

# AN OPEN SOURCE RICH CLIENT SOLUTION FOR ARCHAEOLOGICAL MOBILE GIS

M. A. Brovelli<sup>a,\*</sup>, D. Magni<sup>a</sup>, M. Orlando<sup>a</sup>, F. Rottoli<sup>a</sup>

<sup>a</sup>Politecnico di Milano, Polo Regionale di Como, via Valleggio 11, I-22100 Como (CO), Italy – (maria.brovelli, diego.magni)@polimi.it; mauro.orlando@katamail.com; rotoffili@gmail.com

**KEY WORDS:** Mobile GIS, Graphic Interface, Computer Vision, Visualization, Archaeology, Open Systems

## ABSTRACT:

A Rich Client solution based on Open Source technologies has been implemented in order to realize a mobile GIS application for the exploration of archaeological data. The solution, named RiCOMGIS, is divided in two separate modules: MapGenerator and Front-End Producer. MapGenerator acts as the server-side GIS part of RiCOMGIS. It is basically a Java application ad hoc implemented, which makes use of UMN MapServer Java MapScript. Front-End Producer is responsible for the generation of the graphical user interface (called RiCOMGIS Front-End), produced at run-time using the functionalities offered by the Open Source platform OpenLaszlo. RiCOMGIS Front-End acts also as a client by communicating with MapGenerator through an AJAX (Asynchronous JavaScript and XML)-like mechanism, in order to acquire information from the server or to ask for the generation of only those elements of the interface that have to be updated. RiCOMGIS has several advantages in respect to traditional mobile GIS. First, its modularized structure results in ease of maintenance. Second, no software must be pre-installed on the user device. Third, the possibility to update only specific elements of the interface, in spite of entirely refreshing it at each interaction, improves the user experience, reducing at the same time the bandwidth consumption, that is an important aspect for client-server architectures and software designed to be utilised on mobile devices. The application is still a prototype and its implementation is in progress. However it presents a stable version that provides GIS functionalities and displays maps of a pre-roman archaeological settlement near Como (Italy).

## 1. INTRODUCTION

New technology and scientific developments have been changing the traditional use of GIS (Goodchild, *et al.*, 2004). Starting from a situation in which data and functionalities were stored in the same machine at the same workspace, we are now moving towards a scenario where we can deal with geography everywhere and at every time. The advantages of such a kind of improvement are evident in most applications, but specifically when a prompt intervention, as in case of emergencies, is required.

Computing technology developments, such as the availability of laptop computers, PDAs (Personal Digital Assistants) and smartphones, equipped with new operating systems and software components, allow people to perform geographic analyses and processing while moving in the field, with a great advantage with respect to simulate or post-process data in the laboratory.

Unfortunately the miniaturization of devices leads to different limitations such as reduced screen size and input interface, microprocessor, primary and secondary storage.

The user interaction interface requires specific implementations, such that developed with our prototype, in order to improve the usability of the mobile GIS.

The computing power limitation can be overcome by using the mobile device in a client-server architecture, as an extension of Web GIS technology (Li, 2007). In that case wireless connectivity and communication are core issues allowing

mobile devices to exchange voice, data and services with remote resources (Peng, and Tsou, 2003).

Communication network congestion and presence of obstacles are the main problems degrading the connection. Moreover, management of massive data sets is difficult and the request-response delay become higher if the hand-held device has hardware (e.g. the CPU) and/or software (e.g. the Internet browser) with low performances.

Furthermore, unstable connection, delays and high response times become even more critical because of the considerable expense that a wireless connection still requires.

A solution could be the usage of an autonomous local wireless network, possibly powered by solar panels with a support battery and a power controller (Brovelli and Magni, 2007).

Different client-server (as well as stand-alone) solutions are available in the free-open source and proprietary world.

In the Open Source environment, jvnmobilegis (Google, 2007), is a cross-platform mobile map viewer sponsored by the Information Center of Natural Resources and Environment of Vietnam. It is developed by the Google Code team with Java Micro Edition technology. It gets data from WMS servers, automatically scales bounding box and image to best fit on the device screen and allows interactive actions on map view (JavaVietnam, 2006).

Mapping of mobile device position can be obtained by using Spot (Skylab, 2007) or Navlet (Hanak-Hammerl, 2005), which

---

\* Corresponding author.

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.

GeoTracing (GeoTracing, 2007) is a software platform for creating multimedial geo-applications that allow the users to tell their "multimedial story" (by means of images, videos, audio files or text) about their movement through the landscape. It has been implemented in J2ME environment; the used technologies are KeyWorx (core technology), Pushlets (for real-time browser events), AJAX (Asynchronous Javascript And XML), and Google Maps.

A mobile version of gvSIG GIS program has been recently (on March, 2008) released in its first version gvSIG Mobile Pilot (BN44) (gvSiG, 2008). The software is being developed by Prodevelop and the Robotics Institute of the University of Valencia in a J2ME environment as an Open Source GIS client to be used in mobile devices with a wide set of capabilities.

Finally, some 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 (Open Geospatial Consortium) specifications and aiming at developing a system to enable users of mobile 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, Transactional Web feature Services) and raster support (WMS) (MOSS4G, 2007).

In the proprietary environment, ESRI (ESRI, 2007) proposes the "traditional" ArcPad, mainly devoted to fieldworkers, and ArcGIS Mobile®, a software developer kit (SDK) installed as part of ArcGIS Server® that allows developers to build customized applications in both stand-alone and client-server architecture.

MapInfo offers different solutions for mobile applications (MapInfo, 2007), mostly server side: MapXtreme® Java™ Edition, MapXtreme® For Windows®, and MapXtreme® 2005 (for .NET developers). On the opposite, at client side MapX® Mobile is proposed, a SDK allowing the implementation of stand-alone or wireless connected applications.

IntelliWhere® family, the global Intergraph's Mobile GIS solution (Intergraph, 2007), is composed by different pieces of software: IntelliWhere® LocationServer is the server side component and delivers and receives location-based information and services; IntelliWhere® OnDemand is the client application that runs on Pocket PC devices and Trimble GeoExplorer® hand-helds.

Autodesk (Autodesk, 2007) has two location service solutions, grouped as Autodesk® LocationLogic™; Autodesk Insight™ is for fieldworkers' crew management and Family Minder is specifically for parents or guardians.

## 2. RICOMGIS IN DETAIL

### 2.1 Implementation of client-server applications based on Open Source software

Since 2005 the Geomatics Laboratory of Politecnico di Milano – Como Campus is dealing with the design and the implementation of mobile GIS Open Source software solutions. The first implemented application is NAMGIS (Nomadic Adaptive Mobile GIS), an Internet-based mobile GIS developed in collaboration with CEFRIEL (Center of Excellence For Research, Innovation, Education and industrial Labs partnership) within the ArcheoMedSat project, financed by the Italian Ministry of Education, University and Research.

NAMGIS is based on a client-server architecture and aims at providing moving users with basic GIS functionalities, improved by properties of context awareness. In detail, users can load the interface of NAMGIS into the browser of their hand-held devices, simply writing the correspondent URL. Then, they can access, browse and query maps and data which describe the territory they are going through. That is possible by means of the following GIS functions:

- browse functions (pan, zoom in, zoom out, zoom by rectangle, predefined views);
- query functions (feature query, attribute-based query, exploded/collapsed query results, query map);
- legend and table of contents for layer management;
- other objects (scalebar, reference map).

Moreover, NAMGIS is improved with properties of user positioning, location of points of interest (POIs) and context awareness.

User positioning is obtained by means of a GPS receiver associated to the hand-held device; acquired user's coordinates are displayed on the map at real time. When that functionality is enabled, the user can choose if constrain the program to draw maps centring them onto the user's position every time it changes (or when it changes more than a given distance) or not. Location of POIs is obtained by means of the RFID (Radio Frequency IDentification) technology. Every POI is provided with a passive RFID tag, while a RFID reader is coupled to the hand-held device. When a user is close to a POI, the RFID reader reads the correspondent passive RFID tag and triggers related information, in particular its coordinates, that are displayed on the map.

The properties of context awareness allow NAMGIS to change accordingly with the conditions of the context where the user is or moves. At the moment two properties have been implemented: the awareness of sunlight conditions, that acts on the interface contrast (high contrast for strong sunlight, normal contrast for low or normal sunlight), and the bilingual support (available languages are English and Italian).

Server-side, NAMGIS elaborates available geographic data producing maps and related information, manages and processes information of positioning, location and context, organises all those contents in a unified and coherent response and sends it to the clients.

The capabilities described above are provided by an application which is the integration between a pure web GIS (NAMGIS Core), implemented by the Geomatics Laboratory, and the SAF (Situation Aware Framework), realised by CEFRIEL.

NAMGIS Core acts as the GIS engine of NAMGIS and is composed by a Java servlet and some classes implemented ad hoc and running inside an Apache Tomcat 5.5.17 servlet container. The application relies on MapServer Java MapScript 4.8.4 in order to retrieve atomic GIS operations, selected and assembled to provide GIS functionalities and to draw maps and related objects, such as legend, reference map, and scalebar (Brovelli and Magni 2007). MapServer is an Open Source environment for developing and building spatially-enabled web applications, released by the University of Minnesota. Here MapScript, i.e. the scripting interface of MapServer, has been adopted in its Java version, in order to facilitate the integration with the SAF, written in Java itself.

Moreover, the servlet generates and manages the web interface and inserts into maps positioning and location information, coming from receivers (GPS) and sensors (RFID) combined to the mobile device.

Acquisition and management of sensors information, as well as available capabilities of context awareness, are provided by the SAF (Brioschi, *et al.*, 2007).

Starting from 2007 the Geomatics Laboratory of Politecnico di Milano – Como Campus planned a new project in order to evaluate the possibility of exploring new quickly spreading programming and development technologies (namely AJAX-like client-server communication mechanisms and client generation through XML-based mark-up languages) to realize a new version of NAMGIS (although without context awareness) specifically thought and optimized to improve the user interaction's experience, to produce a more easily adaptable and maintainable software and to face the problem of low connection bandwidth which generally affects mobile devices. The mobile GIS produced as result of this project, called RiCOMGIS (Rich Client Open source Mobile GIS), is the main topic of this article and will be presented in the following subsections.

## 2.2 RiCOMGIS main characteristics and functionalities

**RiCOMGIS main characteristics:** RiCOMGIS stands for "Rich Client Open source Mobile GIS", a name that clearly summarize the main characteristics of the web GIS developed. These are analyzed in the following.

First, RiCOMGIS is a web GIS that uses a client-server architecture in which the client can be denoted as a rich client. Regarding this point it is necessary to precisely state what we mean for "rich client" since the lack of a standardized definition has produced the effect that this term is sometimes used by different people with different meanings.

For "rich client" we denote here and in the following a client that offers nearly the same deployment ease as a thin application with nearly the same enhanced interactivity and usability of a traditional desktop application, and that lies between fat and thin clients regarding the way the three tiers that generally compose at a conceptual level a web application (namely, view, business logic and data access) are partitioned between the client and the server.

Besides, RiCOMGIS is completely based upon open source technologies and products, in particular: PostgreSQL as DBMS with PostGIS extension to support geographical data, UMN MapServer (through its Java API called Java MapScript) to

generate maps, legends and scalebars, and OpenLaszlo by Laszlo Systems, Inc. for the generation of the user interface. Eventually RiCOMGIS is a GIS specifically developed and optimized for mobile devices.



Figure 1. RiCOMGIS on a hand-held device.

**RiCOMGIS functionalities:** RiCOMGIS provides to the user the following functionalities:

- Pan through drag and drop: the user can click on the main map and drag it to reposition it; a new map that covers appropriately the entire map space is then immediately generated.
- Rectangle zoom: the user can click on a point in the map and start drawing a rectangle including the desired map region, that is then magnified so to fully cover the map space.
- Zoom in and zoom out: by activating the "zoom in" or, alternatively, the "zoom out" function, a panel to choose the desired level of zoom (the possible values are two, three, four and five) is presented to the user and the map is then zoomed according to the chosen level.
- Layers selection: by pressing the appropriate button a list with all the layers names specified in an application's file containing all the default settings for map generation and visualization (the so-called MapFile) is presented to the user, together with two buttons to respectively activate (that is, to make visible on the map) or deactivate all the layers with a single click. Furthermore, each listed layer has on its left side a square that can be checked or unchecked in order to activate or deactivate it independently from the other layers. The list presented to the user has all the layers used to generate the last map already checked.
- Legend visualization: the user can choose to view all the active layers with their associated symbols or colours on the map in a simple scrollable list.
- Reference map visualization: a button allows the user to view on the right-bottom corner of the screen a map of the entire geographical area the user can explore, with a rectangle inside it that includes the region represented on the main map.
- Scalebar visualization: by activating this function the user can see on the bottom of the screen a graphic representation of the main map's scalebar together with the explicit indication of the nominal scale value.

- Query by point: when the user enters the query by point mode, if there is at least one layer from those marked in the MapFile as queriable that is active, he can click on a point of the map and a query is automatically performed returning in a panel on the screen all the features (with their attributes) that have been found in the application's dataset and that are associated with that point. For example, by clicking on a point with the icon of an ancient home building remain, a number of properties such as the era which the remain belongs to, its conservation state, etc. are presented to the user.
- Query by attributes: by pushing the button to activate the "query by attribute" mode, a panel appears on the screen containing on its left the list of all the layers set as queriable in the MapFile. When the user selects one (e.g. the towns layer), its attributes (e.g. the area of towns' territory) are listed on the right so that the user can select one of them by simply clicking on it. Then, a new panel is shown on the screen, containing the attribute selected, a combo box to select the operation (choosing between >, >=, =, <, <=) to use in the query, a text-box to insert the value searched by the query and a button to launch the query. To search for examples all the towns that have an area minor than 30 km<sup>2</sup> it is necessary to select the layer of the towns, choose the operation < and insert in the text-box the value 30. When the query is performed all the features that are at present on the map and that matches the query criteria are shown to the user in a panel on the screen.



Figure 2. RiCOMGIS Front-End: Zoom by Rectangle.

### 2.3 The structure of RiCOMGIS

In order to show how RiCOMGIS reaches the objectives for which it has been designed (improvement in user interaction's experience, ease in maintenance and reduction of bandwidth consumption) it is necessary to look at how it is structured and how the different elements that compose it interact among them.

RiCOMGIS is basically composed by two modules: the server module called MapGenerator and the client generator (which resides on the server, too) called Front-End Producer.

**MapGenerator:** Basically, MapGenerator is responsible for the creation of maps (including the reference ones), legends and scalebars, according to the information delivered by the client through HTTP requests, in addition to provide the information

(e.g. queries results) that the client needs to perform its operations. This latter goal is achieved by sending to it XML response messages embedded into HTTP responses. The code of MapGenerator is organized into several ad-hoc developed Java classes. The two most important are RequestsManager and Engine, which together build up the core of MapGenerator.

In particular, RequestsManager is a servlet running into Apache Tomcat (though also other containers can be adopted without any trouble) that acts as follows:

- it constantly listens for client's HTTP requests and receives them when sent by a client;
- it checks if the received requests are compliant with a specific set of rules (a.k.a. the "application communication protocol") that dictate the name and the number of the parameters the client has to include in the request in order to ask for a specific operation to be performed;
- if the check fails it delivers to the client a HTTP response which embeds an appropriate XML message containing information about the missing or wrong parameters;
- if, instead, the check succeed it invokes a specific method (varying according to the request) of the class Engine and it returns the result (which is always an XML message) to the client, by embedding it into a HTTP response.

The class Engine, instead, is the class that concretely performs the operations required by the client and generates an XML message containing information about the result produced.

To perform the required operations the class Engine uses the functionalities provided by a number of auxiliary ad-hoc implemented Java objects together with those made available through the use of the already mentioned Java MapScript API. This last is used both to generate maps (including reference ones), legends and scalebars that are loaded into the graphical interface of the client, and to manage the communication with PostgreSQL (with PostGIS extension) to load data from the database.

Regarding this latter point it is important to note that the code has been written to be independent from the specific DBMS used (in fact, although RiCOMGIS uses PostgreSQL, also any other DBMS supported by UMN MapServer could be adopted) and from the specific organization of data into the database (considering for example the use of a Relational DBMS this means that the code is written so that there is no need to specify how many tables compose the database and what is their name or their structure). This also implies that the database can be modified or completely replaced with another without the need to change the source code of the class.

Besides, the class Engine reads the information and the settings for the generation of the maps and of the other elements of the interface (legend, scalebar and reference map) from a plain text file that can be easily edited and changed according to specific needs. This is the already mentioned MapFile, a file used in all the application built upon UMN MapServer and so whose syntax is well known to every MapServer user.

Also in this case the class Engine has been developed in a way that allows the user to edit the MapFile (with the only constraint to maintain information about the possibility to perform query

on the layers in each layer's "metadata" section) without any need to change the source code.

**Front-End Producer:** Front-End Producer is the part of the software that generates the client to be loaded into the browser of the user.

This module is structured as a web application residing on the server that the user can invoke by typing its specific URL and that returns a file compatible with Adobe® Flash player together with an HTML page with the code necessary to load the file into the browser (which obviously must be equipped with the Adobe® Flash player plug-in).

This file is then executed producing the user interface of the whole application, which acts also as the client of the RiCOMGIS client-server architecture.

Once the client (called RiCOMGIS Front-End) is active, it starts collecting user's input and, on the basis of it, sends appropriate HTTP requests to MapGenerator, in order to perform the necessary operations, namely the replacement of a specific element (i.e. a map) of the user interface with an updated or new version, or the rendering inside an interface widget (e.g. a scrollable panel) of the information returned, such as the result of a query on the map.

These operation are performed after RiCOMGIS Front-End has received the HTTP responses of the server, extracted from them the embedded XML messages and parsed the XML code so to acquire the information required.

The kind of communication just explained, based on the exchange of HTTP requests and HTTP responses embedding XML messages (a mechanism very similar to AJAX) was the first technology under experimentation in the RiCOMGIS project and it has proved to be very effective in allowing the download from the server of only those elements of the GUI that need to be replaced or alternatively loaded for the first time.

Thanks to this behavior it has been possible to improve the user experience avoiding the continuous refresh of the visual interface and, together with the fact that some information is included in very light XML messages leaving to the client the responsibility for generating their visual representation (such as in the case of the layers list), it has been possible also to reduce at the same time the bandwidth consumption.

It is now important to briefly look inside Front-End Producer. Front-End Producer is basically composed by the software OpenLaszlo, plus several files with extension ".lzx". These files contain the instructions for the generation of the user interface and for the management of the communication with MapGenerator, written in a language called LZX intertwined sometimes with the language ECMAScript (often imprecisely called JavaScript).

LZX is an XML-derived language conceived in order to ease the development of Rich Internet Applications and therefore it is particularly suitable to implement applications with well structured visual interfaces and with support to AJAX-like communication mechanisms, such as RiCOMGIS Front-End.

When Front-End Producer is invoked, OpenLaszlo reads from its execution directory the .lzx files, and thanks to its internal

functionalities it transforms them into a unique Adobe® Flash player compliant file.

This is the file that is returned to the user who has invoked Front-End Producer, together with the HTML page needed to load it into the browser..

At this point it is very important to say that there will be in the future the possibility (now still in beta) to generate DHTML (i.e. HTML mixed with ECMAScript and CSS) code in spite of an Adobe® Flash player compliant file, so to make possible to run the client on all DHTML fully compliant browsers, without any need for players supporting Adobe® Flash files.

The generation of a rich client through the use of an XML-based language such as LZX was the second technology that the RiCOMGIS project aimed to evaluate.

Considering the result produced it is possible to say that this approach has allowed developers to create quite advanced features together with a good-looking, easy to use and complete visual interface with a very limited amount of code.

**Conclusions on RiCOMGIS structure:** The structure of RiCOMGIS just described, in addition to the above mentioned advantages in term of better usability and low bandwidth consumption has also many advantages in respect to ease in software maintenance.

First, the fact that the two main components of RiCOMGIS (MapGenerator and Front-End Producer) have been developed as separate modules that need only to respect the rules prescribed by the application communication protocol for requests formulation, makes possible to change the code of one module without any need for changes in the other module.

Second, the mechanism of acquisition of the client from Front-End Producer avoids the need of an off-line client installation, allows the user to always have the most recent version of the software (so that the newest enhancements and bugs solutions are always available as soon as the code on the server is updated) and facilitates the work of software developers since the whole code of the application resides entirely on the server. Eventually, the fact that MapGenerator is largely independent from the structure of the MapFile and of the data in the DBMS allows to adapt RiCOMGIS to a number of different contexts with minimal efforts.

#### **2.4 An application of RiCOMGIS for archaeology**

The first implementation of RiCOMGIS deals with data and maps of Comum Oppidum, an archaeological site in Northern Italy.

The site is located at the South of the town of Como, on a little hill range, and is partially included in the territory of the Spina Verde Regional Park. The part which is not included into the park has both preroman and roman features and is swallowed up by the urban area of Como now.

Instead, the part which belongs to the park is almost preroman and, since lies in steep and wooded slopes, it is still preserved by urbanisation and can be organised in a network of visiting routes.

This second section of Comum Oppidum was chosen in 2001 as test site for the realisation of the web GIS ArchaeoGEW (Archaeological GIS Explored by Web\*; see Brovelli and Magni, 2003) and maintained for the implementation of NAMGIS and RiCOMGIS.

A visitor can find there a lot of preroman remains, like rupestrian inscriptions, dwelling structures, rocky manufactures, sandstone quarries, tombs, and living structures. They aren't organised in a homogeneous or continuous settlement, but emerge among trees and slopes like spots in the park.

By consequence it is important to place them correctly in a map, in order to aid a tourist to plan a visiting route and to move through the park. The second of these purposes can be well faced by a mobile GIS application (see Brovelli and Magni, 2007): visitors walk through the site with their hand-held devices and see maps and associated information onto their mobile GIS client application. In such a scenario RiCOMGIS acts both as the client application and as the map server; in fact, RiCOMGIS Front-End allows the user to display and explore, in a hand-held device browser, map and data provided server-side by MapGenerator.

The data model of this archaeological instance of RiCOMGIS comprises the following data:

- archaeological remains,
- contour lines,
- lakes,
- boundaries of the Spina Verde Park,
- municipalities,
- continuous urban areas,
- spotted urban areas,
- tourist points (such as historical villas or viewpoints),
- paths (pedestrian paths, cycle paths, cart ways).

All geographic information is in the WGS84 geographic reference system.

Archaeological remains and tourist points are tables of a PostgreSQL database, spatially enabled by PostGIS. Instead, other data are ESRI shapefiles.

Some data are described by associated attributes that can be retrieved by means of query operations. In detail, archaeological remains have information on identification (code, name), location (municipality, locality), positioning (positioning type, coordinates, elevation), archaeological characteristics (type, period, find out manner, exploration level, restoration), tourist exploitation (access path, accessibility, visibility, conservation, data access), and references (bibliography, archaeological group).

Other data provided with associated information are municipalities (attributes are: name, Italian *provincia*, population, surface, density, Spina Verde Park partnership), lakes (name), tourist points (type, code, name, municipality, locality, Spina Verde Park inclusion, access path), and paths (number, name, difficulty level, travelling time).

This application has all the functionalities available in RiCOMGIS, i.e. pan, zoom (by rectangle and by scale factor),

feature query, attribute-based query, layer management, legend, scalebar and reference map.

Moreover, by analogy with NAMGIS, properties of location and context awareness and support for GPS and RFID could be thought for improving RiCOMGIS.

### 3. CONCLUSIONS

RiCOMGIS is a client-server application that makes use of Rich Internet Application technology to provide a versatile web interface to access GIS data and maps from hand-held devices. The application presents a stable version which loads data and maps of an archaeological site in Northern Italy.

However, RiCOMGIS has been implemented to be utilized also for contexts different than archaeology.

RiCOMGIS is provided with a Flash web interface at the moment, but in future it could be improved with a DHTML interface, in accordance with the capabilities of OpenLaszlo, which it is based on.

Furthermore, the implementation of RiCOMGIS has taken into account the possibility to improve the mobile GIS with properties of location and/or context awareness, by means of a GPS receiver, associated or embedded in the mobile device, and/or RFID (Radio Frequency Identification) tags, placed on objects of interest (e.g. the remains of the archaeological site which RiCOMGIS refers to).

Other developments will be focused on the upgrading of GIS functionalities and on the OWSs (OGC Web Services), in order to separate the data providers and the GIS map server. In fact, the GIS server (UMN MapServer Java Mapscript embedded in MapGenerator) can act both as a client of OWSs, loading data and maps provided by other independent remote services, and as a server, presenting maps and functionalities to the user of RiCOMGIS. In this way, the user of RiCOMGIS could free himself from a particular geographic context and choose his favourite context to be loaded and visualised by RiCOMGIS.

### REFERENCES

- Autodesk, 2007. Autodesk - Products - Location Services. <http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=2774137> (accessed 23 Dec. 2007).
- Brioschi M., Brovelli M.A., Corcoglioniti F., Legnani M., Magni D. (2007). Implementazione di un mobile GIS sensibile al contesto archeologico di utilizzazione e prime prove d'uso. *Proceedings VIII Meeting degli Utenti Italiani di GRASS e GFOSS*, February, 15-16, 2007. Palermo, Italy.
- Brovelli M.A., and Magni D., 2007. Open Source mobile GIS solutions for different application fields. *Proceedings 5th International Symposium on Mobile Mapping Technology*, May, 28-31, 2007. Padova, Italy.
- 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.

\* ArchaeoGEW, Politecnico di Milano, Laboratorio di Geomatica. <http://geomap.como.polimi.it/agew/>.

- ESRI, 2007. Mobile GIS. [http://www.esri.com/software/arcgis/about/mobile\\_gis.html](http://www.esri.com/software/arcgis/about/mobile_gis.html) (accessed 6 Mar. 2008).
- Goodchild, M.F., Johnston, D.M., Maguire, D.J., Noronha, V.T., 2004. Distributed and mobile computing. In: *McMaster, R.B., Usery E.L. (Ed.). A Research Agenda for Geographic Information Science*. Boca Raton: CRC Press, pp. 257–286
- GeoTracing, 2007. GeoTracing. <http://www.geotracing.com/> (accessed 24 Dec. 2007).
- Google (2007). jvnmobilegis. <http://code.google.com/p/jvnmobilegis/> (accessed 24 Dec. 2007).
- gvSIG (2008). gvSIG Mobile. <http://www.gvsig.gva.es/index.php?id=descripcion&L=2> (accessed 8 Apr. 2008).
- Hanak-Hammerl R., 2005. NAVLET. <http://www.navlet.org/> (accessed 24 Dec. 2007).
- Intergraph, 2007. Intergraph Home page → Capabilities → Select Product families. <http://www.intergraph.com> (accessed 23 Dec. 2007).
- JavaVietnam, 2006. JVNMobileGIS 1.0 released. <http://www.javavietnam.org/javavn/jvngis/home2.html> (accessed 28 Mar. 2007).
- Li Q., 2007. Opportunities in mobile GIS. In: *Drummond J., Billen R., João E., Forrest D. (Ed.). Dynamic and mobile GIS*. CRC Press, Taylor & Francis Group. pp. 19-33.
- MapInfo, 2007. Products & Services: Mapping. [http://extranet.mapinfo.com/products/products\\_index.cfm?productcategoryid=1](http://extranet.mapinfo.com/products/products_index.cfm?productcategoryid=1) (accessed 23 Dec. 2007).
- MOSS4G, 2007. MOSS4G - Mobile Open Source Software for Geoinformatics. <http://moss4g.telascience.org/> (accessed 28 Mar. 2007).
- Peng Z.-R., and Tsou M.-H., 2003. *Internet GIS*. John Wiley & Sons.
- Skylab, 2007. SPOT. <http://www.skylab-mobilesystems.com/en/products/spot.html> (accessed 28 Mar. 2007)
- SMITA, 2007. SMITA (Atlantic Territorial Mobile Information System). <http://www.smita.eu.com/> (accessed 28 Mar. 2007).

