

OPEN SOURCE MOBILE GIS SOLUTIONS FOR DIFFERENT APPLICATION FIELDS

M. A. Brovelli ^a, D. Magni ^{a*}

^a Politecnico di Milano, Polo Regionale di Como, 22100 Como, Italy - (maria.brovelli, diego.magni)@polimi.it

KEY WORDS: Mobile GIS, mobile mapping applications, user positioning, context awareness, local wireless network, GML

ABSTRACT:

This work deals with use of Open Source technologies for implementing mobile GIS solutions, in order to face the need of moving GIS applications from desktop or web platforms to mobile platforms - like PDAs, smartphones, etc. - accordingly with the great and growing diffusion of mobile devices and location based services.

The implemented mobile GIS is a client-server solution based on a Java servlet inside Apache Tomcat servlet container. The GIS functionalities are obtained by MapServer Java MapScript and by some classes implemented expressly.

The paper will discuss the technological aspects of the mobile GIS and will show some specific applications, in particular for archaeological contexts and for issues of civil protection.

Moreover, other related topics will be treated: how to make the mobile GIS context aware, how to integrate it with user positioning and object of interest location, how to face some problems (slow loading of images, connection costs) coming from the chosen architecture, how to use it within a local wireless network, how to redefine maps and contexts by GML (Geographic Markup Language).

RÉSUMÉ:

Le travail qu'on va présenter regarde l'utilisation de technologies Open Source pour implémenter solutions de SIG mobiles, avec l'intention d'affronter la nécessité de déplacer les applications SIG de plate-formes desktop ou web à plate-formes mobiles, comme PDA et smartphones, en suivant la grande et croissante diffusion des dispositifs mobiles et des LBS (location based services).

Le SIG mobile qu'a été réalisé est une solution client/serveur, basée sur une servlet Java dans Apache Tomcat.

Les fonctions SIG sont données par MapServer Java MapScript et par des classes Java implémentées expressément.

Ce document discutera les caractéristiques techniques du SIG mobile et montrera des applications spécifiques, en particulier pour contextes archéologiques et pour cas de protection civile.

Seront aussi examinés autres sujets reliés aux précédents: comme rendre le SIG mobile sensible au contexte où il est utilisé, comme intégrer le SIG mobile avec fonctions qui donnent la position de l'utilisateur et la localisation d'objets d'intérêt, comme affronter les problèmes qui dérivent de l'architecture choisie (lent chargement des images, coûts de connexion), comme utiliser le SIG mobile avec un réseau local wireless, comme créer une nouvelle définition des maps et des modèles avec le GML (Geographic Markup Language).

1. INTRODUCTION

1.1 Overview

Mobile GIS are systems supposed to be run on all the available mobile platforms, from laptops and tablet PCs to PDAs (Personal Digital Assistants), pocket computers and cell phones, with eventually pre-installed or installable Java Virtual Machine.

The usage of such a kind of systems can be very wide: they can be used simply for navigation purposes (routing and tracking) or recreation ones (visiting parks or areas of natural, cultural interest) or for maintaining inventories characterised by a geographic component, for events and incidents reporting or for directly mapping on the field. In the former cases they are simply read-only mode mobile GIS, while in the latter ones they are supposed to guarantee also transactional functionalities (create, modify and delete), that can be synchronous or not.

Data retrieval, which is the basic process needed also for transactions, can be implemented in stand-alone applications with a preliminary loading phase. In this case the server is a simple datasource. Once the cartographic data have been

modified, they can be uploaded to the server to upgrade the database itself.

On the opposite others architectures consist in a continuous synchronous negotiation in client-server based solution. In that case usually a "three tiered architecture" is implemented, that means that the application is divided into a presentation tier, which is a client component responsible for sending request to the server and to view the results, a business logic tier, consisting of the server side components, including the Web server and application server, and a data management tier, responsible for the management of both spatial and semantic data.

Mobile communication networks and the Internet are the two main communication ways in a mobile spatial information service framework.

Information are transferred via some protocol as WAP (Wireless Application Protocol), HTTP (Hyper Text Transfer Protocol), XML (eXtensible Markup Language), SOAP (Simple Object Access Protocol), and ATP (Active Transaction Pages). Communication network congestion or the presence of obstacles, in some cases makes difficult to get connected and this leads to "connection lost" status. That's the reason why in

some cases an autonomous system, like the one presented in the paper, is preferable.

Very often at the geographic and communication functionalities they integrate also one or more positioning systems, as GPS (Global Positioning System), Cellular Radio Location Tracking Systems or, more recently, RFIDs (Radio Frequency Identifications).

A service which provides the users with information based on their current positions is called Location Based Services (LBS). A location based service is therefore more than a mobile GIS, because in such a service we have to automatically account for data, spatial, user and device contexts. From this point of view the system we have designed and implemented and that is presented in the paper, is an example of location based service. Within this system the mobile GIS component represents the cartographic tool, while the a specific platform provides the system with context awareness.

Referring to the mobile GIS solution, which is the main part of the work we would like to present in this communication, we would like to make some remarks.

First of all it is worth to stress that, from the technological point of view, they take advantages also from the improvement in mobile hardware in the form of lightweight devices and field PCs. In fact mobile GIS developers have heavily to deal with hardware constraints such as the small screen size, the reduced colour palette, the low processor performance, the short battery life, the limited persistent storage and ROM. The latter aspects interact with the data processing and cannot be considered specific for cartographic data. On the opposite the formers are related to the map appearance and readability, an therefore a great attention must be paid in order to simplify the information on the screen without losing functionalities and to maintain as larger as possible the frame in which the map is displayed. Maps show usually a lot of layers (and very often aerial o satellite ortho-photos as background) and by convention most maps use different colors for each. Each layer must be adapted separately and, depending on the specifically architecture adopted, this adaptation can be on the mobile device, on the server side or both. Moreover at different scale different symbols could be used for the same feature. Maintaining these properties on small devices and at same time assuring other properties, such as interactive and continuous panning and browsing, is one of the most important challenge for developers.

1.2 Deriving mobile GISs from Web GISs

Another observation is that nowadays many solutions are available to publish maps on the Web. They are called Internet GIS or Web GIS and they can be used as a basis to derive mobile solutions. The interaction between Geographic Information and the Internet is being standardized by the OGC (Open Geospatial Consortium), which is a non-profit, international, voluntary standards organization composed at the moment of 340 companies, government agencies and universities participating in a consensus process to develop publicly available interface specifications.

Generally, in the Internet, XML (eXtensible Markup Language) is the language used for the definition and the description of applications. The communication is based (in the majority of cases) on the HTTP protocol and the usage of XML makes the Web Services platform and operating system independent. Also for the modeling, transferring, and storing of geographic information a special kind of XML, called Geographic Markup Language (GML), is adopted. It provides an open, vendor-

neutral framework for the definition of geospatial application schemas and features, but it isn't a presentation language for data display and therefore, to obtain presentation, GML must be converted in another XML graphical format.

The functioning of OpenGIS Consortium Web Services (OWS) can be summarized in four steps:

1. the client contacts the server and queries it about its functionalities;
2. the server sends back to the client an XML document containing the functionalities of the supported service.
3. the client asks the server for data.
4. the server provides the data as requested.

In case the service generates maps and makes them available as images or as a series of graphical elements, typically already projected in a given reference and coordinate system and with the already associated symbols and colours, it is called Web Map Service (WMS).

In case the result involves geographic entities (or features), which are represented in GML format and which can be edited and spatially analysed, it is called Web Feature Service (WFS).

This service is composed by the following basic functions:

- the readable description of the available features and the parameters related to the requests accepted by the service (GetCapabilities);
- the description of the features (data scheme) (DescribeFeatureType);
- the supply of "Feature"-type objects (instances) (GetFeature);
- data manipulation (for example operations such as "create", "update" or "delete") (Transaction);
- the application of a lock-request to one or more instances during a transaction (LockFeature).

The first three of the previously seen functionalities are mandatory, the last two are optional. In particular WFSs are described as basic or "read-only" mode, if they support the GetCapabilities, DescribeFeatureType and GetFeature operations and transactional (WFS-T) if they support also the Transaction and LockFeature operations.

As we will see, part of the specifications and the products already available for the Internet GIS can be re-used for the mobile GIS implementation.

Transferring a Web GIS application to a mobile device appears convenient almost for client-server solutions, because the architecture can be already compliant to mobile GIS purposes. But critical aspects seen above interesting mobile device and mobile GISs – mainly small screen size, low processor performance, and number and type of layers drawable in a map – must be taken into account and faced in any case.

2. FREE AND OPEN SOURCE SOLUTIONS

In occasion of the last Free and Open Source Software for Geoinformatics (FOSS4G2006) Conference held in Lausanne in September 2006, a first meeting on Mobile GIS took place. The same day the OSGeo (Open Source Geospatial) Foundation, announced the launch of the mobile GIS open source project, aiming at providing various GIS solutions for a variety of mobile devices.

The main objectives (Reimer, et al., 2007; OSGeo, 2007) are to provide fully Open Source solutions, to assure compatibility with a large variety of mobile devices and portability across different operating systems (Linux, Palm OS, Symbian OS, Windows CE/XP Embedded), to guarantee multi-language support, to build up highly configurable user interfaces, to

integrate the mobile GIS with some positioning systems, to create a bi-directional transfer mechanism for information to and from the mobile device.

Two languages are suggested for development: Java, which should technically work on any OS (operating system) with preinstalled Java Virtual Machine, and C/C++, even if in that case the GUI (graphical user interface) is not portable to various OSs, that require a separately developed GUI.

The idea is to exploit as much as possible all tools already available in the Open Source world, such as development environments for building spatially-enabled Internet applications (e.g. UMN Mapserver, Mapbender, MapBuilder, CartoWeb, ka-map, Geoserver, GRASS Server, MapGuide Open Source) (Blazek, and Nardelli, 2004; Bray, 2007; Camptocamp, 2007; Garnett, 2007; Lime, 2007; Mapbender, 2007; MapTools.org, 2007, Shorter, 2007) and utility cartographic libraries like GDAL, PROJ.4 or GEOS (GDAL, 2007; GEOS, 2007; Wamerdam, 2007).

At the moment there are numerous active projects. In the following we recall just the most known.

One of the first project was that aiming at porting GRASS on handheld Linux Operating System devices (Stankovic, et al., 2004). Another solution, also written in C language and working on Linux OS is the Maemo Mapper, specifically designed for the Maemo platform and Nokia Internet Tablet 770 (Maemo.org, 2006).

J2ME OGC WMS Client is a tool, written in Java 2 Platform Micro Edition (a collection of Java APIs for the development of software for mobile devices), that communicates with WMS compliant with OGC specifications (Skylab, 2006). The product is free for non commercial users and has been tested for a variety of devices, such as Siemens S65, BlackBerry 7290, Palm III, Tungsten T3, Nokia 6600, Sony Ericsson T616, iPAQ Pocket PC.

Similar solutions are WMSMobil J2ME, developed at the University of Regensburg (i-locate, 2006), and JVNMobileGIS 1.0, which gets data from WMS servers, automatically scales bounding box and image to best fit on the device screen, allows interactive actions on map view, supports GetFeatureInfo (OGC WMS specification) (JavaVietnam, 2006).

Mapping of mobile device position can be obtained by using Spot (Skylab, 2007) or Navlet (Hanak-Hammerl, 2005), which get positioning data from any NMEA External GPS receiver, connected via a serial cable or via Bluetooth. Navlet need a Java Virtual Machine (JVM) implementing an appropriate profile.

Finally two interesting ingoing project are: SMITA (Turistic Information System for Mobile Devices in the Atlantic Area) and MOSS4G (Mobile Open Source Software for Geoinformatics). SMITA is an Interreg IIIB (Atlantic Area) European project, based on OGC specifications and aiming at developing a system to enable users of mobile computing devices to access tourism, cultural and citizen information in all partner regions (SMITA, 2007). MOSS4G proposes to create, at the beginning on a Windows CE platform and using C/C++ programming languages, a GPS enabled application with data exchange and remote services (WFS-T) and raster support (WMS) (MOSS4G, 2007).

3. THE IMPLEMENTED MOBILE GIS

3.1 Architecture and general characteristics

Since last year a mobile GIS powered by Open Source solutions is being implemented within a project funded by the Italian

Ministry of University and Research (MiUR) and focused on the use of geoinformatics to support archaeology for both research and recreation exploitation.

Although conceived for archaeological issues, the mobile GIS can be adapted to other fields, like civil protection or fleet management; so, its architecture and technical characteristics are presented at first.

At the moment, a first version of the mobile GIS is completely implemented with basic GIS functionalities and functions to acquire and display information of user positioning (by GPS) and location of POIs (Points Of Interest, by RFID).

The mobile GIS works actually with a client-server architecture: the GIS engine lives inside a remote server and the mobile devices, which act as clients, access it through a wireless network (the Internet or other).

Geographic data and other information are also server side, excepting positioning, location and context data that are collected by sensors and receivers combined to the handheld device.

In a "pure mobile GIS" scenario (i.e. with sensors and receivers data acquisition disabled or not set) it's enough for a user to open its browser and access the mobile GIS link.

On the contrary, in a "LBS" scenario (i.e. the complete installation and configuration of the application) some drivers have to be installed on the mobile device in order to retrieve data from GPS receivers or proximity sensors like RFID tags.

For both the scenarios data and maps aren't stored on the client and the mobile GIS works in a read-only mode.

This solution has some critical aspects to be faced, deriving mainly from the strong dependence on the connection. For reducing and controlling these problems, autonomous networks can be suitable: in the next paragraphs we will see how the implemented mobile GIS was tested with a local wireless network.

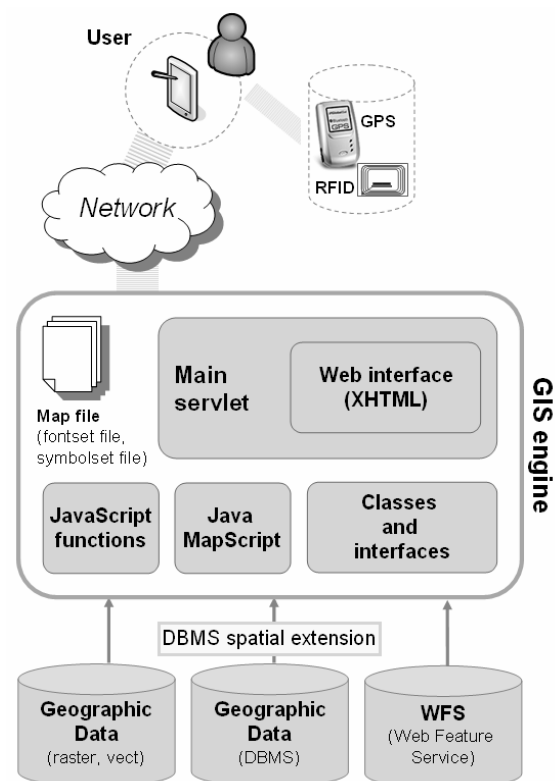


Figure 1. Mobile GIS architecture

3.2 The GIS engine

The GIS engine is a Java application implemented with MapServer Java MapScript 4.8.4 and running inside an Apache Tomcat 5.5.17 servlet container.

MapServer is an Open Source environment for developing and building spatially-enabled web applications, licensed by the University of Minnesota (UMN).

MapScript is the scripting interface to MapServer, generated using SWIG (Simplified Wrapper and Interface Generator) (Gillies, 2006), and Java is the language of MapScript adopted for this mobile GIS (other versions of MapScript are PHP, PERL, Python, Ruby, and C#).

The core of the application is a servlet which sends requests to and processes responses by Java MapScript, providing in this way GIS functionalities; it acquires information about data loading, configuration and rendering from a map file.

Moreover, the servlet generates and manages the web interface - loaded by the users into their browser and corresponding to a HTML template file in a standard CGI MapServer-based web GIS - and inserts into maps inputs coming from sensors (RFID) and receivers (GPS) combined to mobile device.

In detail, at the present step of the mobile GIS implementation the servlet manages these operations:

- the mobile GIS initialisation, by loading the map file, creating and visualising the web interface, and defining and initialising in it a usual HTML form, thought to manage the communication between the user's browser and the mobile GIS;
- the communication with the form and the mobile GIS throughout whole work session;
- map drawing and saving;
- map browsing operations (zoom factor evaluation, zoom by point, geographic extent updating);
- query operations (feature query, query by attributes, toggling collapsed and exploded query results, query map generation);
- dynamic legend writing and updating;
- scalebar drawing and saving;
- reference map drawing and saving;
- predefined views selection and activation;
- retrieving and visualisation of GPS coordinates;
- retrieving and visualisation of RFID coordinates;
- writing the web interface and setting its layout.

Some Java classes and interfaces have been implemented around the servlet in order to separate the call of a functionality (made within the servlet), its processing (within classes) and the definition of variables and parameters (within interfaces).

In particular, the process of a functionality consists in the execution of a set of operations, implemented within a class, required to call a Java MapScript method or a function implemented within the class itself.

At the current step of the mobile GIS implementation, these classes and interfaces are included into the application:

- MapFile: class for creating and saving map images, setting layer status, making pixel to coordinates conversions, drawing new layers and features;
- navigaCarta: class for map browsing functionalities: pan, zoom, predefined views;
- queryCarta: class for features and attributes queries;
- TOC: class for dynamic legend generation and management;
- scriviTempl: class for writing the web interface and its on/off elements;

- datiIntegrazione: interface that collects variables required by predefined views and query by attributes.
- varMS: interface that collects all other variables used in the mobile GIS.

3.3 The map file

The map file loaded by the mobile GIS is a standard MapServer's map file and all information about data loading and configuration is managed there or into the fontset file and the symbolset file, that set labelling and graphics respectively.

So, accordingly with MapServer capabilities, the mobile GIS can load several types of vector files and raster files, tables from spatial DBMSs and features retrieved by OWS.

The map file defines also other objects of the mobile GIS: the scalebar, the reference map and the query map.

Moreover, it contains parameters and metadata necessary for filling in the dynamic legend and specific layers for displaying coordinates coming from GPS and RFID. This last function is made by inserting GPS or RFID points as inline features of specific layers.

3.4 The web interface

The web interface is loaded by users within their browser and allows them to send requests to and obtain results from the GIS engine.

The web interface is an XHTML page generated by the main servlet at each session and contains a HTML form with the parameters used for setting inputs to be sent to the GIS engine.

A set of JavaScript functions, mapped within the GIS engine by the varMS interface, completes and improves GIS and graphic functionalities.

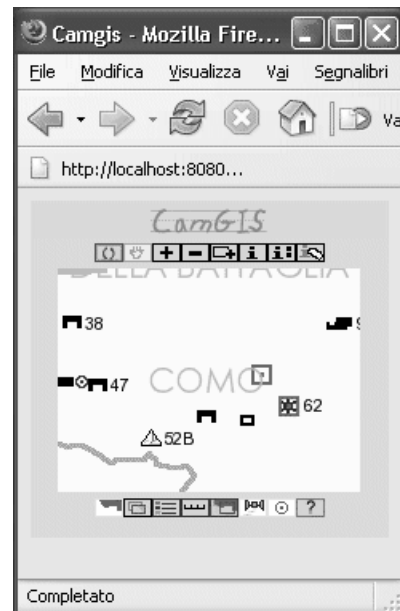


Figure 2. Screenshot of the web interface of the mobile GIS, with GPS and RFID data visualisations enabled

Since a typical client of the mobile GIS is a handheld device, the web interface has to be created and displayed accordingly. By consequence, it must fit small screen dimensions and face problems of page and image slow processing and loading. First ones have been faced by:

- reducing the map as little as possible, searching a mediation between small dimension (and consequently space saving) and map readability;
- avoiding text strings where possible and preferring icons to identify functionalities;
- splitting contents and objects (e.g. map, dynamic legend, scalebar or query results) in different sections of the web interface, turnable on or off on demand;
- presenting long lists (such as query results) with collapsed or exploded mode.

The problem concerning slow processing and loading of page and images is not fixed yet. In fact, it can be certainly reduced by decreasing the number of graphic elements (symbols and icons) to be loaded, but not solved completely as long as mobile device processors have low performances.

AJAX (Asynchronous JavaScript And XML) solution could be an interesting way to go, but its compatibility with browser for mobile devices has to be checked in advance.

3.5 Mobile GIS objects and functionalities

Within the web interface of the mobile GIS it is possible to display and toggling (turning on or off) these GIS objects:

- main map;
- dynamic legend;
- scalebar;
- reference map;
- list of predefined views;
- query results;
- query map.

Only the main map is displayed when the mobile GIS is initialised; at each session toggling an object is independent of toggling the other ones.

At the moment the mobile GIS has the following functionalities:

- map refreshing/updating;
- pan, zoom in, zoom out (by mouse click);
- feature query;
- query by attributes;
- explode/collapse query results;
- clear query results;
- display GPS-based user's position;
- display RFID-based POI location.



Figure 3. Screenshot of the module for attribute-based queries

First eight items are basic GIS functionalities, while the two last are LBS functionalities and make the mobile GIS aware of the context where it is used.

Really, other capabilities of context awareness have been thought for this application, like awareness of brilliance/luminosity, of the adopted mobile device or of the contents deepening.

At the moment, only the awareness of brilliance/luminosity and its effects on setting the contrast of the interface is being implemented, while other purposes have been postponed to next steps of the project.

3.6 Some words about the platform of context awareness

The mobile GIS, we said, acts as a location based service and has some capabilities of context awareness.

These functionalities are managed by a platform of context awareness or SAF (Situation Aware Framework), which controls informative fluxes coming from sensors and receivers up to the web interface (web application).

The description of the platform lies outside the topic of this paper, so only a short presentation will be given here.

The SAF was implemented by CEFRIEL (ICT Center of Excellence For Research, Innovation, Education and industrial Labs partnership) for a previous project and it has a Web based architecture organised in three logical levels (Brioschi, et al., 2006).

- The “Sensor management level” manages sensors to obtain location data; at this level user positioning and POI location are acquired by GPS receivers and RFID readers and tags respectively.
- The “Context framework” aggregates and manages context information coming from different sources, like positioning and location or brilliance/luminosity.
- The “Web application” delivers services, adapting contents and layouts to the context data collected by the context framework, and interacts with the proper mobile GIS.

The platform is composed by two independent processes, the location server and the context server, and by sets of Java classes and XML documents for every logical level.

Within the web application level an adaptation framework collects information from lower levels and, basing on a special adaptation model, sets what to present to the user and how to present it.

We assigned to the combination of SAF and the mobile GIS the name “Context Aware Mobile GIS” (CAMGIS).

4. USE CASES OF THE MOBILE GIS

4.1 Archaeological mobile GIS of Comum Oppidum

The implementation of CAMGIS belongs to a project thought for archaeological purposes.

By consequence, the first instance of the application has been planned and implemented for an archaeological context, i.e. the preroman settlement of Comum Oppidum, within the Spina Verde Regional Park, near Como (Northern Italy).

The archaeological mobile GIS of Comum Oppidum presents all the characteristics described above, so only some information about its geographical and archaeological data, and a description of a test made on-the-field with an autonomous local network will be presented in the next rows.



Figure 3. The mobile GIS displayed in a pocket PC

This mobile GIS loads a data subset extracted by the database of the Web GIS ArchaeoGEW (Brovelli, and Magni 2003); thus it can be considered as derived from an existent Web GIS (see Section 1.2).

In detail, data are both proper archaeological data - i.e. archaeological remains, loaded from a table of PostgreSQL 8.0.1 by PostGIS 0.9 – and background data, available both as ESRI shapefiles and tables of PostgreSQL again.

In order to display coordinates coming from GPS receivers, WGS84 has been chosen as reference system; by consequence all data have been converted from its native reference system, i.e. the Italian Gauss Boaga (West zone), to WGS84.

The archaeological mobile GIS was tested on-the-field with an autonomous local wireless network (see Figure 4) based on a Cisco Aironet 1300 access point (with Power Injector BLR2T associated) and powered by a solar panel with a support battery and a power controller (AGF Italy and Helios technologies).

The user's mobile device was a HP iPaq 5450 (with upgrade to Windows Mobile 2003), with a RFID reader (CF-II ACG 13.56 MHz), inserted at top of the mobile device, and a GPS receiver (GlobalSat BT-338), connected via Bluetooth.

The server was a Toshiba T5500 1.66 GHz, with Windows XP OS onboard and powered by own battery.

Passive RFID tags were placed near or on (where possible) archaeological remains.

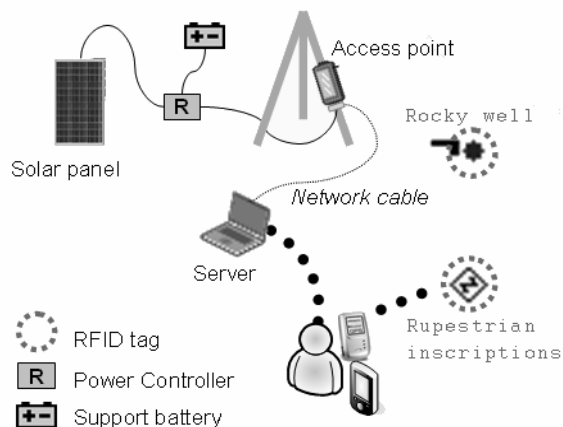


Figure 4. Autonomous local wireless network for the archaeological mobile GIS of Comum Oppidum

In this infrastructure, a user (as a client) walked along a section of the archaeological site, connecting and using the mobile GIS and its associated sensors, in order to know her/his own position (by GPS) and the archaeological remains locations (by RFID).

The test confirmed problems concerning slow loading of page and images we said above (about 20-40 seconds depending on the page to be loaded) and pointed out some critical aspects regarding the rate of page refreshing, GPS, and RFID.

The first one is due to the reloading of the web interface every time the GPS user position changes. In fact, mainly if the user is walking, the GPS position changes more quickly than the page is refreshed by the mobile device processor, and the user experience is made worse. By consequence, we think that a solution with a controlled page refresh is suitable: in this case the page is reloaded only if a condition is verified, for example if the user position varies more than a given distance.

Critical issues of GPS concern the accuracy of the adopted receivers and possible side effects, e.g. due to the Kalman filter embedded inside the receiver software (Brovelli, et al., 2007). They will be studied during other tests with other GPS receivers also.

At the end, RFID presents some open issues due to the real effectiveness of that technology for archaeological contexts. In fact, a passive RFID tag can be intercepted by the RFID reader only within a distance of no more than 10-20 cm. Is this limitation too hard?

We think that this question has an affirmative answer if the RFID method is seen only for location purposes and for outdoor archaeological contexts where monuments and remains are punctual and without a complex level of exploration, i.e. where it's enough to know "that is the archaeological evidence" and not to know its structural or decorative characteristics, like mosaics or particular architectonic elements.

But if the RFID technology is thought for indoor and/or complex contexts or, in general, not only for location purposes but also for information retrieving purposes, we think it can be useful and interesting. A big rock with many inscriptions, all marked with RFID tags, could be a significant example: a visitor doesn't use her/his RFID reader for localising the rock, but for retrieving information about every inscription on it.

Of course, the range of passive RFID technology remains a critical issue and could be bettered with the dual active RFID technology. However, the second one has other items to be considered, because it's more expensive and its tags require supplied power and maintenance.

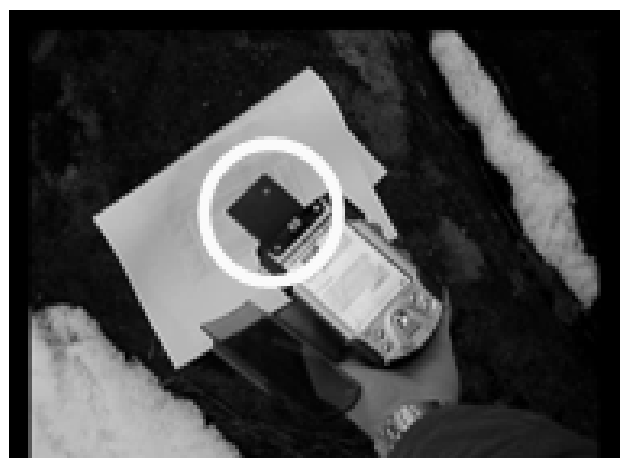


Figure 5. A user is reading a RFID tag with its mobile device. The circle shows the RFID reader.

4.2 Other use cases

The implementation of two other instances of the mobile GIS are just started and haven't, at the moment, a mature level of completion, so they will be only mentioned in this paragraph.

The first one is another archaeological mobile GIS, implemented for the site of Meta Sudans at Rome, very close to Coliseum.

In this case the archaeological site is a urban Roman one and it is being excavated. So, a possible user can be not only a tourist, but also an archaeologist employed in the excavation. By consequence there are two possible user's profiles. At the moment the mobile GIS cannot match the requirements of an archaeologist user, because it is a read-only mobile GIS. In fact, an archaeologist needs to do transactional operations, like data entry or editing, and some other tools supporting her/his on-the-field activity, like reporting systems or excavation blogs and forums.

The second one is a mobile GIS to be adopted in use cases of civil protection and its geographic context is the district of Como (Northern Italy). Also for this application both a read-only user or a transactional user can be thought.

Geographic information loaded in the mobile GIS is composed by background data, transportation data, at-risk industries, vulnerable elements (like schools, churches, recreational centres), and resources of civil protection (like first aid points, hospitals, fire-brigades, police offices or flooded areas).

5. NEXT DEVELOPMENTS

Future developments of the project will involve several open issues seen above, the improvement of the application with new properties of context awareness and a new implementation of the "model of the environment" made with GML.

Critical aspects that will be faced on next months are those required for improving GPS and RFID capabilities and for bettering the user experience, thus to check the web interface with new mobile devices, OSs and browsers, to speed up loading of page and images, and to fix some minor bugs of GIS functionalities (e.g. readability and length of dynamic legend).

The context awareness of the application, we said, could be upgraded with new purposes, like awareness of brilliance/luminosity, adopted mobile device or contents deepening. At the moment only the first one is planned and being implemented already with a new skin and a new map file for high contrast situations.

With the "model of the environment" we refer in general to three elements: the map (i.e. the environment which the mobile GIS refers to and where it is used in), the location model (that describes the user's position), and the context model (that describes specific conditions which the user is in when he uses the mobile GIS).

Actually these three models are distinct and described with different technologies: XML for location and context model, geographic data (ESRI shapefiles, raster images and rows of spatial tables) for the map. But map, context model and location model aren't so far and conceptually they can be considered as specific aspects of a unique model of the "world" or "environment" where a user of the mobile GIS "lives". So, can be interesting, in our opinion, to unify the three models in a unique one, with a technology that can inherit the information given by the existent models, and merge them.

The GML has been signed out as the technology more suitable to do it.

The implementation of the unique GML-based model isn't started yet, but it will be a topic of the project for next months. After a first analysis, we can say that the context model is the model that is less translatable in GML, because that language is not conceived for issues of context awareness. Instead, the description of the location model and the map seems to be well matched by GML capabilities.

At the end it's important to underline that a GML-based "model of the environment" is right to be transmitted through the Web, in particular by WFSs (see Paragraph 1.2). In fact, it's possible to think a scenario where a GML model is shared by a service that acts as a "model provider" and acquired by a "model user" by means of a WFS GetFeature. Then the "model user" uploads the model into his mobile GIS, that becomes centred on the environment described by the model. In an archaeological use case, the agency that manages an archaeological site can be the "model provider", publishing the model of the archaeological site on its website and allowing it to a WFS. Instead, visitors of the archaeological site can be "model users", because they could download the model of the archaeological site and upload it into their mobile GISs.

6. CONCLUSIONS

At the current step of its development, the implemented mobile GIS presents basic GIS functionalities, user's positioning and location of POIs. The two last functionalities can be considered, together with other ones (like the adaptation of the web interface to brilliance/luminosity conditions) as properties of context awareness: in this way the mobile GIS becomes a context aware mobile GIS.

The first instance of the application is for an archaeological site and has been tested on the field with an autonomous local wireless network.

Some open issues coming from tests and new implementations will address next activities of the project, in order to better the user experience of the mobile GIS, to improve its context awareness and to unify the map, the location model and the context model in a unique model based on the GML language.

ACKNOWLEDGEMENTS

We want give a special thank to M.Brioschi, F.Corcoglioniti, and M.Legnani, of CEFRIEL (ICT Center of Excellence For Research, Innovation, Education and industrial Labs partnership of Milan), which actively participate at the project implementing the context awareness platform and the connection among the proper mobile GIS, mobile devices and associated sensors (RFID reader and tags) and receivers (GPS).

The work is partially funded by the MiUR (Italian Ministry of University and Research), within the FIRB 2003 project "ArcheoMedSat: progetto e sperimentazione di rilevamento satellitare e web gis per la valorizzazione di monumenti e siti archeologici del Mediterraneo tra Oriente e Occidente".

REFERENCES

Blazek R., and Nardelli L., 2004. The GRASS Server. Proceedings. of the FOSS/GRASS Users Conference 2004, Bangkok, Thailand, 12-14 September 2004.

- Bray R., 2007. MapGuide Project Home. <http://mapguide.osgeo.org/home> (accessed 28 Mar. 2007).
- Brioschi M., Brovelli M.A., Legnani M., and Magni D., 2006. Implementazione di un mobile GIS sensibile al contesto archeologico di utilizzazione. *Atti X Conferenza Nazionale ASITA, Bolzano, Italy, 14-17 November 2006, Vol. I, pp. 471-476.*
- Brovelli M.A., and Magni D., 2003. An archaeological web GIS application based on MapServer and PostGIS. *The International Archives of Photogrammetry and Remote Sensing.*, vol. XXXIV part 5/W12 Commission V, pp. 89-94.
- Brovelli M.A., Realini E., Reguzzoni M., and Visconti M.G., 2007. Comparison of the performance of medium and low level GNSS apparatus, with and without reference networks. *Proceedings of the 5th International Symposium on Mobile Mapping Technology, Padova, Italy, 28-30 May 2007.*
- Camptocamp SA, 2007. Welcome to the Home of CartoWeb! <http://www.cartoweb.org/> (accessed 28 Mar. 2007).
- Garnett J., 2007. What is Geoserver. <http://docs.codehaus.org/display/GEOS/Home> (accessed 28 Mar. 2007).
- GDAL, 2007. GDAL - Geospatial Data Abstraction Library. <http://gdal.osgeo.org/> (accessed 28 Mar. 2007).
- GEOS, 2007. GEOS (Geometry Engine - Open Source). <http://geos.refractor.net/> (accessed 28 Mar. 2007).
- Gillies, S., 2006. MapScript API Reference. <http://mapserver.gis.umn.edu/docs/reference/mapscript> (accessed 26 Mar. 2007).
- Hanak-Hammerl R., 2005. NAVLET. <http://www.navlet.org/> (accessed 28 Mar. 2007).
- i-locate, 2006. i-locate WMSMobil. <http://www.wmsmobil.de/> (accessed 28 Mar. 2007).
- JavaVietnam, 2006. JVNMobileGIS 1.0 released. <http://www.javavietnam.org/javavn/jvngis/home2.html> (accessed 28 Mar. 2007).
- Lime S., 2007. Welcome to MapServer. <http://mapserver.gis.umn.edu/> (accessed 28 Mar. 2007).
- Maemo.org, 2006. Maemo Mapper. <https://garage.maemo.org/projects/maemo-mapper/> (accessed 27 Mar. 2007).
- Mapbender, 2007. Welcome und Willkommen to the Mapbender Project. http://www.mapbender.org/index.php/Main_Page (accessed 28 Mar. 2007).
- MapTools.org, 2007. ka-Map. <http://ka-map.maptools.org/> (accessed 28 Mar. 2007).
- MOSS4G, 2007. MOSS4G - Mobile Open Source Software for Geoinformatics. <http://moss4g.telascience.org/> (accessed 28 Mar. 2007).
- OSGeo, 2007. OSGeo Mobile Solutions. http://wiki.osgeo.org/index.php/Mobile_Solutions#Mobile_GIS_project (accessed 27 Mar. 2007).
- Reimer S., Bowden T., Loskot M., and FOSS4G2006 Mobile GIS BOF Attendees, 2007. FOSS4G2006 Mobile GIS BOF Minutes. http://wiki.osgeo.org/index.php/FOSS4G2006_Mobile_GIS_BOF_Minutes (accessed 27 Mar. 2007).
- Shorter C., McWhirter B., 2007. MapBuilder Home. <http://communitymapbuilder.org/> (accessed 28 Mar. 2007).
- Skylab Mobilsystems Ltd., 2006. J2ME OGC WMS Client. http://www.skylab-mobilsystems.com/en/products/j2me_wms_client.html (accessed 28 Mar. 2007).
- Skylab Mobilsystems Ltd., 2007. SPOT. <http://www.skylab-mobilsystems.com/en/products/spot.html> (accessed 28 Mar. 2007).
- SMITA Project, 2007. SMITA (Atlantic Territorial Mobile Information System). <http://www.smita.eu.com/> (accessed 28 Mar. 2007).
- Stankovic J., Neteler M., and Flor R., 2004. Mobile Wireless GRASS GIS for Handheld Computers Running GNU/Linux. *Transactions in GIS, Vol 8 (2), pp. 225-233, DOI: 10.1111/j.1467-9671.2004.00177.x.*
- Wamerdam F., 2007. PROJ.4 - Cartographic Projections Library. <http://proj.maptools.org/> (accessed 28 Mar. 2007).