Proceedings of the First International Workshop on Telegeoprocessing: TELEGEO'99. 1999, p. 71-75

Making profit of software interoperability in a web enabled water point inventory distributed system

Santiago Comella, Rafael López, Rubén Béjar, Javier Zarazaga, Pedro R. Muro-Medrano

Departamento de Informática e Ingeniería de Sistemas
Centro Politécnico Superior, Universidad de Zaragoza
María de Luna 3,   50015 Zaragoza
{prmuro@posta.unizar.es}

To submit to TeleGEO99 (Lyon, May 99)
Draft of 11/8/2007 4:33:00 PM

ABSTRACT

This paper presents the interoperability aspects of a software application used to maintain a water point inventory system and generate official statement files. The application needs to manage tabular data and images, many of them georeferenced, coverages, texts, … The system approach is to integrate several specialized software applications (GIS visualization, database application, word processor, image processor, web, …) and make them interoperate by using open system standards.

1 Introduction.

Geographic Information System (GIS) constitute one of the fastest-growing market sector of the software industry and for good reason: GIS data are valuable for a wide range of users, from city planners to tax assessors [Gol97]. This fast-growing market has created a increasing necessity of systems heavy customized to user's requirements. There are also other requirements shared with other markets of the industry: Improved development productivity, interoperability and openness between application, and web capabilities. GIS technology has focused in four big trends to achieve all these requirements [Thr98]: OpenGIS, spatial functionality within database management, Internet technology, rapid development software tools such as OCX. All these trends are sharing the underlying concept of interoperability.

The basic software approach for interoperability is the use of components and objects based on open standards for component integration such as CORBA or COM/DCOM. This approach allows the construction of software systems based on components (that may have an independent live) and the use of COTS elements that can interoperate together sharing computing services and data. That interoperability strategy facilitates adaptability, scalability and integrability [Mag98].

We present in this paper the interoperability aspects of a software application used to maintain an inventory of water points and generate official statement files. The application needs to manage tabular data and images, many of them georeferenced, coverages, texts, … The system takes the advantages of several specialized software applications (GIS visualization, database application, word processor, image processor, web, …) and integrates them making profits of some of their individual functionalities working now for a new joined application. On the other hand, this technology make very easy for the new application to provide services that could be used by still unknown software applications.

The rest of this work is organized as follows. First of all some word semantics is established and an overview of the IPA, a water point inventory system, is briefly presented to provide the basics for the application problem. Next, the technological interoperability approach for the IPA is illustrated which is followed of its architectural overview with some example scenarios. The paper finishes with the conclusions and bibliography.

2 Dealing with the integration of different geodata and applications

This work is about making a set of unrelated applications look like a whole-integrated system in a GIS context. First of all, there are some related concepts with similar meaning that can create confusion: Interoperability, Openness and Integration. They are really quite simple with clear enough relations: a system must be open in order to interoperate with it, and when two or more systems interoperate in a tight enough way they can be considered integrated. According to the definition of the Oxford dictionary, integration means “putting together parts of a whole”, however, we would rather use it as “bringing together independent parts”. The main difference lies in the autonomy of the parts, i.e., are the parts intended to brought together and only meaningful as a whole, or do they have a meaning by themselves? [Vch98]. For “open system” we will adopt the definition of the USA Department of Defense (DoD) Open system Joint Task Force:

> *A system that implements sufficient open specifications for interfaces, services, and*

supporting formats to enable properly engineered components to be utilized across a wide range of systems with minimal changes, to interoperate with other components on local and remote systems, and interact with users in a style that facilitates portability" [DOD96, DOD97].

Finally, we define interoperability, extending the definition given by the OpenGIS Consortium, Inc. (OCG) [OGC96], as:

The ability of digital systems to 1) freely exchange all kinds of information; and 2) co-operatively, over networks, run software capable of manipulating such information.

Before speaking about integration we should wonder if commercial systems in general and GIS packages in particular are "open". Basically: yes but not enough. Most of them export their functionality in a RPC (Remote Procedure Call) interface as DCOM or CORBA, RPC interfaces are "conditio sine qua non" to make a system open, but to make any system really open a higher level specifications for interfaces would be necessary, for example, in compound documents the popularization of OLE (a specification built over COM) has made most of the Windows based applications really open, there are also several high level specifications being developed for different vertical markets, in GIS market the OpenGIS Consortium, Inc. (OCG) is developing the Open Geodata Interoperability Specification (OpenGIS) that is a comprehensive specification of a software framework for distributed access to geodata and geoprocessing resources [OGC96]. OpenGIS is aimed to get interoperability between GIS packages minimizing reengineering of used components. This is a really ambitious target, it will enable an "intergalactic" interoperation: every system OpenGIS compliant will be able to interoperate with everyone else. Unfortunately, OpenGIS is not currently available in any GIS package. Then, current interaction solutions in GIS environments had to be based only in RPCs, however, is desirable to have OpenGIS specifications in mind in order to make solution which will be easily improved to be OpenGIS compliant in the future.

2.1 IPA: a water point inventory system

A good example of interoperability requirements are hydrographic confederations, which deal with activities related to managing, planning and controlling the hydrographic aspects for a river's catchment area. One of their main responsibilities is granting licenses for water exploitations, in this procedure they have to deal with administrative paperwork and GIS analysis in order to prevent over-exploitation of natural resources. To carry out this role the organization maintains an inventory of every water point in their zone of influence. They have taken a census of more than thirty thousands wells, all of them geo-referenced and related to more than twenty relational tables and as many geographical coverages. A great amount of work is involved maintaining this inventory.

We will now describe, in a schematic way, the typical day-to-day process carried out by the confederation, and we will show one of the many different procedures involved in the water point inventory. Suppose that a new water point is registered. First, the point must be georeferenced using any GIS application, this involve to relate the new point with pre-existing information: 250 satellite raster sheets (more than 70 Gbytes); more than 15 vectorial coverages as natural resources, irrigated lands, geological aspects and so on. Second, information about the new point must be introduced, this information can be tabular data stored in relational databases or pictures from the water point that need to be processed using photo image processing tools. Finally, a report of the overall process should be generated, this inform, which comprise both: geographical and tabular data, should be able to be accessed via Internet.

This complex process reflects in the work flow of the office, and it forces the organization to use several sets of applications: GIS packages, relational database managers, word processors, photo image processing tools, etc, with low integration between them. Several problems arise from this form of work: a Clerk should jump from application to application in order to process a single expedient, there was no easy way to integrate outputs from GIS packages and office applications, batch conversions are needed in order to make the well inventory, which are normally stored in a relational table, visible from GIS packages , etcetera. There are other considerations involved in the problem as well: First, data value and concurrent access demanded for a centralized and reliable way to store the water point's inventory. Second, Internet and remote work must be considered, for example: a report must be able to be accessed or a new point to be added by Internet. Third, work should be automated releasing the worker from as much mechanical work as possible.

3 The technological approach proposed for the IPA

Some of above requirements could be addressed by commercial products, for example:

centralized data, reliable storing and common access of data from GIS and office applications could be achieved by data servers with spatial capabilities [Buc98]. Automation of work could be partially got using GIS applications' programming languages, however, this automation is limited to whose utilities and functions supported by these proprietary languages. Finally, Internet access could be achieved using Internet mapping servers. These solutions suite most of user's requirements, but alone are no enough to fulfil expectations from very specialized users, the reason lies in the large granularity of the items used which does not allow a real control from developer. Another drawback of this solution is the great amount of licenses of expensive software needed, you are paying huge packages to use a very limited set of their capabilities.

In general, depending on the grade of customization wished by a client the system should be constructed using larger or smaller grained building blocks. For example, to average users, built packages as proposed in last paragraph could be the best solution. In the contrary, to very specialized users, that needs very tight control of their system, almost everything should be built from scratch (using only LAN and low-level libraries). We have adopted a compromise solution, we could not get enough customization and integration using only pre-made systems, but we do not need (and could not effort) to built entirely the system. Most of the system are based in pre made packages, only a part has been built using GIS specialized medium granularity components, this subsystem acts as the glue of all other subsystems, it integrates everything, to make the system seem a whole.

We will now describe an overview of our system, including possible extensions and alternatives. Next, for further understanding, we will trace some working scenarios with the more usual tasks performed by our system.. Finally we will focus on extensions to web enabling our solution.

4 IPA architectural overview

In the next picture we show the overall architecture of our system. We can initially divide the system in the typical three layers: Data, Application (or system logic) and Client layers. From an Internet view point it is a typical three layer architecture, with a client layer operating as a graphic interface for users, a logic Layer serving and processing requests from remote clients and a data tier working as a data repository. But the application tier can be used also directly as an integrated environment by workers of the confederation to perform special processes, in this operation mode the architecture would be reduced to a simple client/server architecture. This unevenness could have been avoiding developing Internet clients to every functionality in the system, however it only worth to such cases often used and whenever a remote access was needed. In the next paragraphs we will detail every tier in the system focusing later on the application tier and its components.

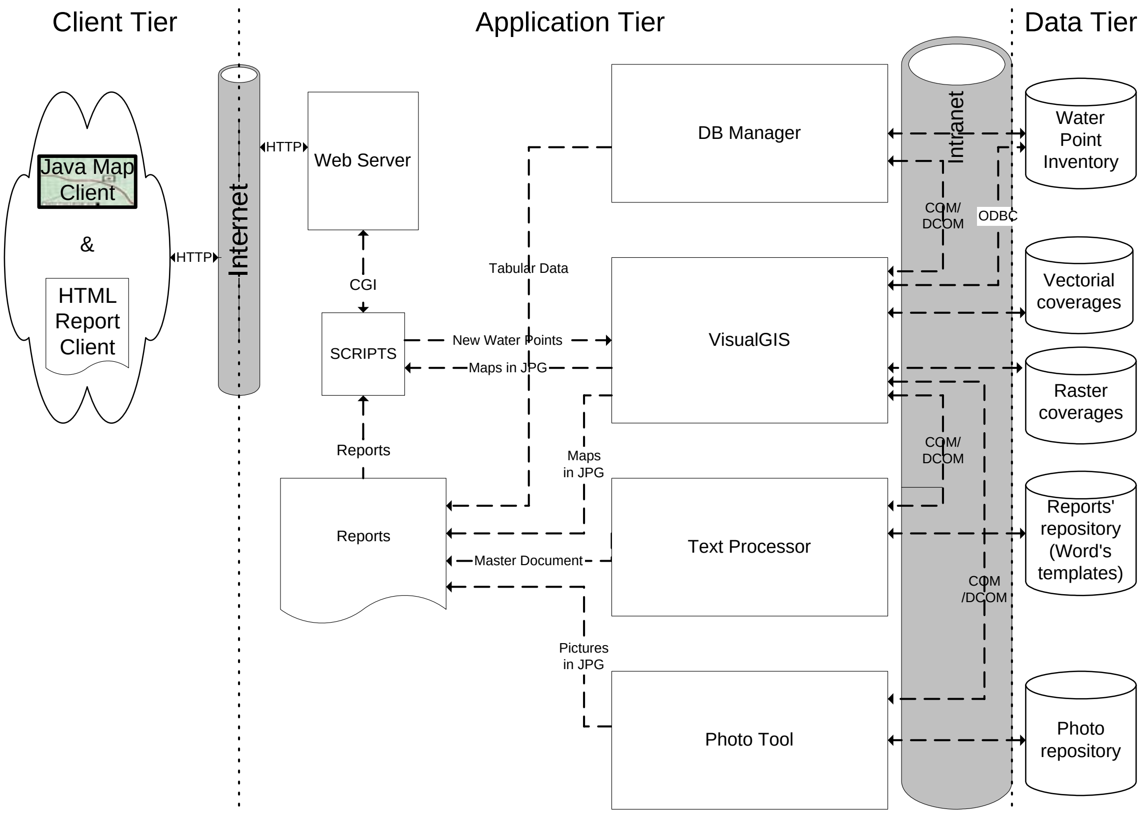

Figure 1: IPA architecture.

The inner layer, called data tier, is a centralized data repository. Of course, the data that is less subject to change can be decentralized in order to minimize data transit through the Intranet. Nowadays, the application layer directly access data storage, we are planning to introduce a data server to improve reliability and transfer performance over the net.

For the client tier, a Java light client has been developed to enable map visualization and graphical introduction of new water points. This client presents an interface to navigate across map layers zooming, panning and so on. It translates user‘s events into commands to the map server (Our VisualGIS) and shows the pictures returned by the server, all in all, not very different to many market Web mapping tools. Additionally, we are now designing a few HTML pages to enable users to query reports over Internet.

The main part of the system is the application layer. As we had already pointed out, it can operate, in addition with a traditional Internet CGI Server, as a web server with mapping capabilities, to enable this operation mode, our VisualGIS component exports its functionality through COM interfaces to let the CGI scripts from the server to control and collect outputs from it. On the other hand, it also has a Graphical User Interface to enable operators from the hydrographic confederation to directly perform some tasks. In both operation modes, it makes transparent to the user the inter - application communication that performs to carry out tasks.

In the section we will focus on the main components of the application layer, explaining their main responsibilities and interactions between them.

4.1 A closer look.

The central item of this layer is the VisualGIS, is almost the only piece of code built, of course it has not been built from scratch, we have used C++ as programming language and an ActiveX[1]'s library of components specialized in map visualization and manipulation. VisualGIS is itself an independent system that can be used as a complete map viewer that can be extended in several ways in order to fit new needs from projects. But the main purpose of the VisualGIS is to act as the glue of all the system, it manages and coordinates all the other software items using their COM interfaces to make every software packages perform the entrusted tasks. Most of the time the worker only uses this package's GUI, running everyone else in background or even remotely (using DCOM). We will enumerate all this "helper" packages explaining the relation with VisualGIS.

The first needed helper is a database manager that is mainly responsible for maintaining the relational tables, but it can also be used to execute complex relational queries. We have implemented a COM and SQL relation between our VisualGIS and the database manager in order to export a relational view as the selection of our VisualGIS (and also to import the selection of our VisualGIS as a relational View). Therefore, we can use the database manager to perform relational queries and our GIS Viewer to make spatial ones.

To carry out our reporting needs we have included a word processor, we could have used an specialized reporting component, but we have enough reporting capabilities using only a commercial word processor which licenses were already paid. We have built the reports using

[1] Activex are components built on top of COM see [Pla97]

the graphical editing capabilities of the word processor and storing them on disk. Whenever reporting is needed, we can command the word processor to load these reports and every involved software to fill their respective fields, for example, the VisualGIS will insert all needed maps, the database manager tabular data and so on.

There are other packages involved in the system as photo image processors, or charting tools, but their behavior is pretty similar to those already explained.

4.2 A typical working scenario

This scenario shows how the IPA tabular data application can interoperate with the VisualGIS application from the user's point of view. Figure 2 shows a typical situation where the user is selecting some points in one of the applications using the mouse, a good integration between both applications implies that the same points be selected in the other, this facilitates the edition tasks providing faster information to the user and produces the effect as if it is a unique application. This interrelation is also used to insert new water points and update some georeferenced information in the database coming from geographic coverages, and in general to provide a graphical view of tabular data whenever the information is georeferenced.

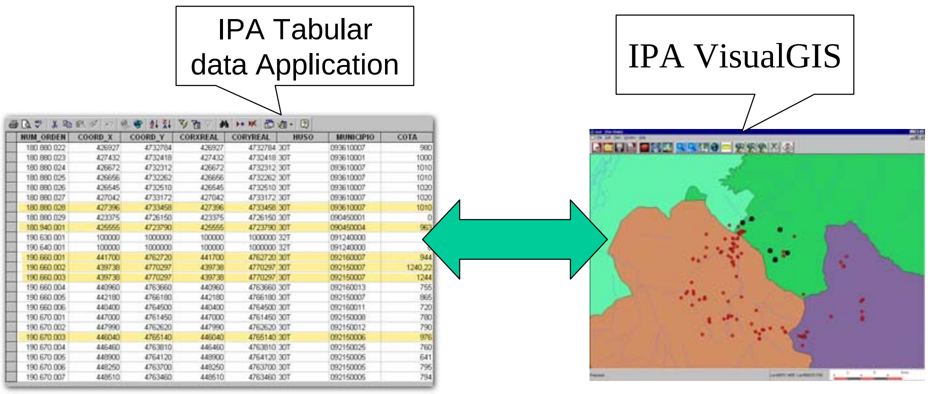

Figure 2: Water points are selected and the selection is synchronized between both application components

4.3 A web powered scenario

Figure 3 shows another typical scenario using the GIS visualization component from a web-based application. In this scenario, a request is made in a web page to show an image of the geographic area around a water point. The request is propagated by means of CGI to the VisualGIS component which has this tasks as one of its services. VisualGIS must process the request performing some computation in several vectorial and raster coverages. The output of the service is sent back to the web client.

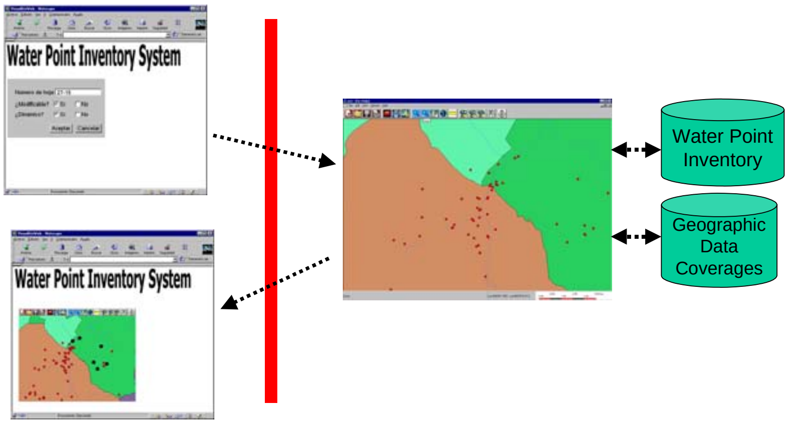

Figure 3: Request for the graphic visualization of a water point.

5 Conclusions

This paper has shown a practical experience of the use of interoperability to develop a software application for a water point inventory system. We already had some components supporting much of the needed functionality but we had to add more functionality and make all the components work together in a way more transparent to the user. The technical approach taken was to develop the least possible components from scratch and making them interoperate with the rest of available components (writing the appropriate interface layer using interoperability standards). Some of the new system functionalities are now in a prototype state in the *Hydrologic Planning Office* of the *Confederación Hidrográfica del Ebro* (this office is in charge of the water point inventory system of the Ebro River in Spain).

OpenGIS standard will be a good technical decision in a short future but we did not followed it to create de software because the lack of commercial components. We used the standard COM to interoperate instead of CORBA because some of the commercial components we used already supported that standard. Standard SQL was also used for database querying.

The possibilities provided by the basic interoperatibility standards have improved our expectations. In this sense, our developing work has made apparent to us that the possibilities for scalability, adaptability and integrability are now much more tangibles. We still have some work to do with the system and we a looking for more functionalities to be incorporated.

6 References.

[Buc98] R. Buckley, "Oracle SDC: the new features ", *Mapping awareness,* Oct-98.

[Thr98] G. Ian Thrall, "The big four", *Geo Info Systems*, v.8 pag.31, Sep-98.

[Mar98] P. Marshall "GeoMedia 2.0: A powerful tool for GIS developers" *Federal Computer Week;* Falls Church, Jul-6-98.

[Gol97] H. Goldstein, "Mapping convergence GIS joins the enterprise", *Civil Engineering,* V 67, pp 36--39, Jun-97.

[Koc98] V. P. Kochikar, "The Object Powered Web", *IEEE Software*, pp 57, May/June-98.

[CB98] S. Coetzee and J. Bishop, "A New Way to Query GISs on the Web", *IEEE Software,* pag.31, May/June-98.

[Vch98] A. Vchovsky, "Interoperable and Distributed Processing", *Taylor & Francis*, 1998.

[DOD96] DoD, "Directive 5000.1 Defence Acquisition", 15 March 1996.

[DOD97] DoD, "Regulation 5000.2 Mandatory Procedures for Major Defence Acquisitions Programs (MDAPs) and Major Automated Information System (MAIS) Acquisition Programs", 8 October 1997.

[OGC96] Open GIS Consortium, "The openGIS™ Guide", 1996.

[Pla97] D. S. Platt "The essence of COM with ActiveX", *Prentice Hall*, 1997.