Principles of Service-Oriented Architecture and Web Services Application In Order to Implement Service-Oriented Architecture in Software Engineering

Mehdi Khanbabaei, Mehdi Asadi

1Young Researchers Club, Shabestar Branch, Islamic Azad University, Shabestar, Iran.
2Department of Computer Science, Islamic Azad University, Khamneh Branch, Khamneh, Iran.

Abstract: Service oriented architecture (SOA) is an Evolutionary distributed computing from the design hypothesis based request/response for synchronous and asynchronous applications. For an application, business logic or private function became modular and presented as services for consumer/client application. These services have free connection that service interface is independent implementation. For success in business, organizations require information within the organization and Meta information. However, the existence of huge information repositories created by traditional applications prevents information flow between the external and internal parts. SOA is an approach that replaces these huge sources by loose coupling services and satisfies the required information flow and finally increases speed in these organizations. In this paper, first we study the fundamentals of SOA and then describe web services application in software engineering. In addition, we show that web services can be used as initial block in program creation according to several indicators like accessibility, quality, cost and efficiency. After analysis of required services for organization, we present new service oriented analysis and design method. Finally, we express a solution for optimal use of the Web Service in SOA.

Key words: SOA, Service Oriented Architecture, Web service, Platform, Service, Software Engineering

INTRODUCTION

Currently, SOA is accepted as an architectural style for organizations. It enables them to have maximal speed and minimal circulation of information flow. Yet, wherever information technology has been developed there are a number of problems and obstacles to achieve this end. It must be borne in mind that using SOA does not warrant success per se. What warrants success is that IT architects must be familiar with the advantages and disadvantages of this technology and be aware of how to use it having in mind the investors of the organization (Koch, 2005). SOA is a new technology which delivers software functions in the form of service. The services acquired can be called on by other applications and be used for constructive new services (Chatterjee et al, 2004). SOA is an advancing technology that can introduce dramatic changes in constructive distributed applications. The services consist of discrete elements with predefined and certain interfaces that converse with each other through XML messages. Using SOA, one can come up with solutions which are not limited to the scopes of organizations and companies.

Service-oriented architecture can be investigated from different perspectives: every person or beneficiary has a conception of SOA depending on his position. As follows, the perspectives of IT managers, business managers, designers and implementers of system will be discussed (Bell, 2008):

- IT managers: A style of architecture that includes patterns and rules which culminate in the development of characteristics like modularity, packaging, loose coupling, reuse and Composability. It structurally consists of a service-provider and a service-requester.
- Business managers: a set of services that organizations wish to provide for their clients or partners.
- Designers and implementers of information systems: a style of programming which uses unanimously-agreed and technology-independent standards and supports the intractability between software elements regardless of their platform and implementation technology.

The principles of SOA:

This section will point out some of the principles of SOA, which must be taken into account in implementing programs based on this technology (Coenen, 2006).

Reusability:

In SOA, services are designed in a way that they can be used in the near or distant future. These abilities are acquired due to the use SOA makes of platform-free standards and also due to the creation and use of service interface. Services are divided into two categories: a group is considered leader and acts as the orchestrator of
other services to which several references are made and the other group is basic services, which do not do any calling.

**Standardized Service Contract:**
Services that need to converse with each other must use a contract to communicate which includes address, name, function and service-specific input and output data. This contract is the only way to recognize the receivers of service to codify which one must be very careful.

**Loose coupling:**
One of the principles of SOA is loose coupling, that is, services must not have intense dependence on each other and be as independent from each other as possible. A service might need the information of other services and use them and at the same time keep itself independent from how the others are implemented.

**Hiding Of Internal Implementation:**
Only the part of service which can be observed by service-receivers is considered service interface. It is neither necessary nor useful for users and service-receivers are aware of the internal implementation of service because the need to change the internal implementation technology which might be increased must be invisible to the users. The users must only be aware of the service performance not the way it is implemented (Valipour, et al, 2009).

**Service Composability:**
Services must have the capability to be combined with each other to make optimal use of service orientation. Services couple with each other, creating various granularities to make them able to cover function and performance of a sub-system.

**Service Autonomy & Statelessness:**
One of the main principles of SOA is its autonomy from the context in which it is executed. This means a service must not follow the condition in which it is executed because if it saves its current state it might get into trouble when responding to other requests and lose its autonomy and the ability to be used simultaneously.

**Discoverability:**
The interface defined for the service must have the ability to be derived, discovered and extracted by its users. Violating this principle makes SOA, which tries to step over the organizational boundaries, lose its whole concept. One other benefit of this technology is the discretion given to the user in choosing service-providers which compete on the basis of several factors including quality, cost and support.

**The virtual SOA platform:**
The objective of a virtual SOA platform is to separate service from implementation and make services independent from applications so that all of its components can be used for implementing other platforms. Another objective is that service designers can order what service they need without there being any relations between service and implementation. Virtual SOA includes a blueprint that involves expansion and implementation of new services. This blueprint provides a guideline for expansion and implementation of applications, which ensures that the published services follow structural rules consonant with the ideas of customers and managers.

If it is possible that some of various programs are able to share the same structure, one might ask how a general architectural style can be introduced which covers all these applications. This style might be created by different ways. This is to say that a technical environment might be public, being a set of policies or a series of frameworks or practices. The components of a virtual SOA must include the following:

- The host and client environment
- Firmware
- Capital and assets
- Broadcasting and extraction
- Managing service layers
- Security structure
- Monitoring and measuring
- Diagnostics and troubleshooting
- Managing customers and partners
- Web service protocols
- Managing authentication
The Use Of Web Services in SOA:

A web service is a software system that is specified by URI (Universal Resource Identifier) and its general interfaces are defined and interpreted using XML; other systems can interact with web service by a determined behavior and in accordance with its definition. Web services are among the most important recent developments in software field which will certainly have remarkable results for designing and implementing applications and paving the way for the new generation of web-based applications. Web services can be used for implementing a SOA and create the constructive blocks of the function which are accessible through the internet. The designers of SOA typically use web service standards, used extensively in industrial and commercial contexts, to implement their services. These standards remarkably increase the interoperability of commercial systems.

A web service has an interface that indicates how the web service should be run by machine. Another way to converse between systems through web services is by using SOAP messages, which typically use HTTP and XML and work in combination with the other standards related to web. In addition, web services can be considered as a programming interface that has the ability to follow WSnn protocols.

Web services are independent of software platform and can cover the accessible systems in a way that make them available through the internet or enable them to be used for displaying systems and new applications. Some of the applications that can help implement SOA based on web-service-based technologies are Jini, CORBA or REST. Figure 1 shows the components of SOA (channabasavaiah, et al., 2003).

![SOA Diagram](image)

**Fig. 1:** SOA components Every constructive block of SOA might play one or two of these roles:

Service generator: service generators create web services and publish the interface related to web service and information needed to access it in the registry service. Every service generator must take some fundamental decisions including:

- Which services should be displayed?
- What compromise there must be between security and ease of access?
- What criterion does service pricing rest on?
- Which branch of service must be listed in the interface?
- What sort of agreement of commercial partners is needed to be used in the service?

The service registries indicate which services are available and who is receiving the service. Transactional interfaces are divided into public and private ones. The public interface is made accessible to the public through the internet and in a cost-free way while private interface is accessible for a limited number of people like users in a commercial company. Service generators must estimate and calculate the amount of information to be exchanged and provide high validities for services.

Based on business model, brokers can lead to the maximization of requests and the increase in enrolling for receiving the service. UDDI (Universal Description Discovery and Integration) directory is a way for publishing and extracting information about web services. One can refer to ebXML (Electronic Business using eXtensible Markup Language) and also interfaces based on MDR (ISO/IEC 11179 Metadata Registry) standard as the other interfaces of service.

Service consumer: the service-consumers or the clients of web services find what they need in registries of the brokers and through various transactions of search and connect to the service generator to request the service they need. Obviously enough, if service generator has the capability of providing several services simultaneously, the service-consumer can have access to multiple services at the same time.
Although there are a number of negative points about using web services as the basis for SOA, all of them are soluble issues. For example, as regards efficiency, one can use stronger processors or for solving the issue of security, infrastructures based on asymmetric encryption can be used. Table 1 summarizes the properties of a good service that uses features of SOA and web services.

Although using SOA have many benefits like flexibility, reusing applications, helping organizations to make the maximum use of applications. It enables commercial companies to react quickly to the rapid changes of commerce world; it presents some challenges and concerns that are summarized in table 2. Although these challenges might be worrying, using this technology has gained a broad acceptance due to the advantages the use of SOA and web-services provides.

Table 1: Properties of services implemented using web services and SOA.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Technologies used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standardization</td>
<td>Due to the existence of various heterogeneous protocols and non-comprehensibility of service, it is all the more necessary to use standardization methods.</td>
<td>W3C( World Wide Web Consortium) OASIS( Organization for the Advancement of Structured Information Standards) IETF( Internet Engineering Task Force) JCP( Java Community Process) OMG(Object Management Group)</td>
</tr>
<tr>
<td>Addressing</td>
<td>One of the most fundamental requirements of using web services in SOA is addressing because there is no address available for web service directories on the web. SOAP messages must include addressing information of the final machines.</td>
<td>WS-Routing WS-Referral</td>
</tr>
<tr>
<td>Managing metadata</td>
<td>Includes information needed to construct body and headers of messages so that the requestor can access the service he wants.</td>
<td>XML Schema WSDL WS-Addressing WS-MetadataExchange</td>
</tr>
<tr>
<td>Policy</td>
<td>Announcement of policy is the only way through which the requestor can see whether or not the service-provider can satisfy his needs.</td>
<td>WS-Policy Work Group IBM TAM-eB TFIM( Tivoli Federated Identity Manager)</td>
</tr>
<tr>
<td>Security</td>
<td>Security mechanisms are necessary to face destructive threats. These mechanisms evolve around encryption, authentication and authorization</td>
<td>WS-SecureConversation WS-Trust WS-Federation XML-based technologies</td>
</tr>
<tr>
<td>Reliability and Messaging</td>
<td>It is a mechanism which ensures that messages are sent and received by the requestor in a certain time interval.</td>
<td>WS-Reliability WS-ReliableMessaging WS-Eventing WS-Notification</td>
</tr>
<tr>
<td>Transactions</td>
<td>It provides an opportunity for entire success or failure of several directions on persistent data.</td>
<td>WS-Transactions family WS-Context WS-CoordinationFramework WS-TransactionManagement</td>
</tr>
<tr>
<td>Orchestration</td>
<td>In orchestration, a central controller distributes the workflow among several agents (service, worker, system, etc). One of usages of the concept is to break large processes into smaller components under the supervision of main orchestrator.</td>
<td>OASIS WS-Business Process Execution Language</td>
</tr>
</tbody>
</table>

**Service-oriented Analysis And Design Methodology:**

Looking at the articles and books published about SOA, one can see that in the discussion related to methodology the most frequent activities and literature are related to IBM. Two main reasons for this are the precedence of this company in presenting and providing support for the well-known and unique RUP methodology and this company is pioneer in the field of information architecture systems and the high quality of its services. Since SOA is a “style” of architecture it can be designed and implemented in various organizations in different ways to satisfy different needs. Thus, we must not expect to have only one certain methodology to support this approach. IBM has so far introduced three methods for SOA each one of which has its own capabilities and specific conditions and are produced for specific needs. These three methods are:

- The RUP methodology expanded for SOA
- SOA modeling and architecture (SOMA)
- Modernizing legacy services using SOA
Table 2: challenges and concerns regarding service-orientation and the use of web services in SOA.

<table>
<thead>
<tr>
<th>Challenges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>SOA is not a simple application that can be run in a matter of minutes. A software methodology to guide electronic commerce that faces some difficulties. Implementing a commercial plan using SOA needs training and is done in slowly but steadily way.</td>
</tr>
<tr>
<td>Cost</td>
<td>Although using SOA in long term leads to save on costs, it causes high costs in short term usage. The extra costs include promoting the web and security technologies, training in, learning new system, and managing organizational system.</td>
</tr>
<tr>
<td>Network Degradation</td>
<td>Loose coupling of services and the coupling of several services together lowers the efficiency of web. Occurrence of any problem in each one of these services leads to the breakdown of the whole application. To prevent these breakdowns and increase the accessibility, control of the web must increase.</td>
</tr>
<tr>
<td>Security</td>
<td>Due to open communications which exist besides web services in sharing information among the staff, customers, partners and others, the security of organization is compromised and in need of strong security mechanisms.</td>
</tr>
</tbody>
</table>

Importance of these three methods lies in the fact that they are the perfect collection introduced in SOA and the other methods like Agile SOA, OASIS Adoption Blueprint, and MSOAM are considered subsets and derivations of the methods provided by IBM. The almost similar procedures used for these methodologies are as following:

**Service Design:**

Defining the new service is the first and most important stage in this methodology whose realization requires information received from users to cover the needs they have requested. The three ways to define the service are:

- General definition of the new service on the basis of existing needs: in this method the whole service is designed in a way that it would satisfy both the present and requested needs and can follow the principle of reusability.
- Defining the service based on ready information of the system: there is no need in this method to implement the service and a new interface is just designed to use for functions and the information of the existing system. This kind of services lack logic or code and their lifetime depends on system’s lifetime.
- Combining the existing services to define a new service: this method is the simplest and cheapest method for defining a service. The services in this method must be ready for combination in advance and the designers should try to make the best use of the full potential of the basic services.

**Choosing Software Platform:**

The best technology and platform for implementing services in this stage are selected based on criteria in consonance with the specific performance of each service. In addition, security factors, accessibility, efficiency, and reliability must be taken into account and attempts must be made to reach these critical factors.

**Implementing Services:**

In this stage, implementation and establishment are launched in terms of the products acquired from the previous stages. Issues like configuration, managing versions and satisfaction with services must be taken into account. Before the service is fully implemented, a pilot version of service and, in case of satisfaction of users with it and impeccability of implementation, the final version of the service is provided.

**Publishing Services:**

After reaching success in designing and implementing the service it is time for publishing the service for use. In order for the users to recognize and receive the service easily, the technical information, service procedure, service order, and other critical properties of the service must be documented in a WSDL file and made accessible to service-receivers. The publication of services is done through local method for the users inside the organization and through the internet for public users who are not inside the organization.

**Service Contract:**

To validate the service, a contract between publishers and service-receivers is signed that refers to three principles of reliability, security, and efficiency.

**Reliability:**

Measurable standards must be provided for each service to measure its reliability. 99 percent accessibility, for example, is one of the criteria appropriate for this end. Another criterion is guarantee of message delivery altogether.
Security:
Authentication and authorization are two major criteria in service security. Authentication is to examine the authenticity of the agent who requests the service and is done usually through controlling usernames and passwords. Authorization controls the levels of access and the authorities that the authorized agent might have.

Efficiency:
One of the criteria for gauging this, which has a high importance, is the answer time. This latter refers to the time passed after a service-provider responds to a request and is normally calculated in terms of seconds or milliseconds. One other criterion is operational capability which is the number of responses a service provider gives in a specific time like 100 responses in a second.

Solutions Offered For Service-Orientation:
- Persistent communication methods like RPC must be used in communicative protocols, that is, methods based on message must replace transient methods.
- As for security mechanisms like authentication and access control, platform-independent methods must be used to maximize the communicative ability and interaction between services.
- The service interfaces must observe the principle of well formed as far as possible through WSDL to make the service available to the beneficiaries and service-users.
- In an organization, information of the operation process must be kept on service-receiver or customer side in order to minimize the storage volume of the information and to enhance the speed of the organization.
- Platforms capable of supporting various versions of a service must be used in order to utilize the version that best suits the service if necessary.

Conclusion:
In this paper we express SOA principles and services properties that implemented by web services and SOA. Then we classify current service orientation and web service application challenge in SOA. For solving these challenges, we present new method with 5 phases and describe benefits of usage. We use message passing for communication methods instead of transient methods. For reducing volume of stored information in organization, we present process status for operation kept in client side. We use platform independent methods for security mechanisms that maximize communication and minimize delay.

REFERENCES
Alcedo Coenen (sidebar by Dr. Chris Harding), 2006. SOA Magazine Issue II: Copyright, November December 2006.