A Service-oriented Architecture for GIS Applications

Atif Naseer, Hossam I. Aldoobi and Basem Y. Alkazemi
GIS Technology Innovation Center, Umm Al-Qura University, Makkah, Saudi Arabia

Keywords: Geographical Information System, Service Oriented Architecture (SOA), Software Architecture.

Abstract: Many Geographic Information Systems (GIS) are currently built according to the N-Tier architectural style. Despite the advantages of this style in terms of simplicity for the GIS business domain, it lacks many capabilities related to several key characteristics that might affect reusability and extensibility of GIS systems. Service-oriented architecture (SOA), on the other hand, can better serve the continual changing nature of GIS business requirements by facilitating faster and more reliable implementation of business requirements. This paper analyzes the current state of one of the commonly used GIS applications, namely GRASS, to define a potential road map for re-factoring its architecture to match our proposed SOA style in this paper.

1 INTRODUCTION

Geographic Information Systems (GIS) are widely used by many users nowadays. With the rapid development of technology, GIS systems are now used as a daily tool by many people. However, most people face a lot of difficulties in designing and implementing GIS systems. Some of these difficulties relate to GIS data, GIS services, GIS system development and interfacing of GIS systems with other applications. GIS introduces some methods and services to visualize, analyze and manipulate GIS data. Different organizations develop their own data model and structure.

Nowadays many GIS systems are used to achieve the goals of their users. Most of the systems follow N-Tier architecture for GIS application development. There are several advantages to N-Tier architecture, the main advantage being that all the business logic is defined once in a business, and the presentation layer uses that logic to represent information. Any changes in one layer can be easily available in all the applications. Despite the advantages of this style in terms of simplicity for the GIS business domain, it lacks many capabilities related to several key characteristics that might affect reusability and extensibility of GIS systems. Also it is more difficult to set up and maintain the physical layers as it results in high communications overheads as well.

SOA is a software engineering methodology for constructing a distributed software system. The SOA-based model can better serve the continual changing nature of GIS business requirements. In this paper we analyze the current state of one of the commonly used open source GIS software programs, GRASS. GRASS is one of the open source tools used for geospatial data management and analysis, maps construction and processing, modelling and visualization of spatial data (GRASS GIS, 2015). GRASS is used to develop desktop GIS applications. Most of the vendors describe their GIS system from a business perspective. They do not describe the architecture of the system, but only show the functional aspects of the systems to the users.

Here we discuss an SOA-based model and map the services of GRASS onto this model. We will identify the services which are currently not present in GRASS and try to identify those services and fit them into appropriate layers of proposed architecture.

The paper is structured as follows. In section 2, we discuss GIS application architecture in brief. In section 3, we discuss the Service Oriented Architecture of software engineering. Section 4 highlights analysis of GIS application architecture. In section 5 we map the GRASS on Proposed SOA-based Framework. We discuss future research directions and conclude the paper in section 6.

Naseer A., Aldoobi H. and Alkazemi B..
A Service-oriented Architecture for GIS Applications.
DOI: 10.5220/0005556501510155
In *Proceedings of the 10th International Conference on Software Paradigm Trends* (ICSOFT-PT-2015), pages 151-155
ISBN: 978-989-758-115-1

2 GEOGRAPHIC INFORMATION SYSTEM APPLICATION ARCHITECURE

A Geographic Information System (GIS) is a computer system that has an N-Tier architecture like most other computer information systems (Jankovic and Milidragovic, 2013). Most of the systems used for GIS application development follow the N-Tier architecture. This architecture is mainly composed of three layers, namely *Human Interaction Layer, Processing layer and Information Management Layer*, as shown in Figure 1.

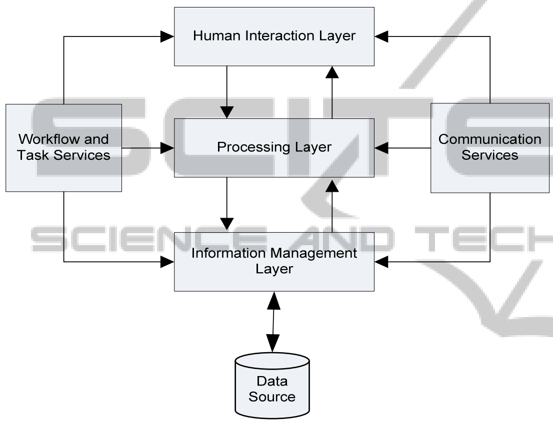

Figure 1: Typical N-Tier GIS Architecture.

The main advantage of this n-tier GIS architecture is that it separates the system functionality into three different layers, and these functions interact only through interfaces. The developer can modify the functionality of any layer with little impact on others. So this architecture provides flexibility, maintainability, reusability, scalability and increased performance. (Luaces, Brisaboa, Paramá and Viqueirs 2005) presents some special functional requirements for GIS applications. For example:

- The datatypes required to represent geographic information are different from ordinary datatypes called Special data types.
- Geographic information have different analysis and visualization procedures.
- Geographic information is typically large in terms of its hierarchical structure.
- Geographic information database transactions are much more complex than a transaction of standard relational database.
- There are two diverse conceptual views of geographic space i.e. field-based model and object-based model.

Some of the examples of GIS systems that follow N-Tier architecture are as follows:

- MapWindow GIS is an open source GIS application and set of programmable mapping components. These components can be reprogrammed to perform different tasks. The MapWindow GIS is built upon Microsoft .NET technology (MapWindow GIS, 2014). It provides a human interaction layer to build user interfaces, it has some processing functions that can process the data and it also has a data storage mechanism. So the MapWindow application includes standard GIS data visualization features, attribute table editing, shape file editing and data convertors (MapWindow GIS Desktop Overview, 2013).
- ArcGIS is used for creating and sharing web maps, and web mapping apps, searching for GIS content in organizations, creating groups, and sharing links to GIS apps, maps, and layer packages (ARCGIS 10.3.1 FOR SERVER, 2015). ArcGIS provides different types of services like real-time data processing, querying, visualization and editing.
- SEXTANTE is a spatial data processing framework that is used by Quantum Geographic Information System (QGIS) for analysis. SEXTANTE contains many tools like graphical modeler, command line interface and batch processing interface. It allows the QGIS interface to run a large number of analysis algorithms. (The SEXTANTE framework, 2015).

3 SERVICE-ORIENTED ARCHITECTURE

Service-oriented architecture (SOA) is a new development in distributed computing. It allows software components called "services" to interact with each other to complete any task. In service-oriented application, each activity is implemented as a separate service (Kushwaha, Amundson, Koutsoukos and Neema, 2007). Many applications can be created from a composition of these services, and the services can also be shared among many applications.
Service-oriented architecture (SOA) has become a very popular architecture paradigm for designing and developing a distributed system. According to Bianco et al. (Phil, Rick and Paulo, 2012), there are many

definitions of a service in SOA but none are universally accepted. A service:

- Can be deployed independently.
- Is used in the network by name or locator instead of absolute network address.
- Implementation is encapsulated and can only interact through the interface.
- Can be discovered by user through a directory service.
- Is dynamically bound so a user can locate and bind the service at runtime.

In SOA, the services are interact among themselves through a standard language called XML. According to Papazoglou et al. (Papazoglou, 2003), this architectural approach allows multiple applications running on different technologies to communicate with each other. An SOA service is usually a business function implemented in any language that has a well-defined interface and is accessed and used by other components. Papazoglou et al. Papazoglou, 2003) state that this feature is inherited from the software engineering concept of modularity, like modules, objects and components. A service is a complete business function that is reused by many other applications.

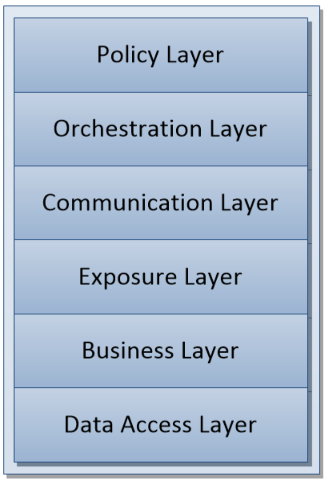

Figure 2: Architecture Layers of SOA-based Framework.

Here we discuss the SOA-based model proposed in a previous work by Alkazemi (Alkazemi, Baz and Grami, 2012). This framework functions as an integration facilitator for applications. It is composed of six layers. Figure 2 shows the proposed architecture.
The descriptions of these layers are as follows:
Data Access Layer: This layer provides the database connectivity service to the application. This layer hides the underlying database from the application.
Business Layer: This layer is responsible for executing the basic functionality that represents an organization's business needs. This layer provides the services related to the business layer of any application. It will be responsible for managing all the modules and encapsulating them accordingly so that every module of an application behaves as a standalone system.
Exposure Layer: This layer is responsible for exposing the available applications from the application layers to services (e.g. web services, com components). All applications are therefore decoupled from their underlying environment and made available through the request-response interaction mode.
Communication Layer: This layer provides the services related to communication pattern and routing protocol. It enables service discovery and interaction which helps applications in establishing communication. It also defines the policies according to different standards used in the application. For example, web services interact by exchanging SOAP messages over HTTP protocol. So this layer is responsible in establishing communication between services. This layer is also referred as Enterprise Service Bus (ESB) layer.
Orchestration Layer: This layer is responsible for defining the business process. With the help of this service system the sequence and dependencies between all the services required by the application are established. Figure 3 shows how the web services collaborate with each other.

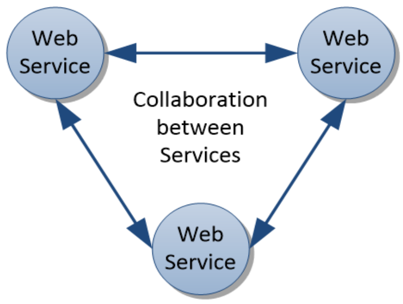

Figure 3: Collaboration between services.

Policy Layer: This layer is responsible for defining the user rights and privileges to access the application. Any application administration can define the privileges and policy of the application by using this service.

4 ANALYSIS OF GIS APPLICATION ARCHITECTURE

Most GIS applications are built using N-Tier architecture. There are several advantages and disadvantages of N-Tier architecture. We proposed an SOA-based framework in the previous section. There are several advantages of SOA-based systems. Some of them are:

- The SOA environment is open and interoperable, as the services are reusable and have standardized components.
- SOA creates an infrastructure for application development, so an application based on SOA provides users with only the functionality they need (Sahin and Gumusay, 2008).
- Data is not stored locally for any processing and is used from various different sources. For this reason the quality of output is increased.
- A system developed using the SOA approach can be adapted to changing requirements and technologies. Also these systems are easy to maintain and are more consistent (Aydin, 2007).
- The services in SOA-based architecture are loosely coupled, which means we can easily change or update any component without making changes to other components.
- The services in SOA-based architecture are encapsulated so the user can get the services without knowing the internal implementation of the service.

There are numerous characteristics of N-Tier and the SOA model that make them a successful model for an application. In this section we make a comparison between N-Tier and SOA according to the following characteristics: Language/Technology Dependency, Extendibility, Modifiability, Data Exchange, System Integration, Maintenance, Separation of Concerns, Coupling, and Encapsulation.

Table 1: Comparison between N-Tier and SOA model.

Characteristics	N-Tier	SOA
Language/Technology Dependency	Dependent	Independent
Extendibility	Hard to extend	Fully extendible
Modifiability	Major changes to source code	Supports plug and play effectively
Data Exchange	Internally	Internally/ Externally
Maintenance	Affects most of the system	Only one service is affected
Separation of Concerns	Homogeneous	Heterogeneous
Coupling	Tightly coupled	Loosely coupled
Encapsulation	Partially encapsulated	Fully encapsulated

We have generated a comparison matrix of N-Tier and the SOA-based model. Table I below presents this matrix.

5 MAPPING OF GRASS ON PROPOSED SOA-BASED FRAMEWORK

In the above section we define the SOA-based framework with different layers. Each layer provides a specific service which helps in developing any GIS application. In this section we discuss how an application can be built in GRASS GIS using the above-mentioned SOA-based framework. We try to map the services provided by GRASS GIS onto each layer of the proposed SOA-based model.

Data Access Layer: In GRASS GIS, the function db.connect allows the user to set database connection parameters. Values are stored in the mapset's VAR file (GRASS Database Connection, 2013). By adding this service to GRASS the user can add, access and retrieve data from any type of database. This layer provides the compatibility services between DBMS and GRASS.

Business Layer: GRASS GIS supports lots of business applications. It can provide a service that serves different types of businesses related to archeology, agricultural, geophysics, public health, remote sensing etc. This layer helps in developing a component-based GIS using GRASS.

Exposure Layer: To wrap GRASS functionality as a service, one mandatory requirement is to invoke GIS functionality by a programmable interface (e.g. shell scripts or an interface for programming languages). GRASS has a function "—interface-description" for process description.

Communication Layer: GRASS uses the Open Geospatial Consortium (OGC) standards and protocols for data sharing and processing (Open Geospatial Consortium, 2015).

(Xueming, Shifeng and Steve, 2004) presented the inter-object and intra-object communication mechanism. When a client object requests a method on a server object, it goes to the Object Request Broker (ORB). The ORB then invokes the method on behalf of the client. The ORB takes care of locating the server object, establishing a connection, invoking the method, getting the result, and closing the communication session.

Orchestration Layer: GRASS is not an SOA-based model so it cannot establish the sequence and dependencies between all the services required by the application. There should be some wrapper that reads all the services of GRASS and establishes the sequence and dependencies of all the services.

Policy Layer: "Free software projects are gaining interest even in the proprietary GIS industry due to their stability and the transparent development process" (Open Source GIS, A GRASS GIS Approach, 2007). GRASS is a desktop-based application and provides access to users according to the policy developed by the system administrator. One of the main advantages to convert GRASS from desktop to web-based service is that GRASS will identify the users from their IP addresses and display the relevant graphs and information for that user.

6 CONCLUSIONS AND FUTURE WORK

In this paper we highlight the fact that most systems follow N-Tier architecture for GIS application development. Despite the many advantages of N-Tier architecture, there are lot of shortcomings. We introduce the SOA-based framework for GIS application development and discuss how an application can be built in GRASS GIS using the proposed SOA-based framework. We try to map the services provided by GRASS GIS onto each layer of the proposed SOA-based model.

The next step in this project is to translate the design of the proposed architecture into a real working system and conduct a number of experimental works to generate results from the proposed system. Also we will develop some other applications from different domains using the proposed SOA architecture.

ACKNOWLEDGEMENTS

This work was funded by KACST GIS Technology Innovation Center at Umm Al-Qura University under grant "GISTIC-13-OP04".

REFERENCES

GRASS GIS 2015. Available form: <http://en.wikipedia. org/wiki/GRASS_GIS>. [9 April 2015]

Jankovic Dejan, Milidragovic Radmila, 2013 "Integration opensource GIS software for improving decision-making in local community", Acta Technica Corvininesis - Bulletin of Engineering. Vol. 6 Issue 4, p73-76.

Luaces M. R., Brisaboa N. R., Paramá J. R. and Viqueirs J. R. 2005, "A Generic Framework for GIS Applications. Web and Wireless Geographical Information Systems", 3428/2005:p94–109.

MapWindow GIS 2014. Available from: <http://en. wikipedia.org/wiki MapWindow_GIS> [31 Dec 2014].

MapWindow GIS Desktop Overview, 2013, Available fromhttp://www.mapwindow.org/apps/wiki/doku.php?id=mapwindow_4_overview [23 March 2013]

Arcgis 10.3.1 for Server, 2015, Available from: <http://www.esri.com/library/brochures/pdfs/arcgis-server-functionality-matrix.pdf>, [May 2015]

The SEXTANTE framework, 2015 Available from: http://www.sextantegis.com, [23 February 2015]

Kushwaha, M.; Amundson, I.; Koutsoukos, X.; Neema, S.; Sztipanovits, J, 2007: "OASiS: A Programming Framework for Service-Oriented Sensor Networks," COMSWARE 2007. 2nd International Conference on, vol., no., pp.18, 7-12 Jan 2007.

Phil Bianco, Rick Kotermansk, Paulo Merson, 2012 "Evaluating a Service- Oriented Architecture", Available from: http://www.sei.cmu.edu/reports/ 07tr015 .pdf [15 Dec 2012].

Papazoglou MP, 2003 "Service-oriented computing: concepts, characteristics and directions". In: Proceedings 4th international conference on web information systems engineering.

Alkazemi, Basem Y, Baz Abdullah, Grami Grami M, 2012, "Refactoring Legacy Software Systems into SOA Compatible Style to Support e-Business Development in Enterprise Organizations: The Case of Umm Al-Qura University's Systems, International Journal of e-Education, e-Business, e-Management and e-Learning" 2012, Vol.2(6): 465-472 ISSN: 2010-3654.

Sahin K and Gumusay M. U. 2008: "Service oriented architecture (SOA) based web services for geographic information systems", Paper presented at the XXIst ISPRS Congress, Available from: http://www.isprs.org/congresses/beijing2008/proceedings/2 pdf/5 WG-II-5/03.pdf, 2008.

Aydin G, 2007, "A Service Oriented Architecture for Geographic Information Systems Supporting Real Time Data Grids", Ph.D. Thesis, Indiana University, IN, USA, pp. 12-47.

GRASS Database Connection, 2013 Available from: <http://grass.osgeo.org/grass64/manuals/db.Connect .html> [14 Dec 2013]

Open Geospatial Consortium 2015, Available from: http://en.wikipedia.org/wiki/Open_Geospatial_ Consortium, [27 April 2015]

Xueming Wu, Shifeng Zhang and Steve Goddard, 2004, "Development of a Component-based GIS using GRASS", Proceedings of the FOSS/GRASS Users Conference (2004).

Open Source GIS, A GRASS GIS Approach, 2007, Available from http://www.academia.edu/5753114/ Open Source_ GIS_A. GRASS_GIS_ Aproach_3rd, [August 2007]