International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 1
ISSN: 2454-4248
149 – 155

Survey on Choreography for Web Services

[1] M. Mohana Devi,
M.C.A., M.Phil.,
Department of Computer Science,
Selvamm Arts and Science College (Autonomous)
Namakkal (Tk) (Dt) – 637003.

[2] Mrs. D. Ananthanayaki,
M.C.A., M.Phil., Assistant Professor, Department of Computer Science,Selvamm Arts and Science College (Autonomous)
Namakkal (Tk) (Dt) – 637003.

[3]Mrs. K.K.Kavitha,
M.C.A., M.Phil., SET., (Ph.D).,
Vice Principal, Head of the Department of Computer Science,
Selvamm Arts and Science College (Autonomous)
Namakkal (Tk) (Dt) – 637003.

Abstract:- Web service choreography is used to interchange the message between source and destination. The transfer message is based on the data type and transmission sequence. For many years, organizations have been developing solutions for automating their peer-to-peer collaborations, within or across their trusted domain, in an effort to improve productivity and reduce operating costs.

Web Services are a key component of the emerging, loosely coupled, Web-based computing architecture. A Web Service is an autonomous, standards-based component whose public interfaces is defined and described using XML. Other systems may interact with a Web Service in a manner prescribed by its definition, using XML based messages conveyed by Internet protocols.

Keywords – *Message, Peer-to-peer, domain, XML, Web service, Protocols.*

*****

I. INTRODUCTION TO WEB SERVICES

A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-process able format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

Web Services represents a new architectural paradigm for web application. Web services implement capabilities that are available to other applications via industry standard network and application interface and protocols. An application can use the capabilities of Web Services by simply invoking it across a network with having to integrate it.

Service oriented architecture (SOA) is a component model. The different functional units of applications called services were linked through well-defined interfaces and contracts. It is an independent of the implementation services, hardware platforms, operating systems and programming languages.

A web service is a software module which is designed to perform a certain set of tasks.

- The web services can be searched for over the network and can also be invoked accordingly.
- When invoked the web service would be able to provide functionality to the client which invokes that web service.

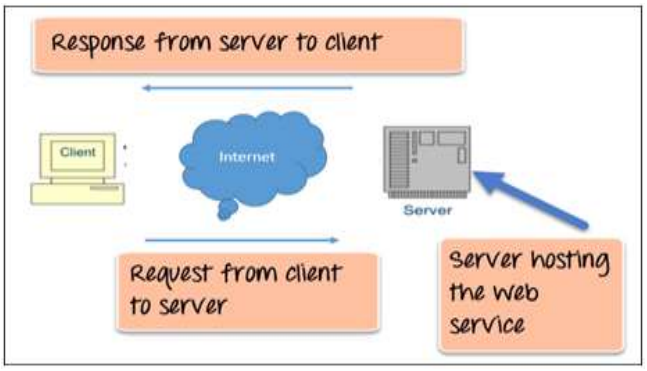

Fig.1.1 Web Services

These requests are made through what is known as remote procedure calls. Remote Procedure Calls(RPC) is calls made to methods which are hosted by the relevant web service.

Web Service Components

In order for a web service to be fully functional, there are certain components that need to be in place. These components need to be present irrespective of whatever development language is used for programming the web service. Let's look at these components in more detail.

SOAP (Simple Object Access Protocol)

 ISSN: 2454-4248
149 – 155

SOAP is known as a transport-independent messaging protocol. SOAP is based on transferring XML data as SOAP Messages. Each message has something which is known as an XML document.

Here is what a SOAP message consists of

- Each SOAP document needs to have a root element known as the <Envelope> element. The root element is the first element in an XML document.
- The "envelope" is in turn divided into 2 parts. The first is the header and the next is the body.
- The header contains the routing data which is basically the information which tells the XML document to which client it needs to be sent to.
- The body will contain the actual message.

The diagram below shows a simple example of the communication via SOAP.

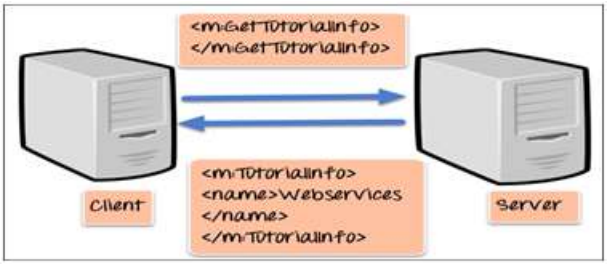

Fig.1.2.1 Simple Object Access Protocol

WSDL (Web Services Description Language)

A web service cannot be used if it cannot be found. The client invoking the web service should know where the web service actually resides. Secondly, the client application needs to know what the web service actually does, so that it can invoke the right web service. This is done with the help of the WSDL, known as the Web services description language.

The WSDL file is again an XML-based file which basically tells the client application what the web service does. By using the WSDL document, the client application would be able to understand where the web service is located and how it can be utilized.

The important aspects to note about the above WSDL declaration are as follows;

1. **<Message>** - The message parameter in the WSDL definition is used to define the different data elements for each operation performed by the web service. So in the example above, we have 2 messages which can be exchanged between the web service and the client application, one is the "Tutorial Request", and the other is the "Tutorial Response" operation. The Tutorial Request contains an element called "Tutorial" which is of the type string. Similarly, the Tutorial Response operation contains an element called "Tutorial Name" which is also a type string.

2. **<Port Type>** - This actually describes the operation which can be performed by the web service, which in our case is called Tutorial. This operation can take 2 messages; one is an input message, and the other is the output message.

3. **<Binding>** - This element contains the protocol which is used. So in our case, we are defining it to use http (**http://schemas.xmlsoap.org/soap/http**). We also specify other details for the body of the operation, like the namespace and whether the message should be encoded.

UDDI (Universal Description, Discovery and Integration)

UDDI is a standard for describing, publishing, and discovering the web services that is provided by a particular service provider. It provides a specification which helps in hosting the information on web services.

It defines a means to publish and, more importantly, discover (or search for) information about Web services, including WSDL files. After browsing through an UDDI registry for information about available web services, the WSDL for the selected services can be parsed, and an appropriate SOAP message can be sent to the service.

Web Service Architecture

Every framework needs some sort of architecture to make sure the entire framework works as desired. Similarly, in web services, there is an architecture which consists of three distinct roles as given below

1. **Provider** - The provider creates the web service and makes it available to client application who wants to use it.

2. **Requestor** - A requestor is nothing but the client application that needs to contact a web service. The client application can be a .Net, Java, or any other language based application which looks for some sort of functionality via a web service.

 ISSN: 2454-4248
149 – 155

3. **Broker** - The broker is nothing but the application which provides access to the UDDI. The UDDI, as discussed in the earlier topic enables the client application to locate the web service.

Fig.1.3 Web Service Architecture

The above diagram showcases how the Service provider, the Service requestor and Service registry interact with each other.

1. **Publish** - A provider informs the broker (service registry) about the existence of the web service by using the broker's publish interface to make the service accessible to clients

2. **Find** - The requestor consults the broker to locate a published web service

3. **Bind** - With the information it gained from the broker(service registry) about the web service, the requestor is able to bind, or invoke, the web service.

Purpose of WS-CDL

Business or other activities that involve different organizations or independent processes are engaged in a collaborative fashion to achieve a common business goal, such as Order Fulfillment.

For the collaboration to work successfully, the rules of engagement between all the interacting participants must be provided. Whereas today these rules are frequently written in English, a standardized way for precisely defining these interactions, leaving unambiguous documentation of the participants and responsibilities of each, is missing.

The figure below demonstrates a possible usage of WS-CDL.

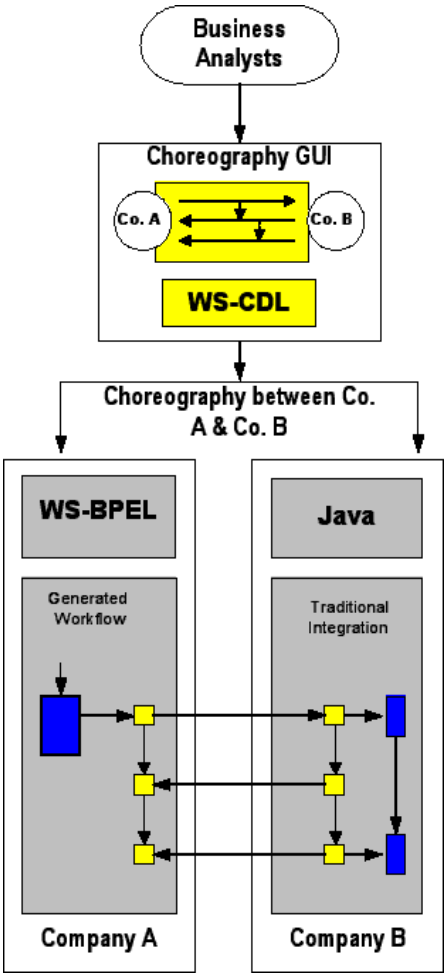

Fig.1.6 Usage of WS-CDL

- Company "A" relies on a WS-BPEL solution to implement its own part of the choreography

- Company "B", having greater legacy driven integration needs, relies on a J2EE solution incorporating Java and Enterprise Java Bean Components or a .NET solution incorporating C# to implement its own part of the choreography

Similarly, choreography can specify the interoperability and interactions between services within one business entity.

II. REVIEW OF LITERATURE

Service choreographies are service compositions that implement distributed business processes without need for centralized coordinator, thus reducing the number of exchanged control messages and simplifying the distribution logic. Choreography composes services in an abstract way by means of expected role that each service plays in the composition. Building the choreography is usually two step

 ISSN: 2454-4248
149 – 155

process. Service choreography is a description of the peer to peer externally observable interactions that exist between the services; therefore, choreography does not rely on a central coordinator. A choreography model describes multiparty collaboration and focuses on message exchange; each web service in a choreography knows exactly when to executes its operation and with whom to interact.

Adam et al. [1] introduced the Multi Agent Protocols (MAP) choreography language for implementing the choreography interconnection models. This choreography language based on formal foundation and π-calculus. It demonstrated how choreographies service can be specified, verified and enacted over a distributed peer-to-peer network. The methodology is detailed in the following sections and implements the motivating RedShift scenario. The RedShift scenario is taken from the AstroGrid science use-cases and involves retrieving and anlaysing large-scale data from the multiple distributed resources.

Zahra et al. [2] discussed the web service composition mechanisms and related formal methods and verification issues. To design the composition of services by using Web Service Choreography Definition Language(WS-CDL). Here we present a method using first order logic notation for partial order planning problems. This method can be used for interactive systems that all participants having common understanding of interaction rules. It consists of three parts precondition, action and effect. The main specifications of choreography that represent its behavior such as, activities and ordering structures, interaction activity, channel type, variables.

III. PROPPOSED ARCHITECTURE

Introduction

This chapter presents and discusses the architecture for the web service choreography based on meta search engine developed for the betterment of the data access, data accuracy, and to increase the time efficiency in retrieval of data. It explains the retrieval of data from various search engines simultaneously by giving a single keyword and role of the administrator for updating the new search engine.

Proposed architecture for web service choreography

In the existing architecture, the merging the results retrieved is fully depends only on the page rank algorithm. The architecture for the proposed system provides the user a wide range of accessing the search results effectively from various search engines simultaneously. The choreography architecture elements are easily connected to simplify the accessing time in different search engines and to produce effective results.

Web service is key area that makes it available on internet and uses a standardized XML messaging system. XML is used to encode all communications to a web service. There have been many concepts in XML that focus on the efficient communications under content-aware web services architecture.

The proposed architecture for service selection through multiple choreographies provides an abstraction of the details of the service selection and adaption. The architecture is based on a model execution engine similar to the meta-search engine, a runtime environment to support automatic coordination various search engine results based on the page rank algorithm. As shown in the following diagram, our architecture is structured with the following modules:

1. User.
2. Request and Response Handler
3. Search engine manager
4. Search Engine
5. Service Choreographer.
6. Result Manipulator.

User:

This is the initial stage where the user enters with the given credentials for entering into the user interface. The user interface has the text box where the user enters the keyword that to be searched in the various search engines and the results will be displayed in the same after extracting from various web services that are predefined by the developer. The user can request a new search engine to the administrator for the betterment of the results.

Admin:

In this, if the user request for a new search engine to be added in the choreography, the admin has the rights to view the available search engine in the database and can able to add a new predefined search engine for more possible outcomes to the user. Once the admin adds the new search engine, its corresponding web services will get automatically initiated with its roles and responsibilities which are predefined.

Search Engine Manager:

This is the central module in which the process starts with the choreography concept and connection of the available web services. In this module, the ‘n’ number of different search engines will be available and based on the selection by the administrator it gets connected with its

 ISSN: 2454-4248
149 – 155

corresponding web services and also gets interconnected with the other web services. When input keyword is recognized by the available search engines, the corresponding web services will get automatically initiated and process the request.

Search Engine Selector

If the number of component search engines in a meta search engine is very small, say less than 10, it might be reasonable to send each user query submitted to the meta search engine to all the component search engines. In this case, the search engine selector is probably not needed. However, if the number of component search engines is large, as in the large scale Meta Search Engine scenario, then sending each query to all component search engines will be an inefficient strategy because most component search engines will be useless with respect to any particular query. For example, suppose a user wants to find 50 best matching results for his/her query from a meta search engine with 1,000 component search engines. Since the 50 best results will be contained in no more than 50 component search engines, it is clear that at least 950 component search engines are useless for this particular query.

Result Manipulator:

After the processing the request by the web services, the uniqueness in the results will be stored in the database for avoiding the data redundancy. To achieve this process we need to get interconnected all the web services through mapping. The stored results in the database will be compared to each other so that the repetition of the results from the various search engines will be removed. The Result manipulator will gather all the unique data and will display them in a separate column for better view of the search results.

Meta Search Engine

Meta Search Engines can be classified into two types.

- General purpose Meta Search Engine
- Special purpose Meta Search Engines.

The former aims to search the entire Web, while the latter focuses on searching information in a particular domain (e.g., news, jobs).

Major Search Engine Approach:

This approach uses a small number of popular major search engines to build a Metasearch engine. Thus, to build a general-purpose Metasearch engine using this approach, we can use a small number of major search engines such as Google, Yahoo!, Bing (MSN) and Ask. Similarly, to build a special purpose Meta Search Engine for a given domain, we can use a small number of major search engines in that domain.

Large scale Meta Search Engine approach:

In this approach, a large number of mostly small search engines are used to build a Meta search engine. For example, to build a general-purpose Meta search engine using this approach, we can perceivably utilize all documents driven search engines on the Web. Such a Meta search engine will have millions of component search engines. Similarly to build a special purpose Meta search engine for a given domain with this approach, we can connect to all the search engines in that domain. For instance, for the news domain, tens of thousands of newspaper and news-site search engines can be used.

Web Service

A Meta search engine is a search engine that collects results from other search engine. Web service offers such functionality and then presents a summary of that information as the results of a search. Most search engines available on the Web provide only a browser based interface; however, because Web services start to be successful, some of those search engines offer also an access to their information through Web services. Two types of search engines are observed, one that acts like a wrapper for the HTML pages returned by the search engine and other one is build upon the Web service offered by the search engine but this difference is visible only when looking at the internal processing of the service. It is difficult to distinguish them from the outside as they implement the same interface.

Advantages of Web Service Meta Search Engine

We attempt to provide a comprehensive analysis of the potential advantages of meta search engines over search engines. We will also focus on the comparison of Meta search engine and search engine.

Increased Search Coverage:

Meta search engine can search any document that is indexed by at least one of the search engines. Hence, the search coverage of a meta search engine is the union of those search engines. Meta search engine with multiple major search engines as components will have larger coverage than any single component search engine. Different search engines often employ different document representation and result ranking techniques, and as a result, they often return different sets of top results for the same

 ISSN: 2454-4248
149 – 155

user query. Thus, by retrieving from multiple major search engines, a Meta search engine is likely to return more unique high quality results for each user query.

Better Content Quality:

The quality of the content of a search engine can be measured by the quality of the documents indexed by the search engine. The quality of a document can in turn be measured in a number of ways such as the richness and the reliability of the contents. General-purpose meta search engines implemented using the large-scale meta search engine approach has a better chance to retrieve more up-to-date information than major search engines and meta search engines that are built with major search engines.

Good Potential for Better Retrieval Effectiveness:

More unique results are likely to be obtained, even among those highly ranked ones, due to the fact that different major search engines have different coverage and different document ranking algorithms. The result-merging component of the meta search engine can produce better results by taking advantage of the fact that the document collection of major search engines has significant overlaps. This means that many shared documents have the chance to be ranked by different search engines for any given query. If the same document if retrieved by multiple search engines, then the likelihood that the document is relevant to the query increases significantly because there is more evidence to support its relevance. In general, if a document if retrieved by more search engines, the document is more likely to be relevant.

IV. SYSTEM IMPLEMENTATION

Modules Description

The architecture is based on a model execution engine similar to the *meta-search engine*, a runtime environment to support automatic coordination various search engine results . As our implementation is structured with the following modules:

- Admin Module
  - Login Module
  - Request Module
  - Update Module
  - Logout Module
- Guest Module
  - Login module
  - Search Module
  - Track Module
  - Logout Module

Admin Module

In this, if the user request for a new search engine to be added in the choreography, the admin has the rights to view the available search engine in the database and can able to add a new predefined search engine for more possible outcomes to the user.

Once the admin adds the new search engine, its corresponding web services will get automatically initiated with its roles and responsibilities which are predefined.

Login Module

Login Module presents site visitors with a form with Username and password fields. If the User enters a valid username/password combination they will be granted access to the website.

Request Module

Request ModulePrompt the user to choose the search engine .

Update Module

Once the admin choose their search engine to click the Add button in order to add the search engine.

Logout Module

Once the user update the search engine to logout the session.

Guest Module

This is the initial stage where the user enters with the given credentials for entering into the user interface. The user interface has the text box where the user enters the keyword that to be searched in the various search engines and the results will be displayed in the same after extracting from various web services that are predefined by the developer.

Login Module

Login Module presents site visitors with a form with Username and password fields. If the User enters a valid username/password combination they will be granted access to the website.

Search Module

The User needs to enter the keywords to be searched in the search tab

Track Module

Track Module will display all the results corresponding to the search in different search engine

Logout Module

Once the search is done the user needs to logout the session via logout button.

 ISSN: 2454-4248
149 – 155

V. CONLCUSION

Web Services represents a new architectural paradigm for web application. Web services implement capabilities that are available to other applications via industry standard network and application interface and protocols. An application can use the capabilities of Web Services by simply invoking it across a network with having to integrate it.Web Services are a key component of the emerging, loosely coupled, Web-based computing architecture.

A Web Service is an autonomous, standards-based component whose public interfaces is defined and described using XML. The Web Services specifications offer a communication bridge between the heterogeneous computational environments used to develop and host applications. The future of E-Business applications requires the ability to perform long-lived, peer-to-peer collaborations between the participating services, within or across the trusted domains of an organization.

The Web Services Choreography specification is aimed at the composition of interoperable collaborations between any type of participant regardless of the supporting platform or programming model used by the implementation of the hosting environment. A choreography description is the multi-participant contract that describes this composition from a global perspective.

The goal of this proposal is to shed some light on the reasons that make building Meta Search engines, especially large scale Meta Search engines, difficult and challenging. Modern search engines providing interfaces that allow external applications to issue Web search queries that are actually processed using their large scale computing infrastructure.

This paper proposes a robust Meta search engine, which can communicate to heterogeneous platform using choreography techniques for web services. As much progress has been made in advanced Meta Search engine technology, several challenges need to be addressed before truly large scale Meta Search Engine can be effectively built and manage.

In our work contains we are using limited number of search engines, results based on availability of free version of API codes. In future, we can have the web based application can be upgraded with the paid API codes from various search engines and can be launched in a free version with limited search engines and a paid version as a full choreography web based application.

REFERENCES

[1] Adam Barker, ChristoperD.Walton, and David Robertson. Choreographing Web Services.IEEE Transactions on Services Computing. Vol. 2, No. 2, April-June 2009.

[2] Zahra Madani, NaserNematbakhsh. A Logical Formal Model for Verification of Web Service Choreography.Proceedings of 2009 12th International Conference on Computer and Information Technology (ICCIT 2009).

[3] Howard Foster, Sebastian Uchitel, Jeff Magee, JeffKramer. Compatibility Verification for Web Service Choreography. Imperial College London, 180 Queen's Gate, London SW7 2BZ, UK {hfl, su2, jnm, jk}@doc.ic.ac.uk

[4] FestimHalili and Agni Dika. Choreography of Web Services and the impact of estimation of execution plan. 2012 International Conference on Information Technology and e-services.

[5] Jan Terpak, PavelHorovcak, MAtejLukac. Mathematical models creation using orchestration and choreography of web services. Institute of Control and information of production processes.

[6] Andreas Weib, VasiliosAndrikopoulos, Michael Hahn, and DimkaKarastoyanova. Enabling the Extraction and Insertion of Reusable Choreography Fragments. Institute of Architecture of Application Systems (IAAS) University of Stuttgart, Stuttgart, Germany.

[7] Thomas Kothmayr, Alfons Kemper, Andreas Scholz, JorgHeuer. Instant Service Choreographies for Reconfigurable Manufacturing Systems- a Demonstrator. Chair for Database Systems.