Australian Journal of Basic and Applied Sciences, 9(16) Special 2015, Pages: 179-183

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Service Computing

[1]Dr. M. Thiyagarajan, [2]Chaitanya Krishnakumar, [3]Dr. V. Thiagarasu

[1]*Professor Emeritus and Dean Research, Nehru Group of Institution, Kuniamuthur, Coimbatore- 641008.*
[2]*Department of Computer Applications, Nehru Institute of Information Technology and Management, Thirumalayampalayam, Coimbatore – 641105*
[3]*Assistant Professor of Computer Science, Gobi Arts and Science College, Gobichettipalayam-63845*

ARTICLE INFO

Article history:
Received 10 March 2015
Received in revised from 20 March
Accepted 25 March 2015
Available online 10 April 2015

Keywords:
Web Services, Service Computing, JSP, XML, PROLOG, Graph isomorphism.

ABSTRACT

We aim at providing web services after three stages, customers classification, authentication and service data entity. The implementation is done through JAVA XML and prolog programmes details are given as in an illustration.

To Cite This Article: Dr. M. Thiyagarajan, Chaitanya Krishnakumar, Dr. V. Thiagarasu, Service Computing. ***Aust. J. Basic & Appl. Sci.,*** *9(16): 179-183, 2015*

INTRODUCTION

Services computing refers to a flexible computing architecture that packages functionality as a suite of interoperable routines that can be used within multiple, separate systems from several business domains. Service computing requires loose coupling of services with operating systems, and other technologies that underlie applications. Functions are separated into distinct self-describing and autonomous units, or services, which developers make accessible via pre-defined interfaces over a network in order to allow users to combine and reuse them in the production of applications. These services communicate with each other by passing data in a well-defined, shared format, or by coordinating an activity between two or more services. Service Computing includes the concepts of Service Oriented Architecture, Mashups, Software as a Service, and Cloud Computing. (Dr. Geetha).

Service Oriented Architecture (SOA) is one domain that is emerging very rapidly and changing the way we write web applications. Recent years have seen a tremendous growth of the service-oriented applications. More and more companies are investing heavily in Service Oriented Architecture. The principal benefit which the companies derive by implementing Service Oriented Architecture is, to redefine the existing infrastructure and reforming the legacy applications as consumable services. This helps them to reduce the cost of maintaining the system and support the end users with legacy applications over the network. Service Oriented Architecture contains the functionality of interoperable services also known as Web Services. These services are platform independent and they can be consumed over a network. Service oriented architecture is a new concept. In a more efficient scenario, common tasks would be shared across all three processes. This can be implemented by decoupling the functionality from each process or application and building a standalone authentication and user management application that can be accessed as a service. This would be a simple example of SOA. In reality, enterprise SOA is much more difficult because services may be deployed across multiple domains of ownership. SOA defines how to integrate widely disparate applications for a Web- based environment and uses multiple implementation platforms. By implementing SOA, the companies can run more effectively. Companies with large legacy back-end systems, like Google, Amazon have used this approach to modernize existing infrastructure.

The present paper implements three independent services. These services can interact with each other. The whole work has been divided into three modules.. Each service can interact with each other thus defines the term service computing. one service module categorize the users based on his rights. Second service authenticates the user against his set of resources. Third service publishes the set of resources authenticated for that set of users. All the

Corresponding Aluthor: Dr. M. Thiyagarajan**,** Professor Emeritus and Dean Research, Nehru Group of Institution, Kuniamuthur, Coimbatore- 641008.
E-mail: m_thiyagarajan@yahoo.com

services are developed using java and converted as a service to enforce reusability.

Section 1: basic concepts:

Definition 1: Service Oriented Architecture:

Service-oriented architecture (SOA) is an approach used to create an architecture based upon the use of services. Services (such as Restful Web services) carry out some small function, such as producing data, validating a customer, or providing simple analytical services. In addition to building and exposing services, SOA has the ability to leverage these services over and over again within applications (known as composite applications). SOA binds these services toorches tration, or individually leverages these services. Thus, SOA is really about fixing existing architectures by addressing most of the major systems as services, and abstracting those services into a single domain where they are formed into solutions.

In addition to building and exposing services, SOA has the ability to leverage these services over and over again within applications (known as composite applications). SOA binds these services toorchestration, or individually leverages these services. Thus, SOA is really about fixing existing architectures by addressing most of the major systems as services, and abstracting those services into a single domain where they are formed into solutions.

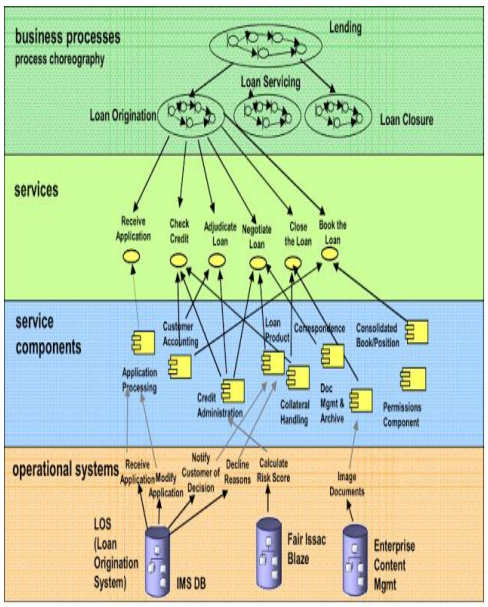

Fig.1: Service Oriented Architecture.

One of the keys to SOA architecture is that interactions occur with loosely coupled services that operate independently. SOA architecture allows for service reuse, making it unnecessary to start from scratch when upgrades and other modifications are needed. This is a benefit to businesses that seek ways to save time and money.

One of the keys to SOA architecture is that interactions occur with loosely coupled services that operate independently. SOA architecture allows for service reuse, making it unnecessary to start from scratch when upgrades and other modifications are needed. This is a benefit to businesses that seek ways to save time and money.

Definition 2: Web Services:

Web Service Architecture Working Group defining a Web service as ”a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition using XML based messages conveyed by Internet protocols”. (http://www.w3.org/TR/ws-arch/wsa.pdf) 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-processable 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.

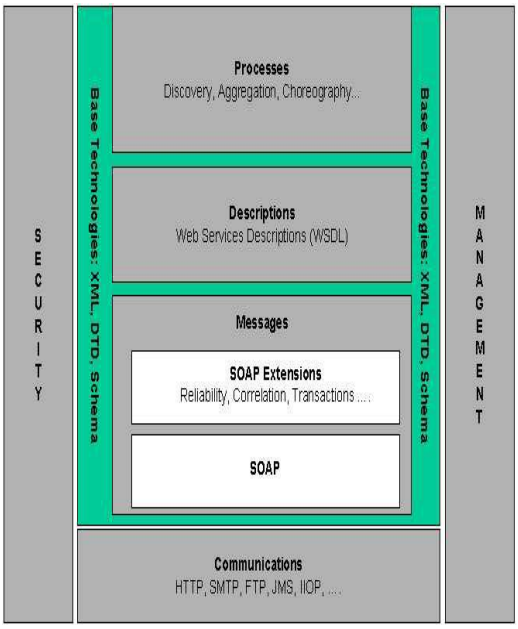

Fig. 2: Web Services Architecture.

XML solves a key technology requirement that appears in many places. By offering a standard, flexible and inherently extensible data format, XML significantly reduces the burden of deploying the many technologies needed to ensure the success of Web services.

Definition 3: SOAP:

SOAP provides a standard, extensible, compostable framework for packaging and exchanging XML messages. In the general case, a SOAP message represents the information needed to invoke a service or reflect the results of a service invocation, and contains the information specified in the service interface definition. When using the optional SOAP RPC Representation, a SOAP message represents a method invocation on a remote object, and the serialization of in the argument list of that method that must be moved from the local environment to the remote environment.

Definition 4: WSDL:

WSDL describes Web services starting with the messages that are exchanged between the requester and provider agents. The messages themselves are described abstractly and then bound to a concrete network protocol and message format. Web service definitions can be mapped to any implementation language, platform, object model, or messaging system. Simple extensions to existing Internet infrastructure can implement Web services for interaction via browsers or directly within an application. The application could be implemented using COM, JMS, CORBA, COBOL, or any number of proprietary integration solutions.

A *registry* is an authoritative, centrally controlled store of information. UDDI is often seen as an example of the registry approach

Definition 5: Graph Isomorphism:

Two Graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ that have the same sets of vertices $V_1 = V_2 = \{1,2,\ldots..n\}$ are isomorphic, if there exists a permutation π on vertices $\{1,2,..,n\}$ so that $(u, v) \in E_1 \leftrightarrow (\pi(u), \pi(v)) \in E_2$.

Two graphs *G1* and *G2* are said to be isomorphic, if a one-to-one permutation or mapping exists between the set of vertices of *G1* and the set of vertices of G2, with the property that if two nodes of *G1* are adjacent, so are their images in *G2*. The graph isomorphism problem is therefore the problem of determining whether two given graphs are isomorphic.

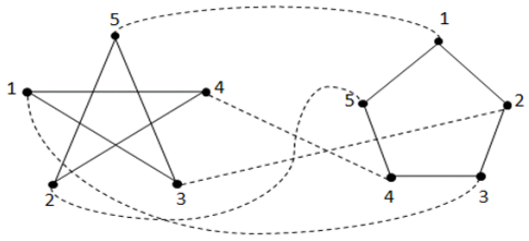

Fig. 3: Relabelling the vertices of graph.

Section 2: web service access of rights as service computing:

Service computing enables the interaction of many independent service units. Each service is a stand alone functions. Thus we can invoke a service just like a function call. Thus these independent services can be invoked for anywhere. Thus the modifications require change only in that particular services thus ensures reliability.

We have developed a program to illustrate the service computing. Three independent services have been developed .thus choreographer just needs to invoke the service at the time of their need. For our scenario, we have created user categorization service, user authentication service and resource provision. Thus a user enters the system he is categorized into a particular class. authentication service is used for resource authentication. The output will be the list of documents .The resource provider service will display all the documents in the list.

Section 3: illustration of sample service:

User categorization:

A user log in to the system. His data will be available in the database. From the database his category will be retrieved. We can modify this for set of rules. For example, we can group the customers with their registration code KL,TN,KA etc. Thus we can say input the registration and we will get the group we belong. We need to modify only the conditional data for any further change thus ensures reliability. This can be used by any group for this group of information thus ensure reusability.

This service has been developed using java and constructed as a service. our sample program will enter username and password. program will consult the database and retrieves the result.

User authentication:

This service will authenticate the user against a set of documents. We deviate from the traditional approach of authentication by random key generation. Thus a user class will be provided with a secret key and he will be authenticated using that secret key. Here we use the graph isomorphism property for authentication. We are generating a random graph on the run. So each class will have independent graphs. Thus for the access, he must have the right key for the right access. Here in our program, we have developed a graph for three set of classes

Pseudo code for zero knowledge using graph isomorphism:

1. Graph G1 with n number vertices .
2. Graph G2,G3,G4 represents the isomorphic graphs for C1, C2,C3 based on the access right of the users.
3. A Client randomly generates a=0 or a=1 and a random Graph H.

4. Server then reply with a random b value.(From the b value we are performing a hash function and takes a bit 0 or 1 as b)
5. Based on a and b client will give σ as response to the server,
6. If (a==b) $\sigma = \rho^{-1}$
Else if (a=0 && b=1) $\sigma = \rho^{-1} \circ \pi$
Else $\sigma = \rho^{-1} \circ \pi^{-1}$
end if
7. If (σ (H) =G2) Access for C1 users
Else if (σ (H) =G3) Access for C2 users
Else if (σ (H) =G4) Access for C3 users
End if
8. End pseudocode

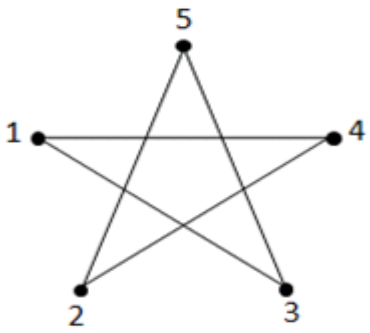

Fig. 3: G1

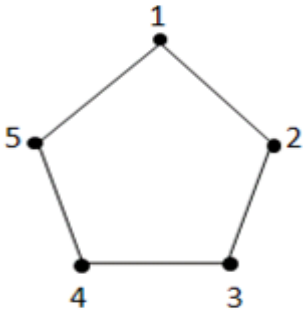

Fig. 4: G2 .

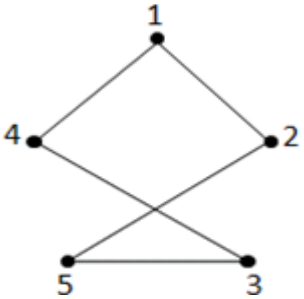

Fig. 5: G3.

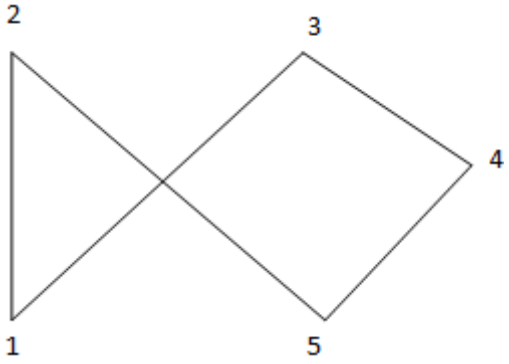

Fig. 6: G4.

Eric Ayeh in his MS Thesis [5] prescribed an algorithm which incorporates both zero knowledge and Graph Isomorphism. Graph G1, G2, G3, G4 are public. User will generate a random graph H from G1 and if the user is able to generate any of the Graph G2, G3, G4 from H, he is given the corresponding access. Thus authentication is implemented using Graph Isomorphism.

For our purpose we have provided an additional property of receiving the list of documents from the database. It will retrieve only the file IDs of the document. Thus after the execution of service, we will have our list of authenticated documents. Thus all these files id will be stored in the memory stack for further purposes.

Resource provider:

Almendros-Jiménez *et al* (2008) have presented a proposal for the implementation of the XPath language in logic programming. They described the representation of XML documents by means of a logic program. In this context, rules and facts are used for representing the document schema and the XML document itself. In particular, they described as to how to represent indexes of XML documents in logic programs. That is, rules are stored in many memories by using two kinds of indexes, one for each XML tag, and other for each group of terminal items. One can query a logic programs which represents an XML document by means of the XPath language. This evolves the specialization of the logic program with regard to the XPath expression. Finally, they explained as to how to combine the indexing and the top-down evaluation of the logic program

THIS SERVICE WILL retrieve THE XML DOCUMENT for display. This service will have the list of document id in their kitty. This xml documents will be retrieved from the database . with proper styling this will be displayed. XML provides a standard way to define the structure of documents that is suitable for automatic processing. This enables the development of generic tools that parse documents and extract their content as well as their structure. Restrictions on the structure of a document can be specified by Document Type Definitions

(DTDs) or XML SCHEMAs (however, neither of these provide any semantic information). XML has been widely adopted as the foundation for data representation and formats on the Web. Many parsers and toolkits exist for different programming environments, which implement the XML related standards.

Table 1: Comparison of classic Algorithm and Zero Knowledge using Graph Isomorphism

Features	ZERO KNOWLEDGE	
	Classic Algorithm	Graph Isomorphism
Time Complexity	0.032s	0.009s
Space Complexity	0.8MB	0.79MB
KEY Extraction	Easy	Difficult
Cheated verifier	No way to handle	Randomized bit to stop the cheated verifier

Conclusion and Future Work:

Thus three independent services have been created. It can be used for general purpose. We have implemented the services using java. We have performed the categorization, authentication and retrieval of documents using the xpath language. We have made connection call for invoking the categorization and then to the authentication. Work is going on for invoking the resource provider module. Our final aim is to perform the automaton of these services.

REFERENCES

Almendros-Jiménez, J.M., A. Becerra-Terón and F.J. Enciso-Baños, 2008. "Querying XML Documents in Logic Programming", Journal of Theory and Practice of Logic, programming, 8(3).

Dr. Geetha, G., Rohith Dhand, "AN Automated Trust based security frame work for maximizing authorization control in web services", professional Lovely University.

http://abdennourinsat.blogspot.in/2012/04/soa-tutorial.html

http://nsl.cse.unt.edu/~dantu/cae/attachments/Eric_Ayeh_MS_thesis.pdf.

http://researcher.watson.ibm.com/researcher/view_group.php?id=152

http://www.w3.org/TR/ws-arch/wsa.pdf