



Plug-In Enabling SOAP to Wsdl Conversion

Megala Kandasamy1 , Sindhuja M2

2 Asst Prof NEW Prince Shri Bhavani College of Engineering & Technology

megalakandasamy@gmail.com1 sindhumano12@gmail.com2

ABSTRACT: In computing, a plug-in is a set of software components that adds specific capabilities to a larger software application. This paper presents the idea of an Eclipse Plug-in which will enable the conversion from SOAP Web service to a WSDLful Web service. The plug-in will use Eclipse PDE. This plug-in will take WSDL File as its input and provide the user a WSDLful template on Spring MVC Framework.

KEYWORDS . Eclipse, SOAP, WSDL, L, MVC

I. INTRODUCTION

A Web service is a method of communication between two electronic devices over the World Wide Web. Web services are used to request for and access infrastructure services in the cloud. Today, IT business organizations are highly dependent on Web services for achieving a rich set of features and functionality in their respective projects. SOAP, till now has been a highly used protocol for creating Web services on the network. However, the programmers and developers currently working in SOAP domain are facing the problem of scalability, access to external systems, independent installable components and composition of services.

These problems are eliminated in WSDL to a great extent. Hence, there is a need of a system/tool which would successfully convert existing SOAP services to WSDLful services.

A. BASIC CONCEPTS .

ECLIPSE :

The Eclipse is an Integrated Development Environment (IDE). It is an IDE for Java Developers contains what you need to build Java applications. It is considered by many to be



the best Java development tool available. The Eclipse IDE for Java Developers provides superior Java editing with validation, incremental compilation, cross-referencing, code assist; an XML Editor; Mylyn; and much more.

Eclipse also provides an extensible plug-in system i.e. Plug-in development environment (PDE). It uses plug-ins to provide all functionality within and on top of the runtime system.

SOAP :

SOAP means Simple Object Access Protocol. It is a protocol specification for exchanging structured information in the implementation of Web Services. SOAP is a simple XML-based protocol to let applications exchange information over HTTP. SOAP can form the foundation layer of a web services protocol stack, providing a basic messaging framework upon which web services can be built. This XML based protocol consists of three parts: an envelope, which defines what is in the message and how to process it, a set of encoding rules for expressing instances of application- defined data types, and a convention for representing procedure calls and responses.

SOAP has three major characteristics:

1. Extensibility (security and WS-routing are among the extensions under development),
2. Neutrality (SOAP can be used over any transport protocol such as HTTP, SMTP, TCP, or JMS) and
3. Independence (SOAP allows for any programming model). WSDL :

WSDL means REpresentational State Transfer. The REpresentational State Transfer architecture is an architecture that describes how the Web should work. WSDL defines a set of architectural principles by which you can design Web services that focus on a system's resources, including how resource states are addressed and transferred over HTTP by a wide range of clients written in different languages. If measured by the number of Web services that use it, WSDL has emerged in the last few years alone as a predominant Web service design model. In fact, WSDL has had such a large impact on the Web that it has mostly displaced SOAP- and WSDL-based interface design because it's a considerably simpler style to use.

One of the key characteristics of a WSDLful Web service is the explicit use of HTTP methods in a way that follows the protocol as defined by RFC 2616. WSDL uses HTTP GET,PUT,POST and DELETE methods for handling client requests.



WSDL :

types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the operations listed in the WSDL file using XML or HTTP.

WSDL is Web Services Description Language. It is

Standard WSDL document structure is composed of :

iii) Controller : The controller interprets the mouse and keyboard inputs from the user, informing the model and/or the view to change as appropriate.

<Types>: A container for data type definitions used by the web service. This element defines the data types that are used by the web service.

an XML-based language that is used for describing the functionality

<Message>: A typed definition of the data being communicated. Each message can consist of one or more parts. The parts can be compared to the parameters of a function call in a traditional programming language.

offered by a Web service. A WSDL description of a web service (also referred to as a WSDL file) provides a machine-readable description of how the service can be called, what parameters it expects, and what data structures it returns. It thus serves a roughly similar purpose as a method signature in a programming language.

<portType>: A set of operations supported by one or more endpoints. This element can be compared to a function library (or a module, or a class) in a traditional programming language.

WSDL is often used in combination with SOAP and an XML Schema to provide Web services over the Internet. A client program connecting to a Web service can read the WSDL file to determine what operations are available on the server. Any special data

<binding>: A protocol and data format specification for a particular port type. This element defines the data format and protocol for each port type.



MVC :

Flexibility in large component based systems raise questions on how to organize a project for easy development and maintenance while protecting your data and reputation, especially from new developers and unwitting users. The answer is in using the MVC architecture. The Model-View-Controller (MVC) is a software architecture pattern that separates the representation of information from the user's interaction with it.

i) Model : The model manages the behaviour and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller).

ii) View : The view manages the display of information.

Today, many MVC frameworks exist viz. Spring, Struts, Grails etc. which can be used to develop variety of applications.

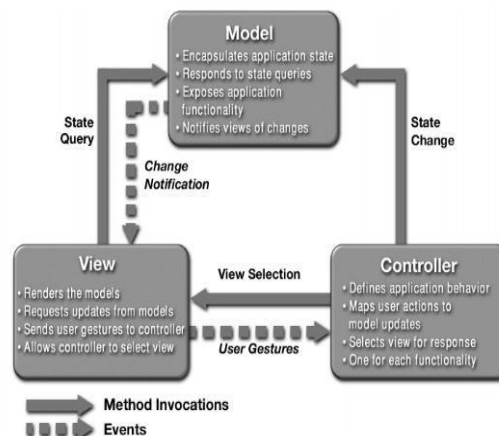


Fig.1 MVC Framework

B. SOAP Versus WSDL

Many discussions of SOAP versus WSDL focus on the point that encoding services as SOAP/WSDL makes it difficult to expose the semantics of a web service in order for it to be easily and widely understood, so that many different providers can potentially offer the same services.

In Table 1.1 we compare these along six dimensions:

The location where servers providing the service can reside; how secure the interaction is; whether transactions can be supported; how dependent the protocol is on HTTP technology; the extent of development tools and support



required; the efficiency of the resulting implementations; and finally the software development productivity to be expected using each. We conclude from this analysis that for most requirements SOAP is an overkill; WSDL interfaces are simpler, more efficient and cheaper to develop and maintain.

Table 1.1 SOAP vs WSDL

Basically, WSDL and SOAP are two architecturally different technologies. The problem of SOAP to WSDL conversion can be better understood by being aware of the real time drawbacks of SOAP protocol which can be listed as follows:

SOAP messages are always addressed to an endpoint, which is implemented by a SOAP router, also called dispatcher.

	augmented with additional security layers		can be addressed only by SOAP
Efficiency	XML parsing required	XML parsing can be avoided by using JSON	WSDL is lighter and more efficient
Transaction	Can be supported	No support	Situations requiring complex multi-request/multi-party transactions need SOAP
Technology	Can work without HTTP, e.g. using message queues instead	Relies on HTTP	WSDL is for pure internet communications and cannot mix other transports
Tools	Sophisticated tools required (and are available) to handle client and server development	No special tools required if using JSON	WSDL is lighter and easier to use
Productivity	Low, due to complex tools and skills needed	High due to simplicity	WSDL is faster and cheaper

	SOAP/WSDL	WSDL	Comments
Location	Some endpoints can be behind corporate networks on non-HTTP connects, e.g. message queues	All endpoints must be on the internet	Complex B2B scenarios require SOAP
Security	HTTPS which can be	Only HTTPS	Very stringent security needs

SOAP is a protocol construction set used to write own application protocols. The protocol describes the structure of the request and the reply. The description of the request and the answer represents a rigid framework that cannot be easily changed. In SOAP the possibility for the gradual evolution of existing Web services is limited.



In SOAP, the letter baskets of the co-workers cannot be addressed directly.

SOAP addresses a service and not a resource. Moreover, it does not use global address space.

SOAP requires greater implementation effort and understanding on the client side.

SOAP requests use POST and require a complex XML request to be created which makes response-caching difficult.

The drawbacks of SOAP mentioned above are eliminated in WSDL

due to the following advantages:

There are many Application development MVC Frameworks for various programming languages such as .NET, Java, PHP, ASP.NET etc. Grails, Apache Struts, Stripes and Spring are some of the Java MVC frameworks. There are two broad classifications to be aware of viz. Server-side and Client-side frameworks.

MVC frameworks are typically the most popular frameworks in the Java web development world and are structured around the framework of a re-usable web application.

Struts 2 is a Configuration framework because if you look at the number of XML artifacts that you have to create to configure the framework (approximately 2-7 files) it's very configuration-heavy. Stripes, however, is on the opposite end of the spectrum. It professes that it's a very Convention over Configuration framework. It leverages not only coding conventions but also class reflection to help it understand what the different components are of the application. There is only one configuration file: web.xml. In the middle is Spring MVC, which started off as a Configuration framework but with every release it has moved more and more towards Convention. It currently has probably just two configuration, and in Spring 3 it is supposed to be even less.

In the real world, there are many reasons why people adopt WSDLful URLs. The most prominent reason is search engine optimization (SEO) because it's theoretically easier for a search engine to traverse your website and content and understand the categorization and hierarchy. So, a framework that helps create WSDLful URLs is an immediate benefit to sales and marketing. Will the framework generate the client-side validation? Will it generate the JavaScript out of the developer's view? Will it adhere to the validation that the developer defined? As a Java developer, you don't want to fight with a bunch of different JavaScript variations. So, what kind of JavaScript help is included? This dovetails



into Asynchronous JavaScript + XML (AJAX) or background processing within the web. Spring provides complete support to develop WSDLful applications. Hence, we will be using Spring Framework to develop our plug-in.

Spring's primary features are dependency injection (DI) and aspect-oriented programming (AOP). DI makes it possible to tie software components together loosely. Aspect-oriented programming is often defined as a technique that promotes separation of concerns within a software system. Aspect-oriented programming enables you to capture functionality that's used throughout your application in reusable components.

i) Spring Work-Flow :

Spring's Web MVC framework is designed around a DispatcherServlet that dispatches requests to handlers, with configurable handler mappings, view resolution, locale and theme resolution as well as support for upload files. The DispatcherServlet is an actual Servlet (it inherits from the HttpServlet base class), and as such is declared in the web.xml of your web application. Requests that you want the DispatcherServlet to handle will have to be mapped using a URL mapping in the same web.xml file. The default handler is a very

simple Controller interface, just offering a ModelAndViewhandleRequest(request, response) method.

Following fig.2 shows Spring's work-flow.

The Client requests for a Resource in the Web Application.

The Spring Front Controller, which is implemented as a Servlet, will intercept the Request and then will try to find out the appropriate Handler Mappings.

The Handle Mappings is used to map a request from the Client to its Controller object by browsing over the various Controllers defined in the Configuration file.

With the help of Handler Adapters, the Dispatcher Servlet will dispatch the Request to the Controller.

The Controller processes the Client Request and returns the Model and the View in the form of ModelAndView object back to the Front Controller.

The Front Controller then tries to resolve the actual View (which may be Jsp, Velocity or Free marker) by consulting the View Resolver object.

Then the selected View is rendered back to the Client.

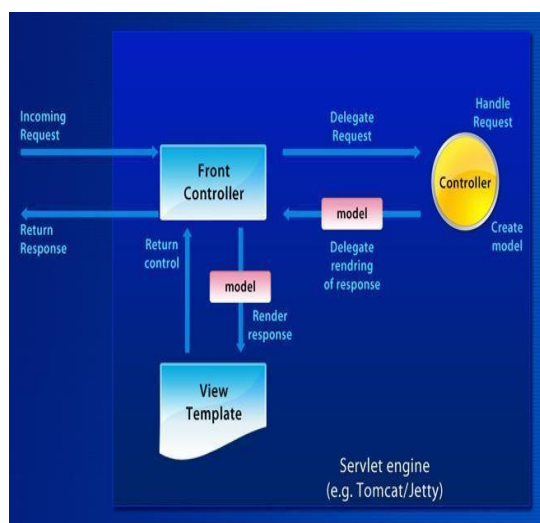


Fig. 2 Spring MVC

Proposed System Architecture : Fig.3 shows the system architecture.

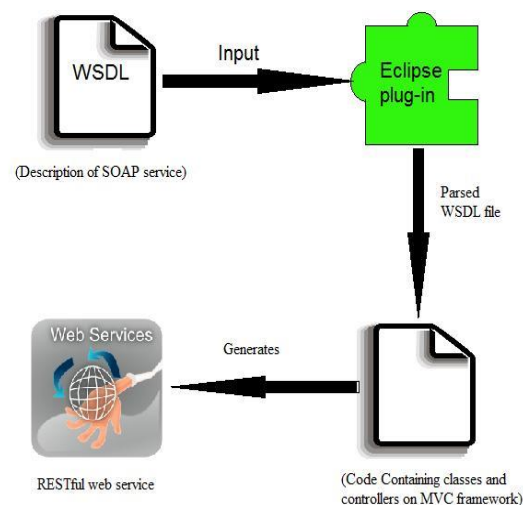


Fig. 3 System Architecture

WSDL is an XML document. XML makes it possible for developers to expose valuable resources in a highly interoperable fashion, where a resource is any type of application or data store used within an organization. The XML Web services architecture defines a standard mechanism for making resources available via XML messaging. Being able to access a resource by simply transmitting XML messages over standard protocols like TCP, HTTP, or SMTP greatly lowers the bar for potential consumers. This architecture

makes it possible for any consumer with XML support to integrate with Web service applications. The XML schema definition simply



tells you what XML messages may be used but not how they relate to each other.

A message exchange is also referred to as an operation. Consumers must be aware of these operation groupings on a particular portType, since it impacts the way they write their code. Consumers must also know what communication protocol to use for sending messages to the service. A binding specifies the concrete details of what goes on the wire by outlining how to use an interface with a particular communication protocol.

The Web Services Description Language (WSDL) provides an XML

grammar for describing these details. WSDL picks up where XML Schema left off by providing a way to group messages into operations and operations into interfaces.

WSDL plays an important role in the overall Web services architecture since it describes the complete contract for application communication. WSDL definitions are used to generate code that knows precisely how to interact with the Web service it describes.

The WSDL file, which is an input to the Eclipse Plug-in, is parsed to give an object that contains necessary description that can actually be mapped into appropriate java classes and methods. This can be achieved by modifying & using traditional SAXParser or DOM Parser,

depending on our requirements, to handle WSDL elements.

While the WSDL file is being parsed, the plugin creates a Dynamic Web Project which has MVC support, in this case, Spring. This Web Project then consumes the parsed output object and maps it into appropriate java classes, methods & data types using

<portType>, <message> & <Types> of input WSDL file respectively responsible for composing a WSDLful service. The user can modify the contents of this generated java code as per his requirements to enhance its features.

Thus the output of the Eclipse Plug-in is a web project that contains the appropriate mappings from SOAP to WSDL on spring MVC framework.

II. FUTURE ENHANCEMENT

This plugin would be particularly beneficial for companies that work in SOAP and are willing to move to WSDL. Rich-client applications are enabled in this Eclipse plugin. Traditionally, the Eclipse platform is designed to serve as an open tools platform. However, it is architected so its components could be used to build just about any client application.



The output of the primary eclipse plugin would be a project in an MVC framework with all the methods in the WSDL file mapped to appropriate URLs and methods in appropriate controllers.

The template is based on an MVC architecture and thus, it can be extended and updated to any other MVC framework as and when required, providing flexibility. There are various MVC frameworks in java available in market such as Struts2, Spring, JSF, Wicket, Stripes etc. The future scope involves targeting other frameworks than the one in which the project has been implemented.

III. CONCLUSION

The basic functionality and feature of this concept of Eclipse Plugin is to convert SOAP services into WSDL services that can be used in WSDLful applications. The plugin takes whole WSDL file as an input which describes the functionality offered by SOAP service to be converted to WSDL. Hence the complicated steps of mapping each module of SOAP to corresponding code are eliminated. .

WSDL services are Lightweight. In WSDLful interface, the requests and responses can be short. SOAP requires an xml wrapper around every request and response. The prime benefit of conversion to WSDL service is API flexibility

and simplicity. By converting SOAP service to corresponding WSDL service, the user can utilize all the advantages of WSDL services such as scalability, access to external systems, Independent Installable Components, Human readable format of documents & Composition of services. A detailed and thorough knowledge of concepts like SOAP, WSDL, WSDL & MVC frameworks is required in order to implement this idea of an Eclipse Plugin.

REFERENCES

*1+ "Enterprise Cloud Computing", by Gautam Shroff.

*2+ "Spring in Action", third edition, by Craig Walls. ISBN 9781935182351

[3] <http://www.eclipse.org/>

[4] <http://www.w3schools.com/soap/default.asp>

[5] <http://www.ibm.com/developerworks/webservices/library/ws-restful/>

[6] <http://msdn.microsoft.com/en-us/library/ff649643.aspx>



[7]
<http://static.springsource.org/spring/docs/2.O.x/reference/mvc.html>

[8] <http://tomcat.apache.org/>

[9].Anisetti, M.; Ardagna, C.; Damiani, E. Fine-grained modeling of web services for test-based security certification. In Proceedings of the 8th IEEE International Conference on Services Computing, Washington, DC, USA, 5–10 July 2011; pp. 456–463.

[10]Frantzen, L.; Tretmans, J.; d. Vries, R. Towards model-based testing of web services. In Proceedings of the International Workshop on Web Services—Modeling and Testing, Palermo, Italy, 6 June, 2006; pp. 67–82.