

Application Development with Service Oriented Architecture

Emine Doğaç¹, Rıdvan Saraçoğlu^{2*}

Received : 08/07/2019 Accepted: 09/08/2019

Abstract: Today, it is one of the most invested areas of organizations to ensure the interoperability of existing corporate applications and newly developed or purchased applications. Developed systems are designed to be shaped according to the needs of the businesses that use it. Classical systems are automation systems that operate from point to point and from other platforms in the enterprise without interaction. Unlike these, the service-oriented architecture (SOA) contains a structure that is ready to interact. In this study, a system has been designed in order to reduce the integration costs between the applications in the long term by using SOA approach, to shorten the business processes in the institution and to reduce the economic costs. In this study, work pieces were published as a method of service over the service layer. Thus, the platform independent main application is accessible to each work piece. At the same time, these work pieces can come together and be reconfigured according to the request of the enterprise. Some of the advantages achieved by using SOA are: Providing improved work processes for services and actors, ensuring collaboration, utilizing from existing and reuse with savings, providing business flexibility by responding to changing business needs.

Keywords: Application Development, Service Development, Service Oriented Architecture

1. Introduction

Service Oriented Architecture (SOA) enables users to reduce integration costs between implementations in the long run, while shortening business processes in the organization and reducing economic costs. Most importantly, it prevents you from writing more codes and re-use of coding. Thanks to these features, which come from the working principle of virtualization, the information is delivered to the right users at the right time and speed. Thus, it also contributes to information reliability very well.

There is a need for an infrastructure requirement which is agile, nimble, changing and transforming, adapting to new demands in a short time and providing communication between heterogeneous structures. SOA provides benefits to the business unit, other than its software vision in the information technology infrastructure.

By reducing operation costs to a large extent, allowing the business to evaluate the savings it has made to improve the information system. The benefits of SOA in terms of the business unit are as follows; the process cost is low because the services are autonomously designed (Autonomy concept), and it is easy to integrate for it proposes industrially standardized principles and also it is simply adapted to the changes in business processes as its components are easy to maintain and replace. It is possible to introduce quickly new business functionality to the market since it is liable to be connected to new systems with less effort through its service interfaces conforming to industry standards and its ability of integration with partner companies [1].

For a long time, institutions and organizations have perceived Information Technology IT departments as more cost items into which business resources are allocated. Generally, the return on

investment was considered as positive. There had been significant benefits about accrual, collection, order, workflow, staff and automation of standard processes such as public relations, scaling and productivity. Stacks of paper and personnel have been replaced by high-capacity host servers and main frame that keep everything tidy and stable. While change is seen as a problem in a less or controllable environment, in organizations that want to adapt to the changing world and markets, large and accelerating changes have begun to force IT departments. Particularly difficult to manage, using different technologies and company products, disconnected departments and functions have begun to produce new data silos. It is difficult to access new data silos that are formed by IT departments for new projects, so it has to create rule engines, re-interface, re-analysis and re-data silos for each new job. While the amount of investment in IT departments has increased, recycling has started to decrease. IT departments have created a perception that creates more cost in organizations than value creation. With SOM, collecting software developed for different platforms used by corporations on a single platform will both facilitate the follow up of the software and reduce the cost. In order to meet the changing needs within the organizations and institutions and to have a real demand-based structure, the basic condition is the feature; flexibility and agility. These two vital features are the first answer in the IT world with the concept of SOM. In every automation to be created with SOM, which is a software architecture designed to solve demand-based solutions for unified enterprise problems, it can be ensured communication between the automation of heterogeneous institutions can be used dynamically reconfigurable, can be used alone or in combination with other services. In the automation that we will establish with SOM technology, each thread will be designed as methods of a service and can be integrated in other projects of institutions / enterprises. By converting small threads into services, a reusable structure will be obtained. Therefore, maintenance costs will be reduced to very

¹ Electrical and Electronics Engineering, Yüzüncü Yıl University, Van, TURKEY

ORCID ID : 0000-0001-7736-5399

² Electrical and Electronics Engineering, Yüzüncü Yıl University, Van, TURKEY

ORCID ID : 0000-0002-5853-4545

* Corresponding Author: Email: ridvansaracoglu@yyu.edu.tr

low levels. The database layer will be accessed only through the security protocols of the services and maximum security will be provided and the IT departments will be able to access the database, thus avoiding large and independent data silos. The applications will thus be flexible, service-oriented and agile.

2. Analysis of Field Literature

SOM is not really a new topic. When we look at the 1980s, this subject started with development studies of object approach and the SOM approach has begun to take shape in the 90s with the emergence of portals in recent years, SOM has emerged as a technological infrastructure with workflow and process management tools. These methods are preferred because their efficiency and success are high and costs are low.

Channabasavaiah et al. (2004) highlighted that the managers of the organization suppress for the more efficient use of corporate resources and supported the idea that SOA is one of the biggest solutions to this problem [2].

In the work of Srinivasan and Treadwell, web services and SOM style have been seen as a new basis. He explained the basic concepts, relationships and benefits of SOM [3]. Lee et al. found that the SOM structure is built up of many services and that integrated applications are based on the SOM structure [4].

Beklen stated that SOA is an approach with unique rules that are implemented in different fields such as finance, insurance and public, and requires detailed analysis and modelling in order to define services well. SOA has been successfully implemented for different sectors all over the world and has increased the development level of architecture. It is stated that the solution to application problems by layers abstraction and using the methods of platform-independent integration are the most important advantages of architecture. Large-scale software projects are developed by applying software engineering disciplines to solve different business problems in various sectors. However, each institution has used different technologies and approaches in its infrastructure. In some projects, applications were developed without using any architectural approach. Such situations have transformed applications, which have grown and become more complex over time, to the applications that cannot be integrated with each other, with difficult maintenance, repetitive work, or platform-dependent applications. SOA, which aims to solve such problems by isolating the application layers from each other, has been examined under subheadings such as architecture, service, and definition [5].

Üstündağ, in his study, said that providing the interoperability of existing corporate applications and newly developed or purchased applications is one of the areas where organizations invest most. As a result of this study, the systems that are integrated with each other is an application developed with .Net and an existing SAP (System, Application & Products) application. As a result of his study, the systems that are integrated with each other is an application developed with .Net and an existing SAP application. He argued that the importance of developing such integration layers as different layers could be better understood if the number of applications interacting with each other increased according to the size of the organization, business volume, and complexity of the processes [6].

Dongsu et al. reported that SOA facilitates the collection, organization, and maintenance of institutional solutions in order to respond rapidly to changing needs. In an application using SOA, reusability is high. Every service created can be reused even in different projects. In case of a change in business requirements,

this change only has an effect on the application side. In other words, the relevant change is made only in this service and the applications that use this service have revealed that applications are not affected by the change to a great extent [7].

In their study, Çopur argued that SOA has an architectural style for creating software applications that use the services offered in a network like a web. Applications in SOA are built based on services. It promotes a loose connection between services so that they can be reused. A service is an application of well-defined business functionality, and such services can then be consumed by customers in different applications or business processes. Using SOA, businesses can attain significant effectiveness in development costs and quickly adapt to changing business conditions by reusing and restructuring existing services in the development of new applications. SOA has demonstrated better integration of enterprise IT resources, including pre-isolated application silos and legacy systems [8].

Bussler, in his study, revealed that the SOA structure provides facilities for the solution of problems such as data increase, development of service-oriented technologies, integration studies, process management, security principles, and corporate information systems [9].

Kreger, found out that it reduces application complexity by defining Web services and SOA service interfaces, as well as providing Web services, full-time integration, and interoperability of legacy applications [10].

Herand stated that in order to be flexible and agile, the software needs to be developed in a structure that is not fragile, portable, uncomplicated, unnecessary repetition, easy to understand, organized and controllable. This desirable structure can be achieved by applying a certain methodology, following a certain methodology, after careful understanding of the objectives of SOM, which aims to minimize the effect of change. Otherwise, he thought that the use of SOM could create a used drug effect without reading it. It should be ensured that the useful parts of the old systems used before the transition to SOM in institutions should also be used in the newly developed SOM based system. In this way, the old systems, which were obtained at great costs, are prevented from becoming completely unusable. In this study, SOM also emphasizes the importance of strategy determination and selection during implementation. After determining the conceptual services, determining the strategies for the development of these services, conducting technical and financial analysis between the identified strategies and in line with these analyzes one of the strategies has been decided to be implemented. It was defended that this way reduced the likelihood of making mistakes at the decision stage. The importance of SOM management platform selection is emphasized when performing SOM-based service development and management [11].

In their study, Jeng and An argued that a large-scale SOM-based project was more successful with System Dynamics modeling methodology. System Dynamics is a methodology that examines and manages complex feedback environments such as business goals and social systems. In this study, various SOM scenario analyzes based on System Dynamics modeling have been performed. These different studies address the overall impact of SOM on different factors and what they will yield in the long run. Two major features of SOM are rapid development and many solutions have identified high degree of reusability in textures and improvements [12].

O'Brien recommends the use of managerial activities in the SOM projects, which include studies on the scope, cost and effort to be spent during the management of a SOM project. In addition to technical, social, cultural and institutional aspects, he also examined the level of maturity with the experience of the

organization and defended the idea of trying to estimate the scope, cost and effort by considering the different types of SOM projects [13].

Kaya stated that SOM is used in many different areas thanks to its features and advantages. It can be easily adapted to different platforms. Using SOM in applications provides great advantages in terms of time in development, but creating services at the beginning of the process is not very advantageous in terms of time. The most prominent feature of SOM is its reusability, a service can easily be used in different applications without any changes. Having a description of a service is enough to use it. To use a service, it is enough to send the necessary variables [14].

In the Tarhan's study, a system will need to be transformed one by one in its existing methods, and in many places need to be recoded. Newly developed infrastructures may require a higher labor force than the development costs of closed-loop applications. The reason for this is the adaptation times of the methods that can communicate in the same layer and scope under different additional broadcast layers (shells), which will be made available to other systems. Although there are development environments that can easily perform related transactions, it is known that related periods lead to increases in business development times.[15]

Menekşe stated that the advantages of the SOA strategy will be mentioned in the coming sections. Thereby, it is not wrong to say that advantages will be valid for the solution proposed. The SOA based solution here firstly models all processes correctly, determines the real- roles of the scenario, states all work routines for all these roles. After this, the solution models all processes as services.[16]

3. Materials and Methods

The basic idea of SOA is service. Service is defined as a discrete unit of business functionality that can be done through a service agreement. The service agreement specifies the interaction between the service provider and the service user. These are:

- Service interface
- Interface document
- Service rules
- Quality of service
- Performance

SOA services are described in standard definition language, allowing multiple platforms and protocols. They have a common interface. By supporting similar processes, they conduct the activities by communicating with each other [17]. In SOA, each function is defined as a service. All services operate autonomously [18].

SOA is an approach that enables different services to work together by creating certain structures, namely, service is in service. Normally in multi-layer applications, one layer must always implement on another layer. In this approach, the hierarchy needs to be smooth. To give a simple and classic example, let's assume that your project has a data layer and a business layer. The Business layer invokes the data layer, handling database operations, thus becomes unilateral and project-based. However, if you create the data layer as a service, it stops being one-sided and allows access for users from anywhere. In this way, it does not matter where and/or on which platform the application runs. SOA eliminates dependencies in point-to-point integrations [19].

The lifecycle of SOA is as follows; Model (identify, design, analyze requirements), Merge (create, merge, test), Engage (people, processes, information integration) and Manage (manage applications and processes, monitor process, business unit, and

information technology coordination).

In the long run, SOA aims to reduce integration costs between applications through re-using services, adopting standards and increasing interoperability between applications; it is also intended to create business processes by using services as lego stones when necessary and thus to fulfill business needs faster [20].

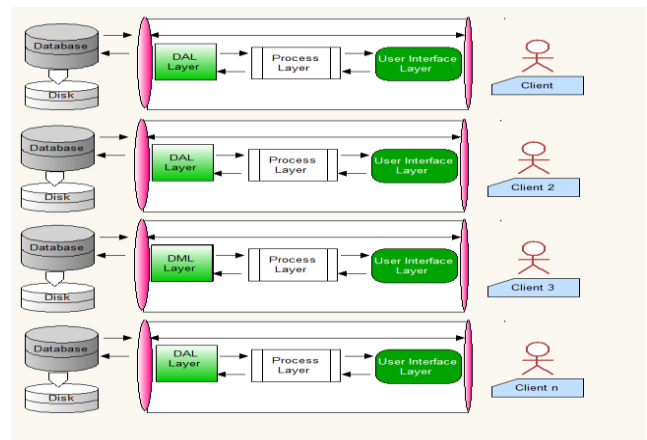


Figure 1 Service Oriented Pre-Architectural Building Model

Figure 1 illustrates the point to point architecture before using the Service Oriented Architectural design pattern. In this structure, the client is first connected to the data access layer via the user interface and then to the database. In each application developed / to be developed, if any CRUD (create read update delete) transaction is used over the database, if they use the same database (no enterprise allows uncontrolled connection), they must be connected to the database separately or different independent and incompatible data for each application. must form bases. Processes (methods / functions) that perform the same or similar work within the application must be written separately for each application.

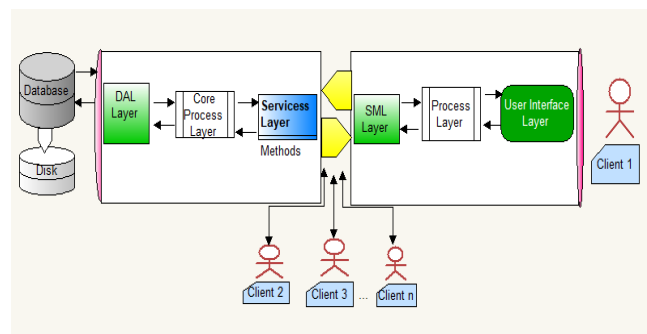


Figure 2 Service Oriented Post-Architectural Building Model

In Figure 2, with Service Oriented Architecture, each client connects to the same database from different interfaces and saves them from independent and incompatible data silos. The reusability principle, which is one of the basic principles of objective programming with unlimited use of each process (method / function), will be applied not only within the limits of the software but also within the architecture of the software. Our main application, which reaches the external environment through services, will be platform independent and the integration of our basic application with each application will be ensured due to the service structure.

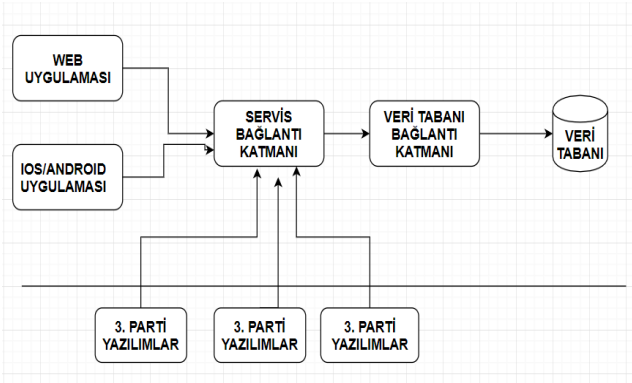


Figure 3 SOA General Structure

Figure 3 shows an example of how multiple applications use the same workpieces (methods) from different interfaces by calling the same services. The use of these one-time workpieces by numerous applications has an impact on maintenance and software costs. At the same time, the database was separated from all applications and this structure created the security of the database. By using a single database with SOA, heterogeneous enterprises/ institutions and organizations can get rid of incompatible data silos and allow all data to be stored in the same database

4. Developed Services

In the structure we created with SOM, the same database has controlled features for more than one project, eliminating the need to rewrite the tables, fields, procedures, triggers or indexes for each project and eliminating the problem of incompatible data silos. In a software developed with the traditional point-to-point software architecture, the need for re-code development in each layer of scripts was needed to develop new screens that were subsequently required. In the project developed with SOM, it is seen that the methods in the services are brought together and the institutions like lego can create the new screens they need only with the development they will make in the user interface layer. For application created with SOM, database layer is created on application side for CRUD operations to database. In line with the analyses, the required threads are minimized to the maximum level in accordance with the reusability principle and a work layer is formed. Each thread is presented as a method of web service over the service layer. The user interface layer is created externally and the services are integrated. In the established automation, each thread as a service method and a structure that can be integrated in other projects of institutions / enterprises is obtained. The project has been functioned and process oriented with SOM. Designed for performance and change. The platform has not been agreed and provided platform independent integration. Independent blocks, objects, and services that work together, not application blocks, are created. An important advantage is that incremental development and lifecycle take the place of a single and long development process.

Figure 4 shows the service layer image of SOM. Each workpiece written in the main application was published as a method of service over the service layer. Therefore, the platform independent main application is accessible to every thread. At the same time, these threads came together to create new screens without re-creating rules and database layer.



Figure 4 Service Methods

5. Conclusions

It has been revealed that, through SOA, the same thread can be used in different projects without the need to redevelop the thread since the threads are written as the methods of service. It is also seen that the cost of code development due to reusable service methods may decrease. Moreover, it is evident that thanks to the structure of application developed by SOA, integration can be made with each project since it responds to every request of the

end user with notation such as SOAP (Simple Object Access Protocol), json, etc. Hence, it provides platform independent development opportunity by being integrated to each platform. It was seen that a meaningful and consistent data source could be created by using a database and that the incremental code has been developed to give more control over the application and that independent components, objects, and services work together. It has been also seen that the services originating from the SOM structure have been brought together by information technology employees like lego, new screens and new applications have been developed.

References

- [1] B. Şenyurt, (2015.04.04). "What is SOA Architecture?", (Accessed on: 2018.07.14), <https://www.buraksenyurt.com/post/SOA-Nedir>
- [2] K. Channabasavaiah, K. Holley, M. Edward, Jr. Tuggle (2004). Migrating to a service-oriented architecture, *IBM Journal of Research Development*,3-22.
- [3] L. Srinivasan, and J. Treadwell, 2005. An Overview of Service-oriented Architecture, *Web Services and Grid Computing*, HP Software Global Business Unit
- [4] Y. Lee, C. Ma, S. Chou, (2005). "A Service-Oriented Architecture for Design and Development of Middleware", *Proceedings of 12th Asia-Pasific Software Engineering Conference (ASPEC'05)*.
- [5] A. Beklen, (2009). "Service Oriented Architecture Concept, Technology and Design," Institute of Science and Technology, Maltepe University.
- [6] S. Üstündağ, (2006). "Service Oriented Architecture," Institute of Science and Technology, Ege University.
- [7] K. Dongsu, and B. Doo-Kweon, (2010). "Bridging Software Product Lines and Service-Oriented Architectures for Service Identification using BPM and FM", 9th IEEE/ACIS International Conference on Computer and Information Science. Yamagata, Japan.
- [8] D. Çopur, (2011). "An Application of Service Oriented Architecture (SOA) Approach.," Institute of Science and Technology, Baskent University.
- [9] C. Bussler. (2013). "B2B integration: Concepts and architecture": Springer Science & Business Media.
- [10] H. Kreger. (2003). "Fulfilling the Web services promise". *Communications of the ACM*, 46(6), 29-ff.
- [11] D. Herand, (2013). "Development of a methodology for SOA applications on business processes" (PhD Thesis), Institute of Science and Technology Sakarya University.
- [12] Jeng, JJ. and An, L., (2007). "System Dynamics Modeling for SOA Project Management", *IEEE International Conference on Service-Oriented Computing and Applications (SOCA'07)*. California, USA.
- [13] O'Brien, L., (2009). "A framework for scope, cost and effort estimation for Service Oriented Architecture (SOA) projects", *Australian Software Engineering Conference*. Sydney, Australia.
- [14] Kaya M., (2009). "Service Oriented Architecture" (Master Thesis) Doğuş University, Institute of Science and Technology, İstanbul
- [15] H. Tarhan, (2017). "Development of an enterprise resource planning software and intelligent decision support system tools.," Institute of Science and Technology, Yıldız Technical University.
- [16] G. Menekşe, (2011). "A case study with SOA on supply chain process in turkish automotive industry", Institute of Science and Technology, Atılım University.
- [17] D. Çopur, (2013). "Development of a methodology for SOA applications on business processes.," Institute of Science and Technology, Sakarya University.
- [18] T. Francis, E. Herness, R. High Jr, J. Knutson, K. Rochat and C. Vignola. (2004). *Professional IBM WebSphere 5.0 Application Server*.
- [19] A. Arsanjani, (2002). "Developing and Integrating Enterprise Components and Services: Introduction". *Communications of the ACM*, 45(10), 30-34.
- [20] M. P. Papazoglou and W.-J. Heuvel. (2007). *Service oriented architectures: approaches, technologies and research issues*. *The VLDB Journal—The International Journal on Very Large Data Bases*, 16(3), 389-415.