurements from 94 locations in Cambridge. Environments ranged from indoor urban to outdoor suburban.

The CPS results exceed FCC's E911 requirement by some margin with locations accurate to within 125m, 83% of the time [CPS99b]. CPS is going ahead with their trials of Universal Mobile Telecommunications System (UMTS). Early simulations of CPS on UMTS already demonstrate accuracy better than 20 meters. CPS is confident of achieving the target of 5 meters [CPS99a]. The CPS product requires some software in the mobile terminal plus both software and hardware in the base station controller.

Ericsson also offers their Mobile Positioning System for network operators. It supports CGI+TA positioning method, E-OTD method, or assisted GPS (A-GPS). The accuracy of the system depends on the method used and varies from as high as 10 meters for A-GPS to 300-1000 meters for CGI-TA. Ericsson provides a software development kit for the development of positioning applications. In 1998, Swedish operator Telia conducted trials with the Ericsson's system to evaluate its effectiveness in responding to GSM subscribers' emergency calls [MPS98].

According to many market researches, rapid growth of location services is expected along with development of location-aware applications in the very near future [Mob99, Nok99, SveG99]. In the next chapter, we will try to shed light upon geographical information and Geographical Information System.

# 3 GEOGRAPHICAL 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 [AGI96], GIS can be defined as: a computer system for capturing, storing, checking, integrating, manipulating, analysing and displaying data related to positions on the Earth's surface. Another variant by ESRI: GIS is an organised collection of computer hardware, software, geographic data, and personnel design to efficiently capture, store, update, manipulate, analyse, and display all forms of geographically referenced information [Glos98]. 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 specialised GIS, which will work in mobile environment. However, the environment limits the features of the geographical system. We have examined these constrains in Chapter 2.

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. 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, which is used, for example, in DEM (digital elevation model data of United States Geological Survey).

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 co-ordinates as an 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 centrelines 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, you 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.

## Table 3.1: Raster vs. Vector

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

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 [AGI96], 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 [Glo98], 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). 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 modelling 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 modelling 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 a coverage can be organised into subclasses, each having similar properties.

There are many ways for geographic information representation and the most basic of these 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 you do not need to use maps outdoors. But the present range of diverse mobile devices open up opportunities to use digital maps anywhere you need. Unfortunately, these mobile devices have limited resources and a quite small screen, which makes work with maps more difficult. 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 modelling 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 modelling, 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, you 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 areal photograph. If the image has a geographic dimension, it could be georeferenced, i.e. registered to a common geographic coordinate system. Then, it could be used in conjunction with geographic data sets of the same area. A georeferenced graphic image or drawing can be presented, for example, as a background or an overlay of a geographic data in a view. However, you can not identify a geographic feature or query a location from it; though a visual determination of co-ordinates 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.

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.

# 4 XML TECHNOLOGY

XML is a subset of the Standard Generalized Markup Language (SGML) intended to make it more usable for distributing materials on the World Wide Web. SGML has been the standard, vendor-independent way to maintain repositories of structured documentation for more than a decade, but it is not well suited to serving documents over the WWW [SGML89]. Defining XML as an application profile of SGML means that any fully conformant SGML system will be able to read XML documents. However, using and understanding XML documents does not require a system that is capable of understanding the full generality of SGML. XML is a much-restricted form of SGML. XML differs from SGML primarily in simplifying the formalisms of SGML in order to ensure that an XML parser is simple enough to embed in even lightweight software, including software for mobile devices. From HTML point of view, it differs primarily from it in allowing the user to specify new tags, marking types of elements not foreseen in the HTML specification, and making it possible for common off-the-shelf browsers and other software to handle such user-defined element types usefully [Spe97].

XML was created and developed by the W3C XML Working Group, which includes leading industry companies such as Adobe, ArborText, DataChannel, Inso, Hewlett-Packard, Isogen, Microsoft, NCSA, Netscape, SoftQuad, Sun Microsystems, Texcel, Vignette, and Fuji Xerox as well as experts in structured documents and electronic publishing. [W3C98a].

In order to appreciate XML, it is important to understand why it was created. XML was created so that richly structured documents could be used over the web. The only viable alternatives, HTML and SGML, are not practical for this purpose. HTML comes bound to a set of semantics and does not provide arbitrary structure. SGML provides arbitrary structure, but is too difficult to implement for a web browser.

## 4.1 WHAT IS XML?

The Extensible Markup Language (XML™) is the universal format for structured documents and data on the Web [W3C98b]. Structured documents are documents that contain both content (words, pictures, etc.) and some indication of what role that content plays. Almost all documents have some structure. A markup language is a mechanism to identify structures in a document. The XML specification defines a standard way of adding markup to documents [XML98].

XML is really a meta-language for describing markup languages. In other words, XML provides a facility to define tags and the structural relationships between them. Since there is no predefined tag set, there can not be any preconceived semantics. All of the semantics of an XML document will be defined by the applications that process them.

The basic concept of an XML document is the element. Elements can be nested at any depth and can contain other elements (sub-elements). An element contains a portion of the document delimited by two tags: the start tag at the beginning of the element, with the form <tag-name>, and the end tag at the end of the element, with the form </tag-name>, where tag-name indicates the type of the element (markup).

A document type declaration can be attached to XML documents, specifying the Document Type Definition (DTD), that is, the set of rules the XML documents may follow. The document type declaration is composed of two parts: the element declarations and the attribute list declarations. The element declarations specify the structure of the elements contained in the document and their type. The attribute list declarations specify, for each element, the list of its attribute names, types and (possible optional) default values.

Nowadays, there are quite a many additional technologies and mechanisms for XML in development: Namespaces, XML Schema, XPath, XLink, XPointer, XSL, DOM, RDF and so on. They are intended to advance and enhance XML capabilities. Note only about namespaces. Namespaces is a mechanism that allows using in a XML document multiple markup vocabularies, XML languages [Nam99]. Later, we will discuss the problem of using WML with other XML languages.

Below, we will try to shed light upon some emerging XML-based vector formats resembling, in some respects, the format for map representation.

## 4.2 XML-BASED VECTOR FORMATS

Up to now, XML has been used to identify the structure of text elements in Web documents. Now there is a growing demand to use XML to define the syntax for graphical information in Web documents as well.

In storing a visual image as a mathematical description of its lines, curves, shapes and colours, a vector graphics format is compact and efficient. More importantly, the same image can be rendered at optimal resolution on a computer screen or in print.

Representing graphical data in a structured manner has many potential benefits:

- Extracting text from graphics for indexing.
- Building indices of graphical components.
- Distinguishing between informational graphics and navigational graphics.
- Dynamically generating or modifying graphical elements.
- Controlling the positioning and layout of graphics through style sheets.
- Creating interactive graphical elements with scripting.

An XML-enabled vector-based graphical language can include all kinds of information that might be delivered along with the graphic or be available depending upon what request was made. It can provide the ability to add "extra" information about graphical data for "free." Adding metadata to graphics can open the door to new applications without significantly adding to the document's file size or download time. A graphic might supply metadata such as author, title, date and copyright. It might supply the aspect ratio or physical size for a stand-alone print. It might supply business information such as fees for usage [Rei99].

In summary, an XML-based vector graphics language would make Web graphics lightweight elements of a Web document, and bring with it greater functionality and flexibility.

### 4.2.1 SVG

Scalable Vector Graphics (SVG) is a language for describing two-dimensional graphics in XML. SVG allows for three types of graphic objects: vector graphic shapes (e.g., paths consisting of straight lines and curves), images and text. Graphical objects can be grouped, styled, transformed and composed into previously rendered objects. The feature set includes nested transformations, clipping paths, alpha masks, filter effects and template objects.

More than two years in the works, SVG represents a collaborative effort by some of the biggest companies in the computer world, such as IBM, Microsoft, Apple, Xerox, Sun Microsystems, Hewlett-Packard, Netscape, Corel, Adobe, Quark, and Macromedia to find a workable cross-platform solution to Web imaging [SVG99].

According to SVG requirements, the graphics in Web documents will be smaller, faster, more interactive, and be displayable on a wider range of device resolutions from small mobile devices through office computer monitors to high-resolution printers [SVG98].

SVG stands for Scalable Vector Graphics, an XML grammar for stylable graphics, usable as an XML Namespace [Nam99].

To be scalable means to increase or decrease uniformly. In terms of graphics, scalable means not being limited to a single, fixed, pixel size. On the Web, scalable means that a particular technology can grow to a large number of files, a large number of users, a wide variety of applications. SVG, being a graphics technology for the Web, is scalable in both senses of the word.

SVG graphics are scalable to different display resolutions, so that for example printed output uses the full resolution of the printer and can be displayed at the same size on screens of different resolutions. The same SVG graphic can be placed in different sizes on the same Web page, and re-used in different sizes on different pages. SVG graphics can be zoomed to see fine details, or to aid those with vision implements.

SVG format can also integrate raster images and can combine them with vector information such as clipping paths to produce a complete illustration.

Since all modern displays are raster-oriented, the difference between raster-only and vector graphics comes down to where they are rasterised; client side in the case of vector graphics, as opposed to already rasterised on the server. SVG gives control over the rasterisation process, for example to allow anti-aliased artwork without the ugly aliasing typical of low quality vector implementations [Tho99].

Most existing XML DTDs represent either textual information, or represent raw data such as financial information. They typically provide only rudimentary graphical capabilities, often less capable than the HTML <img> element. SVG proves a rich, structured description of vector and mixed vector/raster graphics; it can be used standalone, or as an XML namespace with other grammars.

### 4.2.2 VML

VML (Vector Markup Language) is a markup language based on existing HTML capabilities which allows vector graphical information to be integrated with the text and other data typically found in HTML pages. VML is an application of XML. Its format defines stateless structured semantics that not only meet the fundamental needs of rich editibility and interchange of graphical data in HTML-based authoring tools but also offers a high-level of support for manual authoring.

VML was designed to provide a textual way to describe vector graphics that can be easily cut and pasted for use in a wide variety of authoring tools. It's also written to be integrated into existing HTML 4.0 markup, (which, of course, must be well formed), and to simplify the process of editing text-based vector graphics.

VML was written primarily to be integrated into editors and for use as an export format. VML's markup is frighteningly verbose, but note that most likely the code will be generated by an application, not written by hand.

VML has "built-in" provisions that can be used to describe objects that may be further edited. It also contains a few MS Office-centric features, such as the adjustable handles used to manipulate graphics in Microsoft Word.

VML uses "`v:`" to define its own namespace for VML tags [Nam99]. This should not be confused with v-space in CSS. The "`v:`" is actually an extension from a CSS2-compliant rendering engine. The namespace prefix never appears within the CSS information.

The <shape> element is a VML tag that defines a path for drawing the object. The *stroke* attribute determines whether the outline of the object is visible and the strokecolor attribute makes the outline red. The *fill* and *fillcolor* attributes determine whether an object is filled and with what colour. Below, there is the simple example of a VML code, which generate a five-point star shown in Figure 4.1.

```
<v:shape
style='top: 0; left: 0; width: 250; height: 250'
stroke="true"
strokecolor="red"
strokeweight="2"
fill="true"
fillcolor="green"
coordorigin="0 0"
coordsize="175 175">
<v:path v="m 8,65 l 72,65,
92,11,112,65,174,65,122,100,
142,155,92,121,42,155,60,100 x e"/>
</v:shape>
```

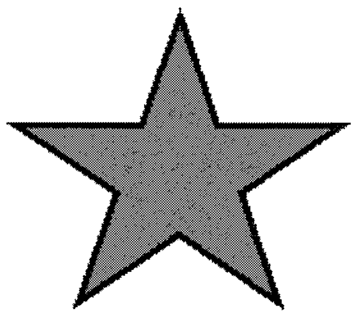


**Figure 4.1: A five-point star described with VML**

The *coordsize* attribute defines how many units there are along the width of the containing block. The coordorigin attribute defines the coordinate at the top left corner of the containing block.

For example, if a group was defined as follows, the containing block would be 300 pixels wide by 250 pixels high (assuming that the parent element of this group was not another group):

```
<v:group style='width: 300px;
height: 250px' coordsize="1000,1000"
coordorigin="-500,-500" />
```

The coordinate system inside the containing block would range from –500.0 to 500.0 along the x-axis and –500.0 to 500.0 along the y-axis with 0.0, 0.0 right in the centre of the rectangle. Any shapes inside the group will be positioned and sized according to this local coordinate system. No matter how the width and height of the group is changed, the local coordinate system inside will remain the same.

The rationale behind this is that the vectors defining a shape can be specified in a local coordinate system. If the containing block for the shape is changed, the outline of the shape will be automatically scaled to the new box. Similarly, shapes within the local coordinate system of a group will be automatically scaled if the containing block of the group changes.

Although the VML specification does not provide a complete DTD, it does include DTD snippets throughout the document. It would be more efficient to design your custom "snippets" required for your project's XML templates, and then merge them together to create a complete DTD or schema, if validation is desired.

Other VML features include:

- Re-editing (after cutting and pasting elsewhere in the HTML page or into another application) with no loss in output quality;
- Handle adjustment;
- Stateless graphics (grabbing the multipointed star which lies under the circle) ;
- Parametric-based graphics (or formula based).

While the core engine is based on the Microsoft Office graphic model, the co-submitters suggested improvements that were incorporated into the VML spec, such as the refinement of the parameterized paths and the compactness of the path description.

### 4.2.3 PGML

The Precision Graphics Markup Language (PGML) is a 2D graphics language that provides precise control of layout, fonts, colour and printing, which will result in Web pages with compelling text, images and graphics, as well as dynamic events and animation. It is designed to meet both the simple vector graphics needs of casual users and the precision needs of graphics artists.

PGML uses the imaging model common to the PostScript® language and Portable Document Format (PDF) as its basis in order to provide simple-to-use graphical objects and precise visual fidelity. This imaging model is lightweight yet robust, comprising a small number of object types, attributes and operators which can be easily leveraged to satisfy the most challenging 2D graphics needs.

The Java 2D API includes the complete PDF/PostScript imaging model together with additional features for Web applications (e.g., transparency and various extensibility features). While PGML certainly can be implemented separately from JDK 1.2, the Java 2D API does offer a potentially straightforward and cross-platform graphics engine on which to build an implementation of PGML.

In PGML, text strings are specified as standard XML character data within an element, using "x" and "y" attributes to define the starting point of the text fragment once it was rendered in the browser. For example, the following will cause the string "PGML Format" to be drawn:

```
<text x="100" y="150">PGML Format</text>
```

Inside a PGML document, the coordinate system can be changed by specifying a transformation matrix. Transformation matrices in PGML work just as they do in PDF. Transformations alter coordinate systems, not the objects themselves. A transformation matrix in PGML specifies the transformation from the transformed (new) coordinate system to the untransformed (old) coordinate system. All coordinates used after the transformation are specified in the transformed coordinate system.

PGML will probably not be generated by hand very often, no more so than PostScript was. It is more likely to be implemented as an export format within Adobe's own product line (e.g., Photoshop and Illustrator). Adobe has also pledged to make PDF/PostScript-to-PGML converters, plug-ins, ActiveX controls and other components available for download on its Web site. Having this kind of functionality "built-in" will undoubtedly simplify the creation and manipulation of PGML graphics on the Web.

### 4.2.4 DrawML

DrawML is a 2D scalable graphics language designed to facilitate the creation of simple technical drawings. Furthermore (and most importantly), DrawML focuses on the process of maintaining and refining a drawing. A drawing should be as easy to update as the document it resides in. The reason for the focus on maintenance is the increased importance of intranets. Up to now internet technology has been used primarily for publishing. People working within an intranet expect to create and change documents on-the-fly.

DrawML drawings are embedded in XML documents in the same way as tables are embedded. Elements from the parent DTD are reused inside drawings. DrawML defines algorithms to handle positioning, resizing and rubberband connections between visual elements.

DrawML is an application of XML 1.0.

## 4.3 BINARY VECTOR FORMATS

### 4.3.1 WebCGM

CGM (Computer Graphics Metafile) has been an ISO standard for vector and composite vector/raster picture definition since 1987. It has been a registered MIME type since 1995 [CGM96]. CGM has a significant following in technical illustration, electronic documentation, geophysical data visualization, and amongst other application areas. WebCGM is a profile for the effective application of CGM in Web electronic documents. WebCGM has been a joint effort of the CGM Open Consortium in collaboration with W3C staff under the W3C-LA project. WebCGM conformance requirements will enhance interoperability of implementations, and it should be possible to leverage existing CGM validation tools, test suites, and the product certification testing services for application to WebCGM. While WebCGM is a binary file format and is not "stylable", nevertheless WebCGM follows published W3C requirements for a scalable graphics format where such are applicable. The design criteria for the graphical content of WebCGM aimed at a balance between graphical expressive power on the one hand, and simplicity and implementability on the other. A small but powerful set of metadata elements is standardized in WebCGM, to support the functionalities of hyperlinking and document navigation, picture structuring and layering, and, search and query on WebCGM picture content.

However, the community of people who want to exchange CGM is fairly focussed – primarily the aerospace, automotive and defense industries. CGM is an industrial-strength format but it has not seen much use on the Web. Also, because CGM is not expressed in XML, applying style sheets to

CGM graphics is difficult. These factors and others seem to point to the need for a vector graphics tagset for XML, one which is able to make use of XML namespaces and stylesheets to mix text and graphics.

### 4.3.2 Flash format

Nowadays, the flash animation is a leading technology for multimedia and interactive web-site building. A designer can develop vector-based pages with Flash Animator software by Macromedia. A user has to download the plug-ins from Macromedia to view web-sites which use flash animation. The file format which is used by this technology is binary optimized for low bandwidth lines vector-centric graphical format. It also allows the use of video files with QuickTime format and audio files with a MPEG coding. Basic features of the flash technology are:

- Vector figures transparency;
- Animated buttons and menus;
- Shape morphing;
- Bitmap support;
- Custom fonts support.

The Flash file format was designed as a very efficient delivery format and not as a format for exchanging graphics between graphics editors. It was designed to meet the following goals [Mic99]:

- *On-screen Display* — The format is primarily intended for on-screen display and so it supports anti-aliasing, fast rendering to a bitmap of any colour format, animation and interactive buttons.
- *Extensibility* — The format is a tagged format, so the format can be evolved with new features while maintaining backwards compatibility with older players.
- *Network Delivery* — The files can be delivered over a network with limited and unpredictable bandwidth. The files are compressed to be small and support incremental rendering through streaming.
- *Simplicity* — The format is simple so that the player is small and easily ported. Also, the player depends upon only a very limited set of operating system functionality.
- *File Independence* — Files can be displayed without any dependence on external resources such as fonts.

- *Scalability* — Different computers have different monitor resolutions and bit depths. Files work well on limited hardware, while taking advantage of more expensive hardware when it is available.
- *Speed* — The files are designed to be rendered at a high quality very quickly.

Flash technology could be used for navigation interfaces, technical illustrations, long-form animations, and other effects for Web sites. A quite interesting example of flash technology for multimedia map representation is located at www.2000.varazdin.com [Var00]. Varazdin 2000 is an academic project pushing the limits of Macromedia Flash software to a new level. It is a vector city map with options such as rotation, zooming, 360 degree navigation, street location by name and many more. It also gives an overview of the historical buildings in the city centre with their images.

Macromedia has made the Flash format open and it is in the process of submitting the Flash vector specification to an appropriate standards body.

## 4.4 COMPARISON OF VECTOR FORMATS

The purpose of this section is to compare formats that were described earlier on. We emphasis the encoding of map images. There are two parameters that we use for comparison: the file size and compressibility. We can group all formats into two major categories: text-based formats and binary formats. At the end of the section, we will analyse which type is better suited to mobile environment.

Test maps were generated from real-life data of Jyväskylä in Finland with the help of ArcInfo software using Windows Metafile format. There are two maps with a different level of details.

- "Map1k" represents the town centre; the area size is approximately 1x1 kilometres; the approximate scale is 1:7672.
- "Map4k" represents the main part of the town; the area size is approximately 5x5 kilometres, and the represented area is 25 times as large as at Map1k; the approximate scale is 1:38461.

Figure 4.2 depicts these maps. The maps contain a street network, water boundaries and land boundaries, without any labels. Following vector graphical formats were examined:

- Windows Metafile format (wmf);
- ASCII Drawing Interchange file format (dxf);

- Encapsulated PostScript file format (eps);
- Scalable Vector Graphics file format (svg);
- Flash file format (swf);
- Computer Graphics Metafile format (cgm).

Following graphical editors were used to convert the Window Metafile format in other different vector formats:

- Flash Animator v. 4.0 for Flash file format (swf), ASCII Drawing Interchange file format (dxf);
- Micrografx Designer v. 7.1 for Computer Graphics Metafile format (cgm);
- Mayura Draw (beta edition) v. 3.6 for Scalable Vector Graphics file format (svg), Encapsulated PostScript file format (eps).

Also the gzip utility program was used to compress files because almost all of them, except CGM and SWF, are text-based and they do not aim to minimise the file size. The gzip tool uses lossless, general-purpose method called Deflate, which consists of two phases: first, a variant of the LZ77 algorithm [ZiL77] is applied, after which, the result is compressed with Huffman coding [Huf52].

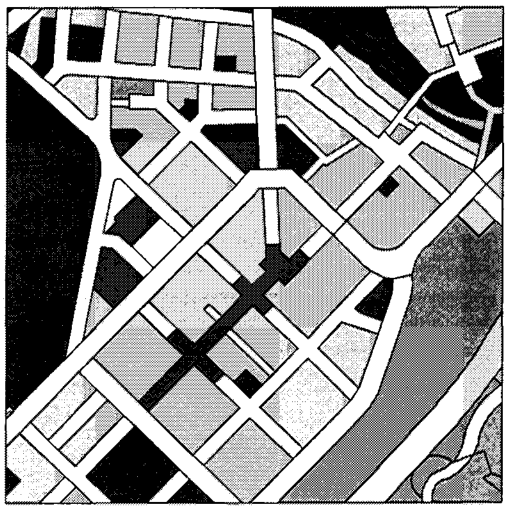
| Map1k.wmf | Map4k.wmf |
|---|---|
|  |  |

**Figure 4.2: Test maps of Jyväskylä**

Note that actually, the file map4k contains information about 25 maps each of them equal to the map encoded in the file map1k. But as Table 4.1 depicts, the average file size of map4k is approximately 13 times larger than the average file size of map1k. And at the same time, the average compressed file size of map4k is approximately 12 times larger than the average compressed file size of map1k. Therefore, if the user intends to view a large region of the city, it makes sense to transfer one large file rather than a sequence of smaller files. By doing so, we will halt the transfer capacity required.

**Table 4.1: Uncompressed and compressed sizes of the vector formats, and the gain achieved with the gzip utility.**

| Vector formats | Map1k | | | Map4k | | |
|---|---|---|---|---|---|---|
| | Original, byte | Compressed, byte | Gain | Original, byte | Compressed, byte | Gain |
| *WMF* | 14434 | 4389 | 0.70 | 195244 | 65355 | 0.67 |
| *DXF* | 57662 | 6259 | 0.89 | 881580 | 82988 | 0.91 |
| *EPS* | 51354 | 11975 | 0.77 | 600711 | 105166 | 0.82 |
| *SVG* | 40995 | 7661 | 0.81 | 615300 | 97802 | 0.84 |
| *SWF* | 6033 | 4624 | 0.23 | 76220 | 51785 | 0.32 |
| *CGM* | 7536 | 5188 | 0.31 | 113292 | 73392 | 0.35 |

What conclusions can we make from Table 4.1? Firstly, it is clear that the compressibility of binary formats is worse than the compressibility of text–based vector formats. This is because binary formats were created directly to achieve the minimal file size and they usually contain only necessary information in a highly compact form. Therefore, binary formats have quite an attractive file size and they are suitable for quick program parse. On the other hand, software development, portability, and expandability are more difficult to achieve with such files. For example, if we changed the length of one field or added a new field in the middle of a record, the rest of data would become corrupt for reading. We would need to modify the software to achieve the desired effect. In order to avoid such problems, the structure of the format should be quite flexible for extension. The situation is the inverse for text-based formats. The parse of text files is performed easily with appropriate software libraries. This applies especially to XML formats with their strict logical structure. If we add or change something in a text format, the program would read this new addition as unknown and would ignore it but the rest of the file would be read in the right way. Therefore, software would work properly with a new version of the file format without any modifications.

What does it mean for mobile environment? If we use binary format and want to add new features for some file format, we would need to upgrade the software at all terminals which use our format. In the case of text formats, the user need only upgrade his software if he wants to use new features but software will work properly otherwise.

## 4.5 BITMAP FORMAT VERSUS VECTOR FORMAT

Note that it is not appropriate to compare the vector format and the bitmap format because they belong to different classes of image formats. However, we can assess the file size for a certain level of the scale. We can transfer the same map on a large scale by one file or several little parts. Below we estimate a better way for file transfer and zooming. Actually, Map1k is a part of Map4k with another value of scale. The scale values were taken from the initial maps in wmf-files. We used 16-colours GIF format as more suitable for simple geometrical images. The uncompressed SWF format was applied to vector files. Table 4.2 shows the approximate file sizes for both vector and bitmap formats for two scales. The upper number is a file size for the bitmap format; the lower number is a file size for the vector format. Because the vector file is the same for both scales, its file size remains the same.

### Table 4.2: File size for two scales

| Scale \ file | Map1k (Kbyte) | Map4k (Kbyte) |
|---|---|---|
| 1:7672 | 13.433 | 323 |
| | 6 | 76 |
| 1:38461 | 2 | 32 |
| | 6 | 76 |

We can draw the following conclusions from the table. Firstly, the sizes of bitmap files are smaller for the larger scale. This is because the map image with streets is very detailed and complex in that scale and the bitmap format encodes information in a more compact form. In contrast, if we choose smaller scale, the vector format becomes more effective, especially for large maps. Secondly, Map4k is the composition of 25 little maps like Map1k. It is clear from the table that for both formats and both scales the size of 25 little files is greater than the size of one big file. The reason for that is the additional file headers and weak effectiveness of compression methods for small files. Therefore, if a mobile terminal has enough resources to store a larger file, it makes sense to transfer one large file rather than several little ones.

Supposing for instance, we have made a map service for mobile users and want to provide an operation such as zooming. In the case of the bitmap format, a new file is required for each scale and the zooming operation is poorly performed because of characteristics of this format. The bitmap format stores the image as the matrix of pixels. Therefore, when the image is zooming out and the size is increasing, we need to calculate new pixels on the basic of the old ones with some method of interpolation. Therefore, the picture quality becomes worse. Because the vector format stores mathematical description of an image, we have to build a new image (make rasterisation) when zooming out. Thus, the vector format enables quality zooming on the client and provides more functionality and is preferable in that case.

## 4.6 SUMMARY

In order to bring the use of vector graphics on mobile terminals, WAP Forum and W3C announced a formal liaison relationship to define next-generation Web specifications that support the full participation of wireless devices on the World Wide Web. Together, the two organisations will face the technical challenge of mobile access to information on the Web. WAP Forum and W3C will coordinate on the future development of XML applications and in content adaptation through the use of vector graphics and style sheets. Instead of developing diverging sets of solutions, it is the intent of both groups to find common solutions that will address mobile requirements [WAPF99]. Therefore, it makes sense to put emphasis on the use of XML and vector formats in mobile environment.

In summary, the vector format is quite an attractive solution for personal navigation systems. It is closer to map encoding than the bitmap format, but requires more power from a mobile device for processing and visualisation. The strong advantage of the vector format is the ability to perform zooming without a quality loss. Also we have seen that for big scales the bitmap format is better. Partitioning of image for transfer does not give an advantage for either vector and bitmap formats.

The XML technology enables the development of flexible and expandable formats and XML is widely used for mobile terminals already [WML99]. Existing XML-based vector formats are just appearing, not yet standardised, and usually designed for graphical and multimedia applications. In the following chapter, we will discuss more detailed work schemes for different formats in mobile environment.

# 5 MOBILE LOCATION SERVICE ARCHITECTURE

In this chapter, we examine the relationship between the architecture of personal navigation systems and a data format we use to transfer information over a wireless network. Each of the formats provides its own level of functionality because of different internal information representation. Therefore, we obtain diverse functionality distributions and schemes of interaction between the mobile client and application server. In addition, if we use location service, interaction with a location server and information flow should be taken into account.

At the beginning, let us look at the conventional definition of the client-server architecture. According to Pressman [Pre97], there are five different configurations for software component allocation:

- *Distributed presentation.* In this rudimentary client/server approach, database logic and the application logic remain on the server, typically a mainframe. The server also contains the logic for preparing screen information, using software such as CISC. Special PC-based software is used to convert character-based screen information transmitted from the server into GUI presentation on a PC.
- *Remote presentation.* In this extension of the distributed presentation approach, primary database and application logic remain on the server, and data sent by server is used by the client to prepare the user presentation.
- *Distributed logic.* The client is assigned all user presentation tasks and the processes associated with data entry such as field-level validation, server query formulation, server file updates, client version control, and enterprise-wide application.
- *Remote data management.* Applications on the server create a new data source by formatting data that have been extracted from elsewhere. Applications allocated to the client are used to exploit the new data that have been formatted by the server. Decision support systems are included in this category.
- *Distributed databases.* The data comprising the database are spread across multiple servers and clients. Therefore, the client must support data management software components as well as application and GUI components.

The balance of the application component is distributed between the client and server based on the distribution that optimise the server and client configuration and the network that connects them.

M. Satyanarayanan [Sat96a] proposes in his work the extended client-server model in which the distinction between clients and servers may have to be temporary blurred (see Figure 5.1). The resource limitations of clients may require certain operations normally performed on clients to be performed sometimes on resource-reach servers. Conversely, the need to cope with uncertain connectivity requires clients to sometimes emulate the function of a server.

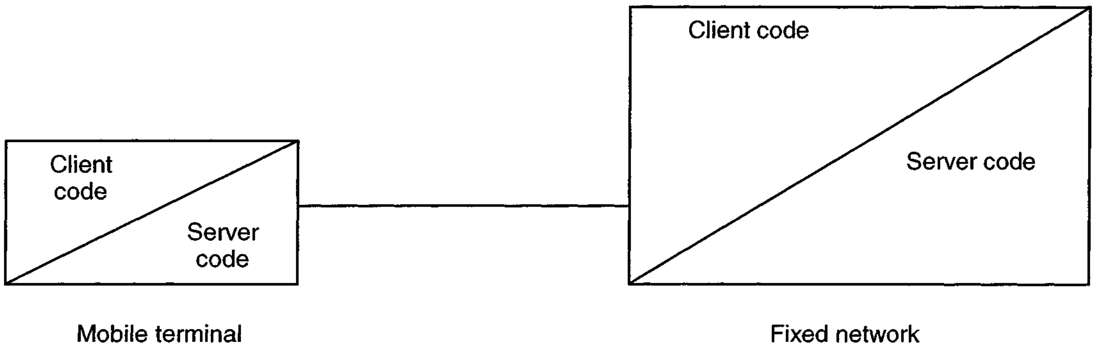



**Figure 5.1: Extended client-server model**

Now, we define types of operations that will be accessible in the navigation service. The functionality we would like to achieve consists of two groups:

- interface functions that are responsible for information displaying and interaction with the user and include image rendering, panning, and zooming;
- analysis functions that perform analysis of topological information and attributes of digital maps in order to provide such services as route optimisation, search of the nearest object and other ones.

Note that we have included into the interface functions only those ones that directly depend on the format type. For example, data input for queries or search is the responsibility of an application and is independent of the format type.

Let us compare the formats with two typical tasks. The fist one is the request for the map of the locality where the customer is located. The customer location is passed to the application server as a parameter. The aim of the fist task is to view the map or perform an interface function. The second task is to perform an analysis function. Note that the result of the second task would be represented either graphically as a map or as a sequence of text (voice) messages. The aim of the second task is analysis rather than the result representation.

Further, we need to determine which information the application should have in order to perform the above-defined functions. The image rendering and panning require just an image of the map in any graphic format. Preference could be given either to the bitmap format or to the vector one depending on particular requirements. If we want the system to choose automatically an appropriate part of the map, positioning of the mobile terminal is required. Self-positioning or positioning of a target mobile station is possible. In the last case, for example, the user could request the location of his/her friend and find the route to him/her. Above, we described zooming for the bitmap and vector formats. As discussed, mathematical description (vector model) is necessary for zooming. The vector or GIS formats could be used in order to support zooming. Finally, analysis functions require topological information or/and attributes. Attributes can be included in the vector format, but such feature is supported in weak form only in few formats. Thus, the GIS format is the most suitable and optimal variant. We have defined possible formats for each operation. Let us shed light upon relationship between these formats before we start defining the system architecture.

As we pointed out above, we could use three data formats for geographical map representation in personal navigation systems:

- Bitmap graphic format;
- Vector graphic format;
- GIS data format.

Figure 5.2 depicts the hierarchy of formats. We have defined the seniority based on facilities and expressiveness of formats with respect to the encoding of geographical information. The GIS format contains the full description of a digital map, geographical coordinates of all elements, relationships between them, and additional information in the form of attributes for each element. We can produce a graphic vector file from a GIS file defining what will be depicted (which layers on the GIS file, which part of the whole map), and in which form. The file will contain an image of the map encoded with geometrical primitives. Usually, the vector format contains a colour for each attribute. In turn, a bitmap file can be produced from the vector format. In that case, we calculate the array of pixels from the vector model for particular size.

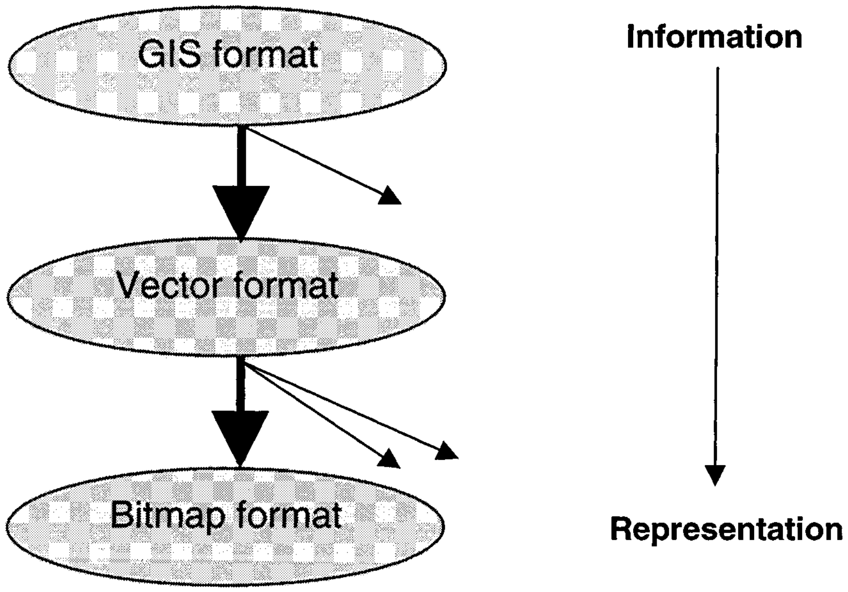



**Figure 5.2: Seniority of formats**

Each format flows out from another according to the seniority along loosing of functionality and additional features. If the bitmap and vector formats are intended for presentation purposes, the GIS format describes geographical data, which can be interpreted in many ways. Therefore, we will obtain different functionality for each format. Depending on the format used, some functions could be performed on the mobile terminal, some on the application server. Table 5.1 shows functional distribution for each format.

We have noted above the major components that form mobile environment. There are two major logical components, which participate in the work of the application. The first one is the application server, which has an access to the database with digital maps. The application server interacts with the client application on a mobile terminal partly through a wireless network. The application requests a service from the server. In reply, the application server provides description of available resources (e.g., types of map). After that, the user chooses necessary configurations of a digital map and sends a request to the server. At the end, the application server compiles the requested file and transfers it to the client. Later, the application may request additional services.

**Table 5.1: Function distribution for each data format.**

| Format | Server side | Client side |
|---|---|---|
| Bitmap | Preparation of the bitmap file which contains the image with necessary information, zooming, performing analysis functions. | Display a map, panning. |
| Vector | Preparation of the vector file which contains the image with necessary information, performing analysis functions. | Display a map, panning, zooming. |
| GIS | Only provide a file with needed information | Display a map, panning, zooming, analysis functions |

The second component is the location service server that, with the methods discussed above, provides positioning functions. According to the GSM specification [LCS00], all positioning mechanisms that were standardised by ETSI require interaction with a fixed network. Location information, obtained from the location server, serves as a parameter for the request to the application server and defines which part of the map is transferred to the client. Figure 5.4 shows the relationships between the major system elements. All interactions are performed through a wireless network. Therefore, they should be very effective and minimal. Further on, we will investigate the information flow for each type of the data formats among all elements.

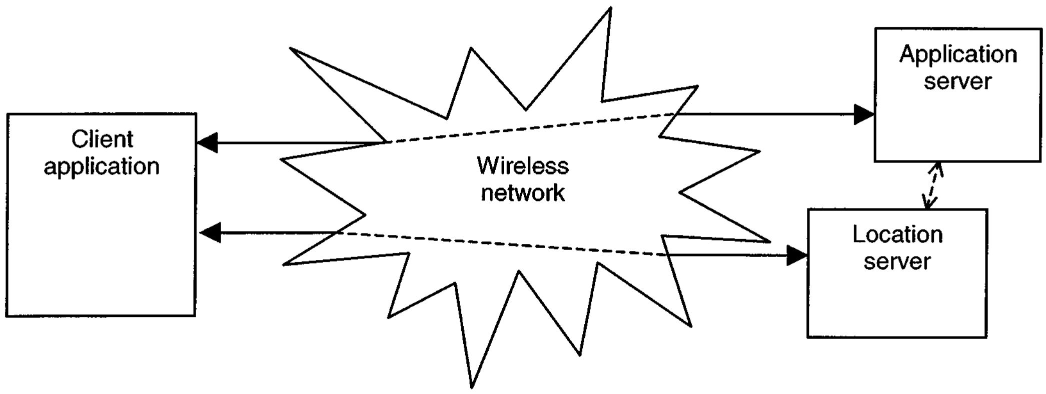



**Figure 5.3: Logical architecture view**

Now, we are able to describe the functions of all the system components based on the conclusion above. APPENDIX D consists of sequence diagrams for each format that show the interaction between the mobile terminal, the location server, and the application server during the tasks that were described above.

Figure D.1 for the bitmap format shows that all operations are performed at the application server. The bitmap format does not allow us to perform many functions at the terminal. Therefore, if we use the bitmap format to transfer information, almost all operations would be running at the server side. Hence, each time, when we need to update the image, a wireless network should be used.

The vector format allows us to perform all interface functions on a mobile terminal, but analysis functions still require calling the server (Figure D.2). For instance, if an adapted version of the SVG format which is able to store text description of graphical elements is used for mobile environment, a quite attractive variant can be obtained [SVG99]. In summary, the bitmap and vector formats enable the presentation type clients. All information is prepared by the server and the client only displays the result and supports the interface with a user.

The GIS format allows performing all functions on a mobile terminal. A user could choose the exact information needed and then the server would generate the file with the ordered information. We see that in the case of the GIS format in which almost all application functions are performed at the client side. In other words, the application server becomes just a data source, a map database. And, as Figure D.3 depicts, location information is not transferred to the application server in that case. It is advantageous to us for reasons of wireless network traffic and security. On the other hand, we can avoid the transfer of location information over a wireless network if the rights for positioning the mobile terminals are given to the application server. It is suitable in the case of the bitmap format and the vector format when the image is formed at the server side. The application server defines coordinates of terminals by itself.

The demand for resources on a mobile terminal increases when we move from the bitmap format to the GIS format but the activity required of the wireless network decreases. The client application should perform not only visualisation as in the case of the bitmap and vector formats but also analysis functions. Therefore, a client terminal should provide enough resources for search algorithms.

Table 5.2 depicts approximate comparative characteristics of the formats in the light of resources usage for described scenarios.

**Table 5.2: Comparison of different format types**

| Format | Frequency of network use | CPU power | File size |
|---|---|---|---|
| Bitmap | High | Low | Small |
| Vector | Low | Middle | Small |
| GIS | Low | High | Large (conventional formats) |

The wireless network latencies should be also taken into consideration. They are quite high for connection establishment in GSM networks. If the system have many sessions during a certain period, latencies could very slow down the work of application, especially, if it requests data from server quite frequently. However, this problem will not be relevant for forthcoming GPRS and UMTS any more.

Let us try to forecast which of the formats is the most appropriate for a geographical information service in mobile environment. According to Moore's Law, each new chip contains roughly twice as much capacity as its predecessor, and each chip is released within 18-24 months of the previous chip [Moo99]. But the bandwidth of a wireless network has a lesser pace of growth and physical limitations.

Figure 5.4 depicts approximately the future trend of growth both for processor power and network bandwidth in a mobile environment. The processor power increases along the vertical axe and the network bandwidth along the horizontal axe. Depending on the properties defined above for different schemes, we have put each data type on this graph. As we see, the GIS format is the most appropriate solution for mobile systems.

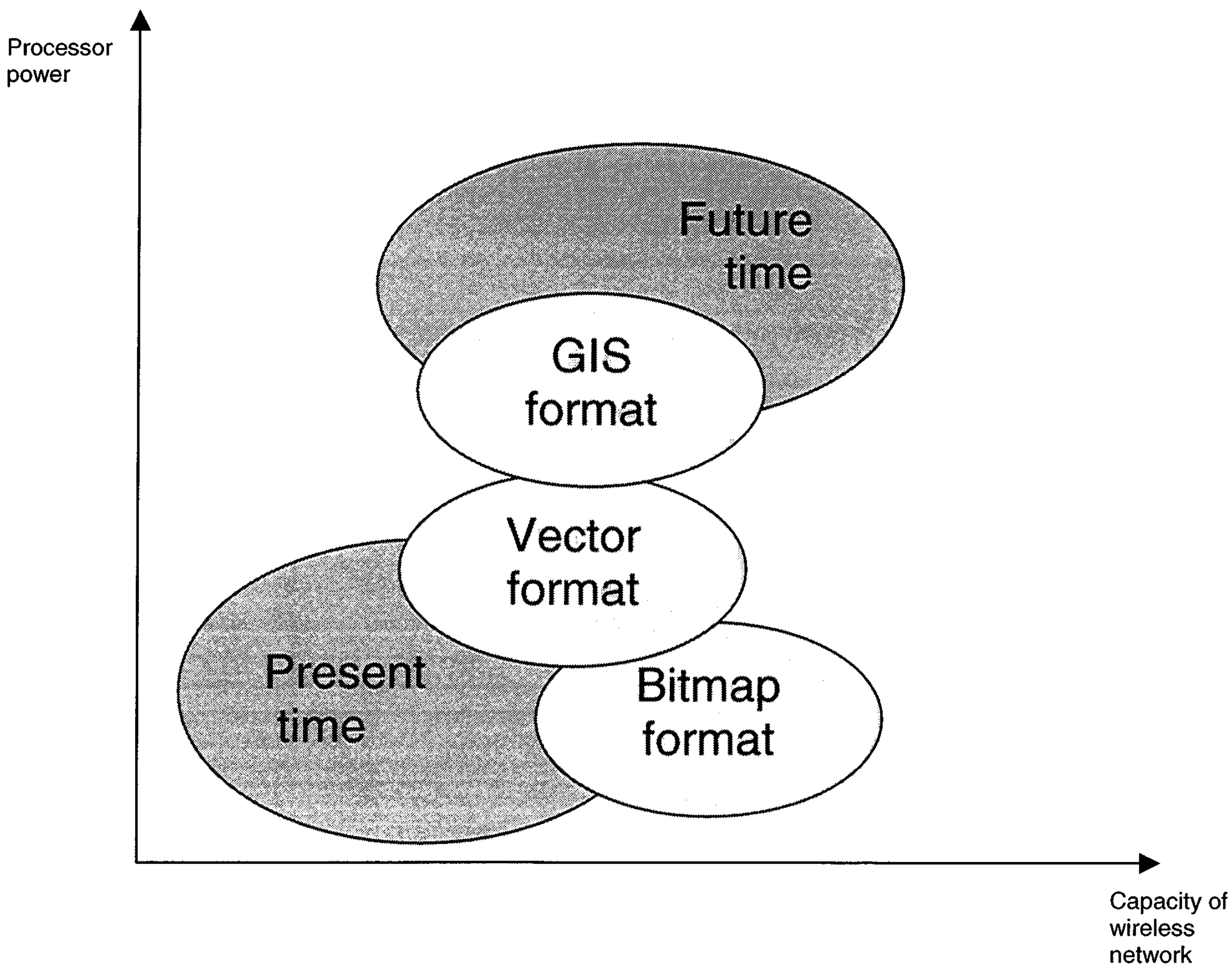



**Figure 5.4: Capacity trends for different formats**

Of course, memory capacity and the amount of local storage space are very important, besides the processor power. In order to store data locally without the network use, we should pay attention to the requirements for the storage capacity. The issue of data compression is arising here. Effective compression methods would help to decrease the requirements. We have already discussed the compressibility of the graphic formats. There are not special algorithms for geographic data compression designed. However, for example, if we took data in the quite verbose XML format, we can achieve even better compressibility than for the compression of original non-XML formats. It is because we can easy extract data of the same type and separately apply a compression algorithm for each type. Hence, compression methods work more effectively and the overall compressibility is higher [Lie00].

Another advantage of the GIS format is the ability to avoid the problem of disconnection. The more autonomous the mobile terminal, the better it can tolerate network disconnection. In environments with frequent disconnections, it is better for the mobile device to operate as a stand-alone computer then as a portable terminal [Fore94].

The problem is that all existent GIS formats were developed for workstations and intended for spatial analysis and map design. They contain many, in our case, unnecessary features and information is often distributed among several files. For instance, if we want to save information about points of interest with one of the widely used GIS formats, ESRI shape files, we will get three files: descriptions of geographical coordinates, an index file, and an attribute file. This is not important for workstation applications, but if we are going to transfer such files over a weak wireless network to a mobile terminal, we need a more appropriate and compact format that takes into account the limitations of wireless environment.

Also one factor that comes up in the mobile environment is the high variety of mobile devices. In the case of graphic formats, we could not be sure that any device would equally show the image with a specific resolution and colours. We can not be sure that some device would even show the image. Therefore, transferring geographical data allows performing visualisation for each device independently. Of course, we need a strong basis to design such a platform-independent format and avoid possible problems like bits order, number formats and others. XML technology can provide corresponding facilities. Although, the price for such flexibility is the increasing of file size because of the tags reiteration, compression could avoid this problem [Lie00].

In the next chapter, we analyse the requirements for the new format and attempt to design it.

# 6 FORMAT DEVELOPMENT

Based on the analysis of using different file formats, we have seen that one of the optimal variants is the GIS format. It is optimal if we would like to provide to the user a rich set of functions and to use the wireless network minimally. Therefore, we will attempt to develop such a format in this chapter.

We would like our new format to possess the following features:

- Interpretation in a wide range of mobile devices;
- Alternative ways for information representation (text, voice);
- Expandability;
- Structure flexibility.

XML-technology meets our requirements for the format development. The logical structure of the XML-based language is easy for program processing. Software can interpret the content of an XML file in an appropriate way for each device. XML is already widely used in the Internet and for WAP services (WML). Therefore, the new mark-up language could be easily integrated in current WWW/WAP services.

Firstly, we will analyse typical use cases of our format in order to specify its contents. We have to define the structure of data we want to describe by our format, the elements of this structure and relationship between them. Then, all entities and their attributes will be described. Based on this, an XML tag tree will be built and the attributes of each tag will be defined. At the end of the chapter, we discuss some topics, which are relevant to the problem of format implementation.

## 6.1 PROBLEM ANALYSIS

We apply the analysis of use cases in order to design the data model of our system [Gor98]. This model should take into account the geographical nature of our problem domain. Therefore, approaches from the conventional GIS analysis could be used. We will apply methods from coverage database design, which are used to develop a data structure for GIS applications. The analysis allows us to define all entities in our problem domain and relationship between them. Following steps are necessary to achieve a complete and effective data model [ARC94]:

1. Create the model of the work performed by the user for which 'location' is a factor.

2. Define and catalogue entities, geographic and related non-geographic features, identifying their attributes and relationships.

3. Determine the most effective spatial representation for the geographic features: point, line, or area.

4. Match the representation of each geographic feature to one of the coverage feature classes: point, arc, node, route-system, linear event, polygon, or region.

5. Organise geographic features into coverages. Coverages allow the grouping of similar geographic features. Certain relationship in the coverage must be honoured. It is also during this step that you will ensure that the attribute definitions are complete and annotation and lookup tables are defined.

Our major problem is to transfer data to a mobile user. We should make it as effective as possible and take into account all limitations of the mobile environment. Firstly, let us view the problem from a customer point of view.

We need to define who our customers are and what they need. According to Pressman [Pre97], the scenarios, often called use cases, provide a description of how the system will be used. We must define the types of people that will use the system. At first, we should note that it is very important to find out whether our user is a pedestrian or a driver, because each type of user needs corresponding information and services. For instance, if you are a pedestrian and do not use a car, you would not need to know the state of roads, traffic congestion and other information related to roads. On the other hand, we should take into account almost all road properties when route optimisation is being performed for a driver. Of course, the user can change his role during the trip. The organisation of profiles would help in such situations and change the client application settings depending on the current role.

Consider that there are two major modes, which correspond to the types of functions we defined early. The first one is viewing geographical information in the form of a map and the additional text description of objects; in other words, interface functions are performed. The second mode is a query to the system in order to define the location of some object in the city or find the route to some object. For the last mode, we perform analysis functions (for example, find shortest way to the airport or the nearest bus stop). Let us elaborate each scenario.

I. Map viewing:

   A. Connect to the server and perform authentication.

   B. Input your own location. There are two possible ways. If user's terminal has an access to location services, the system automatically would resolve the location. Otherwise, the user should enter his location by himself or choose the region to explore.

   C. Select the type of information needed (e.g. only a street network, or also restaurants, hotels, railway stations and so on).

   D. View the map while performing such operations as panning and zooming.

   E. Read relative information such as addresses, objects' names, and their descriptions.

II. Queries:

   A. Connect to the server.

   B. Input your own location.

   C. Input parameters of the query (for example, a full street address, or only street names, or the name of the object).

   D. View result(s) of the query. The most obvious way is to show results on the map, while performing such operations as panning and zooming. However, it is possible to display the results as the sequence of text messages that describes the path verbally.

   E. Read relative information (e.g. the length of a route, possible problems, or an approximate time on the route).

We see that there are roughly three kinds of necessary data: data for visual map representation, data for analysis, and descriptive data related to the objects. Therefore, the entities should include three groups of attributes.

Further, let us define all these entities. The main object, we deal with, is a city, which lies on top of the entities hierarchy (see Figure 6.1).We need a very simple form of map representation at the beginning stage. Thus, a street network will be the basis of city representation, the frame of the new format. A street network will serve for two functions:

- streets representation;
- network analysis.

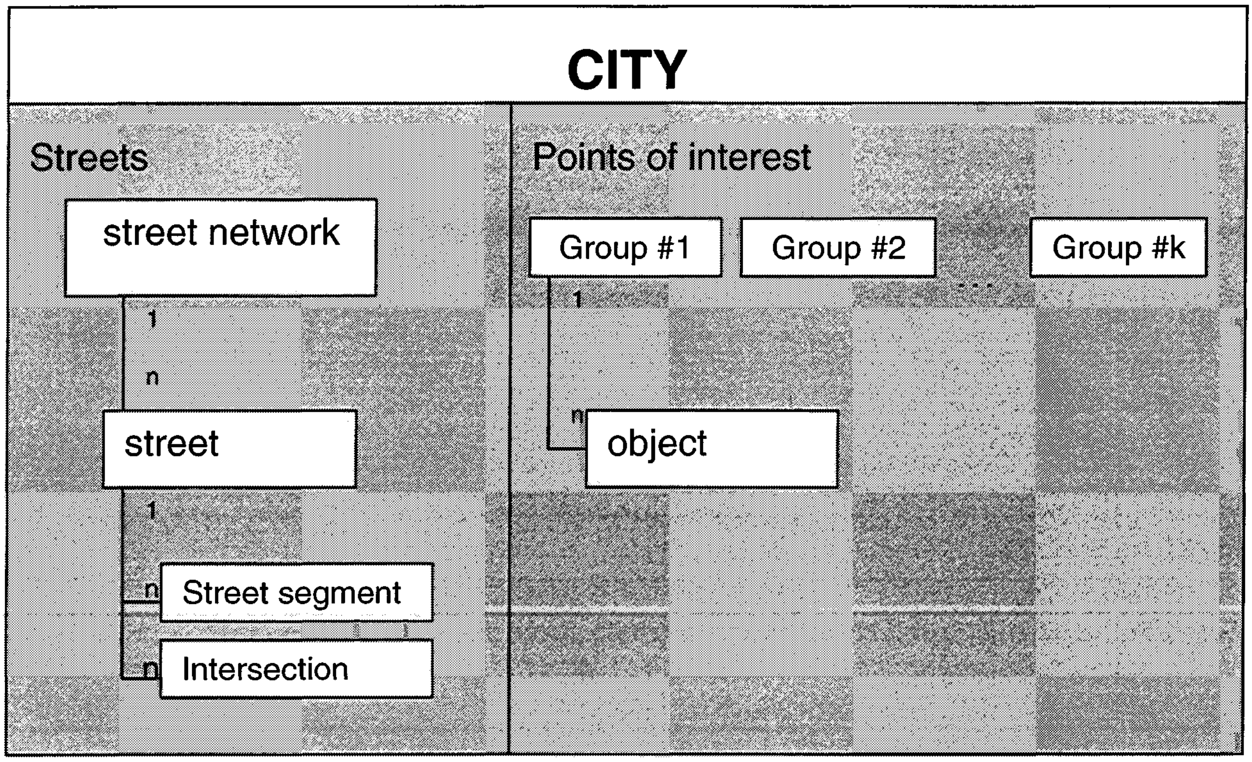



**Figure 6.1: City components for graphical representation**

A street network includes intersections and the road segments between them. Together, they form the framework for street definition. For example, a collection of road segments forms a street. Since addresses are based on the street network, features that include an address as part of their attributes can be related to the street network. A street segment is the stretch of roadway between two intersection nodes. Rather than naming each individual street segment, a street name is given to a collection of street segments defined as a street.

Many commercial and government databases store street names and address information for each street segment which become arcs when converted. These rich attribute databases are very useful during coverage automation and update but they are not useful for us when we use this information in the form of a file and process it on a mobile terminal. These attributes are just one of many possible ways for optimisation.

Because the format should contain information about the relationship between streets and nodes for network analysis, the optimal model is the arc-node topology, which was examined in the previous chapters. It allows the keeping of topological information of the street network and does it in an effective way.

Further, the next class of entities is the points of interest. These are various city objects like restaurants, hotels, shops, hospitals, schools, airports and so on. Usually, these places are destinations for people. It is their principal distinguishing feature that we can mark them on a map with a plot or some point sign. It seems reasonable that the use of building footprints is better for such small scale value. However, the problem lays in the lack of suitable geographical data. It is difficult enough to find the proper source of the information. Instead of footprints, it is more effective to apply a point or some simple geometrical shapes for marking objects for example, a rectangle, a circle, or a triangle. It is possible to implement it in such a way that the area of geometrical shapes will relate with the real area of buildings.

There are two ways to calculate the place on the map for object visualisation:

- using precise coordinates of the object;
- using the address, its street name and number, for approximate determination of the location. In this case, we can not store coordinates for each object and calculate them "on-the-fly" when a new image of the map is built. Unfortunately, the coordinate accuracy is decreased for this method. For example, the result will still be satisfactory for a modern area of a downtown, where the size of buildings is meanly uniform. But for old areas of a city where buildings have been destroyed and built during many years and have the huge range of a footprint area, the result would be significantly worse.

Taking into account the small scale of maps, let us consider using the first method. The number of objects is not too large (in comparison, for instance, with the number of streets) and the benefit is quite small. We simplify the algorithm of map building and save the resource on a mobile terminal.

Thus, we have different city objects. Each of these objects has its own name, address, description and geographical coordinates. Objects are organised into groups (layers) depending on their type for better logical structure and optimal work.

And at the end, the last entity is the user, the owner of a mobile device. There are two important features that should be considered: the type of a user (a driver or a pedestrian) and the user's coordinates. Coordinates are dynamic data and they will be obtained from location services at a real time. Therefore, this entity is excluded from the format and will be managed by an application. A user type allows the proper performance of queries and analysis.

Table 6.1 summarises high-level entities with their possible attributes and geometrical primitives for graphical representation on the map.

**Table 6.1: Identified entities and their geographic features**

| Entity | Geographic feature | Attributes | Representation |
|---|---|---|---|
| Street Network | Streets,<br>segments,<br>intersections | Name, type,<br>condition, address range, lanes,<br>type, traffic control devices | Line,<br>line,<br>point |
| Points of interest | Object location | Type, name, address, description | Point |
| User | User location | Type | Point |

Below we elaborate all entities in the hierarchy. Each street is defined by the attributes, which are described in Table 6.2, and contains the set of next level elements, segments.

**Table 6.2: Street attributes**

| Attribute name | Data type | Description | Comments |
|---|---|---|---|
| ID | Alphanumeric code | Identification code | |
| Type | String (predefined set of values) | Type of street | Each type of street has own graphical representation. We should take into account type of streets for analysis. |
| Name | String | Street name | |
| Sequence of segments | Set of segment entities | Sequence of segments which form street | |

The segment of a street has more specific attributes, which include an address range, type, direction, and lengths. Table 6.3 depicts these attributes.

## Table 6.3: Segment attributes

| Attribute name | Data type | Description | Comments |
|---|---|---|---|
| ID | Alphanumeric code | Identification code | |
| Direction | String, (predefined set of values) | Type of street traffic | It's very important for path search. |
| Lengths | Numeric value | Length of the street segment | |
| Left lower number of buildings;<br>left upper number of buildings;<br>right lower number of buildings;<br>right upper number of buildings. | Numeric value | Address range | These values represent the range of building numbers on both sides of a segment |
| From-node | Pointer to the node entity | The number of from-node (a intersection) | |
| To-node | Pointer to the node entity | The number of to-node (a intersection) | |
| Sequences of vertices | Set of vertex entities | The sequence of vertices which form one segment | Vertices are needed if segment is not a straight line and have a complex form |

Further, we describe the node attributes which form the basics for the street network. Actually, nodes are only used for visualisation because they contain geographical coordinates. Table 6.4 shows attributes for the node.

**Table 6.4: Node attributes**

| Attribute name | Data type | Description | Comments |
|---|---|---|---|
| ID | Alphanumeric code | Identification code | |
| Type | String, (predefined set of values) | Define type of points. There are three types: vertex, from-node, to-node. | |
| Coordinates | Coordinate | Geographical coordinates of a node | We should define the type of coordinate system |

Thus, we have described all components, which form the street network, the backbone of our format. A number of street networks could exist for one city. We can divide them by city regions, especially for large cities, in order to reduce network traffic, because it is unlikely that a user would need a full city map. Further, we will describe components for points of interests. As was noted above, all points of interest are divided into groups by their type. Table 6.5 shows the description of a group.

## Table 6.5: Group attributes

| Attribute name | Data type | Description | Comments |
| --- | --- | --- | --- |
| ID | Alphanumeric code | Identification code | |
| Name | String | The name of a group | It means that the name will reflect the type of objects, which are grouped (for instance, hotels, restaurants, etc.) |
| Set of objects | Set of object entities | Sequence of objects which form a group | |

Now, we can define the object entity, which contains description of a particular object. Note that objects from different groups can be quite different and it is difficult to define common attributes, which fit for all types. Therefore, we will describe very general attributes, but new attributes can be added for particular applications. Table 6.6 shows a possible set of attributes.

We have now described the main attributes of entities. Of course, practical implementation would show the optimal set of entities and attributes, these are just an initial version.

## Table 6.6: Object attributes

| Attribute name | Data type | Description | Comments |
|---|---|---|---|
| ID | Alphanumeric code | Identification code | |
| Name | String | The name of an object (restaurant name, hotel name, the number of a school and so on) | This attribute is depending on an object type. |
| Address | Address | Street address of an object includes the street name and number. | It's very important for path search. |
| Description | String | Text description of an object. | For instance, if it is a restaurant text description can include menu, prices, open hours and other necessary information. |
| Coordinates | Coordinates | Geographical coordinates of an object | We should define the type of coordinate system |

## 6.2 FORMAT DESIGN

We have elaborated the structure of data that will be used for geographical information service. The next step is to define XML tags for each unit of this structure. In other words, we have to transform the model obtained into an XML tags tree based on the hierarchy of city entities (Figure 6.1) and attributes we defined above [Abi99]. Now, we are able to define necessary XML tags and their attributes. Figure 6.2 depicts the tree of XML tags, which were obtained from the analysis model in the previous section. The tree includes two main branches that correspond to the street network and groups of points of interest. High-level tags group information about streets, which are defined with the arc-node topology, and information about points of interest, which are defined with the node topology. The typical document will include one block with street network information and several blocks for each type of points of interest. These tags also can be used as analogue of layers or coverages in the context of conventional GIS. We will refer to our format as GMML (Geographic Mobile Markup Language).

Let us pass on to the next level of the street network branch. As was noted above, each street is described by its name and type and it contains a sequence of segments. Further, each segment is defined by to-node, from-node and vertices according to the arc-node topology. Thus, all location information is included only in nodes and vertices tags. If we directly store information in such a way, some nodes would be described two times in different streets with a common intersection. Therefore, all nodes and vertices are described in the independent block marked by the <points> tag. Now, when a segment has been described, only references to nodes are used. In such a way, we also separate off geometrical information, and all values of coordinates. If we remove this block from the street network block, descriptive attributes and topological relationships between streets would remain intact. In that way, the system can provide to some clients only topological and descriptive information without unnecessary and bulky geometrical data. Such clients could produce results in the text form without graphical visualisation.

As was noted above, all city objects are partitioned according to their type. One peculiarity is that we can use references to streets from the street network section for street description in the same way as references to nodes are used for segment description. Depending on specific needs, we can easily extend the object tag and add some extra sections such as telephone numbers, electronic addresses, and other useful information.

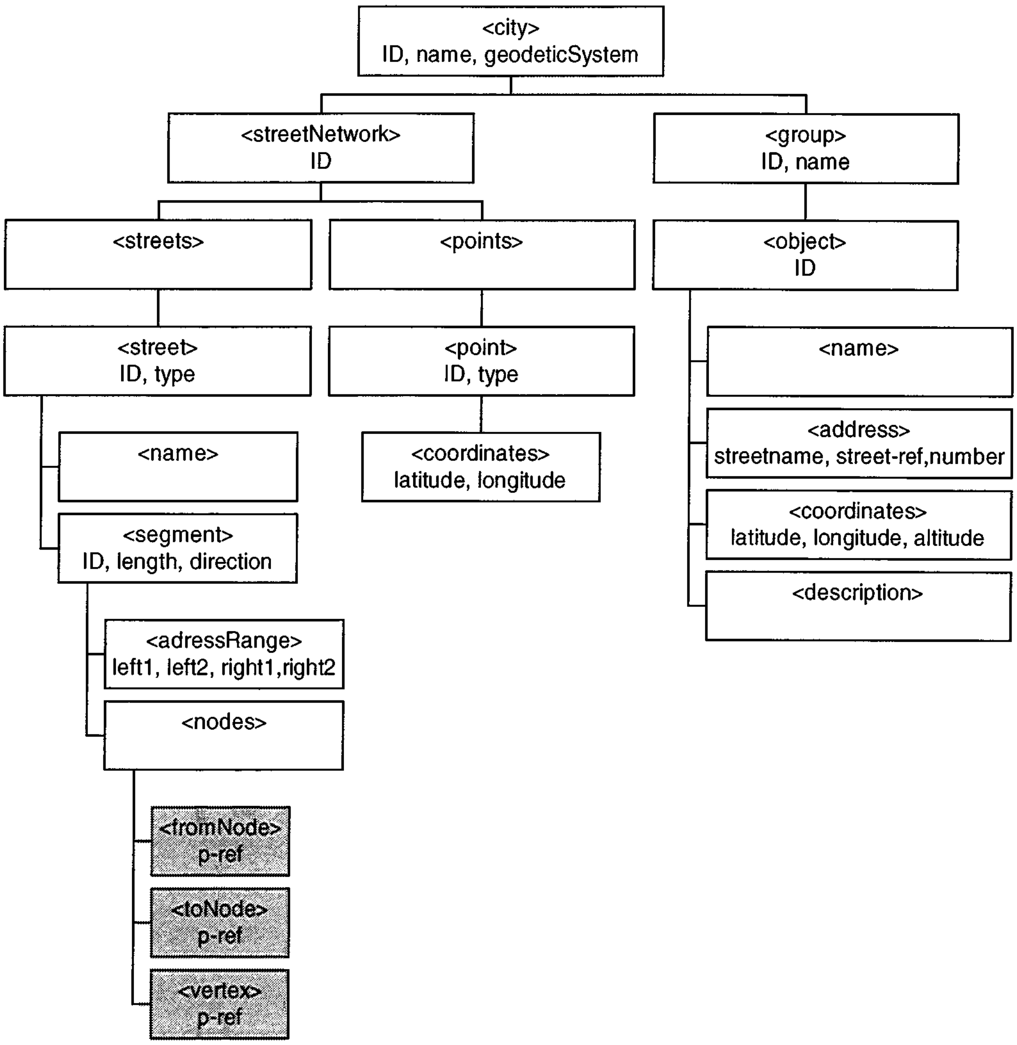



**Figure 6.2: XML tree**

## 6.3 TAG DESCRIPTIONS

This section contains description of all the tags we defined above. The description of each element includes an outline of a tag, its attributes, content description, and references to child elements. APPENDIX B contains DTD of our format. An example of a simple XML file that contains description of three streets, two shops and two hotels is shown in APPENDIX A.

### 6.3.1 CITY ELEMENT

The city element is the root element in any GMML document and must be described. The element shows that document contains data of some city.

The city element has the following attributes:

- id – identification code of the city.
- name - name of the city.
- geodeticSystem – the geodetic system which is used to specify geographical coordinates of all elements in the document. The values of latitude and longitude in the coordinates element are represented in the reference frame of this geodetic system. Only WGS 84 geodetic system is supported at the initial stage [WGS84]. Thus, the single default value is "wgs84".

The city element has the following child elements:

- streetNetwork - The streetNetwork element is defined in section 6.3.2. The streetNetwork element does not need to be described or can be described any number of times.
- group - The group element is defined in section 6.3.14. The group element does not need to be described or can be described any number of times.

These child elements must be described in the order: streetNetwork, group.

### 6.3.2 STREETNETWORK ELEMENT

The streetNetwork element is the child element of the city element and does not need to be described or can be described once. The streetNetwork element specifies a street network of some city region or the city street network. Separate street networks can exist for drivers and pedestrians. The street network is described with arc-node topology. Therefore, it includes the description of nodes and description of streets which refers to nodes description. Because the nodes description contains geographical coordinates, they are used for map visualisation. If an application does not require visualisation, the nodes description could be omitted.

The streetNetwork element has the following attribute:

- id – identification code of the street network.

The streetNetwork element has the following child elements:

- streets - The streets element is defined in section 6.3.3. The streets element must be described once.
- points - The points element is defined in section 6.3.12. The points element does not need to be described or can be described once.

These child elements must be described in the order: streets, points.

### 6.3.3 STREETS ELEMENT

The streets element is a child element of the streetNetwork element and must be described once. The streets element is a container tag, which contains the set of streets that form the street network.

The streets element has no attributes.

The streets element has the following child element:

- street - The street element is defined in section 6.3.4. The street element must be described once or more times.

### 6.3.4 STREET ELEMENT

The street element is the child element of the streets element and must be described once or more times. The street element specifies a street by describing street segments which form the street.

The street element has the following attributes:

- id – identification code of the street;
- type – type of the street.

The street element has the following child elements:

- name - The name element is defined in section 6.3.5. The name element must be described once.
- segment - The segment element is defined in section 6.3.6. The segment element must be described once at least.

These child elements must be described in the order: name, segment.

### 6.3.5 NAME ELEMENT

The name element can be the child element of the street element or the object element and must be described once. The name element specifies the name of a street or some object.

The name element has no attribute. The name element can contain any sequence of characters permitted by the XML specification [XML98].

### 6.3.6 SEGMENT ELEMENT

The segment element is the child element of the street element and must be described once or more times. The segment element specifies a part of the street between intersections.

The segment element has the following attributes:

- id – identification code of the street;
- length – length of the segment;
- direction - possible driving/travelling direction.

The segment element has the following child elements:

- addressRange - The addressRange element is defined in section 6.3.7. The addressRange element can be described once.
- nodes - The nodes element is defined in section 6.3.8. The nodes element must be described once.

These child elements must be described in the order: addressrange, nodes.

### 6.3.7 ADDRESSRANGE ELEMENT

The addressRange element is the child element of the segment element and can be described once. The addressRange element specifies address ranges on the both sides of the street.

The addressRange element has the following attributes:

- left1 – lower number on the left side;
- left2 – upper number on the left side;
- right1 – lower number on the right side;
- right2 – upper number on the right side.

The content of all attributes is integer numeric value. The addressRange element has no contents.

### 6.3.8 NODES ELEMENT

The nodes element is the child element of the segment element and must be described once. The nodes element is the important element for the street network because it connects the streets definition with the nodes definition. It helps to define common intersections among the streets for analysis. The nodes element specifies the references to the nodes, which form the segment. Therefore, all referred nodes must be defined in the points element with appropriate identification codes.

The nodes element has no attribute.

The nodes element has the following child elements:

- fromNode - The fromNode element is defined in section 6.3.9. The fromNode element must be described once;
- toNode - The toNode element is defined in section 6.3.10. The toNode element must be described once;
- vertex – The vertex element is defined in section 6.3.11. The vertex element can be defined once or more times.

These child elements must be described in the order: fromNode, toNode, vertex.

### 6.3.9 FROMNODE ELEMENT

The fromNode element is the child element of the nodes element and must be described once. The fromNode element specifies the from-node according to the arc-node topology and refers to the appropriate node definition.

The fromNode element has one attribute:

- p-ref – reference to the point element that define the from-node for the segment. The appropriate definition must be located in the same GMML document inside the points tag.

The contents of the p-ref attribute are a defined alphanumeric code permitted by XML specification [XML98].

### 6.3.10 TONODE ELEMENT

The toNode element is the child element of the nodes element and must be described once. The toNode element specifies the to-node according to the arc-node topology and refers to the appropriate node definition.

The toNode element has one attribute:

- p-ref – reference to the point element that define the to-node for the segment. The appropriate definition must be located in the same GMML document inside the points tag.

The contents of the p-ref attribute are a defined alphanumeric code permitted by XML specification [XML98].

### 6.3.11 VERTEX ELEMENT

The vertex element is the child element of the nodes element and can be described once or more times. The vertex element specifies the vertex, which forms the segment, and refers to the appropriate node definition. Vertices are necessary because a segment can not have the straight form and have to be described with more than two points.

The vertex element has one attribute:

- p-ref – reference to the point element that define the vertex for the segment. The appropriate definition must be located in the same GMML document inside the points tag.

The contents of the p-ref attribute are a defined alphanumeric code permitted by XML specification [XML98].

### 6.3.12 POINTS ELEMENT

The points element is the child element of the streetNetwork element and must be described once. The points element is a container tag, which contains the set of points that form the street network.

The points element has no attribute.

The points element has the following child element:

- point - The point element is defined in section 6.3.13. The point element must be described once or more times.

### 6.3.13 POINT ELEMENT

The point element is the child element of the points element and must be described once or more times. The point element specifies either a vertex or a node.

The point element has the following attributes:

- id – identification code of the point;
- type – type of the point. There are three possible values: fromNode, toNode, and vertex.

The point element has the following child element:

- coordinates - The coordinates element is defined in section 6.3.17. The coordinates element must be described once.

### 6.3.14 GROUP ELEMENT

The group element is the child element of the city element and does not need to be described or can be described once. The group element specifies a group of similar points of interest. A GMML document can have several groups, which can be displayed in any combinations depending on the user preference.

The group element has the following attribute:

- id – identification code of the group;
- name – name of the group. For instances: hotels, restaurants, schools, university buildings.

The group element has the following child element:

- object - The object element is defined in section 6.3.15. The object element can be described once or more times.

### 6.3.15 OBJECT ELEMENT

The object element is the child element of the group element and can be described once or more. The object element describes a point of interest.

The object element has the following attribute:

- id – identification code of the point of interest.

The object element has the following child elements:

- name - The name element is defined in section 6.3.5. The name element must be described once.
- address - The address element is defined in section 6.3.16. The address element must be described once.
- description – The description element is defined in section 6.3.18. The description element can be described once.
- coordinates – The coordinates element is defined in section 6.3.17. The coordinates element must be described once.

These child elements must be described in the order: name, address, coordinates, description.

### 6.3.16 ADDRESS ELEMENT

The address element is the child element of the object element and must be described once. The address element specifies the street address of the point of interest. It can include the reference to the street element from the street network.

The address element has the following attributes:

- streetName – the name of the street on which the point of interest is located.
- street-ref – reference to the street element . The appropriate definition must be located in the same GMML document inside the streets tag.
- number – the number of a building on the street.

### 6.3.17 COORDINATES ELEMENT

The coordinates element is the child element of the object element or the point element and must be described once. The coordinates element specifies the geographical coordinates of the point on the Earth. The values of attributes depend on the geodetic system specified in the city element.

The coordinates element has the following attribute:

- latitude – the latitude of the point;
- longitude – the longitude of the point;
- altitude – the altitude of the point.

### 6.3.18 DESCRIPTION ELEMENT

The description element is the child element of the object element and can be described once. The description element gives a brief description of the point of interest.

The description element has no attribute. The name element can contain any sequence of characters permitted by the XML specification [XML98].

## 6.4 INTEGRATION OF THE FORMAT IN MOBILE ENVIRONMENT

In the previous section, we have described XML format, which can be used for storing geographical information for personal navigation systems. In this section, we will discuss the ways of implementation and using the GMML format and problems that can arise during integration with existing systems. There are two important topics. The first one is the combination of the GMML document and other XML documents, especially, WML documents. The second topic is the processing of GMML documents in WAE, at the client side. In short, what are possible work schemes among standard user agents?

GMML is defined using XML; therefore, it can be used as a standalone, or, according to the XML specification, as an XML namespace with other grammars. GMML is intended to be used as one component in a multi-namespace XML application. This multiplies the power of each of the namespaces used, to allow innovative new content to be created. For example, GMML maps may be included in the document that uses any text-oriented XML namespace - including WML. Figure 6.3 gives an example of WML file with GMML code. We need to use the GMML prefix to avoid conflicts due to the same names in WML and GMML. Similarly, VML is used inside HTML documents with prefix "v" [VML98].

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="c1" title="1 card">
<DO TYPE="ACCEPT" LABEL="NEXT">
<GO URL="#Second_Card">
</DO>
Select <B>Next</B> to display the map.
</CARD>
<CARD>
<GMML:city id='c01'>
<GMML:streetNetwork id='sn01'>
<GMML:streets>
<GMML:street id='s01' type='Major road'>
...
</object>
</group>
</city>
</CARD>
</WML>
```

**Figure 6.3: Example of use GMML with namespace**

Unfortunately, it is impossible to implement in practice for the current version of WAP standard. As was noted early, the binary code is used to decrease the size of WML files and the standard defines corresponding tables of encoding. Therefore, we can not easily extend WML and add our new tags; in addition, we need to define corresponding binary codes for these new tags. After that, software will be able to process the combined files correctly. This example shows us the weak side of binary encoding. Modification of WML is an extremely hard task. One of the possible solutions is dynamic building of binary encoding table for a particular DTD. Of course, the software will be more complex, but modification of existing DTDs will be much easier.

Thus, if we used GMML with existing systems, we would need to transfer our documents separately, with own files. In that case, we could use our own binary encoding table for GMML. Consequently, the WML document should contain a reference to the GMML document. We hope that future versions of WML will provide such a feature for embedding external objects in WML documents.

Further, let us investigate the architecture of applications which will use the GMML format. Because our format is intended for use on diverse mobile terminals, applications will run in mobile environment. We concentrate on WAP environment. There are two ways to use the format.

The first one is a standalone application, which will use WAP to download files and display their content and perform analysis functions; in other words, the application will use only the protocol stack. Thus, we simplify application because it does not need to interact with other modules at run-time. However, we ourselves need to organise the user interface, the interface to the transport protocol, provide access to the functions of location service, and other particular tasks. Such an application will be difficult to integrate as a general information service.

The second type is a fully functional user agent, which will provide a set of functions to other agents, including the micro-browser. As was noted above, WAE specifies user agents for the two primary standard contents: encoded WML and compiled WMLScript. In order to process our GMML files, we need our own user agent that will interpret GMML code content and perform additional analysis functions such as an optimal patch search, search of nearest objects and so on. This would allow us to build flexible information services using WML pages for the user interface. In addition, the user agent would not take care of retrieval files. The WML user agent will directly pass the necessary code to the GMML user agent. In replay, we can display the map if the micro-browser provides an appropriate mechanism or we can generate a bitmap image, any vector image, or just text description. Because GMML does not define any visual attributes, clients are free to decide how contents will be displayed. The WAE specification does not define interfaces between user agents, it is the responsibility of developers. In practice, we need to develop an application for a particular micro-browser. However, the interface to the GMML user agent could be provided via the WMLScript library, as was done with the WTA service [WTA99]. At the present time, there are still many open questions in this field and experiments are necessary to evaluate diverse variants.

Also note, that use of GMML format in the mobile environment only makes sense if used with compression because XML encoding requires additional space for tags and their attributes. As we have said, binary code is the simplest way to reduce size. In that case, all tags and attributes are replaced by code corresponding to the table. After that, we could use general compression methods. However, these methods do not take into account the nature of transmitted information, and the structure of data. Although experiments with vector formats have shown that general methods are quite effective, special methods for XML documents compression could be applied. Xmill is an example of the tool for compression/decompression of XML data, which incorporates existing compressors in order to compress heterogeneous XML data. It achieves about twice the compression rates of general-purposes compressors (gzip), at about the same speed [Lie00]. The main difference is that we have the DTD. In other words, we have information about the document structure. We can convince ourselves that after a particular tag (or its binary representation) there is a limited number of other possible tags or, for instance, text data. Thus, we need much fewer bits to encode the sequence of tags. Work in this direction is being conducted but no common solution is available as far as we know.

# 7 CONCLUSION

WAP protocols and emerging location services provide a good platform for a new class of location-depended applications. Such a system could provide safety, personal assistance services and geographical location services. One of the problems that arise during application development is in finding an appropriate data format for geographical information transfer to a mobile terminal, especially, in case of geographical digital maps. Any particular choice will affect both overall application architecture and functionality of the client side.

In this work, we have analysed work process and information workflow for three types of format. The use of the bitmap format is characterised by a strong server-centric architecture, intensive network traffic, and low requirements for client hardware. The algorithm for bitmap image displaying is elementary and does not require many resources. Necessity to send text description of points of interest makes software more complex and decreases network traffic. The vector format with embedded text descriptions is significantly better for map representation, wireless network is used much less, and location information is transferred only for search queries. But requirements for a mobile station are higher because the visualisation algorithm is more complex in that case. One of the possible candidates is the SVG format that is in development at the World Wide Web Consortium.

The GIS format is the most efficient choice to avoid a frequent use of the weak wireless connection. A customer needs only to download the file that contains information about the necessary region and requested points of interest. This procedure could be performed automatically with the help of a location service. Of course, such a solution demands more computational power and memory to process the file than for a simple bitmap format. At the same time, it makes the customer more independent from a wireless connection and provides a wider range of facilities.

Finally, in order to achieve the trade-off between functionality on the client side and 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 client software can represent contents in more appropriate way on a particular device.

Further studies and experiments are needed to get more exact estimates about the format's effectiveness for different types of mobile devices and about integration in the WAP environment. Different compression methods should also be tested to define a more appropriate algorithm.

## REFERNCES

[Abi99] Abiteboul S., On Views and XML, Proceedings of the 18 ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, USA,1999.

[AGI96] AGI GIS Dictionary, The University of Edinburgh and Association for Geographic Information, 1996, at URL address http://www.geo.ed.ac.uk/agidict/.

[Agr99] Agrawl P., Famolari D., Mobile Computing in Next generation Wireless Networks, Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications, USA, 1999

[ARC94] ARC/INFO Data Management, ESRI, Redlans, USA, 1994.

[Bha97] V. Bharghavan, "Challenges and Solutions to Adaptive Computing and Seamless Mobility over Heterogeneous Wireless Networks", IEEE Personal Communications Magazine, Vol. 4, March 1997.

[Ben99a] Benefon, arboreal and geodata introduce the first mobile map service in the world in telecom 99, Benefon Oyj, October 8, 1999, at URL address http://www.benefon.com/pressinvestors/pressroom/1999/mapgb.html.

[Ben99b] Benefon introduces world's first GSM+GPS personal navigation phone, the benefon ESC!, Benefon Oyj, October 1999, at URL address http://www.benefon.com/pressinvestors/pressroom/1999/ escgbrelease.html.

[BXML99] Binary XML Content Format Specification, WAP Forum, June 16, 1999, at URL address http://www.wapforum.org/.

[CIT99] The world's first mobile city guide launched in Stockholm, Telefonaktiebolaget LM Ericsson, February 24, 1999, at URL address www.citykey.com.

[CGM96] MIME type CGM, 1996, see at URL address ftp://venera.isi.edu/in-notes/iana/assignments/media-types/image/cgm

[CPS99a] CPS Forecasts 5 Meter Accuracy for E-OTD Location Technology in UMTS, press release, 1999, see at URL address http://www.cursor-system.com.

[CPS99b] CPS Publishes Test Results at International Standards Body Meeting, press release, 1999, see at URL address http://www.cursor-system.com.

[Dana99] Dana P.H., Global Positioning System Overview, 1999, see at URL address http://www.utexas.edu/depts/grg/gcraft/notes/gps/gps_f.html.

[DrML98] DrawML Specification, Submission to the World Wide Web Consortium, W3C, December 03, 1998, see at URL address http://www.w3.org/TR/1998/NOTE-drawml-19981203.

[Eri98] WAP-The wireless application protocol, Ericsson Review No.4, 1998, see at URL address www.ericsson.com/review.

[Eri00] Ericsson Mobile Positioning System, 2000, see at URL address http://www.ericsson.se/wireless/products/mobsys/gsm/subpages/prod/positioning.shtml.

[FCC99] Third Report & Order, Federal Communications Commission, 1999, see at URL address http://www.fcc.gov/Bureaus/Wireless/Orders/1999/fcc99245.pdf.

[For94] G. H. Forman, J. Zahorjan, "The Challenges of Mobile Computing," *IEEE Computer*, April 1994.

[Fla98] The Flash Open File Format White Paper, see at URL address http://www.macromedia.com/software/flash/open/spec.

[GLONASS] The Ministry of Defence of the Russian Federation Coordination Scientific Information Center Homepage, see at URL address http://www.rssi.ru/SFCSIC/SFCSIC_main.html.

[Glos98] Glossary of GIS Terms, Environmental Systems Research Institute, July 17, 1998, at URL address http://www.esri.com/library/glossary/glossary.html.

[Gor98] Gordillo S., Balaguer F., Refining an object-oriented GIS design model: topologies and field data, Proceedings of the 6th international symposium on Advances in geographic information systems, USA, November 1998.

[GPS96] NAVSTAR GPS USER EQUIPMENT INTRODUCTION, US Coat Guard Navigation Center, September 1996, see at URL address http://206.65.196.30/gps/geninfo/gpsdocuments/gpsuser/gpsuser.pdf.

[GPS00] THE INTERAGENCY GPS EXECUTIVE BOARD, see at URL address http://www.igeb.gov/

[Harr99] Harry T. Uitermark, Peter van Oosterom, Nicolaas Mars, and Martien Molenaar, Ontology-based Geographic Data Set Integration, Proceedings of Workshop on Spatio-Temporal Database Management STDBM'99, September 1999.

[Huf52] Huffman D.A., A Method for the Construction of Minimal Redundancy Codes, *Proceedings of the Institute of Radio Engineers*, Vol. 40, No. 9, pp. 1098–1101, September, 1952.

[Imi93] Imielinski T., Badrinath B. R., Data Management for Mobile Computing, In SIGMOD RECORD, March 1993.

[Imi99] Imielinski T., Navas J. C., "Geographic Addressing, Routing, and Resource Discovery with the Global Positioning System", *Communications of the ACM Journal*, 1999.

[ITS99] ITS Hand book, 1999, see at URL address http://www.nihon.net/ITS/index/indexHBook.html.

[Jin99] Jing J., Helal A., Elmagarmid A., Client-Server Computing in Mobile Environments, in ACM Computing Surveys, Vol. 31, 1999

[Kat94] Katz R.H., "Adaptation and Mobility in Wireless Information Systems," *IEEE Personal Communications Magazine,* 1994.

[Lass99] Lassen LP GPS, see at URL address http://www.trimble.com/products/catalog/oem/lassenlp.htm.

[LCS00] Digital cellular telecommunication system (Phase 2+); Location Services (LCS); (Functional description) – Stage 2. GSM 03.71 version 7.3.0, ETSI, 2000.

[Lie00] Liefke H., Suciu D., XMILL: An Efficient Compressor for XML Data, In Proc.: SIGMOD Conference 2000, Vol. 29, pp. 153-164, 2000

[Mat00] Matt Toon, GIS 2000 - What's Next?, GeoEurope, February 2000

[MIP00] Perkins C., Johnson D., Mobility Support in Ipv6, inernet draft, see at URL http://search.ietf.org/internet-drafts/draft-ietf-mobileip-ipv6-11.txt

[Mic99] Michie D., Introduction to SWF, 1999, see at URL address http://openswf.org.

[Mob99] Mobile Commerce, report, Durlacher Research Ltd, 1999, see at URL address http://www.durlacher.com/fr-research.html.

[Moo99] What is Moore's Law?, see at URL address http://www.intel.com/intel/museum/25anniv/hof/moore.htm.

[MPS99] Ericsson's new mobile positioning system uses standard GSM phones, press release, 1998, see at URL address http://www.ericsson.com/pressroom/Archive/1998Q4/19981105-0039.html.

[Nam99] Namespaces in XML, W3C Recommendation, January 14, 1999, at URL address http://www.w3.org/TR/REC-xml-names/.

[Nar99] Narikka J., Spatial data and applications in a mobile client-server environment, technical report, 1999

[NAV00] Navigation Technologies Corporation homepage, 2000, see at URL address http://www.navtech.com/

[Nok99] The Demand for Mobile Value-Added Services, market study, 1999, see at URL address http://www.nokia.com

[PGML98] Precision Graphics Markup Language (PGML), Submission to the World Wide Web Consortium, W3C, April 10, 1998, at URL address http://www.w3.org/TR/1998/NOTE-PGML-19980410.

[Pre97] Pressman R. S., Software Engineering: A Practitioner's Approach, McGraw-Hill Book Company, Fourth Edition, European Adaptation, 1997.

[Rei99] Rein L., The Quest for a Graphic Standard, see at URL address http://www.xml.com/pub/98/06/vector/intro.html.

[SAR00] Strategy Analytics Research: Personal Digital Device Sales Reached $13.6bn in 1999, press release, 2000, see at URL address http://www.strategyanalytics.com/press/PRDM017.htm.

[Sat96a] Satyanarayanan M., Fundamental challenges in mobile computing, In Fifteenth ACM Symposium on Principles of Distributed Computing, May 1996.

[Sat96b] Satyanarayanan M., Accessing information on demand at any location, IEEE Personal Commun., 1996

[SGML89] Smith, Joan M. "Standard Generalized Markup Language and Related Standards." *Computing Communications* 12/2, 1989

[Spe97] Sperberg-McQueen C.M., What is XML and Why Should Humanists Care?, Proceedings of the Digital Resources for the Humanities Conference, September 1997.

[SveG99] Swedberg G., Ericsson's mobile location solution, Ericsson Review No.4, 1999, see at URL address http://www.ericsson/review/.

[SVG99] Scalable Vector Graphics (SVG) 1.0 Specification, W3C Working Draft, August 12, 1999, at URL address http://www.w3.org/TR/SVG.

[SVG98] SVG Requirements, W3C, October 1998, see at URL address http://www.w3.org/TR/WD-SVG/reqts.html.

[Tho99] Thomas K., Get Ready for the Awesome Power of SVG: Scalable Vector Graphics, November 1999, at URL address http://www.webdeveloper.com/design/design_svg_intro.html.

[Trimb98] All about GPS, Trimble Navigation Limited, 1998, see at URL http://www.trimble.com/gps/diffgps/gpsfram1.htm.

[USP96] U.S. GLOBAL POSITIONING SYSTEM POLICY, THE WHITE HOUSE, Office of Science and Technology Policy, National Security Council, March 29, 1996, see at URL address http://www.navcen.uscg.mil/gps/geninfo/white.htm.

[Var00] Varazdin 2000, see at URL http://www.2000.varazdin.com.

[Vei00] Veijalainen J., Mobile Computing Course slides (TJT L60) held during the spring semester, 2000 at the University of Jyväskylä at URL address: http://www.cs.jyu.fi/~veijalai/mobilecomputing00/index.html

[VML98] Vector Markup Language (VML), Submission to the World Wide Web Consortium, W3C, May 13, 1998, at URL address http://www.w3.org/TR/1998/NOTE-VML-19980513.

[WAPA99] Wireless Application Protocol Architecture Specification, June 16, 1999, at URL address http://www.wapforum.org/.

[WAP99] WAP White Paper - Wireless Internet Today, June 1999, at URL: http://www.wapforum.com/what/WAP_white_pages.pdf

[WAPP99] Wireless Application Protocol – The Corporate Perspective, White Paper, March 1999, at URL address: http://www.nokia.com/corporate/wap/future.html.

[WAPF99] WAP Forum and W3C Working Together to Create the "Mobile-Friendly" Web, WAP Forum, December 8, 1999, see at URL http://www.wapforum.org/new/120899_w3c.htm.

[WAE99] Wireless Application Environment Overview, WAP Forum, June 16, 1999, at URL address http://www.wapforum.org/.

[WCGM99] WebCGM Profile, W3C Recommendation, January 21, 1999, at URL address http://www.w3.org/TR/REC-WebCGM.

[Wies91] Weiser M., "The Computer for the 21st Century", Scientific American, September 1991.

[WML99] Wireless Markup Language Specification, WAP Forum, June 16, 1999, at URL address http://www.wapforum.org/.

[W3C98a] The World Wide Web Consortium Issues XML 1.0 as a W3C Recommendation, press release, 1998, at URL address http://www.w3.org/Press/1998/XML10-REC.

[XML98] Extensible Markup Language (XML) 1.0, W3C Recommendation, February 1998, at URL address http://www.w3.org/TR/1998/REC-xml-19980210.

[ZiL77] Ziv J., and Lempel A., A Universal Algorithm for Sequential Data Compression, IEEE Transactions on Information Theory, Vol. 23, No. 3, pp. 337–343, 1977.

# APPENDIX A. EXAMPLE OF SIMPLE XML FILE WHICH INCLUDES INFORMATION ABOUT 3 STREETS, 2 HOTELS AND 2 SHOPS.


<?xml version="1.0"?>
<!DOCTYPE city SYSTEM "city.dtd">
<city id='c01'>
<streetNetwork id='sn01'>
<streets>
<street id='s01' type='Major road'>
<name>A-la avenu</name>
<segment id='sg01' length='2'>
<addressRange left1='1' left2='7' right1='2' right2='6'/>
<nodes>
<fromNode p-ref='p01'/>
<toNode p-ref='p03'/>
</nodes>
</segment>
<segment id='sg02' length='2'>
<addressRange left1='9' left2='13' right1='8' right2='14'/>
<nodes>
<fromNode p-ref='p03'/>
<toNode p-ref='p06'/>
</nodes>
</segment>
<segment id='sg03' length='3'>
<addressRange left1='15' left2='17' right1='16' right2='18'/>
<nodes>
<fromNode p-ref='p06'/>
<toNode p-ref='p08'/>
</nodes>
</segment>
</street>

<street id='s02' type='Major road'>
<name>Black Street</name>
<segment id='sg04' length='2'>
<addressRange left1='1' left2='5' right1='2' right2='4'/>
<nodes>
<fromNode p-ref='p02'/>
<toNode p-ref='p03'/>
</nodes>
</segment>
<segment id='sg05' length='3'>
<addressRange left1='7' left2='9' right1='6' right2='10'/>
<nodes>
<fromNode p-ref='p03'/>
<toNode p-ref='p04'/>
</nodes>
</segment>
</street>
<street id='s03' type='Major road'>
<name>Pink Street</name>
<segment id='sg06' length='3'>
<addressRange left1='1' left2='3' right1='2' right2='4'/>
<nodes>
<fromNode p-ref='p05'/>
<toNode p-ref='p06'/>
</nodes>
</segment>
<segment id='sg07' length='2'>
<addressRange left1='5' left2='9' right1='6' right2='12'/>
<nodes>
<fromNode p-ref='p06'/>
<toNode p-ref='p07'/>
</nodes>
</segment>

```
</street>
</streets>
<points>
<point id='p01' type='node'>
<coordinates latitude='2' longitude='3'/>
</point>
<point id='p02' type='node'>
<coordinates latitude='3' longitude='6'/>
</point>
<point id='p03' type='node'>
<coordinates latitude='4' longitude='5'/>
</point>
<point id='p04' type='node'>
<coordinates latitude='5' longitude='1'/>
</point>
<point id='p05' type='node'>
<coordinates latitude='11' longitude='12'/>
</point>
<point id='p06' type='node'>
<coordinates latitude='13' longitude='9'/>
</point>
<point id='p07' type='node'>
<coordinates latitude='14' longitude='7'/>
</point>
<point id='p08' type='node'>
<coordinates latitude='18' longitude='5'/>
</point>
</points>
</streetNetwork>
<group id='g01' name='Shop'>
<object id='o01'>
<name>Antilla</name>
<address s-ref='s01' number='10 '/>
```

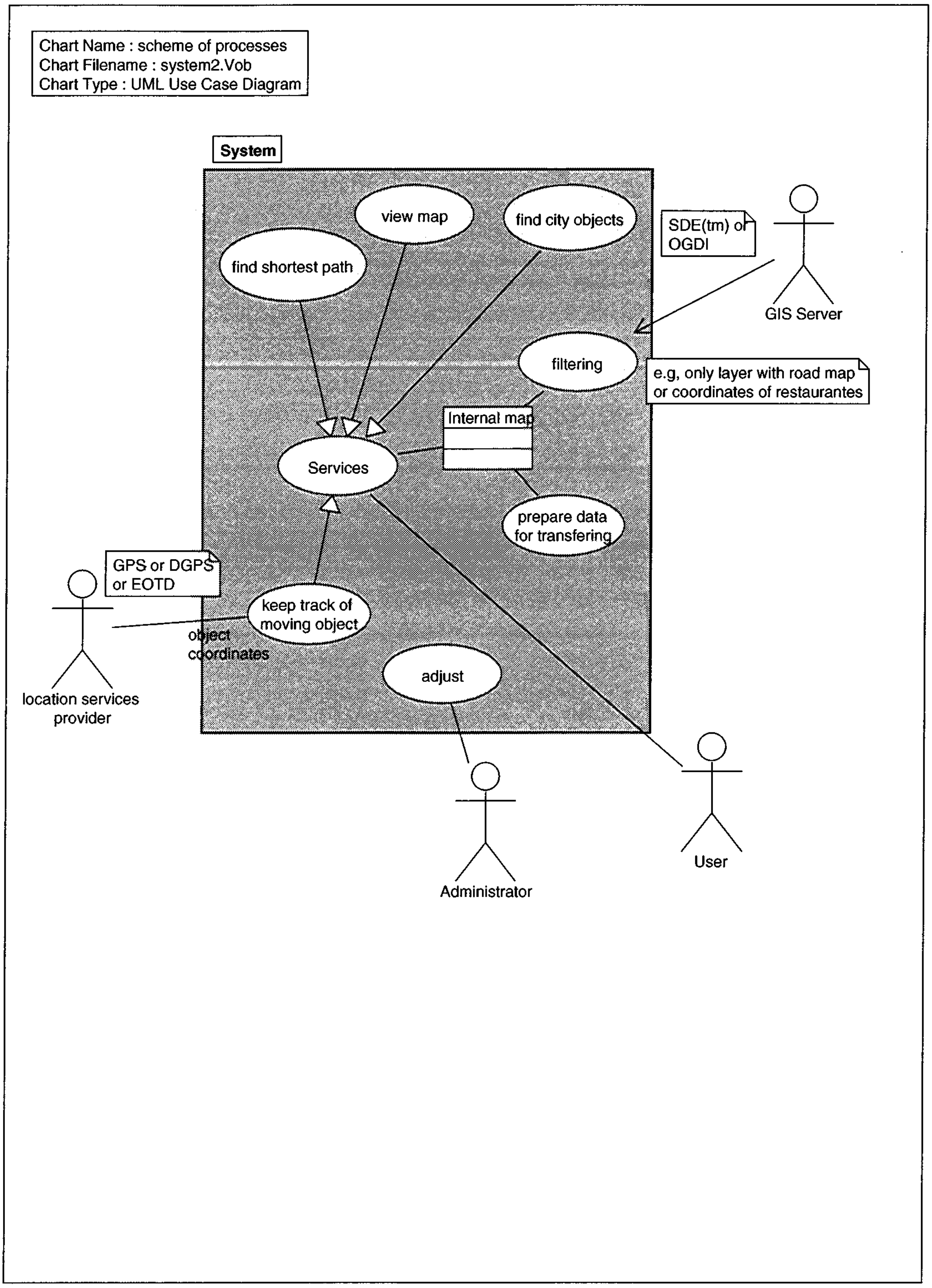
```
<description>
Nice place.
</description>
<coordinates latitude='14' longitude='5'/>
</object>
<object id='o02'>
<name>Sokos</name>
<address s-ref='s03' number='2 '/>
<description>
Very nice place.
</description>
<coordinates latitude='4' longitude='4'/>
</object>
</group>
<group id='g02' name='Hotel'>
<object id='o03'>
<name>Alexandra</name>
<address s-ref='s02' number='4 '/>
<description>
Nice place.
</description>
<coordinates latitude='10' longitude='3'/>
</object>
<object id='o04'>
<name>Scandic</name>
<address s-ref='s03' number='4 '/>
<description>
Funny place.
</description>
<coordinates latitude='8' longitude='5'/>
</object>
</group>
</city>
```

# APPENDIX B. EXAMPLE OF DTD FOR GMML LANGUAGE.

```
<!ELEMENT city (streetNetwork?, group*)>
<!ATTLIST city
            id ID #REQUIRED
            name CDATA #IMPLIED
            geodeticSystem CDATA #IMPLIED>
<!--description of the street network-->
<!ELEMENT sreetNetwork (streets,points)>
<!ATTLIST sreetNetwork
            id ID #REQUIRED>
<!ELEMENT streets (street+)>
<!ELEMENT street (name, segment+)>
<!ATTLIST sreet
            id ID #REQUIRED
            type CDATA #IMPLIED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT segment (addressRange, nodes)>
<!ATTLIST segment
            id ID #REQUIRED
            length NMTOKEN #IMPLIED>
<!ELEMENT addressRange EMPTY>
<!ATTLIST addressRange
            left1 NMTOKEN #REQUIRED
            left2 NMTOKEN #REQUIRED
            right1 NMTOKEN #REQUIRED
            right2 NMTOKEN #REQUIRED>
<!ELEMENT nodes (fromNode,toNode,vertex*)>
<!ELEMENT fromNode EMPTY>
<!ATTLIST fromNode
            p-ref IDREF #REQUIRED>
<!ELEMENT toNode EMPTY>
<!ATTLIST toNode
```

```
              p-ref IDREF #REQUIRED>
<!ELEMENT vertex EMPTY>
<!ATTLIST vertex
              p-ref IDREF #REQUIRED>
<--description of nodes and vertices-->
<!ELEMENT points (point+)>
<!ELEMENT point (coordinates)>
<!ATTLIST point
              id ID #REQUIRED
              type (node|vertex) 'node'>
<!ELEMENT coordinates EMPTY>
<!ATTLIST coordinates
              type CDATA #IMPLIED
              latitude NMTOKEN #REQUIRED
              longitude NMTOKEN #REQUIRED
              altitude NMTOKEN #IMPLIED>
<--description of points of interest by groups-->
<!ELEMENT group (object+)>
<!ATTLIST group
              id ID #REQUIRED
              name CDATA #REQUIRED>
<!ELEMENT object (name, address, description?, coordinates)>
<!ATTLIST object
              id ID #REQUIRED>
<!ELEMENT address (number)>
<!ATTLIST address
              streetname CDATA #IMPLIED
              street-ref IDREF #REQUIRED
              number CDATA #REQUIRED>
<!ELEMENT description (#PCDATA)>
```

# APPENDIX C. USE CASES DIAGRAM FOR PERSONAL NAVIGATION SYSTEM.



# APPENDIX D.

## Figure D.1: Sequence diagram for the bitmap format.

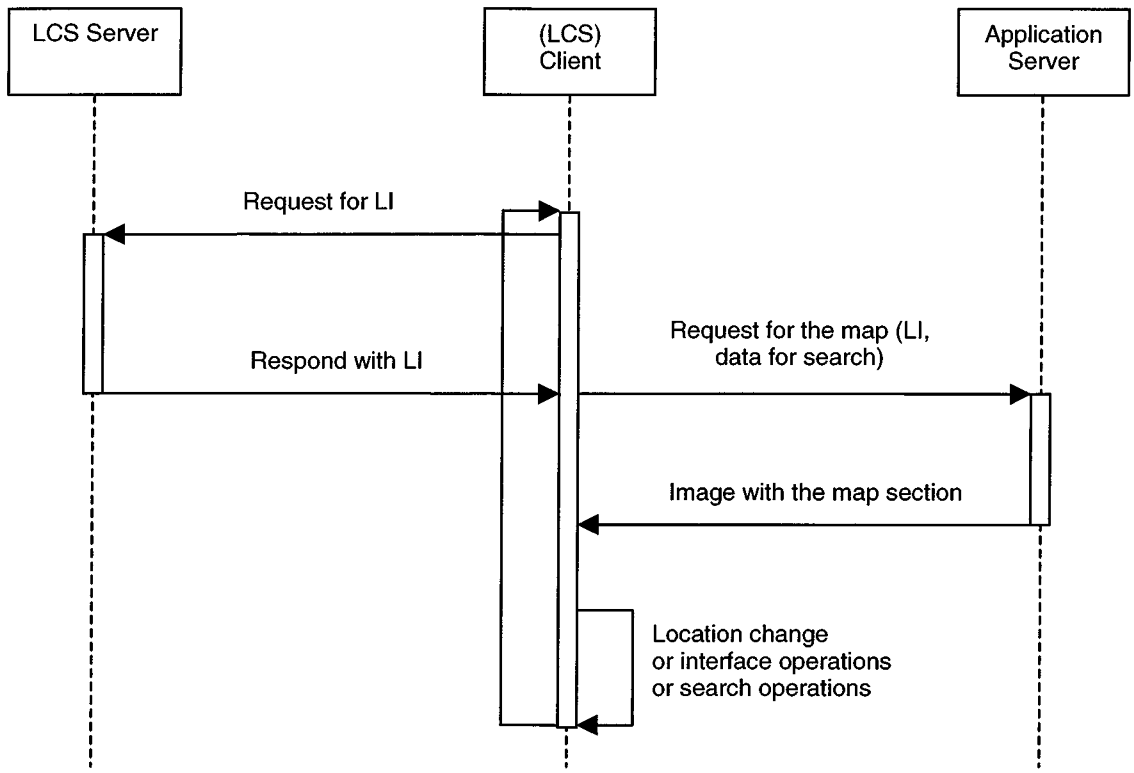
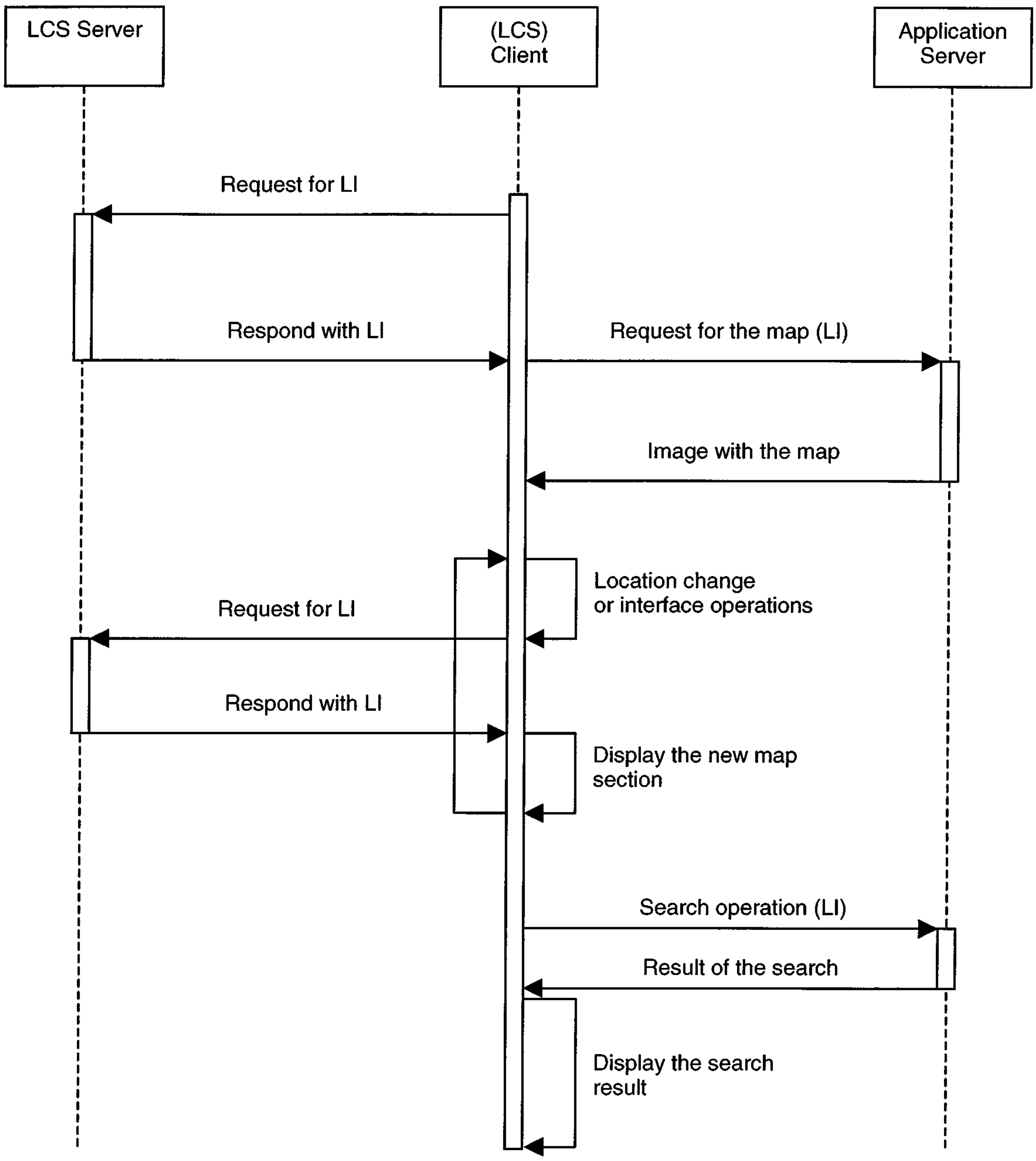


**Figure D.2: Sequence diagram for the vector format.**



## Figure D.3: Sequence diagram for the GIS format.