

## Pattern Authoring: Web Service Oriented Architecture

Nilesh Vishwasrao Patil<sup>Å\*</sup>

<sup>Å</sup>Vishwbharati Academy College of Engineering, Ahmednagar, Pune University, India

Accepted 05 Sept 2014, Available online 01 Oct 2014, Vol.4, No.5 (Oct 2014)

### Abstract

Architecture to loosely coupled architecture. Reusability has been playing the important role in software process and tools development since last decade. Software industry is to achieve intense perfection in throughput and superiority because of reusability in development process. In this paper, I will going to discuss the guideline to pattern authoring for web services with loosely coupled architecture. This pattern will provide framework for creation of web service with highest amount of reusability.

**Keywords:** SOA: Service Oriented Architecture, WS: Web services, WSDL: Web Service Description/Definition Language, W-SOA: Web Service Oriented Architecture, ESB: Enterprise Service Bus, MQ: Messaging Queue.

### 1. Introduction

Now a day the IT industries is undergoing in significant changes because of the development of new technologies. The Software industries is almost moving from tightly coupled architecture to loosely coupled architecture because of there are a lot of advantages of loosely coupled architecture over tightly coupled architecture. Every electronic business is wants to developed software/systems within time with extra cost benefit and efficient. To achieve such functionality we need to create some general patterns for each time occurring problem.

Reusability is the key to growth in any area including or excluding software field. Reusability means previously developed ideas, systems, software, tools and products reuse for avoid to implementing everything from scratch. Development of software, reuse the knowledge in the form of experience, processes in the form of methods, and product in the form of tools . To achieve systematic way of reusing the existing products, systems, processes and knowledge in electronic business to get maximum cost benefit, time benefit, standard and efficient template.

To reuse the existing products, processes, software's, tools, knowledge etc. we need to find out reusable components which will occurs continuously so that we can reuse those particular components in future for same type of problems. To find reusable components, we should know everything (in and out) about the system and process of implementation.

### 2. Background

Following are some important checkpoint requires designing pattern for web service oriented architecture, we should know the background information of this:

### Service Oriented Architecture

Service Oriented Architecture is collections of services that are able to communicate with each other. In this everything has been done through services. A Service is independent, self-contained “unit of functionality” that will has organized, deployed, published and invoked over the set-up, cluster, web, grid, network etc. once service has been deployed then we can say to that service as interface. This interface can be a web service or MQ (Messaging Queue) series based to integrate systems disparate systems by achieving tightly coupled architecture. SOA is an approach through which integrate enterprise applications by achieving loosely-coupled interfaces/services.

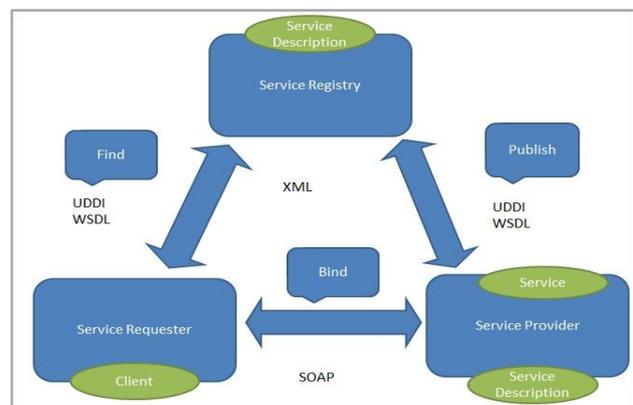


Fig. 1 Service Oriented Architecture

It is considered to be the new phase in the evolution of distributed enterprise applications. In our more and more modest commercial world, in which business needs and enterprises are shifting continually, the quick progress of business applications is a key dynamic in the achievement

\*Corresponding author Nilesh Vishwasrao Patil is a ME student

of a business. SOA has promised to provide enterprises with flexible and extensible architectures that would enable them to adapt their applications easily so that they remain competitive and compliant. There are many of the major software sellers sell Service Oriented Architecture products such as Enterprise Service Buses (ESBs). Examples: IBM Web Sphere Message Broker, Oracle Fusion, Software AG Web Methods, Mule ESB etc.

Even though there is a common acceptance of this concept, a real problem hinders the widespread use of SOA: A methodology to design and build secure SOA-based applications is still needed. Communications among services can involve only simple invocations and data passing or complex coordinated activities of two or more services. It has also provides characteristics like Abstraction, Encapsulation, Reusability, Loosely coupled architecture, integrate disparate systems, Discoverability etc. The main implementation platform for Service Oriented Architecture is web services and messaging queue based.

### Web Services

A Web service is a process of communication between two electronic devices over a network. It is a software function provided at a network address over the web with the service always on as in the concept of utility computing. Web service is a kind of service which is language independent so that it is interoperable between different communications protocol, different operating systems, heterogeneous systems and various programming languages. Now a days web services has been playing the important role in current electronic business Web service is also plays the important role in Enterprise Application Integration.

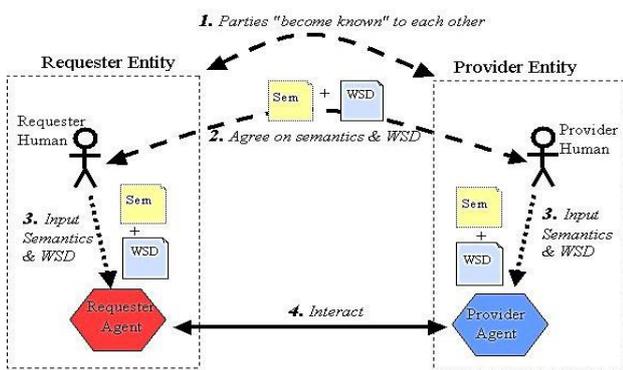


Fig. 2 Web Services

For implementing web service we need web service description/definition language and referred as WSDL. The sample WSDL is as shown in the following fig 3.

### Pattern

Pattern-based approach allows a wide, platform independent view on Service Oriented Architecture that still contains all related information about the technical comprehension replacements. Software pattern provide reusable solutions to recurring design problems in the

specific context. Pattern is becoming part of Object Oriented software development. Pattern plays is very important role in reusability. It will provide general template to continuously occurring problem to get cost and time benefit.

```
<?xml version="1.0"?>
<definitions name="StockQuote"

targetNamespace="http://example.com/stockquote/service"
  xmlns:tns="http://example.com/stockquote/service"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:defs="http://example.com/stockquote/definitions"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

  <import namespace="http://example.com/stockquote/definitions"
    location="http://example.com/stockquote/stockquote.wsdl"/>

  <binding name="StockQuoteSoapBinding" type="defs:StockQuotePortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="GetLastTradePrice">
      <soap:operation soapAction="http://example.com/GetLastTradePrice"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>

  <service name="StockQuoteService">
    <documentation>My first service</documentation>
    <port name="StockQuotePort" binding="tns:StockQuoteBinding">
      <soap:address location="http://example.com/stockquote"/>
    </port>
  </service>
</definitions>
```

Fig. 3 Sample WSDL

A pattern is a reusable solution to a common problem within a given context. It is vital for clients to generate instance of created patterns for complex problems, so that they can reuse that pattern instance in a different scenario simply by altering some of the configuration values. A pattern has been provides a recyclable key to a common problem which encapsulates a tested approach to solving a common architecture, design, or deployment task in a particular context. When we choose to use a pattern instance, most of the design, configuration and development work have already been completed for us, and we will require less development, testing, and operational efforts to saves time and money.

A pattern is similar like a template in C++ but it is also somewhat different than template.

### 3. Pattern for Web Service Oriented Architecture

As we have seen above about Service Oriented Architecture, Web Service and Pattern in background topic. Pattern is not the exactly solution to given problem, its need some changes in the configuration values and business logic as per requirements. For this I am going to give one real life example, in my class there are eighteen students and for wishing the birthday of each students our college use same content of information by changing the name of person and date of birth. A pattern is also same

reusable solution to continuously occurring problems by changing some configuration values so that our time, efforts, money will be save and addition to this we will get standard, no inconsistency in the solution. Another example is Mail merge in Microsoft word we just need to change the address of the sender.

In this paper we are going to discuss about the web service oriented architecture (W-SOA) pattern creation. We have to follow best practices and standard to create pattern as efficient as possible because each pattern instances will be create from this. When we have to create the web service, exposed the process, program, flow or interface as service by using loosely coupled architecture that means service oriented approach. The abstract view of creating pattern as shown in following fig. 4

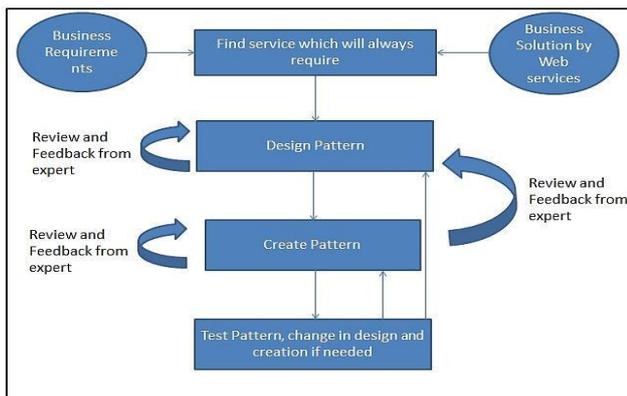


Fig. 4 Abstract view before creation of pattern

In this paper we are going to implement the pattern for web service oriented architecture (W-SOA) so that we should be keep an eye on the following checklist first and the go in detail.

- Analyze electronic business requirements.
- It may have multiple modules or interfaces which we may have exposed as service
- Find the services which will continuously require
- Design pattern for that web service by involving experienced and expert persons.
- Create Pattern for web service by involving experienced and expert person
- Test pattern, change design and implementation of pattern if not work desired function
- Finally create a instance for that pattern

We must have to follow above checklist for creation of web service pattern based on loosely coupled architecture. To creation of web service pattern catch the service which will always require in future.

The pattern creation is the generic template, if we create pattern instance need to change configuration or basic values and business logic by providing WSDL file. This can be implementing using any object oriented programming language or using any integration tools based on service oriented architecture called Enterprise Service Bus (ESB).

Enterprise Service Bus (ESB) is flexible connectivity infrastructure for integrating applications and services. ESB will power our Service Oriented Architecture by

reducing the complexity, number, configuration and size of interfaces. An ESB has done following functionalities between requester and service: Routing, converting, transforming and handling (business events from heterogeneous sources). ESB will help to achieve all goals of Service oriented architecture. There are many ESB is available in market by various vendors such as IBM Web Sphere Message Broker, Mule ESB (Open source), and Oracle Fusion, Software AG web methods and so on. For creation of web service based on pattern using ESB, it's our responsibility to select ESB as per our business need because each ESB has advantages and disadvantages points.

We can use Java, .NET or any ESB to create pattern for web service oriented architecture (W-SOA), so that create GUI (Graphical User Interface) which will takes all basic or configuration values require for web service. Once provided all configuration values create instance of pattern and pattern will generate solution for your problem. The configuration or basic values can be WSDL, type of web service; security features etc. as per the business requirements. The following fig 5 shows sample GUI for web service pattern.



Fig. 5 Sample GUI for web service pattern

In above sample GUI for web service pattern fig.5, the developer first needs to select the web service description language (it includes port type, operations, URL for service etc.) file which defines the web service, then select type of service it depends on business type. The types of service can be simple service or the service which call another web service etc. The last option is security as per our requirement provide security feature to web service (WS-security). After providing above configuration values, then click on create instance, it will generate web service developed through pattern. Now you will get template to create web service, change the business logic and you will get what you expect. The web service pattern will save your time and money for creation of web services based on loosely coupled architecture.

**Conclusion**

In this paper we have seen how a pattern is a reusable solution to a common problem within a given context. It is key point for the clients to generate instance from pattern

for the complex problems. They can reuse the pattern instance in a different scenario by simply changing the business logic and the configuration or basic values. The pattern is provided framework for implement the web service with highest amount of reusability which will save time and money.

### Acknowledgment

This research paper has completed only because support from each and every one including: Government Polytechnic, Ahmednagar, teachers, colleague, parents, friends, students and ME-II classmate. Especially, my acknowledgment of gratitude toward the following important persons: First, I would like to thanks to Government Polytechnic, Ahmednagar, MSBTE, Mumbai and DTE, Maharashtra for motivate me.

Second I would like to thank Dr. Kureshi Sir Principal, Mrs. Joshi Madam Head of Computer Engineering, Mr. M. C. Kshirsagar Sir, Mr. Prabhudev sir, Mr. Natikar sir, and Mr. Jaypal sir Vishwbharati Academy College of Engineering, Ahmednagar to their support and encouragement

Third, I would like to thank Mr. Prafulla Gavde, Mr. Prasad Deshpande, Mr. Raviraj Pawar, Mr. Prashant Badgular, Mr. Prshant Sarwade, Mr. Dattu Ghane, Mr. Dadasaheb Karanjule and Mr. Prashant Chouke, to provide valuable advices.

Last but not least, I sincerely thank to my parents and S. A. Bhalerao who provide the advice and financial support. This research paper will not be possible without all of them.

### References

- Victor R. Basili et al., Software Reuse: A Framework for Research, unpublished.
- Jitendra Joshi, Nisha Singh and Manjari Kumara. (2012). Web Service oriented Architecture Modeling with Pattern for Electronic Business Organization. IJARCSSE Research Paper [Online]. 2(9), pp. 456-460. Available [www.ijarcsse.com/docs/papers/9\\_September2012/.../V2I900179.pdf](http://www.ijarcsse.com/docs/papers/9_September2012/.../V2I900179.pdf)
- [http://www.ibm.com/developerworks/websphere/library/techarticles/1010\\_stewart/1010\\_stewart.html](http://www.ibm.com/developerworks/websphere/library/techarticles/1010_stewart/1010_stewart.html)

### Author Profile



**Nilesh Vishwasrao Patil** is pursuing Master of Engineering with specialization in Computer Engineering from Vishwbharti Academy College of Engineering Ahmednagar. He has around five years of experience, in which around two years of industrial experience as Enterprise Application Integration (EAI) developer. Now he is working as System Analyst in Government Polytechnic, Ahmednagar since November 2011.