

# SERVICE ORIENTED ARCHITECTURE A PROMISE TO THE FUTURE

A. GÎRBEA<sup>1</sup>      F. ŞIŞAK<sup>1</sup>      L. PERNIU<sup>1</sup>

**Abstract:** *Service oriented architecture has had a great impact on the business world. Its novelty is the different approach, which consists of defining reusable services and then building applications by loosely coupling the services in a four layered architecture. This article describes the early difficulties of understanding and implementing the new architecture, and the evolution, as companies are more and more discovering the advantages of this new concept, alongside with the appearance of dedicated solutions provided by specialised software firms. We will closely analyse the progress so far and show future directions of development, like the manufacturing industry, where SOA hasn't even been taken into consideration so far.*

**Key words:** *architecture, service, SOA, flexibility, reusability.*

## 1. Introduction

The term architecture is used to define a plan for building systems, which meet well and precise defined requirements.

The term software architecture is used to refer to the building of a software system (main components, the relationships between the components and the interface of each component).

The novelty of Service-Oriented Architecture is given by the different approach [2], i.e. building systems that concentrate on constructing applications as a combination of services. Hence some great advantages: agile enterprises, greater flexibility and smaller reaction time. Those newly defined systems will then deliver capabilities as services.

SOA can't be installed and it's not a technology, it's just an approach used to

remodel systems.

Applications built with service oriented architecture will integrate many and very different applications, having common interfaces, assuring reusability and greater maintainability.

When talking about architecture, there are three important facts:

- Stating the relationship between the parts;
- Discovering the main parts of the system;
- Combining the parts in a way that provides value for the level above.

When talking about SOA, there are other important aspects, like:

- *Processes*: high-level business functions;
- *Services*: the units which assure business functionality;
- *Integration*: using existing applications or data as services;
- *Documents*: business information (like purchase orders);

---

<sup>1</sup> Dept. of Automatics, *Transilvania* University of Braşov.

- *Semantics*: the meaning of data that goes through the process;
- *Transformation*: changing information from one format to another;
- Stating the relationship between the parts;
- Combining the parts in a way that provides value for the level above.

## 2. History of SOA

The term *service oriented architecture* has been defined in 1996 [5], but the concept of SOA was defined even earlier. It were finance companies (like banks) and telecom companies which were able to define services and to combine them into applications using distributed technologies (like CORBA or the Distributed Common Object Model - DCOM which is an extension of the Component Object Model - COM - introduced by Microsoft).

Analyzing early attempts, there are two reasons responsible for their failure:

- Most companies didn't have the system programmers and architects to manage these new technologies;
- No one really knew how a good service was meant to be.

Most of the attempts failed even before people were able to see if the services were working or not. Many didn't really know what SOA meant and therefore they couldn't succeed in implementing or using it.

There were other technologies, like Tuxedo (which is service-oriented and stateless), which stayed at the base of some well performing, distributed applications of their day. Some companies were able to successfully implement SOA, but their number was much smaller than the one of failed projects.

In Figure 1 we present the timeline of the SOA evolution.

Later, companies like IBM, Oracle or Sun developed this concept further on and defined the term of SOA.

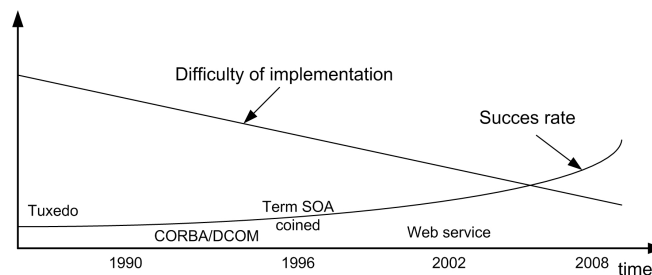


Fig. 1. Evolution of SOA

Looking at today's situation, we have to admit, that things have evolved and web services are easy to use. The technology hasn't become easier but tools and modern environments have brought programmers a great step forward. We can say, that it's possible to develop great services without even knowing what a good service or what distributed technology is about.

Knowledge of distribution and services are today integrated in a platform, which may be based on Java, .Net or even something else.

## 3. What is a Service?

Services are the basic parts of SOA. First of all a programmer has to find out what a good service means [11]. The Organization for the Advancement of Structured Information Standards (OASIS) defines a service as "a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service

*description*”. Before saying that a service is a service, it will first have to implement a set of special characteristics.

When talking about a service, there are two different aspects:

- *Service implementation*: specifies how the capabilities (outputs) of the service will be provided. We are talking here about the internal functionality of the service, which may be based on other existing applications or services or on code written specially for this service. A service will always have internal data used to deliver the specified information as outputs;

- *Service interface*: specifies operations done by the service, inputs and outputs of the service, and the protocols used to define how the capabilities have to be used (for example: data types which have to be provided at the inputs and data typed found at the outputs).

These two parts are totally separated, as shown in Figure 2.

By introducing this separation, consumers won't have access to the implementation of a service [10]. The only thing they have to be concerned about is the interface contract (number and type of inputs and outputs) and once this contract is specified, they may start using the service. The producer of the service may then change the implementation, but this won't affect the consumer as long as the interface remains unchanged.

In addition to the specific structure of a service shown in Figure 2, good services have the following properties [2]:

- *Loose coupling*: this may be the most important characteristic of a service. Coupling refers to the relationships between the service consumer and the service provider. These relationships have to be few, so that they are easy to manage;

- *Modularity and granularity*: business processes are made up of modular services, which can on their turn be composed of other services. Granularity refers to the functional richness of a service. Services

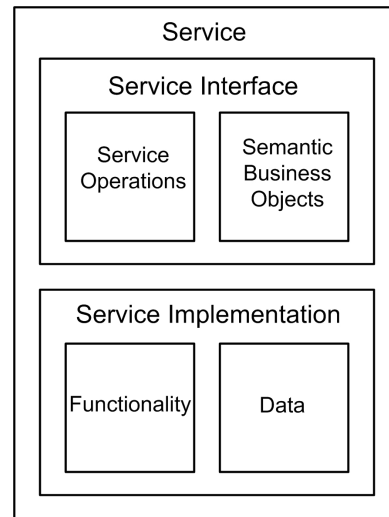


Fig. 2. *Structure of a service*

have to be coarse-grained in order to provide greater functionality within a single operation;

- *Encapsulation*: the internal implementation and data of a service is hidden;

- *Reusability*: a single service may be used in different business processes or accessed by different consumers;

- *Isolation of responsibilities*: every service will be responsible for one or more specific functions. As a result, each function will be performed in only one place. Thus redundancy will be reduced and consistency assured.

## 4. Implementing SOA

### 4.1. SOA Layered Architecture

Figure 3 shows the semantic layers in an SOA environment [7].

At the bottom there is the operational systems layer, also called the *engine room* of the SOA environment. Here the reusability of the services is assured. The registry will store information and metadata about the reusable aspects.

The next layer shows the available services. These may be combinations of

information assets and applications but also external services. The services will provide interfaces and they can be combined into powerful applications.

The third layer shows the business processes that represent the activities of a

company. A *business process* is a series of operations that must be executed sequentially, following a set of rules. People often use the term *orchestration* to refer to the selection, sequencing and execution of operations.

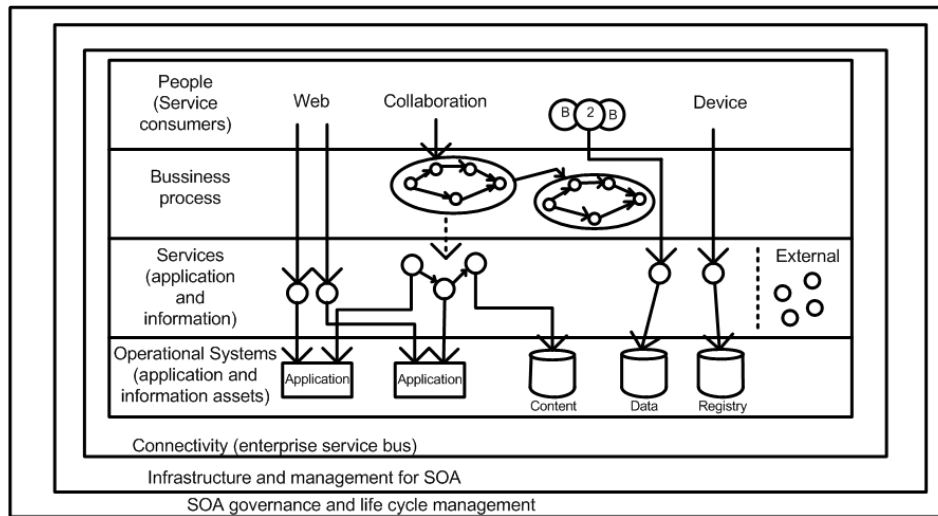


Fig. 3. *The layered architecture of SOA*

The top layer presents the ways or channels that may be used to access the underlying services.

As one can see, apart from the layers we've described until now, in Figure 3 there are also some frames. The *SOA governance and life cycle management* frame refers to the set of policies that consumers and providers of services have to respect and practices that have to be done in order to assure that those policies are implemented correctly.

The layer in the middle, *infrastructure and management for SOA*, refers to the minor extensions for the existing infrastructure in order to support SOA in a more organized way. The stages for structuring and managing the environment are described by the SOA life cycle, which contains the following three steps [8]:

- *Modeling phase*: recognizing the needs

of the business and then designing the services which should perform best.

- *Assembling phase*: defining the business process by assembling newly created but also existent services.

- *Deploying phase*: deploying the business processes into the environment, where they will be used.

- *Managing phase*: Monitoring the business processes from both points of view: provider and consumer.

The inner frame of Figure 3 refers to the *service connectivity*. One of the main advantages brought in by SOA is the flexibility. Much of that is lost if services are connected point-to-point.

Figure 4 displays two different implementations of SOA: on the left side a point-to-point integration, where it is very difficult to change anything. On the right side, this problem is eliminated by using an

enterprise service bus (ESB) which assures loose coupling between services [9]. The bus will just guide the requests to the new services.

Using ESB there is greater flexibility and

consumers and providers become more independent. An ESB provides *service connectivity* which ensures that services are connected and exchanged with the correct data format and message model.

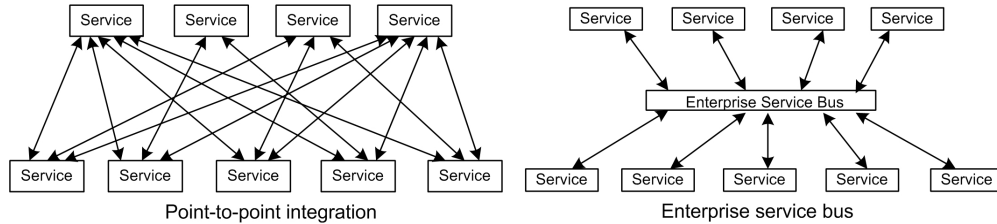


Fig. 4. Assuring flexibility by using an enterprise service bus

**4.2. Registering and Contracting Services**

This is a very important aspect when building SOA based applications. Figure 5 shows the *find-bind-execute* model used by SOA. According to this model, service providers will register their services in a

public registry. Then the consumer will look through this registry in order to find the needed service. Then the consumer will be given a contract and an endpoint address for that service (in order to be informed when the service is changed).

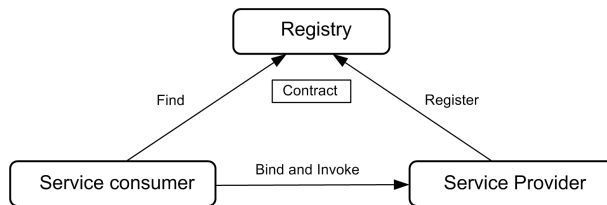


Fig. 5. The *find-bind-execute* paradigm

**5. SOA Solutions Provided By Different Companies**

The most important companies which have build platforms for SOA are: IBM, Microsoft, BEA, Oracle, and Sun. Usually every family of technologies has its own SOA solution, but they can be combined and used without constraints. XML-based technologies like SOAP, BPEL and WSDL are very often used [3].

We will now take a closer look at the solutions provided by two main companies

(we have chosen these two companies because of our experience with their software and because their products have had the greatest impact on the business world).

**5.1. Microsoft**

There are many software packages (software and frameworks) provided by Microsoft in order to help people define services and to build applications [14]. Furthermore they offer instruments for the management of products. There is the *.NET*

*Framework* which is used to build applications on the Windows platform and the extensions of *.NET* enable the development of SOA-based applications.

Another powerful tool is *Microsoft BizTalk Server* which is the main component of the enterprise service bus (ESB). It controls the interaction between web services and enables their combination into complex applications. *Microsoft Office SharePoint Server 2007* is another important software provided by Microsoft which is used to optimize the interaction between people and applications.

Further *Microsoft Windows Vista* simplifies the work with services, and by this it assures greater security and reliability.

There are also some tools used to manage the service lifecycle, like *Systems Center Operations Manager 2007* (displays the health status) and *Microsoft Operations Framework* (offers guidance for realizing greater reliability and availability).

## 5.2. IBM

IBM offers maybe the greatest variety of SOA software [12], [13]. We'll mention only the most important of them.

First there is *IBM Rational RequisitePro solution* which is a very powerful use case management tool. *Rational Software Architect* is a tool based on Eclipse used to model software components. Another important tool is *WebSphere Business Modeler*, which is used to specify the business design: processes, resources and organization. *WebSphere Integration Developer* is also based on Eclipse and has full support for WS-BPEL (Web Services Business Process Execution Language).

Then there is *Rational Application Developer*, which is an integrated development tool for J2EE programming.

On the server side, there are: *WebSphere Application Server* (hosts J2EE and web services) and *WebSphere ProcessServer*

(hosts server environment for the deployment of BPEL-based business process models).

## 6. Implementing SOA in Industry

Up until now we have stated many advantages of SOA, but also the difficulties encountered when trying to implement it. We have seen that SOA has first been introduced in banks and telecom companies [2], but what about using SOA in industry, in factories?

There are two aspects that need to be discussed [4]. In order to present the first one we will refer to utilities and energy companies. They need processes that allow managers to see critical information about the company. These processes have to gather the *right* data, refine it and then provide information for the managers, so that they can take the right decisions. These data refer both to the business aspect and the technical aspect. Very often managers have to take decisions based on wrong information and it's here where the concept of SOA has to come in and solve the problem. The right services have to be defined, which combined together will be able to refine raw data and provide real results [1], [6]. Here's a short list of services which could be implemented: services returning consumer trends according to geographical, time or other variables, services returning KPI (Key Performance Indicators) values from the business aspect but also from the technical/manufacturing aspect, services which should allow the clients to retrieve important aspects regarding their relationship with the company which provides services, (intelligent) services which should be able to predict future market developments, services used for business-to-business communication, i.e. the integration of industrial partners etc. As a development platform we would indicate

the IBM software packages: Websphere Modeler, Integration Developer and Process Server (they are Java based but one can also use other programming languages).

The other aspect refers to the manufacturing process in factories. Nowadays companies have to be very flexible, they may have to cope with situations where clients want them to manufacture small numbers of different pieces. And we have to admit, that the smaller the number of pieces that have to be produced is, the greater the price of one single piece will be.

That's why in manufacturing you companies have to be as flexible as in a bank or in other companies where consumers are people. By assuring greater flexibility and smaller reaction time for changes, companies will be able to outperform their competitors. One example where such services may be used is a flexible line (like the Flexible Manufacturing System - FMS - 200 which is a totally modular and flexible didactic system, developed for an integral training in automation. The technologies integrated and the assembling process with several variants, allow the user to develop the professional skills required by the automated industry of today and tomorrow), where different stations have to perform sequential operations. A bunch of services has to be defined and then combined in order to fulfill the requirements of a piece manufacturing. Then, when coping with different orders the program developers may define other services and in this way enhance the service database. As a result, companies will be able to react even faster. In this second case the services which should be implemented again address both the business and the technical aspect. As the business aspect is very similar to the first case we will only provide some examples for the technical aspect: services which should automatically determine the

equipments and the timetable for the manufacturing of a component, services which should automatically detect supply problems or needs, services which should provide the programs needed to manufacture a component or only a part of a component etc.

Finally we'd like to mention one of the most important problems encountered when developing Service-Oriented Architecture (SOA), especially in industrial environment. This aspect involves meeting the security challenges, because the responsibilities of SOA security refer both to the service providers and the consumers. Recently, many solutions which address these challenges have been implemented, such as the Web Services Security Standards (WS-Security and WS-Policy). However, those standards are insufficient for the new generation of Web technologies, including Web 2.0 applications.

The services in this area should extend the WS security standards, e.g. by adding intelligent data mining techniques.

## 7. Conclusions

Service oriented architecture can be seen both as an opportunity and as a challenge for companies from all areas. The business managers will have to discover the needs of their companies and the aspects that will be improved by using SOA. Then, the main services that will assure the progress of the company will have to be defined.

One can think of SOA as defining: the capabilities needed to fulfill business needs, organization of the capabilities (services with interfaces), who provides and who uses the services and how consumers and providers communicate through the interface.

Service oriented architecture is not new, companies and people have started seeing the benefits of SOA, but there is still a long way to go. We've seen the advantages

(flexibility, reusability, faster reaction time etc.) and disadvantages of SOA (difficulty of recognizing the right services, difficulty in implementation etc.) and we can state that for sure, in time, the difficulties will be overcome and companies will grow in quality, alongside their SOA projects.

There are already many SOA applications used to manage data and information but it's the industry where future developments have to include SOA. This is the challenge that has to be managed in order to improve efficiency.

### Acknowledgements

This paper is supported by the Sectoral Operational Programme Human Resources Development (SOP HRD), financed from the European Social Fund and by the Romanian Government under the contract number POSDRU/88/1.5/S/59321.

### References

1. Blum, N.: *Journal of Network and Systems Management*. In: *Journal of Network and Systems Management* **17** (2009) No. 1-2, p. 33-52.
2. Erl, T.: *Service-Oriented Architecture: Concepts, Technology, and Design*. Indiana. Prentice Hall PTR, 2005.
3. Hewitt, E.: *Java SOA Cookbook*. Sebastopol. O'Reilly Media, 2009.
4. Hong, H.: *A Low Cost Tax SOA Infrastructure in Grid Applications*. In: *Wuhan University Journal of Natural Sciences* **11** (2006) No. 5, p.1320-1324.
5. Hurwitz, J.: *Service Oriented Architecture for Dummies*. Indianapolis. Wiley Publishing, 2007.
6. Josuttis, N.M.: *SOA in Practice*. Sebastopol. O'Reilly, 2007.
7. Lawler, J.: *Service-Oriented Architecture: SOA Strategy, Methodology, and Technology*. New York. Auerbach Publications, 2008.
8. Marks, E.: *Service-Oriented Architecture Governance for the Services Driven Enterprise*. New Jersey. Wiley Publishing, 2008.
9. Reddy, V.: *Evaluating Legacy Assets in the Context of Migration to SOA*. In: *Software Quality Journal* **17** (2009) No. 1, p. 51-63.
10. Rosen, M.: *Applied SOA: Service-Oriented Architecture and Design Strategies*. Indianapolis. Wiley Publishing, 2008.
11. Tinny, N.: *Understanding IBM SOA Foundation Suite: Learning Visually Examples*. Indiana. Pearson Education, 2009.
12. Wahli, U.: *Building SOA Solutions Using the Rational SDP*. New York. Redbooks, 2007.
13. <http://www-01.ibm.com/software/solutions/soa/>. Accessed: 03-11-2009.
14. <http://www.microsoft.com/SOA/>. Accessed: 06-11-2009.