The Road to Ubiquitous Geographic Information Systems
Roam Anywhere - Remain Connected

Andrew Hunter MNZIS, MSc Candidate

Department of Geomatics Engineering
University of Calgary, Calgary, Alberta, Canada
Phone: +403 220-8149 Fax: +403 284-1980
Email: ahunter@ucalgary.ca

Presented at SIRC 2000 – The 12th Annual Colloquium of the Spatial Information Research Centre
University of Otago, Dunedin, New Zealand
December 10-13th 2000

ABSTRACT

The concept of ubiquitous computing has been with us since the initial development of the computer. However, it is only now that technology has reached a level of maturity that allows the computer to be integrated into our environment in a manner that permits information to be only as far away as our fingertips. The emergence of ubiquitous computing is an indication of the maturity of distributed computing technology. The Internet has provided a means to access these services worldwide, which, given the advances in wireless, Internet and mobile device technologies, has pushed the concept of anywhere, anytime computing to the forefront of GIS research and development. The objective of this presentation is to examine the requirements of ubiquitous GIS in this context, and review some of the enabling mobile application technologies such as speech recognition and wireless communication.

Keywords and phrases: mobile GIS, ubiquitous computing, device independence, speech recognition, wireless communication

1.0 INTRODUCTION

Computing has evolved from massive mainframe computers in out-of-the-way rooms servicing requests from many users, to today's dominant networked environment where many computers can serve one person, from anywhere. That is, the trend has been towards ubiquitous computing.

One of the principal concepts of ubiquitous computing is to make the computer so imbedded in our everyday practices that we use it without having to think about it. Therefore, if computing is to become truly ubiquitous, a number of essential criteria must be met: computers must be small, inexpensive, low-powered and contain convenient displays; the network that supports ubiquitous computing must be robust and efficient; and there must be an adequate number of software systems that support ubiquitous applications.

The computing devices that fall into these categories today include Personal Digital Assistants (PDA's), handheld PC's (H/PCs or Palmtops) and wearable computers, although commercial wearable computers do not meet the low cost criteria at this stage (it is anticipated that this metric may well be offset as a result of productivity improvements obtained through hands-free, eyes-up operation).

The Internet is the logical network backbone between ubiquitous computers, as it is now the mass-market network of choice. However, connection of mobile devices to the Internet can be particularly challenging due to their limited memory, and small displays.

Traditional information systems have embraced distributed technology by making data and networked devices such as printers, which are distributed among separated computers, appear as if they are part of the local computer. Nevertheless, the major difficulty with existing systems is that they have been built for proprietary applications, which restrict their generic application for supporting a broad range of scenarios that are likely to be requested in a truly ubiquitous environment.

The concept of ubiquitous GIS raises a number of issues that need to be addressed, from the client/server perspective, with the primary challenge being to create an architecture that can provide the same functionality no matter what device is being utilized in the field. This paper provides a review of some of the client side components considered necessary for ubiquitous GIS. For more information on the server side, one may refer to Tao and Yuan (2000).

2.0 THE COMPOSITION OF A UBIQUITOUS GEOGRAPHIC INFORMATION SYSTEM

With the convergence of powerful inexpensive hardware, standardized communication protocols and innovative software, it is now becoming possible to deploy GIS functionality in a mobile computing environment.

Leading the ubiquitous GIS movement are utility and infrastructure organisations. However, most GIS applications have yet to reach mainstream field operations, primarily because they tend to be large and complex, and are often difficult to implement in the office let alone the field. In order for a ubiquitous GIS to be successful it must emulate existing field practices, as the purpose of a ubiquitous GIS should be to eliminate repetitive time-consuming tasks and streamline work processes. Traditionally field crews have been provided with little computer training; therefore to eliminate the leap from paper based processes to computer-based applications, processes must be intuitive and transparent; *the invisible servant*.

Goodchild *et al* (1997) describes four criteria, which can be applied to ubiquitous GIS very well: the system must be *distributed*, that is data storage, processing and user interaction can occur at locations that are potentially widely scattered; the system should be *disaggregated*, that is the monolithic systems we have today are replaced by 'plug and play' components, possibly from different vendors, that are designed to interoperate through conformance with industry-wide standards; the system should be *decoupled*, that is the system must be able to access a number of components that may be required to complete a specific task, which may be distributed over many networks; and *interoperable*, which means the system is based on an "open" system such as that promulgated by the Open GIS Consortium (OGS). Of these four criteria, interoperability is the most challenging component, as it involves not just the technical issues but also institutional issues. A brief review of GIS interoperability ensues.

Interoperable Systems
The goal of interoperating GISs is to achieve an automated process that will allow us to use data and software services across the boundaries that their collectors and designers envisioned.

Interoperable GISs can be decomposed into three distinct elements. The first is the technical aspects. This element, deals with the compatibility of data formats and techniques that can be utilised to remove implementation details from a users problem. One method of achieving this is the adoption of object-oriented models, which allow complex objects to be considered as individual units that may relate more directly to a users concepts. The second element deals with semantics. If GIS is to reach a certain level of maturity it is necessary that users and developers alike must adopt a common geographical information language. This implies the adoption of a common worldview as well as common abstractions, feature representations, and metadata (Kottman, 1999). The final element deals with institutional issues. That is an organisations willingness to be open and share versus protection of its interests; the added cost of achieving interoperability versus the benefits and value added by interoperability - which may likely be hidden; the right to know versus the right to privacy and protection of intellectual property; and the impacts of technological change on institutions that have been designed to achieve certain goals.

The benefits of developing an interoperable system are numerous. Interoperability will simplify the interaction between the complex collection of formats and standards that exist in the industry today. Interoperability will create a higher level of agreement of basic data models, which will provide transparency so that the user is no longer required to be aware of a datasets implementation details in order to utilise it. Software packages that are interoperable are likely to be stable since the same principles used in an initial application will need to be maintained in subsequent versions. Interoperability will also require a standardised theory of geographic data, which should ensure stability of software over time.

Ultimately the user community will judge the success of interoperable GISs. If GIS becomes interoperable, GIS will become ubiquitous, embedded in many everyday activities across a wide spectrum of activities.

3.0 FEATURES OF A UBIQUITOUS GIS APPLICATION

Mobile devices differ considerably from traditional desktop computers. They come in a variety of forms and processor types. Screen sizes vary dramatically as do input methods, which may range from a stylus and touch screen, to a barcoder, to speech recognition. Limitations in disk space, memory and battery capacity can impose considerable restrictions on mobile applications. However, the most significant differences between a mobile device and a desktop computer is the intermittent connectivity to business systems and the labyrinth of connectivity options - dial-up, wireless, Local Area Network (LAN), docking, and the Internet.

Mobile software applications should be designed for use by mobile users. The software should provide the user with the ability to gather information, execute functional activities specific to their job, provide quick access to external data, update the data stored on the mobile device, and synchronize the data with the external datasets. The application should be able to be used while in motion; it should be uncomplicated to learn, easy to customize and facilitate self-reliance.

Other desirable features of a mobile application could include:

- Support for open standards – full support for open standards will reduce future dependencies when any changes are required as the needs of users mature.
- Support for a large number of users – the application must be able to handle a large number of users concurrently.
- Support for collaborative work – communication between the mobile clients as well as office users needs to be established and the software to support collaborated group computing needs to be developed.
- Ability to work both on-line and off – the mobile device must be provided with facilities to manage a subset of any database being used while the user is working off-line.
- Support for a wide variety of networks - the application should be provided with the capability of working over various communications networks, such as Internet, dial-up, wireless or serial connections.
- Multi-functional – the application should support local and central database query, as well as the synchronization of information and two-way messaging.
- Integration with other applications - the application must be able to seamlessly integrate with existing information systems, without requiring any changes to be made.
- Security – the application must support standard network security mechanisms that provide full authentication and security for access to the device as well as the network.

Software flexible enough to meet a multitude of application needs will only tie up valuable resources in computing devices such as PDA's that are already resource-scarce. Hence, a preferred implementation methodology is to integrate specific data acquisition, mapping and spatial analysis tools into applications packages or components that are only loaded on an as-required basis (Tao and Yuan, 2000). Therefore, a mobile GIS application should support a number of primary and subordinate functions, some of which are listed below:

Primary Functions: Mapping and navigation (zoom and pan); Data collection, query, and updating; Remote data access and management; Remote functional component access and integration; and Location determination (GPS);

Subordinate Functions: Speech to Text; Automatic time stamping (versioning); Report generation; Two way messaging; and if speech is enabled Telephone communication.

A mobile application needs to offer functionality in a simple package, with the most important requirement being that the application should be able to work in exactly the same environment that a prospective field user is currently working in. This requires that the mobile devices be ruggedized, that is the device can withstand extreme temperatures and also endure the shocks of being dropped on the ground.

Speech recognition functionality has been listed as a subordinate function primarily for the reason that PDA's are yet to contain the necessary resources required to support a speech engine. However, if a mobile system is to be developed for either a wearable or handheld PC, it is the opinion of the author that in order for the application to be truly mobile, speech recognition functionality should be promoted to a primary function.

3.1 Speech Component and Wearable Computers

Traditionally, when field crews require access to information they must stop what they are doing and reference hardcopy drawings or manuals. This method of accessing information reduces their productivity. Conversely, speech enabled computing can improve productivity by allowing the operator to access information while continuing to work. Speech enabled computing also adds significant value when a task involves walking and maneuvering in tight spaces, using tools and ultimately using a computer to complete the task. For more information about the use of speech recognition for GIS data acquisition, refer to Tao and Yuan (1999). A speech enabled computing application coupled with a wearable computer allows the user to maintain an awareness of their physical environment while focusing their attention on a task in their virtual environment. If the user is suddenly confronted by an adverse situation, they can quickly switch their concentration to the physical environment, ignoring the virtual. The result is improved user safety.

Speech technology, particularly speech command technology will enhance data input and improve quality control by allowing the user to immediately review the acquired data. An example of the interaction between the user and a speech enabled mobile device could progress as follows: **User:** 'Hydrant, Type: Clow Premier'; **Computer:** 'Database Type: Clow Brigadier'; **User:** 'Correct Type: Clow Premier'; **Computer:** 'Corrected to: Hydrant, Type: Clow Premier'.

By utilising commands that the user is familiar with, the transition from traditional paper based processes to computerised processing can be simplified significantly, allowing the user to get on with the work at hand rather than becoming frustrated with the device and reverting to traditional work methods. The combination of speech recognition and wearable computers can also increase the flexibility of the system and help to remove the computer from the users centre of concentration to the periphery, allowing the user to concentrate on the task at hand.

3.1.1 Command and Control Speech Recognition

Command and Control speech recognition allows the user to speak a word, phrase, or sentence from a list of phrases that the computer is expecting to hear. For example, a user might be able to speak the command, 'Add a Layer', 'Zoom in', or 'Pan'. In general, Command and Control should be implemented to make an application easier to use; to make features in an application easier to get to; or, to make the application more realistic to use.

Command and Control is typically used to provide answers to questions; activate macros; access large lists; prompting the user for required information and facilitating hands free computing. Many database applications implement command and control functionality as a means of speeding up data entry as it is much easier for users to read data to the computer.

Limitations
Even the most sophisticated speech recognition engine has limitations that affect what it can recognize and how accurate the recognition will be. Typically, the microphone is the primary source of speech recognition errors. In order to minimise microphone based errors it is very important that a good quality close-talk microphone headset is utilised. By placing the microphone close to the mouth background noise can be minimised.

Speech recognition engines are designed to 'hear'. Therefore the speech engine can sometimes interpret background noise as words. There are a number of methods that can be employed so that these types of errors are minimised, for example, commands can be implemented to put the microphone to sleep when it is not in use; or the computer can be given a name which the user must say prior to speaking a command so that the computer knows it is hearing a valid command; or the computer can verify every command with the user. If the user does not confirm the command within a certain time then the computer will not act upon the command.

Command and Control Commands
Prior to a command and control recogniser 'listening' for commands it must be provided with a grammar, or list of commands, to listen for. If the user speaks the command as written in the grammar supplied to the recogniser very few errors will be generated. However, if the user diverges from the supplied grammar there is a good chance that the computer will misinterpret the command that it hears. In order to minimize command recognition errors every endeavour should be made to implement commands that are intuitive to users. Lists of available commands should also be readily accessible from anywhere in the application. As well, the more phonemes (a single distinctive speech sound) different between two commands the greater the likelihood of them sounding different to the computer. Typically, speech engines cannot tell who is speaking, nor can they detect multiple speakers, and speakers with accents or who speak in non-standard dialects will obtain a higher proportion of recognition errors. In order to minimize these sources of errors training of the speech recognition engine should be undertaken by each user of the application.

Command and Control Design Considerations
Speech recognition is not a replacement for the keyboard or mouse. Speech recognition is an ineffective pointing device, just as the mouse makes a terrible text entry device. Generally speaking, every feature in an application should

be accessible from all input devices, keyboard, mouse, and speech recognition. Users will naturally use whichever input mechanism provides them the quickest or easiest access to the feature.

In general the keyboard and mouse do not send erroneous signals to the application. This is not so with speech recognition. The number of voice commands that must be recognized at any given time can be significant. To assist the user in locating the correct command, an application can prompt the user for the most common voice responses through visual aids or text-to-speech. Whenever a voice command is spoken, the application should give some sort of feedback to the user indicating that the command was understood and acted upon.

3.1.2 Text To Speech Engines

Text-to-speech is a process through which text is rendered as digital audio and then 'spoken'. Most text-to-speech engines can be categorized by the method that they use to translate phonemes into audible sound. Text-to-speech is used to communicate information to the user, when digital audio recordings are inadequate. Generally, text-to-speech is more efficient than audio recordings when the audio recordings are too large to store on disk, too expensive to record, or when the application does not know what information is to be communicated to the user. While text-to-speech engines perform adequately when 'speaking' individual words, they can become difficult to listen to when long passages are spoken. This is generally because text-to-speech engines lack human prosody (rhythm and intonation).

3.2 Wireless Component

The wireless component is considered to be the enabling element of a mobile GIS. Wireless data access allows users to be more productive by allowing them to get the information they need wherever they are, and disseminate information between field operators and process management personnel. Wireless networks provide the flexibility and freedom required to seamlessly integrate computing with field-based activities. Ideally, a mobile device should be able to select the network (LAN, the Internet, PCS, satellite, etc.) that best meets the users requirements.

The first commercial wireless networks became available in the mid to late 1980’s and were generally developed for specialist applications. Wireless networks work by superimposing data on radio carriers. By utilizing different frequencies multiple users can coexist in the same radio space.

One of the primary differences between wireless and traditional telephone technology is the method of transmitting information. Normal telephone services are based on circuit-switching technology where a dedicated line is allocated for transmission between two parties. Wireless services are designed around packet switching, whereby messages are divided into packets before they are sent. Each packet is then transmitted individually and can even follow different routes to its destination. Once all the packets forming a message arrive at the destination, they are recompiled into the original message. Most modern Wide Area Network (WAN) protocols, including TCP/IP, X.25, and Frame Relay, are based on packet-switching technologies. Circuit switching is ideal when data must be transmitted quickly and must arrive in the same order in which it is sent. This is the case with most real-time data, such as live audio and video. Packet switching is more efficient and robust for data that can withstand some delays in transmission, such as e-mail messages and Web pages.

A number of difficulties need to be overcome if wireless networks for mobile users are to prompt more extensive use. Firstly channel capacity normally available in wireless networks is considerably less than that which is available in wired networks due to the limited spectrum available, power restrictions and poorer signal to noise ratios. Secondly, security is of greater concern in a wireless network than in a wired network, as information is transmitted through space. Performance and interoperability of wireless networks is also affected by the Media Access Control (MAC) utilised by a network. The MAC protocols used in the dominant cellular systems in the US, Asia and Europe (CDMA and TDMA) differ considerably. Telecom is currently in the process of upgrading its Digital AMPS network to CDMA technology.

3.2.1 Wireless Networks

Currently there are five types of wireless networks: Wireless Local Area Networks (WLAN); Satellite based networks; Wireless Local Loops (WLL); Wireless Asynchronous Transfer Mode (ATM); and Bluetooth which provides a low-cost, short-range (≈10m) radio link for wireless connectivity between computing devices.

Wireless LANs
Wireless LANs are designed to provide coverage in a relatively small area such as a building or office. These networks are installed primarily for their flexibility rather than their bandwidth. Wireless LANS share frequencies, which can result in loss of data through collisions. The choice of frequency depends on the communication method used

(microwave, spread spectrum, infrared). Spread-spectrum technology (used in current GPS technology) is the most secure. Spread-spectrum technology transmits information between devices on wideband radio frequencies. This improves reliability and security by reducing interference among users sharing the same spectrum, however some form of encryption is still required if secure connections are to be maintained. Currently there are two wireless LAN standards being IEEE's 802.11 standard and HIPERLAN.

Wireless Local Loops (WLL)
Long distance communication companies are introducing WLLs in the US as a means to avoid paying charges to local carriers. WLLs provide several MHz of bandwidth from fixed wireless access points that are suitable for use as high-speed Internet access points or data transfer in addition to traditional phone services. WLLs are designed around 900, 1800, and 1900MHz cellular technologies and can service between 10 and 100 users within three to four kilometres of an access point. The Local Multipoint Distribution System (LMDS) is an emerging spread spectrum technology that can support very high bit rates for two way data transfer. LMDS uses the 28 to 31GHz band and has recently been implemented by Clear.

Satellites
Satellite based communication is an obvious path to follow as coverage is worldwide. Traditionally, satellites have been used for phone (stationary) services, paging services (location of users) and data transmission services. However with the advances in antenna design and signal reception it is becoming possible to provide mobile services using satellites. The primary difference between a satellite network and a ground-based network is that it is the satellite that moves through the users cell (transmission range) rather than the user moving through a cell relating to a particular access point on the ground. Therefore the satellite must handoff the user to another satellite before the link is lost.

Iridium is a low orbit system that uses 66 satellites to provide mobile communications. Iridium services commenced in January 1999, but due to technical problems and a lack of users, services were terminated during March of 2000. Although Motorola has advised Iridium that it intends to maintain the Iridium satellite system for a limited period of time to allow subscribers in remote locations to obtain alternative communications.

Teledesic is expected to be operational by 2003. It is planning to launch 288 low orbit satellites. Each satellite will be capable of handling up to 155.52Mbps to and from the ground and 622.08Mbps to and from other satellites. A typical Teledesic terminal is expected to operate a 64Mbps downlink and a 2Mbps uplink.

Wireless ATM (Asynchronous Transfer Mode)
Wireless ATM is a promising high-speed packet based network technology. Information is transmitted in 53 byte packets or cells that can be prepared, transmitted and switched by the network at very high speeds. The major advantage of Wireless ATM is that it is scalable, that is it can be used in both local and wide area networks.

3.2.2 Middleware

Mobile middleware is a layer of software that is used by an application so that it can connect to different wireless networks and operating systems transparently. Middleware does add additional complexity (and initial cost) to a mobile application, but it may also improve reliability and network response times if good optimization algorithms are utilized.

Wireless Application Protocol (WAP)
The Wireless Application Protocol is a standard developed by the WAP Forum, a group founded by Nokia, Ericsson, Phone.com (formerly Unwired Planet), and Motorola. The WAP Forum's membership now includes computer industry heavyweights such as Microsoft, Oracle, IBM, and Intel along with several hundred other companies.

Currently, there are a number of wireless access technology implementations that are not interoperable, which creates significant difficulties for developers who are trying to design an application for a number of mobile devices. WAP allows development of applications that are independent of the underlying wireless technology and appears to becoming the *de facto* world standard for wireless information. It is optimized for small devices and is based on the Internet client/server architecture (see Figure ***1***). WAP provides a Wireless-optimized Mark-up Language, WML, which introduces syntax more appropriate for low resource mobile devices.

WAP was designed to solve some of the problems caused when small, low-powered devices on different platforms try to use low-bandwidth network technology to access services or data-intensive content via the Internet. WAP makes efficient use of the available bandwidth by utilising optimal text compression techniques and binary transmissions to reduce transmission overhead. WAPs main contribution is interoperability over assorted wireless networks using a common set of application and network protocols. WAP will be accessible but not limited to the following carrier technologies: GSM-900, GSM-1800, GSM-1900; CDMA IS-95; TDMA IS-136; and 3G systems - IMT-2000, UMTS, W-CDMA, and Wideband IS-95.

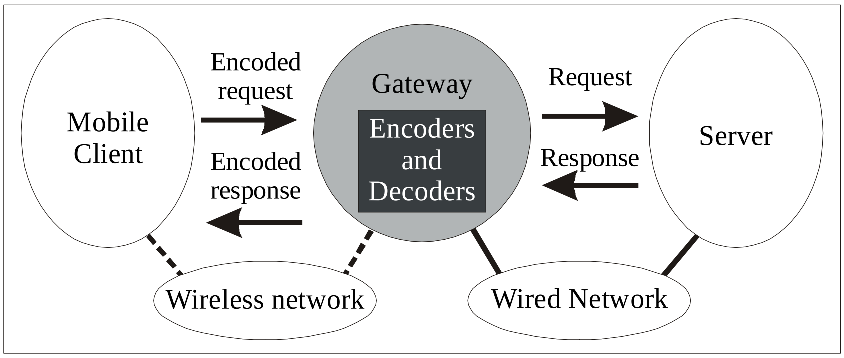

Figure 1: A Basic Wireless Application Protocol (WAP) Architecture

A WAP request is routed through a WAP gateway, which acts as an intermediary between the 'bearer' used by the mobile client (GSM, CDMA, TDMA, etc.) and the computing network that the WAP gateway resides on (TCP/IP in most cases). As depicted in Figure ***1***, the gateway consist of encoders and decoders that process requests, retrieve content or call CGI scripts, Java servlets, or some other dynamic mechanism, and then format data for return to the client. Data is formatted using WML (Wireless Markup Language), a markup language derived from XML. Once the WML has been prepared (known as a deck), the gateway then returns the completed request (in binary form due to bandwidth restrictions) to the client for display and/or processing. The client retrieves the first card off of the deck and displays it on the monitor.

The *deck of cards* is designed specifically to take advantage of small display areas on handheld devices. Instead of continually requesting and retrieving cards (the WAP equivalent of HTML pages), each client request results in the retrieval of a deck of one or more cards. The client device employs logic via embedded WMLScript (the WAP equivalent of client-side JavaScript) for intelligently processing these cards and the resultant user inputs.

At present there are a number of usability issues that are affecting the adoption, and public/industry perception of WAP. Aside from small screens and restricted bandwidth, WAP enabled devices require a new call to a wireless network each time a connection is requested, and keypads are user unfriendly. Because of these weaknesses, designers of WAP services have concluded that they need to optimize a service for each of the different devices and its specific restrictions and interaction techniques. The result is minimal interoperability and high development costs. However WAP is still evolving, and like the Web before it, more time is required so that both hardware and software can be refined in order that all the benefits of WAP technology can be realised.

There are a number of wireless modems that may be plugged into a mobile device, which can be utilized in conjunction with a public wireless network provider. The vast majority offers transmission speeds of 9.6Kbps or 19.2Kbps, although Novatels Merlin II for Ricochet purports to offer 128Kbps. An alternative option is to implement a Virtual Private Network (VPN). This type of system could be based around a router such as the WaveRider NCL 135 which has an over air rate of 1.6Mbps for up to 16km. The primary benefits of this type of system are higher data transmission rates, and no additional costs after purchase and installation of equipment, except for maintenance. The disadvantage of a VPN is the upfront cost of routers and antenna.

3.3 GPS Component

In order to take advantage of the geographic element of GIS data it is logical that a mobile application incorporates a GPS component. In order to meet the goals of interoperability where possible, the GPS module should receive corrected DGPS data in a format such as NMEA 0183. The NMEA data is transmitted in the form of 'sentences', a number of which can be utilised to determine the location of the user and provide an appropriate level of information so that the user can assess positional accuracy in real-time. They include the Global Positioning System Fix Data sentence (GGA) to determine the satellites in use; the Satellites in View sentence (GSV) to determine the azimuth and elevation to each satellite; the GPS DOP and Active Satellites sentence (GSA) for position fix status, HDOP, PDOP and VDOP; and the Recommended Minimum Specific GPS/Transit Data sentence (RMC) to determine latitude and longitude. The NMEA 0183 Standard has evolved into the de facto GPS data transfer standard and is included with all major GPS receivers, thus insuring interoperability with different GPS brands.

4.0 ARCHITECTURE OF A DISTRIBUTED UBIQUITOUS GIS

As established, mobile devices should be small, inexpensive, low-powered and contain convenient displays. The three distinct classes of computers, being Personal Digital Assistants, handheld PC's and wearable computers, have significant differences in terms of operating systems, input/output options and computing resources. As such an interoperable design is imperative.

An XML-based system architecture is given in Figure *2* where the application is delivered to the device via its Browser. A Web Server would receive requests from a WAP or Palm Proxy and dispatch these requests for processing via a Java based servlet engine that acts as a gateway to a back-end distributed system.

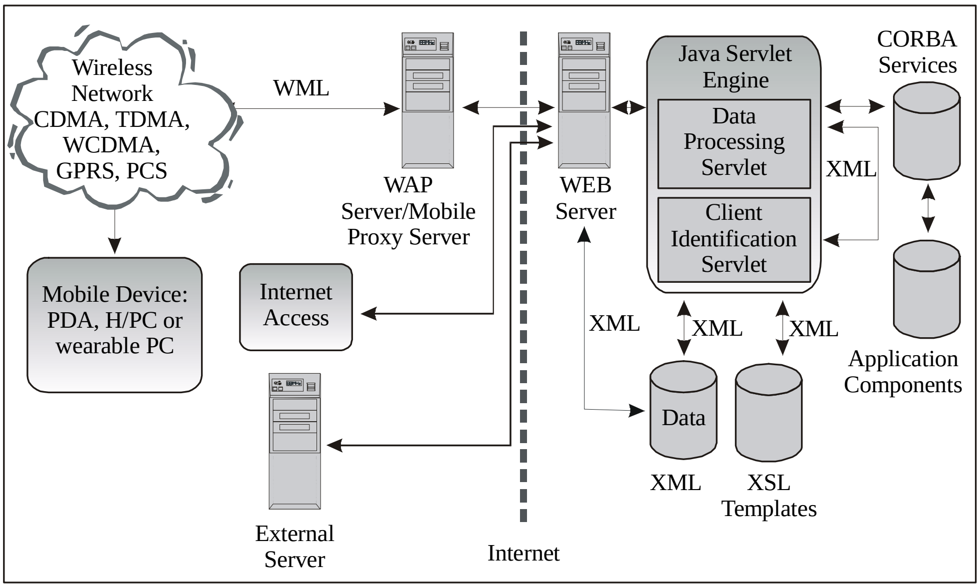

Figure 2: Mobile Distributed Computing Architecture

The servlet engine would be responsible for hosting application servlets and providing them with a standards-compliant servlet interface. An advantage of using a servlet engine instead of running the servlets inside the Web Server process itself is that the servlet engine will provide a standard servlet API and decouples processor servlets from specific implementation details of the Web Server. This increases architectural flexibility as it allows the Web Server to change without impacting the overall system. The servlet engine can also provide various management features that help to shift the load of servlet processing away from the Web Server.

The Servlet Engine requires two servlet components. One component (data processing) acts as a gateway that receives requests from the Web server for processing and forwards these requests to an appropriate service for execution. After processing has been completed it formats the results (XML) and parses them to the Web Server for delivery to the client. The second servlet (client identifier) is responsible for determining the type of client and can therefore notify the data processing component what sort of template should be implemented when formatting results. This will allow a wider range of mobile devices and operating systems to be implemented in the field.

The servlet engine will also require components that allow publication and lookup services to be implemented when the Servlet engine starts up. Essentially when the servlet starts up it will publish itself to a Naming Service to assist with the search for components required to fulfill the processing needs of a client.

The benefit of this architecture is that it is device independent and that it can provide services not only to mobile devices but also to Web users and external servers. The downside to this model, as with most thin client models will be the response time of the application. This factor must be measured against the mobile users need for timely information. As yet PDA's and some H/PC's do not support speech recognition, therefore the users interaction with the device will either via a keypad or pen tablet. This factor severely restricts the users ease of mobility and the speed with which the user can perform tasks on the device.

If the mobile device is a wearable computer it is logical that a greater portion of the processing is carried out on the device. Therefore facilities must be available that allow components to be uploaded to the mobile device if the device has the necessary resources to be able to perform a task locally. Basically a request can be made to the Naming Service for a component that can fulfill a required function. If the mobile device has adequate resources the component would be uploaded to the device. However, if there were insufficient resources on the client, the servlet engine would be responsible for utilising the component. The decision as to whether the component should be uploaded or not could be determined by 'intelligent' load balancing functionality contained in a process management layer on the servlet engine.

5.0 CHALLENGES

The major challenges to ubiquitous GIS relate to the current state of GIS interoperability and wireless technology. At present there are a significant number of wireless technologies that have the capability to service a large number of high demand mobile GIS users. The problem is that insufficient infrastructure is in place that supports these technologies. Nor are there any commercially available wireless devices that take advantage of these emerging services over existing communication networks. This leaves mobile users with two options, the first is to make do with the commercial wireless services that are presently available, which are slow and sometimes dubious; or users can set up their own private virtual network, which will provide more appropriate data transmission rates although start-up costs will be significant and coverage is likely to be limited.

When working with distributed systems security becomes an issue. A firewall for a corporate LAN is normally configured to only permit external access by devices that have an IP address that is within a predefined range, or has been uniquely registered. If a device attempts to establish a connection across the public Internet via an Internet Service Provider (ISP), then it will normally inherit a new IP address – allocated by the ISP for the duration of the session – and thus look like a foreign and potentially unwelcome visitor. This situation is further complicated by mobile devices, which may exist within an IP subnet only for as long as they are in transmission range of that subnet. Firewalls therefore need to become a good deal more sophisticated in their recognition of mobile devices, if the user is to obtain consistent access.

While WAP and HTML protocols are likely to continue as Internet based mobile standards, due to their methods of transmitting map based information (JPEG, GIF, etc.), GIS response time will remain inadequate, effectively ensuring that users will become frustrated with delays in the retrieval of information. This will have a significant effect in terms of global acceptance of ubiquitous GIS. The Scalable Vector Graphics (SVG) standard is the standard that is presently capable of utilizing vector graphics over the Internet, the major benefit of which is scalable graphics, fast download time, and high performance zooming and panning. It would appear that a hybrid standard combining the best features of WAP and SVG is a logical path for ubiquitous GIS applications.

The architecture discussed earlier in this report is delivered via a Web Browser, meaning the application is device independent. However, this architecture does not take into account computing resources that may be available if the client is a high-end mobile device. There are currently a number of load balancing applications available for distributing processing requirements amongst a number of servers. This concept needs to be extended to include the mobile device as a possible source of computing resources. That is, a management layer needs to be built into the mobile architecture that can decide, based on knowledge of processing requirements and processor availability, if a process should be implemented on the client, or if it would be more efficient to implemented it on a server.

Other issues that need to be investigated relate to speech recognition accuracy and positional accuracy. At present good speech engines are averaging 96 to 98% accuracy, which equates to 20 to 40 errors per 1000 words. Is this adequate to ensure user satisfaction? Are error correction tools simple to learn and use? With use, can Command and Control applications adequately adjust to a particular users speech? Experience to date indicates that most speech engines have been designed to work best when the user has a generic North American English accent, and that they can pose problems for those who don't.

Mobile GIS applications are particularly attractive to the utility sector and organisations requiring spatial datasets for asset management. Data custodians within these domains often require sub-metre accuracy. While this is possible with current DGPS technology, at what speed will accuracy specifications such as this no longer be met? Speech recognition is not instantaneous, therefore the user will likely have traveled a certain distance by the time the speech engine has determined that a spoken command represents a feature that must be captured. Is the lag between the issue of a command and the recognition of the command constant? Is the lag constant for every command? All of these issues will have an effect on the positional accuracy of a mobile application.

6.0 CONCLUDING REMARKS

Ubiquitous GIS can be viewed as the seamless integration of information access and presentation across all environments. Therefore, it is necessary to create an open and device independent architecture, which allows information to be presented and communicated with people in all computing environments. In order for ubiquitous GISs to be successful they must be deployed as part of an intelligent enterprise strategy designed to eliminate repetitive time consuming tasks and to empower field crews by providing knowledge based applications which stand-up to everyday practices.

Computing environments with broader acceptance and more common protocols, like operational database connectivity, ActiveX, Java, XML and object-orientated modeling will be the tools of choice for ubiquitous GIS development.

Suitable wireless networks for GIS applications are not yet in place but they are expected to be within the next couple of years. By that time we can expect that real-time access to remote datasets will be commonplace. As yet there are no applications available on the market that provide this feature within a GIS environment. Speech recognition is becoming commonplace within database applications but is not yet available in commercial mobile GIS applications. Interoperability will continue to be an issue; nevertheless gradual steps are being taken by the industry as a whole to resolve this issue.

7.0 ACKNOWLEDGEMENT

The author wishes to recognise that ideas promoted in this paper have been developed through discussions with Dr. C. V. Tao, of the Department of Geomatics Engineering, University of Calgary.

REFERENCES

Anywhereyougo.com @ http://www.anywhereyougo.com/ayg/ayg/Index.po (25/09/00)

Goodchild, M. F., Egenhofer, M. J., and Fegeas, R. (1997) *Interoperating GISs Report of a Specialist Meeting Held under the Auspices of the Varenius Project Panel on Computational Implementations of Geographic Concepts* Santa Barbara, California.

Kottman, C., (1999) *Chapter 4 The Open GIS Consortium and Progress Toward Interoperability in GIS Interoperating Geographic Information Systems* edited by Goodchild, M., Egenhofer, M. J., Fegeas, R., and Kottman, C., Kluwer Academic Publishers.

Licklider, J.C.R., (March 1960) *Man-Computer Symbiosis*. IRE Transactions of Human Factors in Electronics, Volume HFE-1, pp. 4-11.

Microsoft Speech API 4.0 Documentation (1998)

Tao, C.V., and S. Yuan (1999), *Speech Recognition and Its Implications to GIS Data Collection*, Proceedings of 1999 ASPRS Annual Conference, Portland, Oregon, May 17-21, pp. 1060-1069.

Tao, C.V., and S. Yuan (2000), *Development of Network-based GIServices in Support of Online Geocomputing*, International Archives of Photogrammetry and Remote Sensing, Amsterdam, July17-22, Vol. 33, Part B4, pp. 1058-1065.

Technical Resource Connection, Inc. (2000) *TRC's Guide to Mobile Distributed Computing with XML* White Paper 29 pg.

Weiser, M., (1991) *The Computer for the 21st Century*, Scientific American, September, pp. 94-104.

Wilson, J. D. (1998) *GIS Goes Mobile*, GIS World, Vol. 11 No. 12. pp. 54-57

Wireless Development Network @ http://www.wirelessdevnet.com (21/09/00)

Xybernaut Corporation (1999) Speech-Activated Wearable Computing: A Natural Fit White Paper 8 pp.