

A Web Map Service for Mobile Computers

Clodoveu A. Davis Jr.¹, Yuri Jorge Kimo², Fátima L. P. Duarte-Figueiredo²

¹Departamento de Ciência da Computação –
Universidade Federal de Minas Gerais (UFMG)
Av. Pres. Antônio Carlos, 6627 – Belo Horizonte – MG -- Brazil

²Instituto de Informática –
Pontifícia Universidade Católica de Minas Gerais (PUC Minas)
Rua Walter Ianni, 255 – Belo Horizonte – MG -- Brazil

clodoveu@dcc.ufmg.br, yurikimo@gmail.com, fatimafig@pucminas.br

Abstract. *One of the Grand Challenges for computing today involves providing access to large volumes of distributed multimedia data. Mapping and personal navigation are among the most interesting applications involving this challenge, since they require access to large volumes of data, both static (such as street maps) and dynamic (such as traffic conditions). Such applications are currently the focus of much attention, considering that hardware is improving and wireless networks are becoming ubiquitous. Access to online maps is mainly done nowadays through proprietary and thematically limited geographic information services, such as Google Maps. A more interesting alternative for Web-based map access is to use data sources from spatial data infrastructures based on the Open Geospatial Consortium's standardized Web services, such as the Web Map Service (WMS). WMS, however, imposes a higher communications overhead and power consumption to the mobile device, and has a limited scalability potential. This paper presents a proposal for the a WMS client architecture that presents significant gains over the current approach, especially regarding the data transmission overhead imposed by OGC Web services.*

1. Introduction

The convergence of mobile computing with Web-based repositories of geographic data and with geotechnologies, such as the Global Positioning System (GPS), has created a favorable setting for new mobile geographic information applications. Decreasing hardware prices and the integration of GPS into mobile phones and PDAs has provided many people with reasonably precise and costless data on personal location practically all the time. Considering the growing availability of wireless networks, including WiFi, GPRS and 3G, users are increasingly looking for useful location-based services and applications.

However, most geographic applications available today for mobile phones and PDAs are based on static geographic datasets, either downloaded to the device or available through regular Web access to sites such as Google Maps. Even though such datasets are important and functional, users look forward to being able to select among other available datasets, through which they can (1) fulfill specific needs, (2) have

access to more specialized local data, and (3) combine various information sources for better decision-making.

We observe that, in the mobile computing arena, geographic information services are still proprietary and limited. Companies such as Google, Yahoo!, Garmin, and Nokia have developed generic solutions that offer digital map browsing and provide simple mobile applications such as routing and locating nearby points of interest and landmarks. These applications require data to be updated constantly, reflecting today's intensive urban dynamics. Such updating is not easy, much less for global companies such as the ones mentioned. As a result, online maps and applications always fall behind reality in many aspects, thus leading to errors and loss of user confidence. Local data sources tend to be much more reliable (Davis Jr. and Fonseca 2005), although by nature such sources tend to be numerous and diverse.

There is also an important limitation regarding the possibility of integrating data from other sources in mapping environments. The integration of new data in a Web map is a privilege of its creator, although developers are increasingly able to create Web resources in which thematic data are plotted on top of a generic Web map backdrop, using JavaScript-based application programming interfaces (APIs), achieving the so-called *mashups*. Such information sources, however, remain scattered all over the Web, and there is no simple solution to integrate them all, enabling users to select and activate/deactivate new data as layers in a desktop geographic information system (GIS). A more interesting solution for discovering and accessing numerous geographic information sources through the Web is the use of service-oriented architectures (SOA). In SOA, data providers publish metadata on available datasets using a *catalog service*. Clients search this catalog, identifying services of interest (an activity known as *discovery*). The client can then connect (*bind*) to the selected service(s), get from them a list of available data classes (or *layers*), and select which ones are required for display. Data is then sent to the client for visualization (Tu and Abdelguerfi 2006; Davis Jr. and Alves 2007).

The SOA approach to data sharing has been conceived as an interoperability solution, initially by the Open Geospatial Consortium (OGC) (Percivall 2003), then redeveloped by the W3C (W3C 2002). Many connections and message exchanges are required, as described in the previous paragraph, since it is necessary to provide a neutral data exchange environment through the network. This approach has countless advantages. Clients do not need to know in advance which data is available at which server, since that information is available at the catalog. Servers can go offline, and replicas can take their place by changing the catalog. If there are no replicas, the user can select a similar service as a data source. Clients do not need to install specialized software, a simple Web browser often suffices. The precise format in which data is coded and stored becomes irrelevant; data transfer between client and server uses the Geographic Markup Language (GML), an OGC standard based on XML, thus technologically neutral. Furthermore, database managers, application services, and viewing applications can be developed independently from one another, relying only in the standardized interface between them.

OGC's approach to SOA is based on standardized services, which are specialized according to the type of data they provide. The most important ones are the

Web Map Service (WMS), the Web Feature Service (WFS) and the Web Coverage Service (WCS). WMS produces complete maps, rendered at the server and transmitted as images to the client. It can also answer to certain basic queries about the content of the map. WFS provides individual features, usually as vectors, selected from the underlying spatial database, to be rendered at the client side. WCS provides images and regular grids. Unlike WMS, WCS provides only non-interactive data access. In this paper, we are particularly interested in WMS, due to its wide support and its potential for user interaction, in which the user can decide on various visual elements of the map and have access to many basic GIS functions.

The OpenGIS Reference Model has become the core of service-based spatial data infrastructures (SDI) (Bernard and Craglia 2005), such as the ones implemented in projects such as INSPIRE (INSPIRE 2002) and in many similar initiatives throughout the world. Open source development projects provide increasing support for the creation of new OGC-compliant geographic data services, viewers and catalogs. Some of the most widely known are MapServer, GeoServer, gvSIG, OpenLayers, and GeoNetwork. Geonetwork's Web site lists numerous interoperable geographic information services¹, most of which can be accessed by a client that is able to interact with a WMS server.

While noticing the increase in geospatial data available on the Web and in the variety of spatial data sources available through OGC-compliant Web services, we observed a series of hurdles and limitations for their use by clients running on mobile computers (Alves and Davis Jr. 2007). Communications overhead can be significant, considering the various types of messages exchanged by client and server, and the fact that messages and data go around in XML. For instance, the total volume of XML messages required to get an image that fills up the screen of a regular-sized cellular phone (128 by 128 pixels) can exceed the image's size, or about 32Kbytes, in the first connection, depending on the server. Overhead diminishes as more images are obtained from the same source, but for mobile computers this kind of exchange means wasted bandwidth and battery.

We also observe that the service-based architecture approach is specially well-suited to solve interoperability issues, which are still quite common and present obstacles to the dissemination of geographic data and applications. Without such an approach, developers would be limited to creating custom datasets, thus incurring in significant data gathering and transformation costs. However, the characteristics and limitations of SOA simultaneously bring forward scalability concerns. Specifically in the case of WMS, these concerns involve the server side, where images are generated, and the client side, which is burdened by communications and processing loads. For the implementation of NASA's Jet Propulsion Laboratory OnEarth WMS server, Plesea (2008) introduced numerous enhancements on the server side to support a large WMS request load, including an image processing pipeline, tile caching implemented as an Apache module, and virtual image datasets, which provide replicated data access. Bergamini et al. (2006) propose a scheme in which WMS data is cached and prefetched using a P2P approach.

¹ <http://geonetwork-opensource.org/>

However, notice that the most successful Web mapping technologies for large numbers of users, such as the Ajax-based Google Maps (Paulson 2005), are based on precompiled base map tiles at predetermined scale levels, with predetermined content. Complementary information, if available, is presented on top of the base maps at display time. This strategy scales up at the server, since it is only required to locate and transmit the necessary map tiles, and benefits from Web and image caches at the client side (Barroso, Dean et al. 2003). WMS, on the other hand, also transmits images, but these images have to be rendered by the server, at the moment of the request, so that the layer selection function can work. Since it would be unfeasible to precompile images for every possible layer combination, the server needs to take on a much larger processing load.

The discussion around the ideal architecture for Web-based geographic applications, including peculiar aspects of mobile applications, demonstrates how poorly prepared we are nowadays to provide massive volumes of data to many people at the same time, given a requirement for a minimal GIS-like interactivity. Considering how geographic information can be basic for many applications, and the fact that mobile computers and cellular phones will quickly outnumber regular computers, the improvement of information dissemination elements helps meeting the first of the five Grand Challenges for Computing in Brazil 2006-2016, which involves managing information over large volumes of distributed multimedia data (Medeiros 2008).

This paper presents our vision on the design and implementation requirements for a WMS setting that improves interactive access to geographic information sources using mobile computers. Our intention is to improve the usefulness of SDI-related tools, while keeping their interoperability potential intact. Specifically, we propose the division of the WMS client's logic into two units, only one of which running in the mobile computer, while the other is hosted by a thick client somewhere else over the network, possibly the mobile user's own PC. While the thick client implements the complete WMS specification, the communication between the thick client and the mobile computer is optimized, using simpler protocols and techniques to improve user experience at the mobile end.

This paper is organized as follows. Section 2 presents some related work. Section 3 explains the kinds of overhead that a WMS client imposes on a mobile computer. Section 4 details our proposal, and shows some practical results and measurements. Finally, Section 5 presents our conclusions so far and lists priorities for future work.

2. Related Work

Spatial Data Infrastructures (SDI) are seen as a new and interoperable approach to creation, distribution and use of geographic information (Davis Jr. 2008; Fonseca 2008). SDI tries to avoid the old view of GIS as an automated map distribution system, which focuses on map production and distribution of existing sources on an "as-is" basis. SDI is an enabler for understanding space. SDI does not simply deliver maps. It disseminates spatial data with associated quality control, metadata information, and semantic descriptions. The SDI user is someone who is able to combine spatial data from different sources to produce new information for a study area. In this view, SDI can play

an important role in the management of the environment and in the sustainable growth of our society (Davis Jr, Fonseca *et al.* 2009 (to appear)). Notice that the wide availability of spatial data, fulfilling throughput and scalability requirements for the envisioned applications, is strategically important to meet two of the five Grand Challenges for Computing in Brazil (2006-2016), namely the management of information over massive volumes of distributed multimedia data, and the computational modeling of complex systems (artificial, natural, socio-cultural, and human-nature interactions) (Davis Jr, Fonseca *et al.* 2009 (to appear)).

SDI comprise a set of policies, technologies and standards that interconnect a community of spatial information users and related support activities for the production and management of geographic information (Phillips, Williamson *et al.* 1999). SDI involves avoiding redundant effort and reducing production costs for new and existent datasets through the sharing of resources. In general, SDI can be seen as a hierarchy of data sources, as defined by Rajabifard *et al.* (Rajabifard, Williamson *et al.* 2000), in which information is consolidated from corporate and local levels, to regional and global levels (Rajabifard and Williamson 2001; Jacoby, Smith *et al.* 2002; de Man 2006). SDI can be implemented by chaining services of different sources (Alameh 2003) and integrating software components (Granell, Gould *et al.* 2005) that can be found in geoportals (Maguire and Longley 2005a).

The expression SDI was initially used to describe a standardized way to access to geographic information (Maguire and Longley 2005b). A SDI implies the existence of some sort of coordination for policy formulation and implementation, along with more complete and standardized metadata, possibly including means to provide online access to data sources. The first generation of SDI focused on granting a broad thematic scope, which is consistent with the current analogy between SDI and other types of infrastructure: fostering economic development by granting access to publicly-available and multiple-use goods or services. Evolution from the first generation of SDI was made possible by the recent expansion of Web-based information systems. In the USA, the Geospatial One-Stop (GOS) Web portal was created to provide widespread access to geographic information, inaugurating the concept of geoportals (Maguire and Longley 2005b; Tait 2005), currently viewed as SDI components. While an SDI is the overarching environment formed by the confluence of several geographic data providers, each of which granting data access through specific Web services, a geoportal provides means to give humans some level of interactive access to these data resources, including Web-based viewers and metadata-based discovery tools (Figure 1).

The use of Web services to grant direct access to data is the most important distinction between first- and second-generation SDIs. In fact, the numerous possibilities that arise from using such services to encapsulate data from multiple sources, and thereby achieve interoperability, have led Bernard and Craglia (2005) to propose a new translation for the SDI acronym: *Service-Driven Infrastructures*. In fact, current SDIs include Web services as one of the possible data access channels, while maintaining links to downloadable data and existing Web applications.

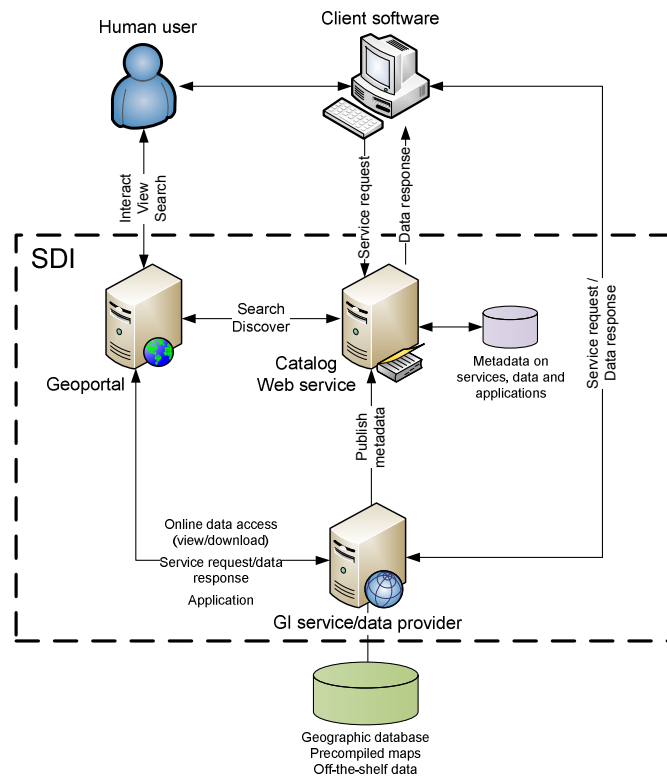


Figure 1 - Geoportals and SDI

The most current view on spatial information infrastructures considers their evolution into the perspective of service-based distributed system architectures, which have been proposed as part of a strategy for developing complex information systems based on reusable components. One of the most interesting approaches in this field is the one of service-oriented architectures (SOA) (Papazoglou and Georgakopoulos 2003). Services, their descriptions and fundamental operations, such as discovery, selection, and binding, form the basis of SOA. SOA supports large applications with sharing of data and processing capacity, through network-based distributed allocation of applications and use of computational resources. In this architecture, services are self-contained, which means that information on the service's description, including its capabilities, interface, behavior, and quality, can be obtained from the service itself, through a standardized set of functions. The Open Geospatial Consortium (OGC) has proposed many standards for Web service-based data access, such as the Web Feature Service (WFS), the Web Map Service (WMS), and several more, including some which are under evaluation at the time of this writing (Klopper 2005).

Although Web services provide a standardized path to interoperability, their cost in terms of computing and communications is not negligible. Web services are usually seen as Web-based enterprise-wide or inter-organizational applications that use open standards (mostly based on XML) and transport protocols to exchange data with clients, thus forming a *loosely-coupled information systems architecture* (Ferris and Farrell 2003; Davis Jr. and Alves 2007). Web services use the Hypertext Transfer Protocol (HTTP) in the application layer (W3C 2002). HTTP only allows the client to perform synchronous calls, which is a problem when dealing with delayed-time transactions and call resume.

In a previous work (Alves and Davis Jr. 2007), our group has studied the requirements posed to OGC Web services (OWS) by local SDI (Davis Jr. and Fonseca 2005). We have observed that local applications can be more demanding than regional- or country-based SDI, as in INSPIRE (2002), because of the potentially larger number of information classes and higher level of detail. We evaluated engineering aspects of OWS specifications and the main OGC services in the context of urban applications. Through these applications, we identified some implementation constraints that are characteristic of GI systems, such as non-standardized fault tolerance mechanisms, clients that are strongly dependent of providers, and others. In response, we implemented a prototype based on the services abstract model for a real-world use case, and tested the usefulness of OGC standards to local SDI, specifically considering issues for client development. One of the services proposed, called the *Data Exchange Service* (DXS), is able to replace local storage and the persistence function of service providers with a third-party neutral service. This benefits mobile computers and thin clients, by not requiring local storage space while processing a Web service. Another proposed service, called the *Client Access Service* (CAS), mediates the connection between client and server. Since the HTTP connection is synchronous, delayed-time services or service chaining can cause the client to continuously poll the server for a response if it does not have a valid IP address, thus wasting network and processing resources. Some other initiatives try to emulate asynchronous calls in Web services through the use of *listeners* that receive responses and forward them to the requester (Ruth, Lin *et al.* 2005), usually over non-HTTP protocols (Brambilla, Ceri *et al.* 2004). Other strategies involve the use of SMTP, the e-mail protocol (Chung, Pan *et al.* 2006). However, as far as we know, no pattern besides CAS enables asynchronous communication using only Web service standards.

There are other limitations in mobile computing and wireless networking that need to be addressed. As wireless networks become popular, the demand for them becomes larger as well. Quality of Service, or QoS, is a concept related to network performance and user satisfaction. QoS parameters like throughput, delay, jitter, availability, and reliability are some indicators that are used to assess the quality of the network support to applications (Duarte-Figueiredo and Loureiro 2007). Wireless networks have some limitations in providing end-to-end QoS to mobile users, including communications failures when the user is moving, and varying energy consumption (Ribeiro, Duarte-Figueiredo *et al.* 2008). Along with network QoS, other mobile computing limitations that are important for our analysis include (1) insufficient bandwidth, (2) limited amount of local memory for temporary storage, (3) power consumption, (4) transmission interferences and interruption, (5) security, and (6) human interface constraints (Forman and Zahorjan 2004).

Our work focuses mainly in three strategies. First, we simultaneously reduce the bandwidth requirements and power consumption by reducing the overhead caused by WMS protocols. Furthermore, shorter interactions mean reducing the risk of interruptions. Secondly, we improve WMS image tiling and transmission strategies, in order to reduce communications overhead, thereby saving power. Finally, we provide faster access to WMS-served images, while reducing the need for local temporary storage, by implementing custom image tiling algorithms. Results can also apply to WCS, since the nature of the transmitted data is similar. In our opinion, WFS would

require a different strategy for optimization, but that discussion is beyond the scope of this paper.

3. OGC's Web Map Service

The Web Map Service (WMS) has been released by OGC in 2000. WMS version 1.1.1, released in 2002, is the most widely supported version, although the most recent is version 1.3 (Open Geospatial Consortium (OGC) 2006). Figure 2 presents schematically the exchanges that take place between WMS clients and servers. The left side of the figure assigns numbers to individual requests, so that they can be more easily identified in the next paragraphs.

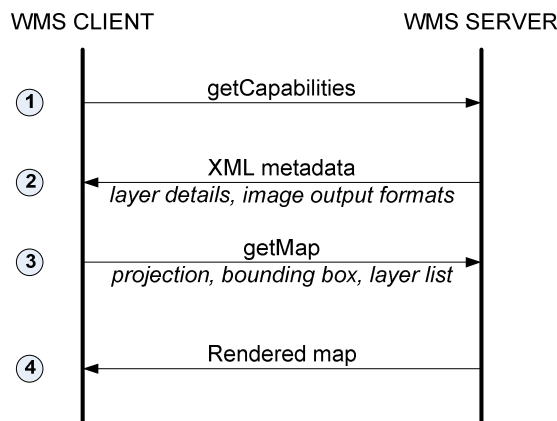


Figure 2 – Web Map Service exchanges

In WMS, as in other OGC specifications, services and their parameters can be discovered using a standard request. Assuming that clients do not know initially which data they wish to access, their initial interaction is to query the service about available data using the `getCapabilities` feature (1). In response, the server generates a long XML document, which contains details on each data layer it has to offer, i.e., metadata on each object class available in its geographic database. Such metadata include title, abstract, keywords, contact information for the technician or institution responsible for the service, access fees, access constraints and more (2). There are also data on the format in which data can be supplied, including traditional image formats, such as JPEG or PNG, and, depending on the service infrastructure, KML, SVG, and XML. Once the services are known, the client has to request a rendered map using the `getMap` feature (3). It is invoked using the traditional HTTP GET request, in which the client passes many parameters such as the projection system, coordinates of a bounding box, list of layers that are to be viewed, the return format, and screen coordinates for the presentation of the rendered map. The server then renders and transmits back to the client the requested map image (4).

At this point, it is important to emphasize that every further interaction between WMS client and server will proceed in the same manner, cycling around steps 3 and 4. If the user decides to pan the image partially in some direction, a new bounding box is specified, sent to the server as a new `getMap` request, a new image is rendered and transmitted.

This method contrasts strongly with current commercial Web map browsers, such as Google Maps, which maintain map tiles pre-rendered at various (and fixed) scales, transmitting only the required tiles at each user request. Intuitively, the WMS approach is much more demanding for the server, both in terms of computing (rendering maps) and communications (image transmission); therefore, WMS services are not nearly as scalable as commercial map browsers.

On the other hand, WMS clients have the freedom to select and compose layers to suit their needs, as well as being able to specify precise viewing scales and to customize output map sizes to fit the user interface. Clients usually implement user interfaces that are more similar to the ones found in desktop GIS, allowing users to activate and deactivate available layers. Furthermore, WMS clients allow users to connect to several different services, thus composing maps with elements from different geographic data providers.

While WMS provides more flexibility and customizability than Web map browsers, its design choices impose a somewhat heavier burden over the server, thus limiting its scalability potential. Clients are not supposed to store the service's metadata permanently, since the capabilities of the server can change in time, and therefore connecting to a WMS server requires a new `getCapabilities` exchange each time. There is no provision for image caching, or image tiling, so that rendering and transmission could be redone only partially, thus wasting the server's resources with reprocessing.

Our proposal, presented in the next section, intends to find a convenient middle ground between Web map browsers and WMS, using an intermediate application server and partially limiting the freedom that the WMS client has in the specification of the service's output.

4. WMS Environment for Mobile Applications

4.1. Solution architecture

Even though there are some Web map browsing applications available today for mobile computers, we have been able to identify only one WMS client, developed by Skylab². The other clients, including Google, Yahoo!, Nokia and Garmin, are restricted to the datasets provided by their developers.

As shown, WMS interactions involve potentially large data volumes at the server, considering the characteristics and limitations of wireless networks and mobile computers. Our approach to this problem involves dividing the WMS client into two components: the *mobile client* (MC), and the *WMS connectivity layer* (WCL). The MC is responsible for user interaction functions, including data visualization. The WCL receives the MC's requests and mediate their distribution to WMS servers. Only the WCL implements interactions according to the full WMS standard, since it must behave as a regular WMS client, from the point of view of any WMS server. Between the mobile client and the WMS connectivity layer, a simpler connection based on sockets has been implemented. The socket connection is much lighter for the mobile computer,

²http://www.skylab-mobilesystems.com/en/products/j2me_wms_client.html

therefore saving network, local memory, and battery resources. Subtracting all WMS connectivity code also helps to keep the MC small. The mobile client has been developed in J2ME, a robust and flexible environment for the development of portable applications for mobile phones and PDAs. The WCL runs on a regular desktop computer, acting as a thick client and fully compatible with WMS, although deprived of a viewing and user interaction interface. The WCL has also been developed in Java, to run as a local application.

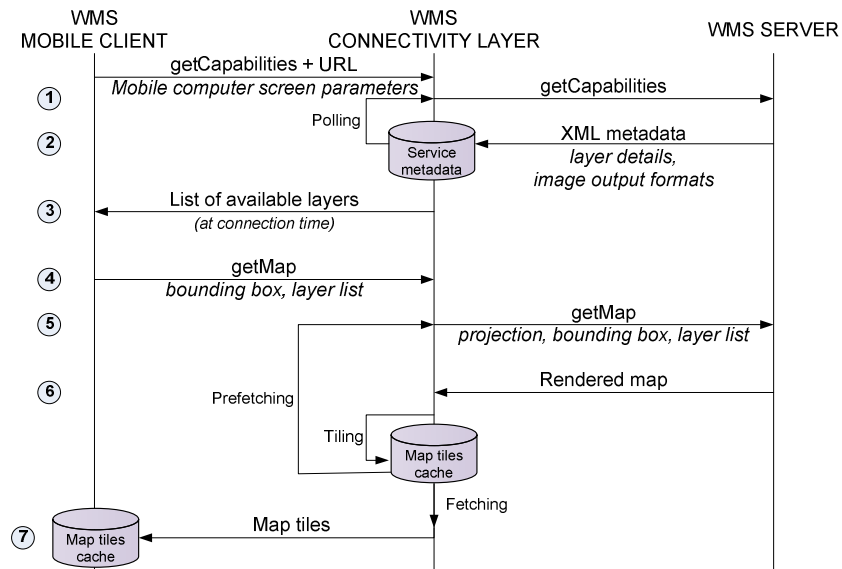


Figure 3 – Proposed WMS environment

Figure 3 presents an interaction diagram in which the communications protocol for our approach is detailed. Steps 1 and 2 perform the same `getCapabilities` interaction presented in Figure 2. When the mobile computer initiates the process, it sends the URL for the `getCapabilities` request, along with its screen parameters (size, resolution, color range). Results are then kept in the WCL, and partially transmitted to the mobile client, using a more compact form (step 3). Our initial tests showed that this simple caching of the layer information from the WMS server can spare the mobile client from a large part of the data transmission, especially when the server manages many data layers. Step 4 corresponds to the map data request, which is sent from the mobile client in a compact form, and translated by the WCL into a full WMS request (step 5). Notice that the projection information is kept in the WCL as a default, in order to avoid redundant transmission. The map is then rendered by the server and returned to the WCL (step 6). At the WCL, the image sent by the WMS server is divided into smaller tiles, which are more adequate for transmission to and display by the MC. Furthermore, the rendered image tiles can be cached at the WCL, in order to avoid redundant computation by the WMS server. Notice also that the bounding box sent in the `getMap` request from the MC is expanded by the WCL, in order to have a larger area available for tiling and transmission to the mobile computer. In our tests, this expansion was set up with twice the original bounding box width and height. This is done as a *prefetching* strategy, whose algorithm will not be detailed here due to lack of space. Finally, map tiles are then transmitted to the client (step 7). There is a tile cache in both the WCL and the MC, even though the latter is subject to stronger limitations.

We adopted a cache invalidation rule in which the tiles farthest from the center of the screen position are discarded to make room for incoming tiles.

The user can perform only four actions that have an effect on the contents of the display: zoom in, zoom out, pan, and change the displayed layers. The latter forces the generation of an entire new image by the server, and is therefore equivalent to restarting the process at step 3 or 4. If the user wants to zoom in or out, the process returns to step 4, and the server has to generate a new image. In the case of a pan, the tile caches comes into play and supplies available tiles with no cost for the server. If the panning action goes beyond the prefetched area, a new bounding box is sent to the server, and divided into tiles upon arrival, once again allowing for some prefetching.

Some commercial mapping tools for mobile computers include a function that allows the user to prefetch tiled areas from map servers such as Google Maps or OpenStreetMap and store them in the mobile computer's memory, in order to allow visualization even in the absence of a network connection. Our strategy allows for that feature also, but allows the user to change layer selections when a connection is available and generally determine some visualization parameters. Our approach also preserves the possibility of getting dynamic data from other OGC servers, combining display data from different sources at display time.

Even though the freedom a WMS user has on the selection of layers and the determination of the bounding box is a desirable feature of the OGC approach, commercial map browsers are much more scalable, since map tiles are pre-computed and stored at several zoom levels. In order to increase the scalability potential of our WMS implementation, and to increase cache hit rates, we intend to establish some fixed zoom levels, and to allow panning in fixed steps. We will also include a layer grouping strategy that allows for a shorter `getCapabilities` response, and an easier layer selection dialog at the mobile end. This grouping strategy will also benefit the tile cache, since indexing must take place considering the combination of layers using to render the image to be tiled. Currently, any modification in the list of layers causes the whole cache to be invalidated and recomputed.

4.2. Experimental results

We developed a prototype for the MC using J2ME, so that it is able to run in any Symbian-based cellular phone. The WCL was also implemented in Java, and installed on an obsolete PC with no special hardware. We performed a set of experiments using a random WMS server found on the Web³, initially intending to check the reduction of data transfer from the mobile computer's standpoint. We have performed a comparison between the Skylab WMS client and our prototype, considering three kinds of interactions in sequence: (1) a `getCapabilities` request, (2) the initial recovery of a layer, and (3) a pan operation involving a displacement the size of a screen width to the right. Experiments were also performed in two different environments. The first one used an emulator from Sun's Java Wireless Toolkit version 2.5.2 (Table 1), and the

³ <http://gdr.ess.nrcan.gc.ca>, a geoscience data repository from Natural Resources Canada (http://gdr.ess.nrcan.gc.ca/wmsconnector/com.esri.wms.Esrimap/GDR_E?SERVICE=WMS&REQUEST=GetCapabilities&)

second one used an actual Nokia N95 cellular phone over a Brazilian carrier’s GPRS network (Table 2). Results indicate the average of three executions.

Table 1 – Test results over the WTK emulator

Action	Skylab (bytes)	Our proposal (bytes)	Reduction (%)
getCapabilities	81,178	18,340	77%
Layer recovery	25,770	34,241	-35%
Pan	70,013	3,308	95%

Table 2 – Test results over Nokia N95 and GPRS network

Action	Skylab (bytes)	Our proposal (bytes)	Reduction (%)
getCapabilities	89,713	21,891	76%
Layer recovery	26,041	36,211	-39%
Pan	52,322	21,463	59%

The increase in transmission volume for the layer recovery operation is due to the overhead associated to each individual transmitted tile. In our proposal, a larger image is requested from the server, and then tiled at the WCL. With this strategy, we can avoid retransmission when the user performs pan operations, which are usually quite common, considering the small dimensions of a typical cellular phone or mobile computer screen. The test results indicate that the overhead in layer recovery would be more than fully compensated by the economy in only one pan operation: while the layer recovery operation implies a 10 Kbyte disadvantage for our proposal, one pan operation implies in an economy of over 30 Kbytes. This gain in the pan operation is a direct result of the caching of image tiles at the mobile computer and at the WCS. Since the image that has originally been requested exceeds the screen size by half the screen size in both directions, when a pan is executed much of the additional image required is already available. Notice also the big difference in the measurements between the emulator and the actual phone. This indicates that, until a more reliable emulator is available, tests should be executed in real networks; on the other hand, this limitation poses a challenge for throughput and response time measurements, since the perceived speed of the network varies throughout the day, due to traffic patterns.

The experiments tried to emulate individually the typical user operations over WMS services. A better approach would be to characterize a typical workload for a mobile WMS user, but there are no established standards for that purpose. Therefore, our experiments must be expanded in the future, using some human-computer interaction or usability techniques, in order to try to assess the impact of our approach on a more realistic situation. The preliminary results showed above, along with the general impression that comes from using both WMS clients in the same environment, ensure us that the gains are significant.

5. Conclusions and Future Work

We have proposed and implemented a fully-functional WMS client which imposes a smaller burden on mobile computers, while preserving most of WMS’s flexibility for user-selected parameters. Of course, the same strategy could be used in the

implementation of WMS clients for more powerful computers, in an attempt to increase the scalability potential of this SOA-based spatial data infrastructure solution.

Preliminary results show that the volume of data transmission between the mobile computer and the server is significantly reduced: in some cases, we have been able to reduce data transmission volume at the first steps of the protocol by as much as 90%. Naturally, less voluminous data transmission implies faster interaction, lower battery consumption, reduced impact on bandwidth, and lower costs. Interactivity is significantly faster, but so far we have not conducted any usability experiments. Anyway, in accordance to our original objectives, we believe that being able to interact with servers on large data volumes, as in our examples, using only a mobile computer while respecting its limitations is an important result that contributes to the first Grand Challenge for computing in Brazil.

Our work for the near future includes a full set of experiments and measurements on the data transmission reduction. The efficiency of the solution will be assessed by monitoring the usual mobile computing parameters, such as bandwidth and battery consumption, and use of resources, such as CPU and memory. Communications latency must also be assessed, since it is an important factor for mobile users, as well as an evaluation of the requirements on the WCL machine. We will also conduct a detailed comparison with SkyLab's WMS client for mobile computers.

At the present stage, we can notice significant gains from using the proposed client architecture, as compared to a regular WMS client. We intend to perform comparisons with commercial map browsers, in order to verify response time and scalability potential. We think that this kind of work makes it possible to design new and improved location-based applications, using a wider range of online geospatial data. We intend to extend this research towards other OGC Web services.

Acknowledgments

Clodoveu Davis' work was partially supported by the Brazilian National Institute of Science and Technology for the Web (CNPq grant 573871/2008-6), CNPq (grants 306370/2006-9 and 551037/2007-5), and Fapemig (CEX APQ 0551-5.01/07), Brazilian agencies in charge of fostering research and development.

References

- Alameh, N. (2003). "Chaining geographic information Web services." *IEEE Internet Computing* 7(5): 22-29.
- Alves, L.L. and Davis Jr., C.A. (2007). Evaluation of OGC Web Services for Local Spatial Data Infrastructures and for the Development of Clients for Geographic Information Systems. *Advances in GeoInformatics*. Davis Jr., C.A. and Monteiro, A.M.V. Berlin, Springer-Verlag: 217-234.
- Barroso, L.A., Dean, J. and Hölzle, U. (2003). "Web Search for a Planet; The Google Cluster Architecture." *IEEE Micro* 23(2): 22-28.
- Bergamini, J.A. and Haungs, M. (2006). *GeoTorrent: Optimizing GIS Web Service Strategies for Interactive Educational Use*. 2006 University Consortium for Geographic Information Summer 2006 Assembly, Vancouver, Washington.

- Bernard, L. and Craglia, M. (2005). SDI - From Spatial Data Infrastructure to Service Driven Infrastructure. Research Workshop on Cross-Learning Between Spatial Data Infrastructures and Information Infrastructures, Enschede, The Netherlands.
- Brambilla, M., Ceri, S., Passamani, M. and Riccio, A. (2004). Managing asynchronous Web services interactions. Proceedings of the IEEE International Conference on Web Services:80-87.
- Chung, S., Pan, J.R. and Davalos, S. (2006). A special Web service mechanism: asynchronous .NET Web services. Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services:212.
- Davis Jr, C.A., Fonseca, F. and Câmara, G. (2009 (to appear)). Understanding Global Change: The Role of Geographic Information Science in the Integration of People and Nature. SAGE Handbook of GIS and Society. Nyerges, T., Couclelis, H. and McMaster, R.B., SAGE.
- Davis Jr., C.A. (2008). Spatial Data Infrastructures. Encyclopedia of Information Science and Technology. Khosrow-Pour, M. Hershey, Pennsylvania, USA, IGI Global. **VII**: 3548-3553.
- Davis Jr., C.A. and Alves, L.L. (2007). Geospatial Web Services. Encyclopedia of GIS. Shekhar, S. and Xiong, H. Berlin, Germany, Springer-Verlag: 1270-1273.
- Davis Jr., C.A. and Fonseca, F.T. (2005). "Considerations from the Development of a Local Spatial Data Infrastructure." Information Technology for Development **12**(4): 273-290.
- de Man, W.H.E. (2006). "Understanding SDI: complexity and institutionalization." International Journal of Geographic Information Science **20**(3): 329-343.
- Duarte-Figueiredo, F.L.P. and Loureiro, A.A. (2007). An end-to-end wireless QoS architecture evaluation. 12th IEEE Symposium on Computers and Communications (ISCC 2007):715-720.
- Ferris, C. and Farrell, J. (2003). "What Are Web Services?" Communications of the ACM **46**(6): 31.
- Fonseca, F.T. (2008). Spatial Data Infrastructure. Encyclopedia of GIS. Shekhar, S.X., Hui London ; New York, Springer.
- Forman, G.H. and Zahorjan, J. (2004). "The challenges of mobile computing." IEEE Computer **27**(4): 38-47.
- Granell, C., Gould, M. and Ramos, F. (2005). Service Composition for SDIs: integrated components creation. 2nd International Workshop on Geographic Information Management (GIM'05), Copenhagen, Denmark.
- INSPIRE (2002). INSPIRE Architecture and Standards Working Group, INSPIRE Architecture and Standards Position Paper. Brussels, Commission of the European Communities.
- Jacoby, S., Smith, J., Ting, L. and Williamson, I. (2002). "Developing a common spatial data infrastructure between State and Local Government -- an Australian case study." International Journal of Geographic Information Science **16**(4): 305-322.
- Klopfer, M. (2005). Interoperability & Open Architectures: an analysis of existing standardisation processes & procedures. OGC White Paper. Consortium, O.G., Open Geospatial Consortium: 26p.

- Liu, Z., Pierce, M.E. and Fox, G.C. (2007). Implementing a Caching and Tiling Map Server: a Web 2.0 Case Study. Proceedings of the 2007 International Symposium on Collaborative Techniques and Systems (CTS 2007), Orlando, Florida.
- Maguire, D.J. and Longley, P.A. (2005a). "The emergence of geoportals and their role in spatial data infrastructures." *Computers, Environment and Urban Systems* **29**(1): 3.
- Maguire, D.J. and Longley, P.A. (2005b). "The emergence of geoportals and their role in spatial data infrastructures." *Computers, Environment and Urban Systems* **29**(1): 3-14.
- Medeiros, C.B. (2008). "Grand Research Challenges in Computer Science in Brazil." *IEEE Computer* **41**(6): 59-65.
- Open Geospatial Consortium (OGC) (2006) "OpenGIS Web Map Server Implementation Specification, version 1.3.0." **Volume**, DOI: Document number OGC 06-042
- Papazoglou, M.P. and Georgakopoulos, D. (2003). "Service-Oriented Computing." *Communications of the ACM* **46**(10): 25-28.
- Paulson, L.D. (2005). "Building Rich Web Applications with Ajax." *IEEE Computer* **38**(10): 14-17.
- Percivall, G., Ed. (2003). *OpenGIS Reference Model*, Open Geospatial Consortium, Inc.
- Phillips, A., Williamson, I. and Ezigbalike, C. (1999). "Spatial Data Infrastructure concepts." *The Australian Surveyor* **44**(1): 20-28.
- Plesea, L. (2008). *The Design, Implementation and Operation of the JPL OnEarth WMS Server. Geospatial Services and Applications for the Internet*. Sample, J.T., Shaw, K., Tu, S. and Abdelguerfi, M. Berlin, Springer.
- Rajabifard, A. and Williamson, I. (2001). *Spatial Data Infrastructures: concept, SDI hierarchy and future directions*. Geomatics'80, Tehran, Iran.
- Rajabifard, A., Williamson, I.P., Holland, P. and Johnstone, G. (2000). *From Local to Global SDI Initiatives: a pyramid of building blocks*. 4th Global Spatial Data Infrastructure Conference, Cape Town, South Africa.
- Ribeiro, A.I., Duarte-Figueiredo, F.L.P., Novy, G., Storck, C., Dias, S.M. and Zárte, L.E. (2008). An artificial neural network approach for mechanisms of call admission control in UTMS 3G networks. 8th International Conference on Hybrid Intelligent Systems (HIS'08):459-464.
- Ruth, M., Lin, F. and Tu, S. (2005). A client-side framework enabling callbacks from Web services. Proceedings of the Third European Conference on Web Services (ECOWS 2005):105-116.
- Tait, M.G. (2005). "Implementing geoportals: applications of distributed GIS." *Computers, Environment and Urban Systems* **29**(1): 33-47.
- Tu, S. and Abdelguerfi, M. (2006). "Web Services for Geographic Information Systems." *IEEE Internet Computing* **10**(5): 13-15.
- W3C (2002). *Web Services Architecture Working Group, Web Services Architecture Requirements*, W3C Working Draft, World-Wide Web Consortium.