A Generic Framework for GIS Applications*

Miguel R. Luaces, Nieves R. Brisaboa, José R. Paramá, and Jose R. Viqueira

Laboratorio de Bases de Datos
Facultade de Informatica, Universidade da Coruña
Campus de Elviña, s/n. 15071. A Coruña. Spain
+34 981 16 70 00 Ext. 1306
Fax: +34 981 16 71 60
{luaces, brisaboa, parama, joserios}@udc.es

Abstract. Geographic information systems (GIS) are becoming more usual due to the improved performance of computer systems. GIS applications are being developed using the three-tier software architecture traditionally used for general-purpose information systems. Even though this architecture is suitable for GIS applications, the special nature and exclusive characteristics of geographic information pose special functional requirements on the architecture in terms of conceptual and logical models, data structures, access methods, analysis techniques, or visualization procedures.

In this paper, we propose a generic architecture for GIS that provides support for the special nature of geographic information and conforms with the specifications proposed by the ISO/TC 211 and the OGC. Our strategy to achieve this goal consists of two steps: (i) we analyze the special characteristics of GIS with respect to traditional information systems, (ii) and we adapt the traditional three-tier architecture for information systems to take into account the special characteristics of GIS.

Finally, we have tried to apply the architecture that we propose in the development of a complete and complex real-life GIS application using commercial tools in the analysis, design and implementation. We describe this application, and we use it to describe the limitations of current commercial GIS development tools by analyzing the differences in the architecture of the resulting system with respect to our proposal.

1 Introduction

Until a few decades ago, manipulating, synthesizing and representing geographic information was restricted to paper maps and these tasks were limited to manual, non-interactive processes. The exponential improvement in the performance of computer-based technologies and the increasing demand for interactive manipulation and analysis of geographic information have created a need for *geographic information systems* (GIS) [1,2]. An important characteristic of geographic information systems is that they are more than tools to produce paper maps. Whereas in traditional cartography the paper map is the database, in a GIS the map is only a projection of a particular view of a geographic database at a given time. This enables the GIS end-user to review an unlimited number of analysis alternatives, and to make maps from different points of view emphasizing different aspects of the information. As a consequence, functional requirements for GIS are vast and go far beyond those of traditional information systems [3,4,5,6].

After many years of research and development, it is now generally accepted that the architecture of general-purpose information systems must consist of three separate tiers, namely: the *presentation tier*, the *application logic tier* (or *business logic tier*), and the *data tier*. The main advantage of this architecture is that it enforces a strict separation of the functionality of the system into three different independent modules that interact only

* This work was partially granted by CICYT (refs. TIC2003-06593 and FIT-150500-2003-588), and Xunta de Galicia (ref. PGIDIT02SIN10501PR).

at well-defined interfaces. This enables a developer to modify each one of these modules of the application with little impact on the others. Therefore, this architecture provides increased performance, flexibility, maintanability, reusability and scalability.

Even though the three-tier architecture for general-purpose information systems is suitable for GIS, the special nature and exclusive characteristics of geographic information pose special functional requirements on the architecture in terms of conceptual and logical models, data structures, access methods, analysis techniques, or visualization procedures. For instance:

– Special data types and operations are needed to represent and manipulate geographic information.
– Geographic information requires many different analysis and visualization procedures.
– Geographic information is typically voluminous with a naturally imposed hierarchical structure.
– Geographic information processing is characterized by transactions that are much longer than a typical standard relational database transaction.
– There are two different conceptual views of geographic space: an object-based view and a field-based view.
– Additionally, each conceptual view of space can be represented in many different ways in a computer.

These and other features impact the overall architecture of a GIS. Therefore, it is very important to determine the special requirements and functionality of GIS applications beyond those of general-purpose information systems in order to design and implement a GIS application framework with appropriate capabilities for modeling, collecting, querying, and visualizing geographic information. This is precisely the main goal of our work: proposing a generic architecture for a GIS framework that provides support for the special characteristics and requirements of geographic information.

The rest of this paper is structured as follows. First, in Section 2 we analyze the special features of geographic information that pose requirements on the architecture of GIS beyond those common to general-purpose information systems. After that, in Section 3, we review the OpenGIS Consortium (OGC) and the ISO Technical Committee 211 (*ISO/TC 211, Geographic Information/Geomatics*) proposals of standards for representing and manipulating geographic information. We also point out in this review the benefits and drawbacks of these proposals. Then, in Section 4 we introduce our proposal for a generic architecture for GIS that fulfills the requirements described in Section 2. Section 5 describes the development of a GIS for the Provincial Council of A Coruña that tries to apply our proposal, and Section 6 is devoted to the analysis of the differences between the proposed architecture and the implemented GIS application caused by the limitations of commercial GIS development tools. We end this work by giving some concluding remarks and describing future work in Section 7.

2 Special Characteristics of Geographic Information

The special nature of geographic information imposes some requirements on the architecture of the information system. We enumerate some of these requirements in this section.

Representation of Geographic Information. The conceptual models used for information systems (e.g., the entity-relationship model) do not have constructs to model application schemas that deal with geographic information. Furthermore, logical models (e.g., the relational model) are strongly geared toward business applications that manipulate large but simple data sets, and do not include functionality to represent geographic information. In addition to data types for numbers, texts and dates, the representation of geographic information on a computer requires new data types such as *point*, *curve*, *surface*, or collection types like *point collection*, or *geometry collection*. Finally, physical models for traditional information systems are unable to represent efficiently geographic information. Summarizing, traditional information systems must be extended at all levels from the conceptual model to the physical model in order to represent geographic information adequately.

Geographic Information Processing. There is a rich set of special transformation, manipulation and analysis techniques applicable to geographic information, which must be integrated within the information system. This must be done by providing an exhaustive set of primitive operations on the data abstractions of the conceptual model, which must be integrated in a query language to retrieve and manipulate the data abstractions of the conceptual model. Finally, problem-solving techniques must be used for the following categories of problems:

– *What is it at this location?*, and *Where is this located?*
– *What is the spatial relationship between these objects?*
– *What location satisfies these requirements?*
– *What will be the situation in the future?*

Visualization of Geographic Information. The visualization of geographic information is a distinctive feature of GIS applications compared with general-purpose information systems. Geographic information has some peculiarities that have an impact on the presentation process:

– Different abstractions must be used for the representation and the visualization of the information.
– It is multi-dimensional.
– It is voluminous.
– It is required at varying scales and sizes.
– It is required from different perspectives.
– It is projected onto a flat surface (i.e., on a computer screen or a paper).

Therefore, there is a need for appropriate metaphors to manipulate geographic information at the user interface of the system. These metaphors must be based on the well-known map metaphor, and must incorporate dynamic operations such as *zoom*, *pan* and the addition and removal of information.

System Architecture. The special nature of geographic information makes more important the fulfillment of some requirements of general-purpose information systems, such as flexibility, extensibility, reusability, scalability, reliability, and security. In order to provide these features, the architecture of the GIS must be based on an extensible DBMS providing geographic information management services, and a collection of modular, highly-distributed, geographic information processing and visualization services.

Other Issues. Geographic information poses other important requirements on the architecture of the system. First, as any other application domain for information system, the field of GIS needs special metadata elements to describe the particularities of the application domain. Metadata in the architecture of a GIS enables to find geographic information data sets, to describe how the information can be used, and to check whether the information satisfies some requirements.

In addition to this, the amount of possible analysis techniques for geographic information is unlimited. Therefore, no GIS development tool can provide all possible operations and problem-solving techniques. As a consequence, the architecture of a GIS development tool must be *extensible* to support the addition of new data analysis procedures. The process of extending the functionality of a GIS development tool is usually referred to as *customization*.

Finally, the temporal component of geographic information was systematically ignored by GIS development tools and the more general research field of spatial databases. This is changing in the last years, and many research efforts has been dedicated to the emerging field of *spatio-temporal* and *moving object* databases [7].

3 International Standards for Geographic Information Systems

Given that each application has a different point of view on geographic information, each developer has defined conceptual models, geographic data models, storage formats, analysis operations, or representation procedures specially adapted to the requirements of the application. As a consequence, there is nowadays a problem of interoperability between the GIS development tools; it is not easy to use one tool to analyze the information collected with another tool.

In order to solve this problem, a number of government, research and industry partners founded the OpenGIS Consortium (OGC) in 1994 to promote interoperability among GIS development tools [8]. Another standards organization that has devoted many efforts to GIS applications is the International Organization for Standardization (ISO) by means of the ISO Technical Committee 211 (ISO/TC 211), named *Geographic Information/Geomatics* [9]. The purpose of both organizations is to create specifications of standards concerning geographic information with detail enough to enable developers to create implementations conforming to these standards that interoperate without problems.

For many years, the OGC and the ISO were working independently to reach overlapping goals, but nowadays both bodies seek to converge toward a common solution. The OGC and the ISO/TC 211 are carrying out a very important task in the field of geographic information systems. They are laying the foundation for a new generation of GIS applications and development tools that will be able to cooperate to a greater extent at many different levels. Furthermore, instead of proposing a monolithic software layer implementing all the functionality of GIS applications, these organizations are proposing to break down the functionality in a vast collection of services with very specific functionality that interact only at the interfaces. This allows a very flexible GIS application architecture because the services implementing the functionality may be running in a single computer, or distributed along a wide-area network, in a totally transparent way.

We have depicted in Figure 1 the architecture of a GIS application with two different user interfaces built over the specifications already published. The first interface consists in a desktop-based GIS for geographic data analysis, and the second one is a web-based GIS for geographic information portrayal. The figure shows the specifications that are already adopted in white boxes, and the missing pieces of the architecture in gray boxes.

The work developed by these organizations has also some little drawbacks. For instance, even though the specifications provide a complete information management tier, much of the processing algorithms and information portrayal techniques must be defined by a developer. No specifications have been defined for the processing tier, and very little work has been produced for the human interface tier.

Another little problem is that the intrinsic nature of any standards organization causes the process of developing a specification to be rather slow. Before an specification is adopted, it must be proposed, written in a draft state, discussed and voted by the membership. This is a lengthy process that cannot be assumed by software companies that must produce novel products at a much faster pace.

Finally, the definition of specifications by multiple groups working independently may cause that the specifications do not match perfectly due to coordination problems. Some concrete examples of this problem are shown in the following list:

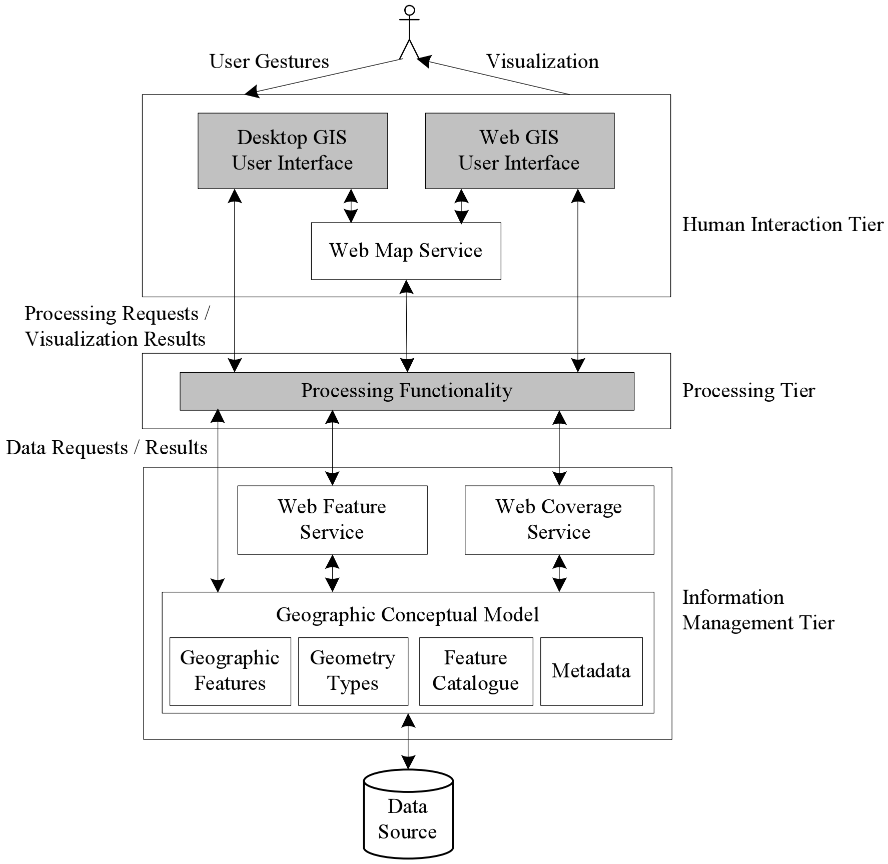

Fig. 1. OGC Services in the Architecture of a GIS.

- *Duplicate definitions.* The specifications for both the Geographic Markup Language (GML) [10] and the Styled Layer Descriptor language (SLD) [11] provide a definition for constructs to specify the visual style of a cartographic object generated from a geographic feature.
- *Missing functionality.* The Web Feature Service (WFS) [12] is defined as a service for querying and manipulating geographic features. However, instead of providing a complete query language like the one defined for the simple feature geometry model [13], the definition imposes some limitations on the types of queries that can be posed over the data source (e.g., the language does not support relational joins, new values in the query result cannot be computed, or the complete set of spatial operations is not available).

In spite of these little drawbacks, the architecture and the specifications proposed by these organization must be considered as a starting point for the development of any GIS application. These specifications are the greatest source inspiration for the architecture described in the following section.

4 A Generic Architecture for Geographic Information Systems

After analyzing the special characteristics of geographic information and the requirements that they pose over the architecture of geographic information systems, we present in this section a description of the components and the interactions in a generic architecture for geographic information systems that meets these requirements. The design of this

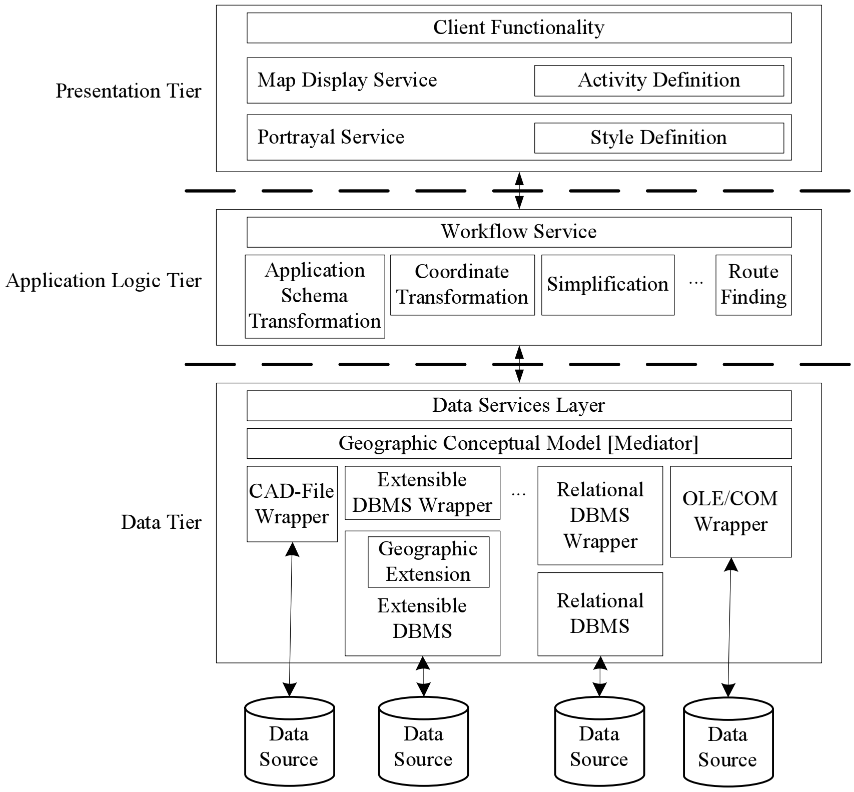

Fig. 2. A Generic System Architecture for GIS.

architecture is heavily influenced by the proposals of the ISO/TC 211 and OGC, and reuses the work of these organizations where their specifications are mature.

Figure 2 shows our proposal of a generic architecture for geographic information systems. The architecture separates the functionality of the system in three independent tiers, namely the *Data Tier*, the *Application Logic Tier* and the *Presentation Tier*. The Data Tier provides data management functionality independently from the software technology. The Presentation Tier is responsible for implementing the user interface of the system, displaying the maps and providing some basic functionality over them. Finally, the Application Logic Tier implements the problem-solving functionality of the system.

In order to enable reusability and flexibility of the system architecture, the functionality of these tiers must be implemented independently of any particular application. The strategy followed consists in finding and isolating the characteristics that are independent from the application schema and functionality from the dependent ones. Then, the independent characteristics of the system are implemented once using generic algorithms. The architecture built with these modules can be used as a framework for GIS applications by filling in the specific details of the application. We describe now each layer in more detail.

4.1 Data Tier

The purpose of the data tier is to provide information management functionality independently from the software technology used to store the data. This tier receives information retrieval and manipulation requests expressed using a query language, evaluates the query, and returns a set of data objects that are represented using an information exchange language.

Considering that there may be many different types of data sources, the internal architecture of the tier must be organized in a mediator-wrapper pattern. The mediator layer consists in a conceptual model for geographic information including data types and a query language for representing and manipulating geographic information, and metadata and catalogue information. Then, a wrapper module must be implemented for each different type of data source. For an extensible DBMS that supports the conceptual model directly by means of an extension module, the wrapper is very simple. On the other hand, for a relational DBMS the wrapper modules uses tables or large objects to store geographic information and implements the query language using memory operations. Similarly, legacy data formats like CAD files must be supported by wrapper modules.

The query language can be a text-based high-level query language (e.g., SQL derivatives), or it can be a low-level query language based on an application programming interface. The same alternatives exist for the query results, which may be accessed using a high-level, text-based descriptive language (e.g. GML [10]), or a low-level, application programming interface (e.g. OpenGIS Simple Features for OLE/COM [14]).

The topmost interface of this tier consists of a collection of *data services* that provide profiles of the conceptual model for specific applications (i.e., subsets of the conceptual model, query and results language, and metadata facilities). These services enable to hide the complexity of the underlying conceptual model by providing, for example, a simplified query language or exchange language. Two examples of services that can be implemented in this layer are the Web Feature Service and the Web Coverage Service defined by the OpenGIS.

4.2 Application Logic Tier

The application logic tier comprises the business logic of the system. This tier must be composed of multiple independent *services*, which are modules responsible of performing well-defined and simple tasks. Each service is defined by giving its interface as a set of operations and a description of the results. When a service operation is invoked, the service module answers the request either by using its internal information, or by building and issuing the appropriate queries to the Data Tier and manipulating the data returned.

The services existing in the architecture cannot be predefined because the functionality necessary for GIS applications cannot be known in advance. We can foresee the need for some services such as a route-finding service in networks, a geographic value simplification service, a coordinate transformation service, or an application schema transformation service. In order to achieve more complex tasks, multiple services must be chained by means of a *workflow service*. This kind of service allows developers and end-users to build a new service by connecting a collection of simple services.

4.3 Presentation Tier

The Presentation Tier is responsible for the user interface of the system that enables data visualization, data manipulation and data entry. The presentation tier receives the user interaction in the form of mouse gestures, keyboard inputs or other device inputs. These inputs are evaluated and the appropriate operations in the application logic tier are invoked. When the results are returned, they are displayed to the user using the appropriate user interface controls and visualization metaphors.

The most important component of the presentation tier is the portrayal service, which is in charge of converting a collection of geographic features into a collection of cartographic objects that can be rendered on a display device. The portrayal process is controlled by a set of style definitions, which must define precisely the way in which each geographic feature must be rendered.

The resulting cartographic objects are visualized using a *map display* service, which uses a *map metaphor* to allow the end-user to manipulate the displayed map. For instance, the map scale is changed by using the *zoom-in* and *zoom-out* metaphors.

In addition to visualization manipulations (e.g., scale and view change), the map display component must allow the end-user to manipulate the cartographic objects displayed in the map to perform geographic operations and to request processing operations from the application logic tier. The *activity module* associates these actions to user interface events that occur in the map. As an example, a developer can associate to the event *click over an element in the map* the action *display element information*. The implementation of the action is responsibility of the developer. However, it is necessary that the operations of the features in the Data Tier are accessible to the developer.

If all the functionality of the system is implemented in the application logic tier, the response time of the system may not be fast enough. In order to achieve faster response times from the system, the presentation tier may implement some functionality using the cartographic objects for the computations. This is implemented in the *presentation functionality* module, and consists in tools for the following tasks:

- *Control of the map scale and position.* This allows the end-user to focus on the map phenomena of interest by zooming and moving the map.
- *Management of the graphical legend.* It allows to understand the map by describing the real-world entities that are described by each symbol. It also allows to the end-user to customize the map by adding and removing additional features.
- *Measure distances and areas.*
- *Display the map context.* For instance, using an overview map, a graphical scale bar, or a north arrow.

4.4 Summary

We have proposed in this section a generic architecture for GIS applications and we have also described briefly each component of the architecture. This architecture proposal provides support for the special characteristics of geographic information and conforms with the specifications defined by the ISO/TC 211 and OGC where possible.

5 The EIEL Geographic Information System

In order to discover the funding needs of each municipality and to propose special action programs to balance the living conditions of the municipalities, each provincial council in Spain is required to conduct, every five years, a survey on local infrastructure and facilities, (named EIEL from the Spanish *Encuesta de Infraestructura y Equipamientos Locales*). The amount of information collected by the survey demands a tool to objectively analyze and evaluate the situation and state of infrastructure and facilities in each municipality.

The province of A Coruña is located in northwestern Spain. With more than one million inhabitants and almost eight thousand square kilometers, it is densely populated with more than a hundred and twenty-five inhabitants per square kilometer. The provincial council of A Coruña decided to broaden the goals of the EIEL for the year 2000. More particularly, these new goals were considered:

- Extend the information to be collected, both in terms of the different kinds of elements to be surveyed, and the amount of information for each particular item.
- Reference the items surveyed to its geographical location or extent.
- Build an information system with the information collected to be used by the provincial council staff, and build a publicly-accessible, web-based information system.

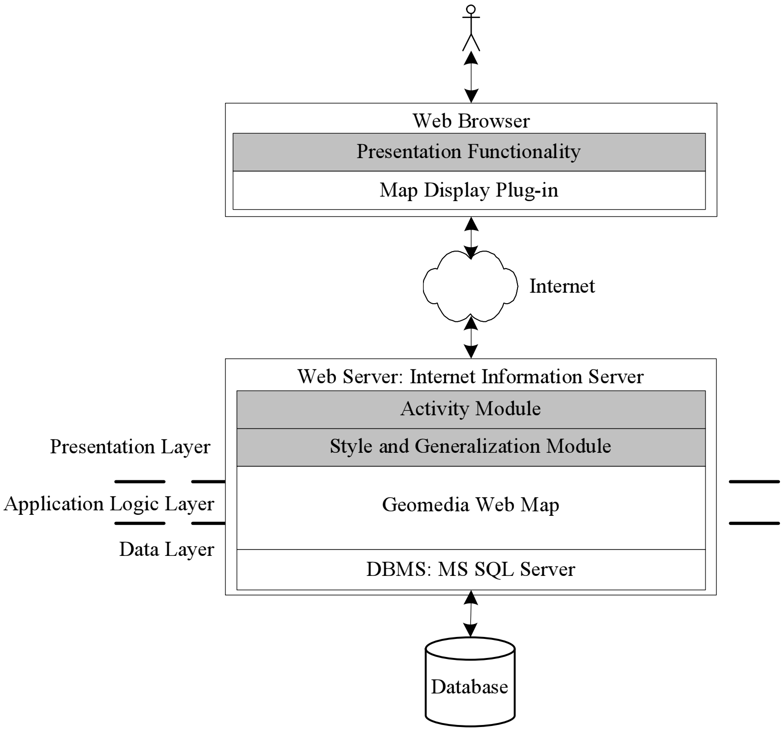

Fig. 3. System Architecture for WebEIEL.

These goals were achieved through a two-year project carried out by the University of A Coruña. A large group of students from the civil engineering school and the architecture school, supervised by a group of professors, collected the data by direct observation or interviewing the responsible staff in each municipality. At the database laboratory of the University of A Coruña, we designed and developed the applications supporting the data collection work flow. Then, we developed a geographic information system to manage and exploit this information [15].

We tried to apply the architecture proposed in Section 4 for the implementation of this GIS. This enables us to prove the applicability of our proposal in a real-world problem. Moreover, since we wanted the system to be ready in as little time as possible, we decided to use existing commercial applications instead of developing new software components from scratch.

Two independent applications were built at the end of the project:

– *A data maintenance tool (GISEIEL).* We designed and implemented an application to enable the responsible staff at the Provincial Council to correct and update the information stored in the GIS.
– *A web-based data exploitation tool (WebEIEL).* This application was designed in order to enable all the staff at the Provincial Council and citizens to browse the information collected by the EIEL. It can be found at http://www.dicoruna.es/webeiel.

Figure 3 shows the system architecture of *WebEIEL*. The components that we had to develop are shown with a gray background, whereas the commercial components are shown with a white background. On the server side, the data is managed by a relational DBMS (Microsoft SQL Server 7.0) and the geographic data model is provided by Intergraph Geomedia Web Map. The information of the web application is served by Microsoft Internet Information Server, the web server supplied by Microsoft. This particular server was chosen because it is required by Geomedia Web Map.

The client side of *WebEIEL* was implemented using dynamic HTML and a map display plug-in provided by Intergraph. This plug-in consists in a Java applet that displays the map information implements many presentation functionality such as the computation of distances and areas. The main reason for choosing a *thick-client* approach for the client was two-fold. First, to provide some interactive functionality in the client such as highlighting selected geographic objects, or displaying a tooltip message when the mouse is over a geographic object. And second, to display the map information using a vector image format that provides a much higher quality than raster image formats.

We had to develop many new modules and customize many existing components in order to implement this system, because the ones provided by Intergraph Geomedia Web Map were not sufficient to meet our requirements. Particularly, we implemented:

– *Style and generalization module.* One of the requirements of our application was that the information displayed in the client should be in a vector format in order to produce high-quality maps. Moreover, the map display plug-in enables enhanced user interactions (e.g., object highlighting, tooltips) if a vector format is used for the map. However, the geographic information collected by the survey was very detailed, and the maps produced directly with this information were very large and were not suitable for web-based visualization due to long transmission times and complex rendering.
  In order to solve this problem, we implemented a module to perform automatic generalization of geographic information. This module uses different representations of the same geographic value at different detail levels, and a set of rules that determine which geographic value must be used for a given map scale. This enables to reduce the size of a map by simplifying geographic values at small map scales. Additionally, this module also enables a developer to easily define map layers, styles, and complete maps.
– *Activity module.* In addition to a definition of the map contents, a developer must also provide a definition of the actions to be performed in response to user interactions. For instance, it is necessary to define the action associated to a user mouse click on a geographic value. We developed a module to facilitate the definition, management and implementation of this functionality.
– *Presentation functionality.* Even though we used a *thick-client* in the client side of the application, we had to provide some client-side presentation functionality such as presentation of alphanumeric information for geographic values, or the management of the graphical legend of the map.

A screen capture of the web-based application is displayed in Figure 4. It shows the graphical legend management on the left, the tool bar on the top, the context information area on the bottom, and the map in the center of the image. A window with alphanumeric information of the selected element is shown overlaid.

6 Analysis of the WebEIEL System

As we already said, in order to reduce the development time, we used commercially-available tools instead of custom-developed software modules in the architecture of the *WebEIEL* application wherever it was possible. This caused the resulting system to be different from the architecture we propose in Section 4. These differences are common to almost any GIS application developed using a commercial GIS development tools. They can be categorized as follows:

– *Incorrect functionality separation.* In the architecture we propose, we have presented a sound division of the functionality into independent software tiers. Moreover, the functionality of these tiers can be further divided into independent software layers. However, in the architecture of the GIS described in Section 5, this principle does not

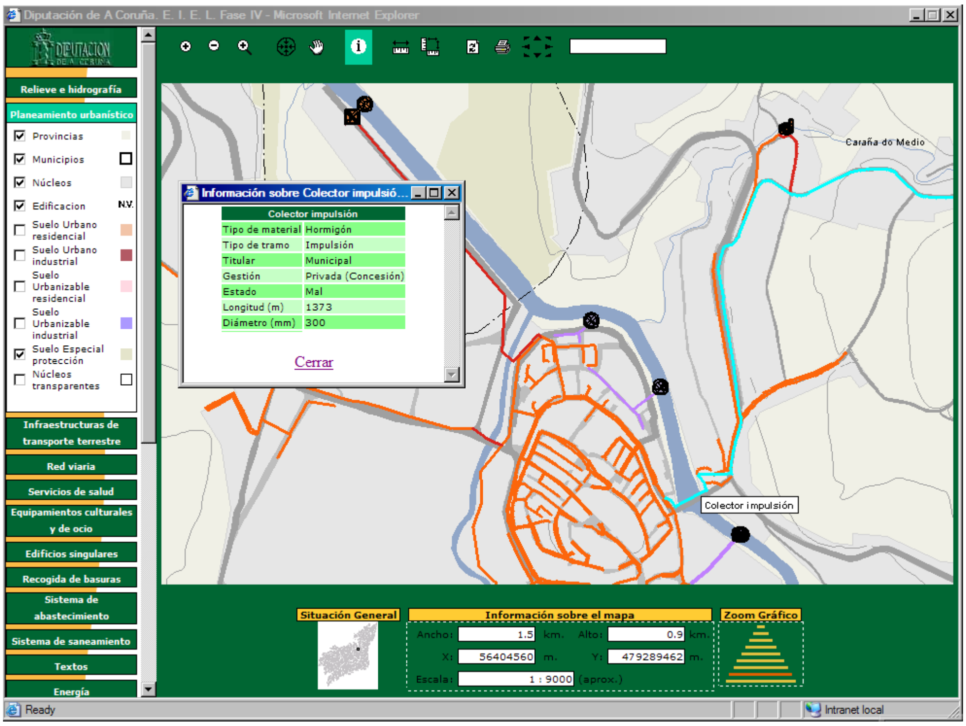

Fig. 4. User Interface of WebEIEL.

hold. Figure 3 shows the separation of the system architecture using thick dashed lines. It can be seen that Intergraph Geomedia Web Map provides in a monolithic software module functionality for the three tiers of the architecture. This is a common problem in many commercial GIS development tools that causes many problems regarding flexibility, scalability, and reusability.

– *Missing functionality.* Another common problem in GIS development tools is that some important functionality described in Section 4 is missing. In our case, the conceptual model for geographic information does not support multiple geographic values in a table.

  Furthermore, the conceptual data model was not implemented using an extensible DBMS. Instead, a relational DBMS had to be used and the geographic values were stored using BLOBs, which caused many efficiency problems. For instance, the computation of a spatial join requires to load both relations in main memory, and perform a selection using main-memory join algorithms.

  Finally, the conceptual data model does not support the declaration of explicit topological relationships. This leads to errors in the geographic information such as error as *gaps* between geographic values that share a border.

– *Technology Independence.* GIS development tools often impose a proprietary data storage format, and proprietary interfaces to the data tier. Sometimes, a proprietary data presentation format is also imposed on the system. The result is that the application is too heavily integrated with the technology and cannot be ported to other software platforms.

7 Conclusions and Future Work

Geographic information is slowly becoming an important element in computer systems. Many applications are being developed for industrial, administrative and research tasks in which geographic information is the central component. Furthermore, geographic information is providing added-value to many applications that did not consider it before (e.g., location-based services).

However, geographic information is a special kind of information that cannot be represented, manipulated and visualized using the methods that were traditionally used for other business and scientific information. Geographic information requires special modeling and analysis methods. The first contribution of our work is an analsysis of the special characteristics that make geographic information system different from traditional information systems.

Furthermore, here does not exist a generic architecture for GIS that takes into account the special nature and characteristics of geographic information and at the same time the well-known requirements for general-purpose information systems. The OpenGIS Consortium and the ISO are working on specifications of standards concerning geographic information. They are laying the foundation for a new generation of GIS applications and development tools that will be able to cooperate to a greater extent at many different levels. The main contribution of our work is a proposal for a generic architecture whose design is based on the analysis of the special characteristics of geographic information with respect to traditional information. This architecture is based on the ISO/TC 211 and OGC proposals, and conforms with their specifications where possible.

The solutions provided by commercial tools do not completely satisfy the requirements derived from the special characteristics of geographic information and the requirements of general-purpose information systems. We have presented the complete development process of a complex GIS for the Provincial Council of A Coruña, and we have compared and analyzed the differences between the architecture we propose and the completed GIS. Commercial systems cause serious efficiency problems and lack of functionality in some cases. Nevertheless, new commercial tools such as Oracle Spatial Option or ESRI Spatial Data Extender are moving towards the right direction by implementing international standards for representing geographic information, and providing a well-structured solution for the functionality they provide.

We have shown in the architecture that there is much functionality that is independent of the particular application of the GIS. This functionality can be implemented in a generic manner, and adapted later on with application-specific details given using high-level languages. However, current commercial GIS development tools do not provide this functionality, and therefore it must be implemented *ad hoc* using programming languages.

The next steps suggested by our work are:

– *Migrate the EIEL GIS to Oracle Spatial Option.* Oracle Spatial Option provides a correct implementation of the functionality proposed for our architecture in the sense that the representation and manipulation of geographic information is performed within the DBMS. In order to analyze the improvement in performance resulting from this feature, we are currently migrating the EIEL GIS to Oracle Spatial Option.
– *Implement the generic modules of the architecture.* Even though we have already implemented some of the generic modules described in the architecture for the project described in Section 5, this implementation is not generic because it uses Intergraph Geomedia Web Map. We plan on implementing the architecture using standards for the interfaces of the system and building modules independent of the supporting technology.
– *Develop tools to create geographic information systems based on the architecture.* Once the generic modules of the architecture are implemented, it is possible to build specific GIS applications by integrating the appropriate modules. It will be possible and desirable to define high-level languages and visual tools that allow a developer to easily select and integrate the modules for a specific GIS.

References

1. Laurini, R., Thompson, D.: Fundamentals of Spatial Informations Systems. The APIC Series Num. 37 - Academic Press (1992) ISBN: 0-12-438380-7.
2. Brisaboa, N.R., Cotelo Lema, J.A., Luaces, M.R., Viqueira, J.R.: State of the Art and Requirements in GIS. In: Proc. of the 3rd Mexican International Conference on Computer Science (ENC), Aguascalientes, Mexico (2001)
3. Burrough, P., McDonnell, R.: Principles of Geographical Information Systems. Oxford University Press (1998) ISBN: 0-19-823365-5.
4. Longley, P., Goodchild, M., Maguire, D., Rhind, D.: Geographic Information Systems and Science. John Wiley & Sons (2001) ISBN: 0-471-49521-2.
5. Longley, P., Goodchild, M., Maguire, D., Rhind, D.: Geographical Information Systems: Principles, Techniques, Management and Applications. John Wiley & Sons (1999) ISBN: 0-471-32182-6.
6. Worboys, M.F.: GIS: A Computing Perspective. Taylor & Francis (1995) ISBN: 0-7484-0065-6.
7. Koubarakis, M., Sellis, T.K., Frank, A.U., Grumbach, S., Güting, R.H., Jensen, C.S., Lorentzos, N.A., Manolopoulos, Y., Nardelli, E., Pernici, B., Schek, H., Scholl, M., Theodoulidis, B., N.Tryfona, eds.: Spatio-Temporal Databases: The CHOROCHRONOS Approach. Volume 2520 of Lecture Notes in Computer Science. Springer (2003)
8. Open GIS Consortium, Inc.: OpenGIS Reference Model. OpenGIS Project Document 03-040, Open GIS Consortium, Inc. (2003)
9. ISO/TC 211: Geographic Information - Reference Model. International Standard 19101, ISO/IEC (2002)
10. Open GIS Consortium, Inc.: OpenGIS Geography Markup Language (GML) Implementation Specification, Version 3.00. OpenGIS Project Document 02-023r4, Open GIS Consortium, Inc. (2003)
11. Open GIS Consortium, Inc.: OpenGIS Styled Layer Descriptor Implementation Specification. OpenGIS Project Document 02-070, Open GIS Consortium, Inc. (2002)
12. Open GIS Consortium, Inc.: OpenGIS Web Feature Service Implementation Specification. OpenGIS Project Document 02-058, Open GIS Consortium, Inc. (2002)
13. ISO/TC 211: Geographic Information - Simple Feature Access - Part 2: SQL Option. Draft International standard 19125-2, ISO/IEC (2000)
14. Open GIS Consortium, Inc.: OpenGIS Simple Features Specification For OLE/COM. Revision 1.1. OpenGIS Project Document 99-050, Open GIS Consortium, Inc. (1999)
15. Brisaboa, N.R., Cotelo Lema, J.A., Fariña Martínez, A., Luaces, M.R., Viqueira, J.R.: The E.I.E.L. Project: An Experience of GIS Development. In: Proc. of the 9th EC-GI & GIS Workshop (ECGIS), A Coruña, Spain (2003)