



# International Journal of Engineering, Business and Enterprise Applications (IJEBA)

[www.iasir.net](http://www.iasir.net)

## Architectural Structure of Geographic Information System

Anoop Singh<sup>1</sup>, Ramanjyot Kaur<sup>2</sup>

<sup>1</sup>Department of Civil Engineering, SBBSIET, Jalandhar, Punjab, INDIA.

<sup>2</sup>Department of Computer Science & Engineering, SBBSIET, Jalandhar, Punjab, INDIA.  
anoophayer@gmail.com<sup>1</sup>, er.ramanjyot@gmail.com<sup>2</sup>

**Abstract:** *Geographic information systems (GIS) have been an active research field for decades now, and they are becoming a very active commercial field due to the increasing demand for interactive manipulation and analysis of geographic information and the exponential improvement in the performance of computer-based technologies. In this paper the generic architecture of Geographic information system is discussed. It is considered that the functionality of a generic architecture for information systems must be divided into three separate tiers, namely: presentation tier, application logic tier, and data tier. This architecture is suitable for GIS applications, the special nature and exclusive characteristics of geographic information impose special requirements on the architecture in terms of conceptual and logical models, data structures, access methods, analysis techniques, and visualization procedures. Thus, in this paper the requirements for GIS and the internal architecture of the GIS have been discussed.*

**Keywords:** *Spatial data, mediator layer, geospatial, metadata, scalable*

### I. Introduction

A geographic information system (GIS) is a computer system designed to capture, store, manipulate, analyze, manage, and present all types of spatial or geographical data. The acronym GIS is sometimes used for geographical information science or geospatial information studies to refer to the academic discipline or career of working with geographic information systems and is a large domain within the broader academic discipline of Geoinformatics[1].

The term GIS describes any information system that integrates, stores, edits, analyzes, shares, and displays geographic information. GIS applications are tools that allow users to create interactive queries (user-created searches), analyze spatial information, edit data in maps, and present the results of all these operations[2][3]. Geographic information science is the science underlying geographic concepts, applications, and systems[4].

GIS is a broad term that can refer to a number of different technologies, processes, and methods. It is attached to many operations and has many applications related to engineering, planning, management, transport/logistics, insurance, telecommunications, and business.[3] For that reason, GIS and location intelligence applications can be the foundation for many location-enabled services that rely on analysis and visualization.

GIS can relate unrelated information by using location as the key index variable. Locations or extents in the Earth space-time may be recorded as dates/times of occurrence, and x, y, and z coordinates representing, longitude, latitude, and elevation, respectively. All Earth-based spatial-temporal location and extent references should, ideally, be relatable to one another and ultimately to a "real" physical location or extent. This key characteristic of GIS has begun to open new avenues of scientific inquiry.

Geographic Information Systems (GIS), a promising branch of Information Systems (IS), have achieved considerable success in recent years. This area of IS has concentrated on the construction of computer based information systems that enable capture, modeling, storage, retrieval, sharing, manipulation, analysis, and presentation of geographically referenced data [5].

### II. Requirements for Geographic Information System

A set of requirements for Geographic Information System have been produced after many years of research [6]. Some of the requirements are discussed in the following:

**A. Flexibility:** The functional requirements and the runtime environment of an information system often change during its lifetime. It is important that the system can adapt easily to these changes. It's also desirable that the information system can be used in different technological platforms with different functionality (e.g., personal computers, mobile devices). Hence, the architecture and the functionality of the information system must be easily adaptable to different platforms.

**B. Extensibility:** In addition to changes to the functionality or runtime environment, during the lifetime of an information system it is often necessary to support and incorporate new requirements and technological advances. The information system must provide a highly customizable framework that can be extended with new features by mean of a programming language or additional software components.

**C. Reusability:** The development process of an information system is costly. Therefore, it is important that the components of the information system can be used again without significant modification as a building block in a different information system from the one that it was originally designed for. This requires the development of generic modules that can be configured for specific tasks by means of high-level languages.

**D. Scalability:** Even though it is possible to estimate the number of users that will use the information system or the computing power needed by the functionality in the information system, these estimations are likely to change during the lifetime of the system. Hence, the system must be designed in a way that it is possible to increase the number of users or expand the capabilities of a computing solution without making major changes to the system.

**E. Reliability and security:** Information systems need to be fault-tolerant and highly-available. Moreover, it has been repeatedly proven that security is a very important requirement for computer-based information systems.

**F. A conceptual model for geographic information:** It is necessary to define a conceptual model that supports the description of geographic information. The model must include abstractions to model real-world geographic phenomena, using both the object-based and the field-based views of space.

**G. Primitive operations on the abstractions of the conceptual model:** An exhaustive set of primitive operations on the data abstractions of the conceptual model must be defined in order to support data querying.

**H. A query language for the conceptual model:** This language is used to retrieve and manipulate the data abstractions of an application schema defined using the conceptual model. The language must be composed by the operations defined previously. The results of queries must be represented by an appropriate language that captures all the abstractions of the conceptual model.

**I. Problem-solving techniques:** The abstractions in the conceptual model and the query language must be used to provide generic solutions for common geographic problems. There must be a precise definition of the problems to be solved, the information needed, and the techniques used to solve them.

**J. Metadata and data cataloguing:** In order to define generic methods to find out and use the information contained in any given application schema, it is necessary to provide models for metadata and data cataloguing.

**K. Logical models for specific technological platforms:** Given that each technological platform has particular limitations for the representation of information, it is necessary to define a logical model for the platform that takes into account these limitations. This model defines data structures for the data abstractions and algorithms for the operations that can be implemented in the technological platform.

**L. Efficient physical models:** Even though the logical model takes into account the limitations of computer systems with respect to the representation of information, it is necessary to define new storage techniques and access methods in order to achieve computational efficiency.

**M. Visualization and interaction metaphors:** 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 grouped in cartographic layers.

**N. Abstractions for information visualization:** The data abstractions used for the storage and processing of geographic information are not suitable for its presentation to the user. Therefore, new data abstractions are needed that support displaying different views of a single geographic object at different resolutions or with different visual styles according to display parameters such as scale.

### III. Architecture Overview

#### A. Description of the Architecture

This section discusses the generic architecture for geographic information systems based on the proposal by the ISO and the OGC[7][8]. The specifications defined by these organizations are used where possible. Architectural Structure of Geographic Information System separates the functionality of the system in three independent tiers, namely the Data Tier, the Application Logic Tier and the Presentation Tier [9]. 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 architectural structure of GIS is shown in Figure 1.

**i) Data Tier:** It provides data management functionality independently from the software technology. Information retrieval and manipulation requests for the data tier are expressed using a query language. Queries are evaluated within the data tier and the result is a set of geographic features that are represented using an information exchange language that is suitable for the conceptual model.

**ii) Application Logic Tier:** It implements the problem-solving and the application specific functionality of the system. The top-most interface of this tier consists of a collection of operations for data processing tasks. These operations represent high-level abstractions of problem-solving techniques (e.g., find a route between two nodes in a network) instead of the primitive operations from the data tier query language (e.g., find whether there is an direct edge between those two nodes). A processing request to this tier is expressed as an invocation to one of

these operations, which is then executed by the application logic tier either by using its internal information or by building and issuing the appropriate queries to the data tier and manipulating the returned data [10].

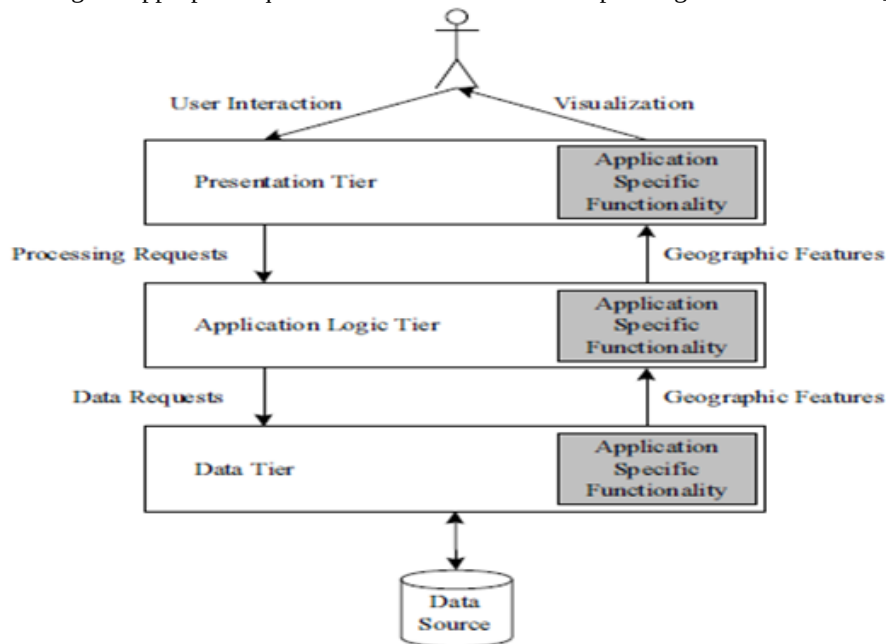


**Figure 1: Layered architecture of GIS**

**iii) Presentation Tier:** It implements the user interface of the system, which enables data visualization, data manipulation and data entry. The presentation tier receives the user interaction in the form of mouse gestures, keyboard inputs or inputs from other devices. 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 [11].

#### **B. Internal Architecture of the Tiers**

Each of these three tiers is not a monolithic software module. Instead, each tier is composed by a set of software layers that divide the functionality of the tier into independent components. Figure 2 shows the internal layers of the tiers in the architecture.

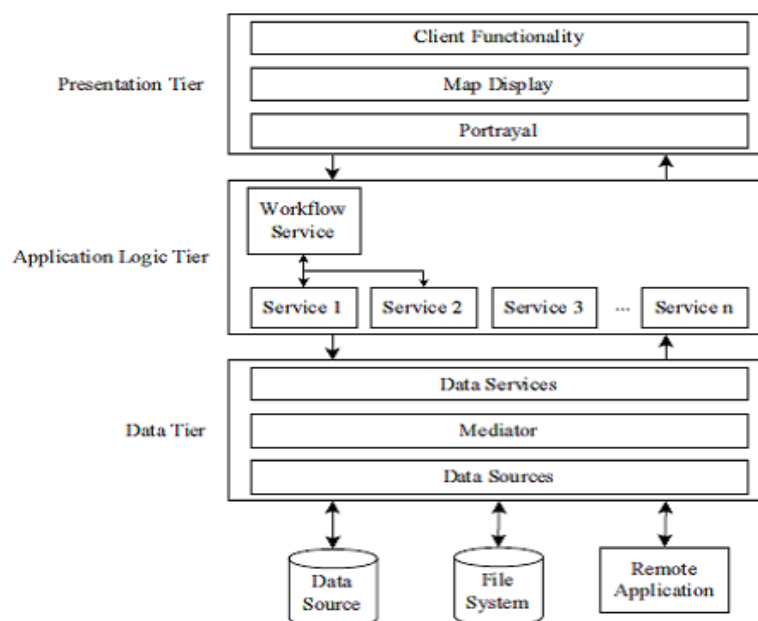


**Figure 2: Internal architecture of Tiers**

**i) Data Tier:** This tier must be internally divided into three independent layers: the data services layer, the mediator layer, and the data sources layer. 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 must offer a single conceptual model geographic information including data types, a query language, metadata and catalogue information. There is a wrapper in the data sources layer for each data source type that deals with the peculiarities of the data source. The data services layer provides a collection of profiles of the conceptual model for specific applications that enable to hide the complexity of the underlying conceptual model (e.g., by providing a simplified query language or a specific exchange language).

**ii) Application Logic Tier:** This tier is not divided into layers like the data tier. Instead, the functionality of the tier must be implemented by a set of multiple independent services. Each service performs a well-defined and simple task, and it is defined by giving its interface as a set of operations and a description of the results. The services existing in the architecture cannot be predefined because the functionality necessary for specific GIS applications cannot be known in advance.

**iii) Presentation Tier:** This tier must be divided into three layers: the client functionality layer, the map display layer and the portrayal layer [12].

**a) Portrayal layer:** This layer 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 styles definitions, which must define precisely the way in which each geographic feature must be rendered.

**b) Map Display Layer:** It is responsible for rendering the cartographic objects on the display device and enabling the end-user to manipulate and interact with the cartographic objects to perform geographic operations and to request processing operations from the application logic tier. The definition of the actions associated to each of the user-interface events is represented as a collection of activity rules.

**c) Client Functionality Layer:** This layer allows for the implementation of the user interface functionality. Additionally, some basic geographic functionality can be implemented in this layer using the cartographic objects for the computations (e.g., client-side map zooms, or measurements). This avoids long processing times involving server queries for simple operations.

#### IV. Characteristics of the Architecture

The architecture of GIS is conceptual in the sense that it can be put into practice using many different implementations [13]. As an example, Figure 3 shows two different implementations of this architecture. Figure 3(a) shows traditional client/server architecture. The client contains the presentation tier whereas the server contains both the application logic tier and the data tier. Figure 3(b) shows an example of a highly-distributed architecture. The presentation tier is implemented by a client computer, and the data tier is implemented by a server computer. However, the services of the application logic tier are distributed among multiple computers [16]. The architecture is flexible to changes in the functional requirements because of the separation of data management functionality from the application logic, and the application logic functionality from the presentation logic. Moreover, using independent services oriented to simple and specific tasks in the application logic tier causes that changes in functional requirements do not require complex software changes. The architecture can be adapted easily to multiple runtime environments because it is highly-modular, as it was shown in figure 3.

The separation of the application functionality into multiple tiers and modules within each tier makes it easier to implement reusable components. Moreover, new functionality can be easily added to the system using new modules for all the tiers, and thus, the architecture provides for high extensibility) [14]. Regarding scalability, the high modularity of the three-tier architecture allows distribution of application components across multiple servers thus making the system much more scalable. Finally, this architecture makes it easier to increase reliability by implementing redundancy at each tier.

It is very difficult to decide which functionality belongs to the application logic tier and which belongs to the data tier. DBMS vendors include more functionality in their products with each release, and tasks that were performed by services in the application logic tier are now carried out by the data tier. However, in order to provide a clear separation of functionality in the architecture, we have determined that the functionality in the data tier must consist of primitive operations oriented to solve multiple problems, whereas the functionality in the application logic tier must be oriented to solve complex and specific problems. For instance, operations to perform basic point-set operations on geographic values, or predicates to check topological relationships between geographic values belong to the data tier. On the other hand, a service that determines the optimal route given a network and two points in the network, or a service that solves allocation problems belong to the application logic tier. The separation of the functionality of the application logic tier in multiple independent services may affect the overall efficiency of the system. A monolithic system can be optimized to perform some tasks efficiently, but the resulting system is barely flexible [15]. On the other hand, a service-based architecture is highly flexible and performs efficiently each individual task, but the overall performance of a complex chaining of services is very difficult to optimize because of the inherent independence of the services. However, we believe that the benefits of service-based architecture outweigh its drawbacks, and therefore, we have chosen this type of architecture.

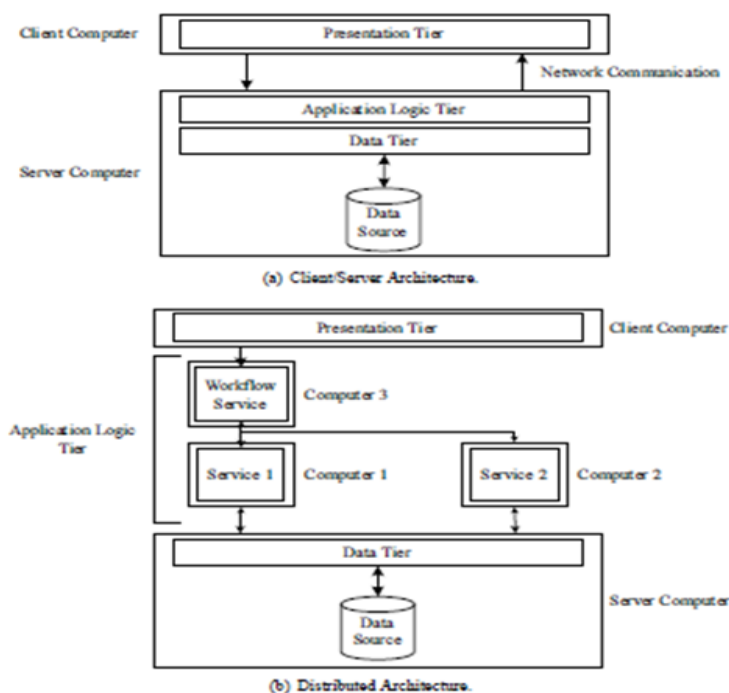


**Figure 3: Two implementations of the architecture**

## V. Conclusion

The architectural design discussed in this paper is inspired by the OGC and the ISO/TC 211. This implies that the three tiers of the architecture are similar in purpose to those defined by the OGC, including the organization of the tasks and the strict separation of the modules that interact only using well-defined interfaces. The different tiers of the architecture have been discussed in the paper. A mediator wrapper architecture for the data tier, and a data services layer to define specific application-profiles of the mediator layer have been considered.

## References

- [1] C. Amil, N.R. Brisaboa, A. Fariña Martínez, M.R. Luaces, M.R. Penabad, A.S. Places, and J.R. Viqueira. Una Interfaz Web para un Sistema Geográfico de Información Turística. In *Actas de la II Jornada de Sistemas de Información Geográfica (JSIG)*, El Escorial, Spain, 2002.
- [2] C. Amil, N.R. Brisaboa, A. Fariña Martínez, M.R. Luaces, M.R. Penabad, A.S. Places, and J.R. Viqueira. Using Geographical Information Systems to Browse Touristic Information. *Information Technology & Tourism*, 6(1), 2003.
- [3] D.W. Adler. IBM DB2 Spatial Extender - Spatial data within the RDBMS. In *Proc. of the 27th International Conference on Very Large Data Bases (VLDB '01)*, pp. 687–692, Orlando, 2001.
- [3] F. Bancilhon. Object-Oriented Database Systems. In *Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pp. 152–162, Austin, Texas, 1988.
- [4] N.R. Brisaboa, J.A. Coteló Lema, A. Fariña Martínez, M.R. Luaces, and J.R. Viqueira. E.I.E.L.: Una Experiencia de un Desarrollo SIG. In *Actas de la I Jornada de Sistemas de Información Geográfica (JSIG)*, Almagro, Spain, 2001.
- [5] N.R. Brisaboa, J.A. Coteló Lema, A. Fariña Martínez, M.R. Luaces, and J.R. Viqueira. S.I.T.P.A.C.: A Territorial Information System for A Coruña Province. In *Proc. of the 8th International Congress on Computer Science Research (CIIIC)*, Colima, Mexico, 2001.
- [6] N.R. Brisaboa, J.A. Coteló Lema, A. Fariña Martínez, M.R. Luaces, and J.R. Viqueira. 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.
- [7] N.R. Brisaboa, J.A. Coteló Lema, M.R. Luaces, and J.R. Viqueira. State of the Art and Requirements in GIS. In *Proc. of the 3rd Mexican International Conference on Computer Science (ENC)*, Aguascalientes, Mexico, 2001.
- [8] N.R. Brisaboa, J.A. Coteló Lema, M.R. Luaces, and J.R. Viqueira. S.I.T.P.A.C.: A Territorial Information System for A Coruña Province. *I+D Computación*, 1(2), 2002.
- [9] N.R. Brisaboa, J.A. Coteló Lema, M.R. Luaces, and J.R. Viqueira. Sistemas de Información Geográfica: Revisión de su Estado Actual. In N.R. Brisaboa, ed., *Ingeniería del Software en la Década del 2000*, pp.77–94. Tórculo, A Coruña, Spain, 2003.
- [10] P. Burrough and R. McDonnell. *Principles of Geographical Information Systems*. Oxford University Press, 1998. ISBN: 0-19-823365-5.
- [11] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison-Wesley, Reading, 1 Edition, 1998.
- [12] J.A. Coteló Lema, L. Forlizzi, R.H. Güting, E. Nardelli, and M. Schneider. Algorithms for Moving Objects Databases. *The Computer Journal*, 46(6):680–712, 2003.
- [13] J.A. Coteló Lema and R.H. Güting. Dual Grid: A New Approach for Robust Spatial Algebra Implementation. *GeoInformatica*, 6(1):57–76, 2002.
- [14] P.P.S.S. Chen. The Entity-Relationship Model: Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.
- [15] N.R. Chrisman. Topological Information Systems for Geographic Representation. In *Proc. of the Second International Symposium on Computer-Assisted Cartography (Auto-Carto 2)*, pp. 346–351, Falls Church, 1975.
- [16] N.R. Chrisman. Concepts Of Space as a Guide to Cartographic Data Structures. In *Proc. of the First International Advanced Study Symposium on Topological Data Structures for Geographic Information Systems*, pp. 1–19, Cambridge, Massachusetts, 1978.