IX Brazilian Symposium on GeoInformatics, Campos do Jordão, Brazil, November 25-28, 2007, INPE, p. 97-108.

A Service-Oriented Architecture for Progressive Transmission of Maps

David Cavassana Costa[1], Mario Meireles Teixeira[1], Anselmo Cardoso de Paiva[1], Claudio de Souza Baptista[2]

[1]Departamento de Informática – Universidade Federal do Maranhão (UFMA)
Av. dos Portugueses SN – 65.085-580 – São Luis – MA – Brazil

[2]Departamento de Sistemas e Computação
Universidade Federal de Campina Grande (UFCG) – Campina Grande, MA – Brazil

david@gia.deinf.ufma.br, {mario, paiva}@deinf.ufma.br, baptista@dsc.ufcg.edu.br

Abstract. *The Internet creates an environment suitable to spatial data share, allowing the users to transmit, visualize, manipulate and interact with them. This environment not only allows new opportunities for geospatial data use, but also introduces new problems that must be managed to permit these data to be used in an effective and useful way. One of such problems is related to the use of these data with small network bandwidth. This work presents an architecture based on Gis Services for the progressive transmission of vector maps in the Internet, anticipating the rasterization process in the server side, thus reducing the amount of data to be transmitted to the client. The architecture proposed uses all the advantages of Gis Services, that are becoming a standard in the construction of Gis applications based on services.*

1. INTRODUCTION

Visualization plays an important role for a better understanding of phenomena in several areas of knowledge and it is so with geographic data. Geographic maps are used long ago for viewing spatial data, helping us to better understand the relationships among those data. In the field of cartography, the visualization process must be formalized through the definition of rules and principles, once different kinds of data can be viewed in different ways [Cecconi 2003]. Such methods and techniques must be used as to optimize their use and validate the data consistency.

Geographic Information Systems (GIS) are computational tools used either to make available and/or analyze information associated with position/localization over the Earth Surface. Currently, the Internet has become a huge content publishing media; being thus a favorable environment for the GIS users to exchange data, perform analyses and present geographic results. Geographic information in the Internet has rapidly evolved with the development of Web technologies.

Internet creates a new geospatial data sharing environment, where data suppliers turn available their geographical databases in a way analogous to the Web pages' textual information, allowing the users to use the Web for data transferring and utilize them for visualizing, analysis and/or manipulation [Bertolotto 1999]. This new configuration supplies new opportunities, for both, the public and the commercial domains, of using geographic data sets. On the other side, new problems arise from such architecture, for example, the availability of large amounts of data stored in repositories lacking efficient transfer methods.

Among the emergent technologies one can cite the Web Services, which allow the building of Web applications flexible, inter-operational and reusable, enabling interactions between those applications. Web Services are software modules identified by an URI that offer services to remote applications called consumers, using the Internet as communication channel [W3C 2002], [W3C 2004].

GIS Services are services of Geographic Information Systems implemented by means of Web Services that perform a specific GIS function which can be integrated as part of one or more applications [Yumin 2004]. GIS Services is often associated to the Web Services based GIS.

Even with the growing Internet users' access to a bigger bandwidth, there are still situations where the transmission cost is a critical factor, as in the case of wireless networks or dialed access, especially in under developed countries or localities where there is no chance of using a dedicated connection. Those adverse characteristics encouraged the appearing of various methods for efficient data transmission, necessary to minimize the response time, enabling the own publication of vector based geographical information in systems with low transmission rates.

Once all geographic data tends to be very bulky in the majority of geographic data banks, the transfer process may force the user to wait a long time or even it may become prohibitive. To solve these problems, techniques such as progressive transfer and generalization can be rather feasible solutions. Such techniques are being proposed, evaluated and combined nowadays aiming at solving or, at least, soften the problem of latency in the Internet geographic map transmission. The complexity of techniques and algorithms meant for solving such problem is one the major motivations for research in that area, once each approach has different focus and scope.

Several methods have being proposed as to progressively transmit vector spatial data [Cecconi 2004], [Bertolotto 1999], [Buttenfield 2002], [Oh 1999], [Han 2004], [Yang 2004], [Yumin 2004]. Such methods are based on the idea that the user, in general, gradually consumes the map's information, starting from low detail levels scaling to higher ones. Based on that evidence, it is possible to generate maps in a certain way, dividing them and gradually transmiting them.

In the progressive transmission, the maps server divides the maps into a low resolution version and a set of incremental versions that, when incorporated into a certain maps' version, generates a more detailed version of that map. The client is in charge of receiving the maps' detail increments in such a level n, and integrating them into the actual version of the map, generating a map version at the level n+1.

Other approaches to that problem exploit the device's constraints, such as the visualization resolution. This can determine the level of details that can be visualized in a specific device. In general, the device can present fewer details than the ones sent by the maps server. The detail level that can not be visualized increases the transmission cost and does not enhance the map's quality being viewed by the client. This way, the removal of this redundant information minimizes the client's response time without barring the map's quality [Liang 2001].

Our work proposes a Web Service Oriented architecture for a geographic map server with support to progressive transmission, using combined techniques of generalization, content adaptation and progressive transmission, in order to efficiently enhance the map transmission process from the server to the client with minimum impact in the data consistency, increasing the application's usability and minimizing the response time experienced by the user by means of modular inter-operational and reusable components, distributed between client and server.

The remaining of the article is organized as follows: Section 2 presents some of the main work related with Web Services and GIS Services. Section 3 presents the used GIS's framework architecture and the transmission maps architecture handled for implementing the service. It also presents the proposed Maps Service, detailing its architecture, usage scenario, service interface and an implemented client application. Section 4 presents the performance data relative to the results and measurements taken with the new proposed architecture and, finally, Section 5 discusses the conclusions of this work, suggesting some feasible future enhancements.

2. SUPPORT ARCHITECTURES

This work aims is to present an architecture based on GIS Services for a map service that also have progressive map transmission capabillities. Before the proposed architecture presentation, we will present two other architectures that used as the basis for our work. The first one is the iGIS architecture. iGis is a GIS framework for Web map publishing. It is used for the map generation that are made available by the GIS Service. Following, we also describe the progressive transmission architecture used bye the proposed GIS Service.

2.1 iGIS Architecture

The iGIS [Baptista 2004] is a framework with a three layer architecture, aiming at implementing a Geographic Information System based upon the Web following the OpenGeoSpatial standard. As a framework, the iGIS allows the rapid applications' development of Web based geographical information systems.

The iGIS architecture was designed according to the MVC (Model-View-Controller) architecture standard in three layers as depicted in Figure 2. In the presentation layer, the Java Server Pages (JSP) technology is used as to implement dynamic pages. Besides that, SVG and Javascript are used for exhibiting the map, visualization tools and the map processing.

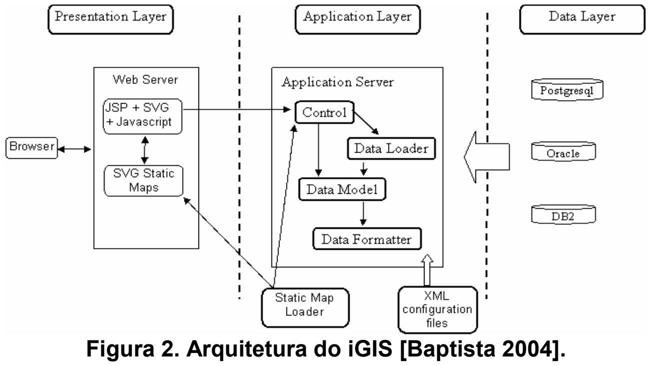


Figura 2. Arquitetura do iGIS [Baptista 2004].

The data layer is represented by a special instance of a database named DataSource. The iGIS can communicate with different DataSources, for example, Oracle, Postgresql, IBM DB2 shapefiles from ESRI. That layer, among others, is responsible for the database independence when handling geographic data. The application layer is responsible for the business logic. That layer manipulates the clients' requests turning them into the OGC model and, finally, gives them a format in an instance of the introduction/presentation layer. This layer is compounded by the following modules: Control, Data Loader and Data Formatter. The data loader is responsible for the data loading from different data sources, which are configured by means of XML files. The control module defines the necessary action sequence for meeting the clients' request, and returns the properly formatted data back to the presentation layer.

The Data Formatter is the module that assures the independence between the data manipulation and presentation. Currently, the iGIS supports formatters for vector and raster data.

As a framework, the iGIS has some extension points. One of them is the Map Formatter, which is useful for implementing a progressive transmission architecture by reusing all the functionalities reminiscent in the framework. Thus, the progressive transmission architecture was implemented within the Formatter package, generating a new sub-pack called "Progressive".

2.2 Progressive Transmission Architecture

As a means of providing the internet with an efficient vector maps transmission and visualization process, is necessary to render the maps into different detail levels. Our approach is based on the progressive map transmission, considering the characteristics of the visualization device, especially its resolution.

The progressive transmission architecture used in this work is detailed in [Costa 2006], and is based on the idea that it is possible to transmit only what can be visualized, progressively, in a similar way as proposed by Liang (2001). However, the used solution is based on a more generic approach that considers the map's low level information, the point's coordinates and the visualization resolution in the device.

By using the maps' progressive transmission, the user must initially work in a low resolution version of the map in order to localize the area that most interest him/her. Then, as soon as a visualization operation is done, e.g., zoom, the user views the interest area with better resolution in an incremental way.

The main idea of the progressive algorithm is to take into account the device's target-resolution. Hence, only the strictly necessary map information to generate a correct visualization in that resolution is sent. That approach removes non visible points and lines.

In the map's rendering process, the points change from map's coordinates to device's coordinates. In this process, some points are transformed and mapped into the same pixel. The used algorithm anticipates that process, removing from the polygon or from a line, consecutive points that would be mapped into the same pixel whenever the map would become rendered.

3. Service-Based Progressive Map Transmission Server

3.1 General Architecture

The progressive transmission GIS Service uses the iGIS framework and its progressive maps generation process starting from spatial database. Firstly, map data is loaded from the database to be published. The published content is defined through XML files that specify the database, the tables and layers of the map to be published. That loading originates several files for the same map. One of them is concerned with the lower resolution map, the others contain the increments and details able to transform a map from a n level to a n + 1 level by means of data integration operations.

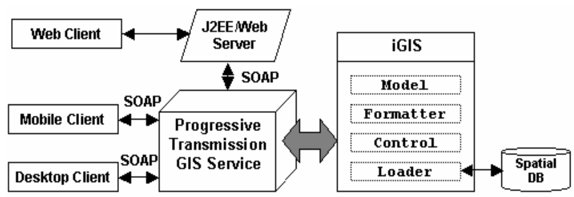


Figure 3. The Proposed GIS Service Architecture

Once the files are generated, the map can be accessed by the GIS Service through their methods, as shown in Figure 3. Initially, the Web Service enables the acquisition of a map in a low version, and later permits the acquisition of a more detailed map for a given resolution level, enhancing its resolution. Yet, there is the possibility for the client to request details of only determined layers of the map, for example request more details about highways; but not so many details of the municipality limits. Interactions between client and map server is done through the SOAP protocol. Thus, the client can be mobile, desktop or even a dynamic Web page. Therein, the iGIS architecture is condensed into just one block, and the implemented GIS Service uses its infrastructure as to provide its services to progressive map clients.

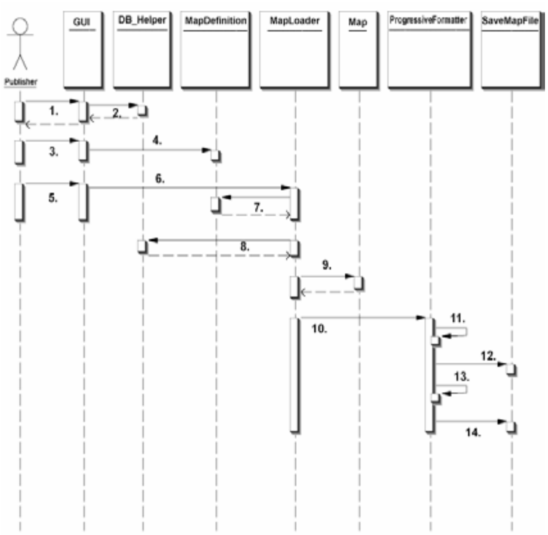
The client must be able of receiving those data and integrate them into the maps they already have in its cache. When receiving data from the increments he/she conducts DOM calls to the SVG map, generating more detailed versions of the related SVG paths, those ones that must be updated while passing from the n level to the n + 1 one.

GIS Services proposes to turn available the maps delivery service supporting progressive transmission, independent of platform or programming language. The only restriction to the client is that it must be able of rendering the map into SVG and update it via DOM operations, which is possible for Web clients, Java implemented Desktop clients, .NET or, for instance, even mobile clients. The service is the same and its use can be adapted according to the application. One advantage of its use by Desktop clients is the performance gain got with respect to the Web client, which is up to 25 times faster in terms of processing and 5 times quicker in terms of total response, results got in our tests. As we will see in the results Section, that happens because, in the case of the Web client, DOM/Javascript operations make the map integration process a little bit slower [8] compared with the desktop client, but not until the extent of not justifying its application also to Web clients.

The GIS Service guarantees the inter-operability, platform and implementation language independency, assuring to the clients higher flexibility. A Java client application may even be in need of more performance as to guarantee higher service quality, for example in the case of critical applications. The implementation of a new client, native for the target platform, can already be sufficient to promote higher performance. Yet, even the data integration algorithm [8] can be modified as to optimize resources usage, in a transparent way for the server, who continues making available its data through its standardized methods with a given format. Furthermore, the progressive transmission service may be integrated in othe Gis systems or can be added other modules to it.

3.2 Maps Generation and client/server interaction Scenario

Initially, maps generation from spatial database must be conducted. This is done through the iGIS Framework, whose task flow is briefly illustrated in Figure 4:



1. Load database maps.
2. Recover metadata from the database.
3. Select map and map layers to be published.
4. Save selected definitions in XML format.
5. Initiate map publishing process.
6. Initiate Map Loader.
7. Recover map definitions.
8. Load spatial data from the database.
9. Create a map's data structure in main memory.
10. Give a format to the map in progressive fashion.
11. Calculate low resolution map.
12. Save low resolution map.
13. Calculate map increments (details).
14. Save map increments into a file.

Figure 4. Progressive maps generation scenario

Once the maps have been generated and their details properly decomposed, the Progressive maps transmission GIS Service starts to work.

3.3 GIS Service Interface

Following, the progressive maps server methods, together with a brief explanation on the behavior of the performed operation, are presented.

String getMapNames() and *String getMapNames(int xMin, int xMax, int yMin, int yMax)*: Obtains the maps' names with support in the progressive transmission stored in the server. The map name will be used by the other methods identifying just a single map. The difference between both methods is that to the second one is assigned as parameter a given rectangular area (box), and only the maps contained within that area will be returned.

String getMapDescription(String mapName): Obtains information on the map with/through the given/defined name. That information is only used by the end user and is not meant to be processed by other services.

String getLowResMap(String mapName): Obtains the map in its lower resolution and less detailed version in the SVG format. Initially, this version is exhibited to the client and, as long as he/she requests more details, the SVG map will be updated via DOM operations, by applying the client's side integration algorithm.

String getLayerNames(String mapName) and String getLayerNames(String mapName, int xMin, int xMax, int yMin, int yMax): Gets the layer names of a given map that will be used by other service's methods. To the second version is assigned as parameter a given rectangular area (box), and only the maps contained within that area will be returned.

String getLayerDescription(String mapName, String layerName): Obtains information about a given legend of a given map. Similar to the information method about the map, those info will only be used by the end user.

String getNextLayerIncr(String mapName, String layerName, int level) and String getNextLayerIncr(String mapName, int level): Gets the details of a given map layer for a given resolution level. The second method is similar to the former, but it gets details on all the layers of a map for a given level. Those details will be processed and integrated to the map already held in the client's cache through DOM operations of the map in SVG.

int getLevels(String mapName): Gets the number of levels (versions) of a given map.

String getMinResolution(String mapName): Obtains the map's lower resolution, allowing the client to adjust his/her viewing window to the less detailed map sent at the beginning of the progressive transmission process.

String getLevelRes(String mapName, int level): Obtains the resolution of a given map level. Similar to the prior method, it helps the client to adjust his/her window when a map is received and integrated, increasing its detail level.

String mapToScreen(double x, double y) and String screenToMap(double x, double y) : Turns a point into real coordinates for a map point in the monitor and in the monitor into a point in real coordinates.

String getDetails(int x, int y): Gets information about a given map point, i.e., information on the objects to which that point belongs to, as for example, rivers, highways and cities existing in that point.

3.4 Reference Application

A desktop client and a Web client were developed as to test the implemented services' functionalities. Additionally, we also conducted a performance evaluation between the Web-based approach, described in [Costa 2006], and the new architecture, based on Web Services. The Axis [Apache 2007] was used to generate the Web Services, together with the Tomcat 5.5 server.

Most of the client application functionalities were previously described, as the application is just a prototype of the implemented services.

Nevertheless, we must highlight the way in which the details request is done. It can be conducted in two ways: synchronously and asynchronously.

In the asynchronous approach, the client is free to perform other tasks while waiting for the increments arrival of a given layer. For example, that is practical in the case the client is interested in the details of the all the map layers. In this scenario, several asynchronous requests of the map's layers details are done via Web services, and as soon as they are received, they are integrated into the map, while details of other layers continue to be obtained, allowing the client to perform other tasks besides the processing and details integration. In addition, the rest of the application consists basically of calls for services previously detailed. The Web client was implemented aiming at comparing its performance against a Desktop client, as will be detailed in the following section.

4. Performance Evaluation

The tests presented herein aim at comparing two progressive transmission approaches, using the same maps and the same technique. The first approach is currently used by the iGIS, described in Section 3.1 and detailed in [Costa 2006], which operates independent ly of the implemented GIS services. The second approach uses the GIS Services architecture proposed in this work. The aim of this evaluation is to compare these two approaches performance and observe the gain obtained with the GIS Services technique, which guarantees flexibility and platform and programming language independence to the client.

Table 1. Maps size and respective transmission times

		Brasil1	Brasil2
Original size (KB)		19139	35847
Quantized Size (KB)		6728	11223
Transm. Time Original Map (s)	**5KB/s (s)**	3827,8	7169,4
	12KB/s (s)	1594,9	2987,3
	25KB/s (s)	765,6	1433,9
Transm. Time Quant. Map (s)	**5KB/s (s)**	1345,6	2244,6
	12KB/s (s)	560,7	935,3
	25KB/s (s)	269,1	448,9

Two different maps were used (Brazil1, and Brazil2, with information on the Brazilian Political Division, whose real and quantized sizes are arranged in Table 1. The real size does not use any technique for efficient map transmission, and the quantized size concerns the map generated by the quantization technique for the chosen maximum resolution. Besides that, the maps transmission times are shown in the Table 1, in order to compare it to other tables that show data relative to those maps using the progressive transmission technique in both approaches.

We intended to measure the extent of the gain via desktop client architecture as compared with the Web client using Javascript processing. That can be verified in Tables 2, 3 and 4. In Table 2, column Incr. represents the increments size for a given map level. Column TJS is the Javascript approach, while column TGS represents the desktop client processing time using GIS Services. The last column represents the Gis Service approach gain with respect to Javascript, meaning how faster the former is than the later. The processing time for the map at level 0 is zero, since the server sends a low resolution map to the client, avoiding any integration processing for map rendering.

Notice that the maps' rendering time was neglected once it did not present much variation in the two approaches.

Table 2. Processing times comparison between clients for Brazil 2 map

Brasil2	Resolution	Incr. (KB)	TJS (s)	TGS (s)	TJS /TGS (gain)
level 0	75x75	1476	0	0	0
level 1	150x150	2876	527,86	20,51	25,74
level 2	300x300	3076	542,28	24,10	22,50
level 3	600x600	3012	555,98	22,88	24,30
level 4	1200x1200	3103	561,13	24,57	22,84
level 5	2400x2400	2917	557,16	24,52	22,72
level 6	4800x4800	2094	540,93	25,53	21,19

In addition, we obtained some additional information concerning the GIS service architecture's real gain compared to the use by Web Javascript clients, weighing the response time (increments transmission time plus their processing time to integrate them into the map). To do that, the results were simulated in low bandwidth networks (hypothetical rates, ideally constant), 5, 12 and 25 KB/s. Values for each network and for each map are shown in Tables 3 and 4.

Table 3. Response times comparison between clients for Brazil 1 map

Brasil1	5 KB/s		12 KB/s		25 KB/s	
	JS	GSD	JS	GSD	JS	GSD
level 0	285,4	285,4	118,9	118,9	57,1	57,1
level 1	506,9	444,4	253,2	190,7	158,9	96,4
level 2	514,4	456,5	254,1	196,2	157,4	99,6
level 3	482,3	419,6	244,3	181,6	155,9	93,2
level 4	376,3	309,5	202,4	135,6	137,7	71,0
level 5	221,9	155,8	138,0	71,9	106,8	40,7
level 6	144,9	78,7	105,6	39,4	91,0	24,8

When comparing the times of Tables 3 and 4 with the times of Table 1, the transmission times of the original and the quantized maps, we observe the usability and efficiency gain of the proposed architecture. Within it, a large step (complete map transmission, in a huge size) is divided in various additional steps (increments transmission), meeting the user's necessities. That can be viewed in Figure 7. The desktop client gain with respect to the Web client is observed comparing both clients' times with a given rate, in Tables 3 and 4. For example, in Brazil 2 map, the desktop client has a total average gain of about 4 to 5 times over the Web client. In the case of the real map transmission, even though it does not demand further details and processing, the initial real map transmission (the only one) takes too much time, being this the main transmission problem of large maps in the Internet. Comparing the progressive transmission techniques with the quantization one, it can be verified the more details are necessary the more advantageous is the use of the progressive approach.

In Tables 3 and 4 we can see that the desktop client gain becomes relatively more evident for higher bandwidths, since the increments transmission time, which is constant for both progressive transmission approaches, has stronger impact in the application than the increments processing time for small transmission rates. Thus, for low bandwidths, a client with large processing capacity can not compensate the problem of efficient data transmission, which is fairly obvious. On the other hand, the use of the progressive transmission technique (irrespectively of the client) for low bandwidths generates great efficiency in the system, for example, when comparing the progressive transmission method with the quantization technique.

Table 4. Response times comparison between clients for Brazil 2 map

Brasil2	Transmission + Processing Times					
	5 KB/s		12 KB/s		25 KB/s	
	JS	GSD	JS	GSD	JS	GSD
level 0	295,2	295,2	123,0	123,0	59,0	59,0
level 1	1103,1	595,7	767,5	260,2	642,9	135,5
level 2	1157,5	639,3	798,6	280,4	665,3	147,1
level 3	1158,4	625,3	807	273,9	676,5	143,4
level 4	1181,7	645,2	819,7	283,1	685,2	148,7
level 5	1140,6	607,9	800,2	267,6	673,8	141,2
level 6	959,7	444,3	715,4	200	624,7	109,3

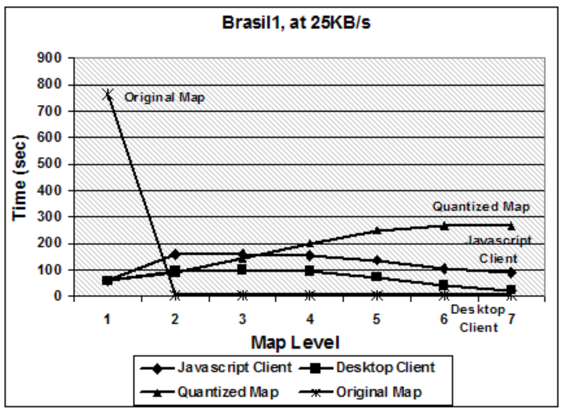


Figure 4. Transmission time comparison for Brazil 1 mp , at 25KB/s, using different approaches

From the tests and the proposed architecture, it was possible to observe some characteristics and advantages derived from the use of the GIS Services when implementing a Map Server with progressive transmission. A higher flexibility in the clients' implementation was reached in different platforms, irrespective of the used programming language. In addition, the asynchrony characteristic fits very well to the problem of progressive transmission: while the client waits for the images' details transmission, the client can perform other tasks in parallel, such as reception and processing of the previously received details. Another benefit derived from the use of Web Services is to provide the client with larger processing capacity by using desktop applications, compared to the scripts running in web browsers.

5. Conclusions and Future Work

In this work was presented a GIS Services architecture for a map server that supports progressive transmission. It was illustrated the use of the Web Services in Geographical Information Systems so as to supply those systems with the advantages of Web Services, according to the Gis' specific characteristics. The proposed architecture of the progressive transmission GIS Service and its operation were illustrated and the Web service's methods were detailed.

The proposed architecture allows different clients to use vector maps, independently of platform or programming languages. Moreover, the characteristic of the progressive transmission provided better usability and better response time as the application client visualizes and interacts with the maps, mainly when low bandwidth is available.

The GIS Services' architecture allowed the implementation of a desktop client, and it was possible to compare the proposed architecture performance (based on GIS Services, desktop), with a Web Gis, which did not make so evident the gains got through progressive transmission due to the many DOM calls that updated the SVG map.

As future work, we expect to further extend the proposed GIS Services, including other techniques in the architecture, besides the Progressive Transmission, such as multi-resolution and buffering, in order to enhance the system's efficiency and usability.

References

Alameh, N.: Chaining Geographic Information Web Services. IEEE Internet Computing, Vol. 07 (September 2003), pp. 22-29.

Apache Axis. Available online at:http://ws.apache.org/axis/java/index.html. Last accessed on: July, 2007.

Baptista, C. S., Silva, E. R., Leite Jr., F. L., Paiva, A. C.: iGIS: A framework for Web Mapping, Eighth East-European Conference on Advances in Databases and Information Systems. Budapest, Hungary,2004.

Bertolotto, M., Egenhofer, M.: Progressive Vector Transmission, ACMGIS'99, p.152-157.

Bertolotto, M., Egenhofer, M.: Progressive Transmission of Vector Data over the World Wide Web. GeoInformatica 5(4):345-373, 2001

Buttenfield, B.: Transmitting Vector Geospatial Data across the Internet. In: M. J. Egenhofer and D. M. Mark (editors), GIScience, 2002, LNCS, pp. 51-64, Berlin.

Cecconi, A.: Integration of Cartographic Generalization and Multi-Scale Databases for Enhanced Web Mapping, Doctor Thesis, Zurich University, 2003.

Costa, D. C., Paiva, A. C., Teixeira, M. M., Baptista, C. S., Silva, E. R.: A Progressive Transmission Scheme for Vector maps in low-bandwidth environments based on device rendering. 3rd International Workshop on Conceptual Modeling for Geographic Information Systems (CoMoGIS'06), ER2006, 2006

Ferris, C., Farrell, J.: What Are Web Services, Communications of the ACM 46(6), 2003, p. 31

Han, Q., Bertolotto, M.: A Multi-level Data Structure for Vector Maps, GIS'04, November 12-13, 2004, Washington, DC, USA, ACM 2004.

Jeong, C. W., Yu, S. D., Kim, M. S., Chung, Y. J., Lee, J. W.: Development of LBS Application using GML. Research Center for Advanced LBS Technology of

Chonbuk University, Korea, 2004, Available at http://www.gmldays.com/gml2004/papers/GML_LBS-Chonbuk.pdf.

Liang, C., Lee, C. H., Lee, J. D., Bae, H. Y.: Scale-Dependent transmission of spatial vector data on the Internet. The 3rd International Conference on In-formation Integration and Web-based Applications & Services, Austria, 2001.

Luo Y., Liu, X., Wang, W., Wang, X., Xu, Z.: Web Service and Geographical Information Integration. COMPSAC Workshops 2004, p.130-133

Oh, Y.: Advanced Progressive Transmission for Spatial Data in Web based GIS, The Second AEARU Workshop on Web Technology, Oct. 1999, pp.41-46.

Panatkool, A., Laoveerakul, S.: Decentralized GIS Web Services on Grid, Proceedings of the Open source GIS - GRASS users conference 2002, Italy, September 2002.

Papazoglou, M. P.: Service -Oriented Computing: Concepts, Characteristics and Directions. wise, p. 3, Fourth International Conference on Web Information Systems Engineering (WISE'03), 2003

Sliwinski, A. Toward Perceived Value-based Pricing of Geographic Information Services. Proceedings of the 7th AGILE Conference on Geographic Information Science, pp.541-549, 2004.

Tao, V.: Online GIServices. Journal of Geospatial Engineering, Vol. 3, No. 2, pp. 135-143, December, 2001.

Tsou, M.H., Buttenfield, B.P.: Client/Server Components and Metadata Objects for Distributed Geographic Information Services. In Proceedings of the GIS/LIS'98, pp. 590-599, 1998

TU, H.: Pattern Recognition and Geographical Data Standardization. The Proceedings of Geoinformatics'99 Conference, Ann Arbor, 19-21 June, 1999, pp. 1-7

World Wide Web Consortium (W3C),Web Services, http://www.w3.org/2002/ws

World Wide Web Consortium (W3C), Web Services Architecture, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211

Yang, B.S., Purves, R.S., Weibel, R., 2004: Implementation of progressive transmission algorithms for vector map data in web-based visualization, International Archives of Photogrammetry and Remote Sensing, Istanbul, Turkey, July 12-23, 2004, 6 p.

Yumin, T., Tianhe, C.: Web-based GIS services in participatory forest management in China. Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International Volume 7, p.4795 - 4798, 2004.