American Journal of Networks and Communications
2018; 7(3): 17-21
http://www.sciencepublishinggroup.com/j/ajnc
doi: 10.11648/j.ajnc.20180703.11
ISSN: 2326-893X (Print); ISSN: 2326-8964 (Online)

Service – Oriented Architecture: Web Technologies

Yadgarova Aziza Anvarovna, Artikbayev Sarvar Dilmuratovich

Department of Electronics and Radio Engineering, Tashkent University of Information Technologies Named After Muhammad al-Khwarizmi, Tashkent, Uzbekistan

Email address:
Sardorjabbarov.sj@gmail.com (A. S. Dilmuratovich)

To cite this article:
Yadgarova Aziza Anvarovna, Artikbayev Sarvar Dilmuratovich. Service – Oriented Architecture: Web Technologies. *American Journal of Networks and Communications*. Vol. 7, No. 3, 2018, pp. 17-21. doi: 10.11648/j.ajnc.20180703.11

Received: May 27, 2018; **Accepted**: June 26, 2018; **Published**: December 18, 2018

Abstract: Service-oriented architecture (SOA) is application architecture in which components or "services", having unified common interfaces, use joint rules (contracts) for definition of how they will access the services and how they will interact with each other. Nowadays this technology is becoming more and more widespread in many fields of IT industry due to the main advantage: capacity to offer effective approach to the solution of one of the most complicated and actual problems – problem of integration of the information resources. Joining the advantages of SOA with the capacities of Grid technology allows providing integration not only of local but of geographically remote information resources.

Keywords: Service-Oriented Architecture (SOA), Geo Information Systems (GIS), Database Management System (DBMS)

1. Introduction

Currently, geoinformation systems (GIS) are widely used to solve corporate users' problems, for example, in such areas as cadastral accounting, town planning, nature management and many others, providing users with spatial data management capabilities, spatial analysis, planning and forecasting problems. At the same time, GIS often act as an instrument for integrating heterogeneous information systems of an enterprise, so software (software) of geoinformation systems should be an integrated part of the corporate information environment [1-2].

Most of the geoinformation systems are implemented within the client-server architecture with a "thick" client: special user-intensive GIS software is installed on user desktops (desktop computers), which implements the user interface, including the tasks of rendering vector and raster spatial data (maps) and application logic, including data processing operations. This software architecture has a number of limitations: from the users' point of view, due to the requirements for workplaces, access to GIS is technologically and technically more difficult. From the developer's point of view, the processes of deployment (implementation) and maintenance (updating) of GIS software are becoming more complicated. It is also difficult to integrate GIS software with other enterprise information systems, which is an urgent task for modern corporate information systems. The desire to overcome these limitations led to the development of software architecture and technologies related to the deployment of application logic (software components) and all data on server computing systems, which is especially important for geoinformation systems due to the large amount of spatial data and resource-intensive processing operations. While it is widely used in geoinformation systems based on client-server architectures with a "thick" client, allowing the spatial data to be placed on the database management system (DBMS) servers the task of placing GIS applications on servers has been unresolved for a long time and is relevant today. The most promising software architecture within the framework of which an effective solution of this problem is possible (from the perspective of the complexity of implementation, maintenance, integration with other information systems, software of multiprocessor computer systems), is a service-oriented architecture (SOA). However, when implementing new types of architectures, corporate information systems software needs to preserve the functionality of existing software, so the task of converting components of existing GIS software when moving from a client – server architecture to a service-oriented with a "thin" client seems relevant. The article analyzes the features of the service-oriented software architecture of corporate geoinformation systems, discusses the possibilities of creating software for multiprocessor computer systems, and proposes a method for solving this task of transforming GIS software into a service-oriented architecture [5-6].

Let's analyze the differences between client-server and service-oriented architectures of GIS software, considering them in the context of the development of software architectures for corporate information systems [3, 5].

2. Main Part

The software of the first corporate information systems consisted of monolithic modules, including all software components (DBMS, application logic, user interaction logic) placed on mainframes. Access to such systems was carried out through "thin" clients, for example, text terminals. Because of the concentration of all software on the server (mainframe), this type of architecture is called a single-tier client by the server architecture (with a "thin" client).

Since the beginning of the 1990s, the two-tier client-server architecture (with a thick client) is beginning to dominate, where data and DBMS are hosted on the servers, and the remaining software components (application logic and graphical user interface) are hosted on desktop computers ("thick" customers). For many types of resource-intensive applications, for example, video, graphics, CAD systems, and geoinformation systems, this type of architecture remains common today. In some cases, the data server may be missing, but this does not change the client software architecture.

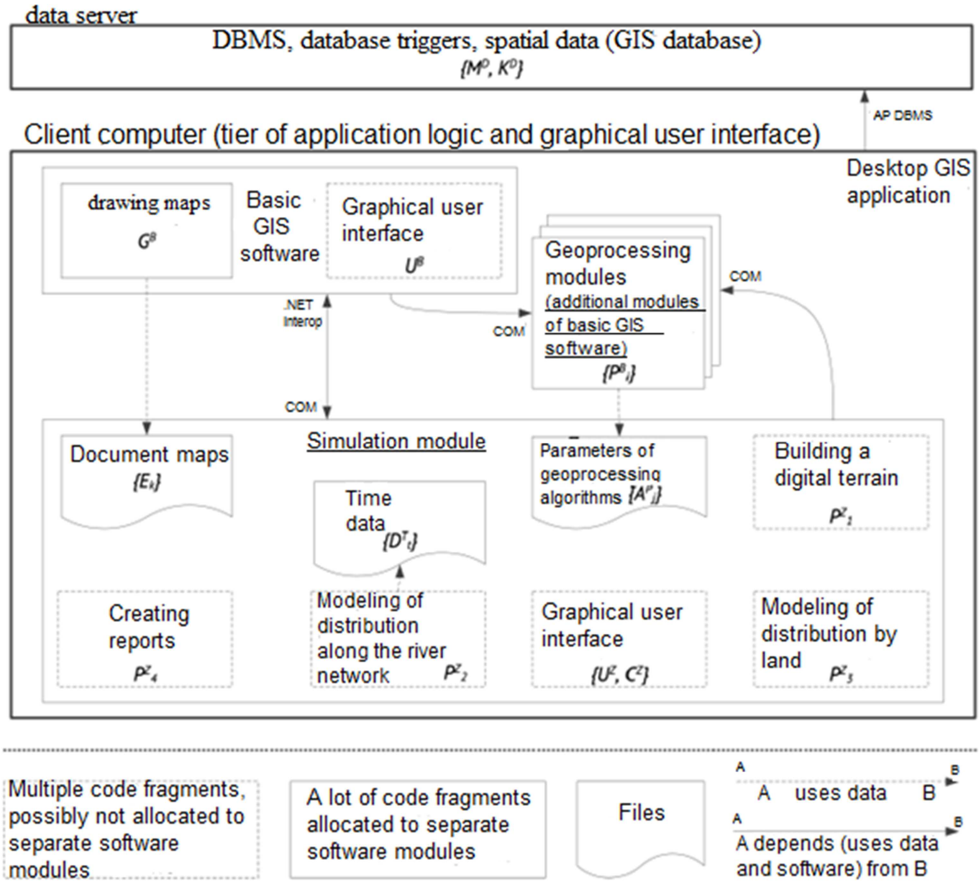

Figure 1. *Placement of components of GIS software for a system for modeling the distribution of oil by land and river network within two-tier client-server architecture.*

Let's consider the typical distribution of software components of geoinformation systems between computing nodes within the framework of a two-tier client-server architecture using the example of a system for modeling the distribution of oil and oil products by land and river network within the Geographic Information System of the Federal Agency for Water Resources (Figure The database server hosts the database management system software and spatial data bases. We denote the set of fragments of the source code of the server software implementing the data models as M_D, and the set of fragments of the source code of the server software implementing the security system as K_D (the superscript D will denote the sets of source code fragments placed on the data servers).

The client computers host the software of the geoinformation system that implements the application logic and the graphical user interface, represented, as a rule, by three components (physical modules): a module of basic GIS software, a geoprocessing module and a module of special GIS software.

The module of basic GIS software (for example, Arc-Map) solves common problems for various geoinformation systems, namely: from drawing card images (a lot of fragments of the software source code realizing this task are denoted as GB), graphical user interface (UB). Geoprocessing modules (for example, ArcToolbox) implement thematic processing of spatial data (a set of corresponding fragments of the source code of the software is denoted as $\{P_{Bi}\}$) as a set of basic operations (for example: intersection, union and other multi-oriented operations on spatial objects) and specific operations for example: interpolation of point values, construction of topological relations of objects of the spatial data base).

As an example of such a geoprocessing module of the modeling system, we can cite PZ3, a module for constructing a digital relief model (a raster of relief heights) based on vector data (isohyps, heights, river network). The third typical component of the GIS client software is special software modules that solve the subject-oriented tasks specific to a particular geoinformation system. In this example, this is an oil distribution simulation module consisting of five logical modules: report generation (P_{Z4}); modeling of oil distribution along the river network (P_{Z2}); modeling of oil distribution by land (P_{Z3}); building a digital terrain model (P_{Z1}); Graphical user interface, which includes the actual graphical interface (U_Z) and the entire application management module (C_Z). In addition to the software components, the modeling module contains three sets of data: parameters of geoprocessing algorithms $\{A_{Pj}\}$, time data $\{D_{Tt}\}$ and map documents $\{E_k\}$. The main disadvantages of using the two-tier client-server architecture in this example are as follows [7-10].

1. To use the system, the end user (client) needs a powerful computer with special software installed on it. To update this software, you need to perform the update procedure for the software modules for all clients, which results in a large amount of time.

2. Despite the fact that the task of modeling the distribution of oil by land and river network is in demand for many government departments and commercial organizations, its solution in the form of a system with a "thick" client does not allow providing access to the functionality of the system to remote users or other information systems.

These shortcomings are partially overcome in the framework of multi-tiered client-server architectures of traditional Internet applications that allow remote users to access GIS functionality without installing special software on client computers. In this case, thick client software is broken down into physical components, which are distributed among several architectural layers - tiers: a tier of data (data files and DBMS placed on data servers), a tier of applications (software components that implement application logic hosted on servers applications) and the tier of the graphical user interface (the corresponding software components are distributed between the servers and client computers or placed entirely on the client their computers). For the oil distribution modeling system, such arrangement of the components of "thick" customers is shown in Figure 2.

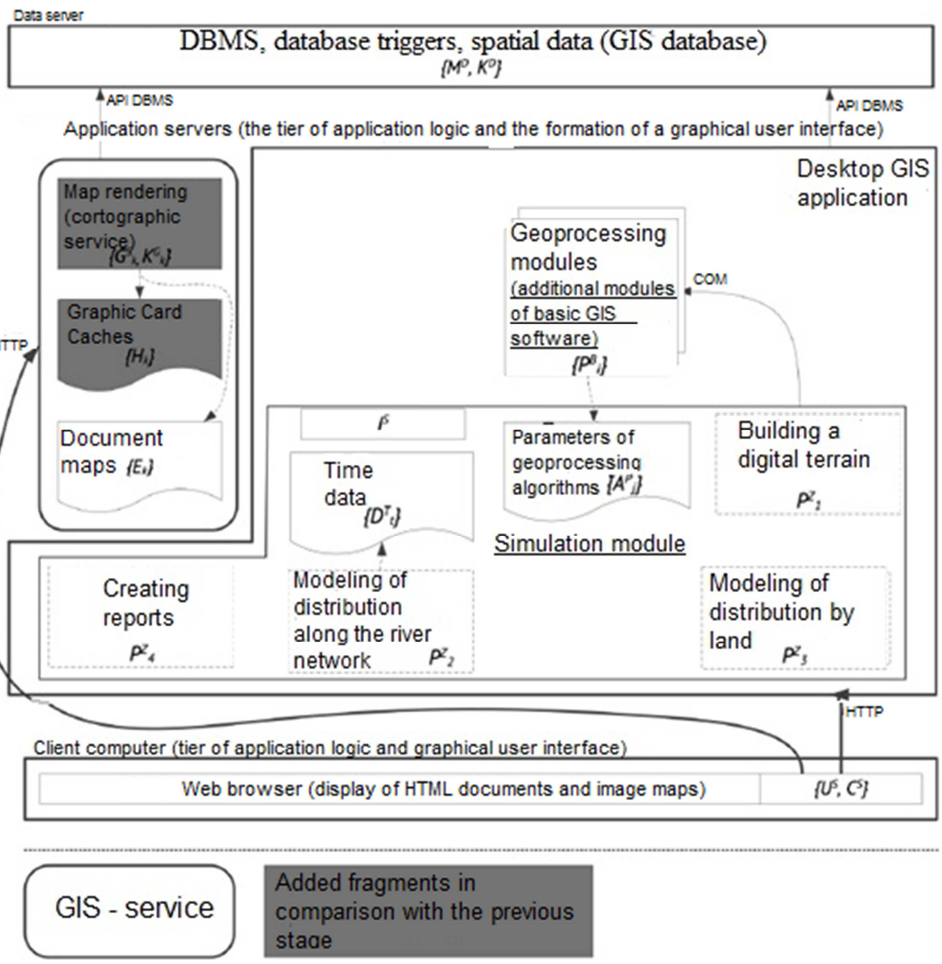

Figure 2. *GIS software components include deployment modeling on land parcels, multi-layer and customer-server architecture within the river network.*

However, in multi-tiered client-server architecture, there are no unified approaches to partitioning the application logic into components and unified methods for implementing component interaction, which greatly complicates system scaling, organization and management of parallel access of users and other information systems to individual GIS subsystems. For example, a server-based GIS application within a multi-tiered architecture (Figure 2), as a rule, is a monolithic software module, so the implementation of the interaction of external information systems with individual logical components of this application is laborious or impossible without splitting the server software into physical modules.

These limitations are overcome within the framework of service-oriented architecture. Unified cross-platform mechanisms of interaction of separate parts of software that are not present in the two-tier and multi-tier client-server architectures, and the decomposition of GIS software into a set of services - loosely coupled executable and ready-to-use modules - allow effectively solving the tasks of integrating GIS with other enterprise information systems by providing a cross-platform network access to the functionality of GIS services to end users and other information systems. In Figure. Figure 3 shows the location of software components for a system for modeling the distribution of oil by land and river network that meets the requirements of a service-oriented architecture: unlike client-server architectures (Figures 1, 2), GIS software components are grouped into a set of services that interact with each other and with others IP through cross-platform protocols (HTTP, SOAP and others). The following section describes the method for converting GIS software when moving from a client – server architecture with a thick client to a service-oriented architecture with a thin client, allowing you to save most of the source code that implements the basic calculation functions (application logic).

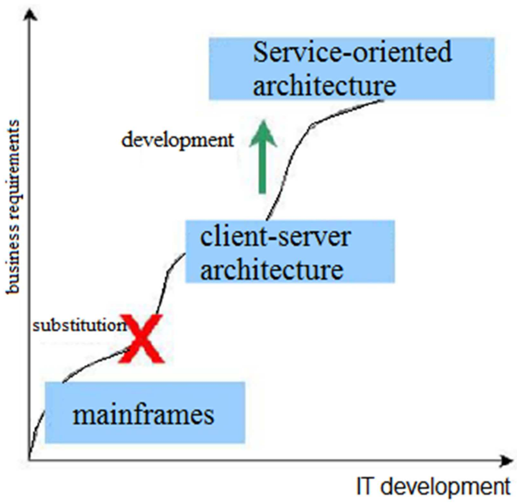

Figure 3. *Evolution of software architectures.*

A classic example of the first stage, mainframe, is the SAP R / 2 system. In the client-server architecture, SAP R / 3 became a popular example, and as a service-oriented solution, SAP is positioning its new integration platform, Net Weaver. "Now SAP produces not only and not so much ERP-system, but rather a specialized shell - a set of tools for fast and effective integration of all that already works in the enterprise," says Timur Aitov, Business Development Director of RBC SOFT. "Service-oriented the approach that underlies the Net Weaver platform makes it possible to dramatically simplify and accelerate the implementation of the corporate information system in any enterprise," he emphasizes.

Service-oriented architecture is based on the technology of Web-services, the ability to self-development of which provides the necessary adaptability of IT infrastructure of the enterprise. However, there is a rather large gap between the obvious potential of Web services and their real capabilities. It is in the basics of SOA on Web services that its main problem lies. The translation of business-critical applications of the corporation to use them within this concept requires the existence of unified standards for Web-services. Existing requirements are clearly not enough to solve complex, complex tasks, and work on the creation of better standards is still very far from committing and in this considerable role is played by the confrontation of the largest giants of the IT industry, such as Microsoft, with the rest of the world of software manufacturers.

In fairness, it's worth noting that after the announcement of Microsoft Visual Studio Team System 2005, as well as IBM's proposals for specific solutions based on SOA, this confrontation has become minimal, and we can expect progress in developing uniform standards governing the use of Web services. For the success of the concept of a service-oriented architecture, there must be a single interface language describing Web services. Recently, work is under way to improve this language - Web Services Description Language (WSDL).

Some of the existing preliminary specifications for Web services today are designed to guarantee the delivery of SOAP messages to the destination and support the stability of the state of long transactions. Nevertheless, according to many experts, it is still too early to talk about the widespread use of such "raw" specifications. The lack of unified standards for Web services does not allow the company to implement critical business processes within the SOA framework and largely hampers the development of this concept.

3. Conclusion

The article proposes the solution of the actual task of transforming the software modules of the existing two-tier client-server software of geoinformation systems with a "thick" client into a service – oriented GIS software with a "thin" client. The method of step-by-step, managed solution of this problem taking into account the specificity of GIS is proposed, which allows to perform the transformation of the components of the initial GIS software, saving most of the source code that implements the application logic, reducing the overall complexity of the system, taking advantage of the service-oriented architecture, placing the GIS software on server computer systems and ensuring access to it through "thin" customers. The proposed method is applied for transformation of modeling problems of natural and techno genic processes - oil spills over land and river network, construction of catchment basins, simulation of flooding zones in flood situations in the geo-information system.

References

[1] Distributed processing of spatial information on water resources in the geoinformation system of Rosvodresursov / Ivanov IG [and others] // Management of water-resource systems in extreme conditions: Sat. doc. Conf. M., 2008. P. 323.

[2] Pavlov SV, Shkundina RA, Usov TM Advantages of the service-oriented architecture of geoinformation systems on the ArcGIS platform (on the example of GIS Rosvodresursov) [Electronic resource] // Proceedings of XV conference of ESRI users in Russia and countries CIS. 1 electron. opt. disk (CD-ROM). 2009.

[3] Matsyachek LA Analysis and design of information systems. M: "Williams" Ltd., 2008. 816 p.

[4] Shekhar Sh., Chaula S. Fundamentals of spatial databases. Moscow: Kudits-Obraz, 2004. 336 p.

[5] Erl T. Service-oriented architecture. Concepts, Technology, and Design. Prentice Hall, 2005.

[6] "Evolution of the Grid Computing Architecture and the Grid Adaptation Model." J. Joseph, M. Ernest, C. Fellenstein. Grid-computing, IBM magazine, Volume 43, N 4, 2004, www.gridclub.ru.

[7] "Grid Perspectives: Grid Computing - The Next Generation of Distributed Computing. What is the difference between grid-computing and P2P, CORBA, cluster-computing and DCE. "Matt Hainos, www.gridclub.ru.

[8] "The Future of European Grids: Grids and Service-Oriented Knowledge Utilities. Vision and Research Areas for the period to 2010 and beyond. Third Report of the Next Generation Grid Expert Group ", www.gridclub.ru.

[9] "SOA Architecture as It Is", by Laurie McWitti http://www.ccc.ru/magazine/depot/06_02/read.html?0104.htm.

[10] "SOA: simple for the majority, difficult for the minority." Sergey Kuznetsov, Review of the November 2007 issue of the journal Computer (IEEE Computer Society, V. 40, No 11, November, 2007).

[11] Service-oriented architecture, http://www.iso.ru/journal/articles/374.html.

[12] "Middleware: a model of distributed systems services", http://citforum.gatchina.net/database/kbd97/3.shtml.

[13] "Convergence of virtualization, grid and SOA" Richard Wirth, "Open Systems", No. 3 2008.