Research Journal of Applied Sciences, Engineering and Technology 8(5): 630-636, 2014
DOI:10.19026/rjaset.8.1015
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: April 29, 2014 Accepted: June 08, 2014 Published: August 05, 2014

Research Article

Dynamic Composition of Web Services based on Event Driven Approach in SOA

[1]P.S. Ashok Kumar, [2]B. Kaarthick and [3]C. Gopal Krishna
[1]Department of Computer Science and Engineering, YDIT, Bangalore-566078, India
[2]Department of E&C Engineering, CIET, Coimbatore-641047, India
[3]Department of Computer Science and Engineering, AIT, Chikmagaluru-577101, India

Abstract: Dynamic Composition of Web services is a Key research in the arena of E-commerce. Enterprise applications are developed as a service provider in the web, some of the challenges we are faced by web services are most likely related to composition of services, security and Quality of Services (QoS). Within these challenges, composition of web service turns out to be a big arena of research, because it supports the integration and Interoperability of B2B applications or enterprise application or E-commerce application. Pragmatically dynamic composition of semantic web services specifies better results compared to the traditional method of discovering candidate services for composition. In this study we present the degree of matching and selection of web services are considered as a salient feature to the dynamic composition of web services.

Keywords: B2B, composition, ontology, OWL, OWL-S, QoS, SOA

INTRODUCTION

Web services are loosely coupled and reusable software components, Where Services are programmatically accessible over the internet. The reusability of software component is a new tendency in software architecture, i.e., reusability minifies the amount of code being written for every specific application. Service Oriented Architecture is one, which helps to achieve the maximum reusability of the components with the set of existing web technologies. Web services are represented in Service-Oriented Architecture (SOA), so specialized and generalized services in an application can be deployed and accessed by other services or business logic.

A web service is a reusable and self-built components, it provides functionalities like furnish specialized services to an application. Web services will render the services to be platform-independent, i.e., the platform independent feature in web services will creates the opportunity in enterprises to develop versatile business activities by combining existing web services. The Composition of web services will combine the existing web services as well as services offered by different service providers (Alamri *et al*., 2006).

The Semantic Web is a repository; it is capable to identify resource, in such a way that machines can understand the services. A Semantic web describes the relationships between objects and the attributes of objects.

Semantic Web is the extension of the World Wide Web; it permits the users to share resources beyond the boundaries of applications and websites. The features of web services and their specifications are accessed based on XML technologies, like WSDL (ChengZhi *et al*., 2010).

METHODOLOGY

OWL based semantic web service composition model: Ontologies are used to describe the conceptualization of a semantic web. Ontology will play an important role in semantic web. Ontologies are used to represents the semantic informational relationship and carry out web information management automatically. Ontology is nothing but a Knowledge based management system, it describe the forms of objects, object properties and relationship between objects, which are possible in a specified domain of knowledge of a semantic web (Deng *et al*., 2008).

Ontology provides a solution to the basic element of the Semantic Web. Ontology is a document or file; it defines the relationship among terminologies. Now a day's researchers have accepted these ontology terminologies in their research study, to improve their research and create their own jargon (Deng *et al*., 2008).

Ontology consists of:

- **Concept:** Conceptual object of the domain
- **Property:** Attribute describing a concept

Corresponding Author: P.S. Ashok Kumar, Department of Computer Science and Engineering, YDIT, Bangalore-566078, India

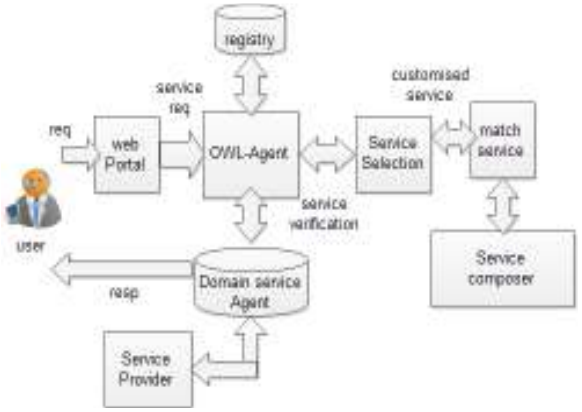

Fig. 1: OWL based web service composition model

- **Relation:** Relationship between concepts
- **Axiom:** Describes the statement based on concepts

Ontology can be defined by:

- RDF (Resource Description Framework)
- OWL (Web Ontology Language)

RDF (S): RDF (S) is a simple lexicon language to communicate the relationships between resources. RDF is a graphical pattern, acquiring a XML-based syntax. The main importance of the RDF is to describe the information in the Web in a machine-accessible form. The RDF model is constructed by triplets, i.e., Object, subject and predicate. In the web RDF (S) resources are identified by Web identifiers i.e., URI (Uniform Resource Identifier). RDF (S) extends RDF with a schema vocabulary. RDF (S) allows us to define basic vocabulary terms such as types, classes, properties, ranges, domains and the relations between them.

OWL (Web Ontology Language): Web Ontology Language is a standard for Web ontologies. OWL allows us to describe the semantics in a machine-accessible form. Owl can be expressed in RDF using an XML vocabulary; it indicates the hierarchies and relationships in between the different resources.

OWL is similar to RDFS, but OWL has a stronger syntax and more expressive vocabulary for defining Semantic Web Ontologies. The Semantic Web is focused on the Activity-centric specification of services, where actions are applicable in state of operations. There is more ability in terms of considering different choices of services based on goals. A goal oriented approach for service composition holds the information about the OWL-S services. Figure 1 represents composition model for OWL based web services, when ever user request a service it select the services from OWL-Agent, that will be identified by Registry. Composing of selected services is matched by match services component. The Domain service agent is a medium to respond to the user service request as well as acknowledgement to service provider.

COMPOSITION OF WEB SERVICES BASED ON EVENT-DRIVEN ARCHITECTURE

An Event-Driven Architecture (EDA) is a software Architecture technique, where automated software tools are used to generate an OWL web services description. EDA focusing only at design level of web services and do not discuss any issues about run time or Dynamic composition of services (Laliwala *et al*., 2006).

Semantic language based tools are built based on architecture, like Apache Axis, maven architecture, etc. QoS is used to specify the semantic information, where services are selected based on QoS, using the QoS

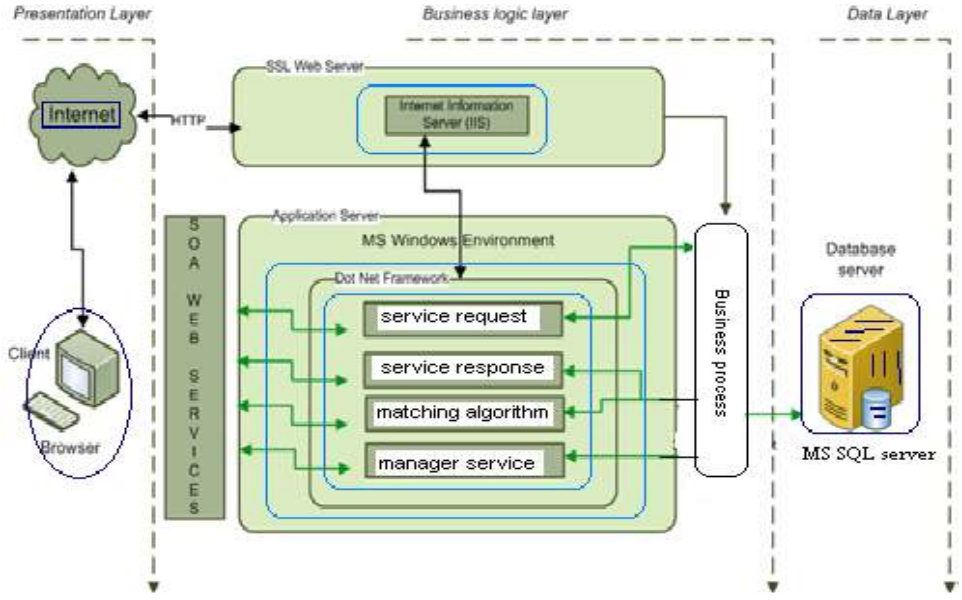

Fig. 2: SOA design

ontology and Protégé has been used for the purpose of annotating QoS related information to the web services. The selection of the services is based upon normalization algorithm (Dustdar and Schreiner, 2005).

Service oriented architecture design: Service Oriented Architecture (SOA) is the heart of a revolutionary computing platform that is being adopted world-wide and has earned the support of every major software provider. At the moment, the technology of Web services are built on existing and emerging Technologies like, HTTP, XML, SOAP or it may be RESTful are in use.

Figure 2 represents the Service Oriented Architecture design of Online Book purchasing and Trading System, it contain different components with 3 tiers, like.

Presentation layer: The presentation layer deals with the presentation logic where the information will be exchangeable between the users. Connection will be established from one user end to other user. The presentation layer provides full flexibility to purchaser to find particular book attributes like, Price, author, publisher etc.

Business layer: The business layer is used for the communication among the modules like, databases and external web services including Receiving Email and receiving MMS from external email server and stores all those information in to databases. When those MMS and Emails will be received on application server and stored into the database then publisher and supplier will be able to look on purchaser data according to their available history.

Data layer: This layer is used to store the information like Book's details, offer and discount etc. Using this data layer the data is sent to presentation layer and modified through the business layer.

For the composition of web services, we consider many QoS attributes, like Reliability, availability security properties and performance properties. Figure 3 represent the various QoS's properties for security issues are represented by ontology, in which the services are semantically selected for composition (Frederico and Jose, 2011).

Graph approach: Graph approach can be used to handle various composite services, whether they are structured as pipelines or acyclic graphs. Our proposed algorithms are based on the shortest path algorithms. We modify the shortest path algorithms to carry on delay confinements.

Construction of graph: Each and every individual service is composed of a node in the graph, with a Cost (Ci**)** and a Value (Vi).

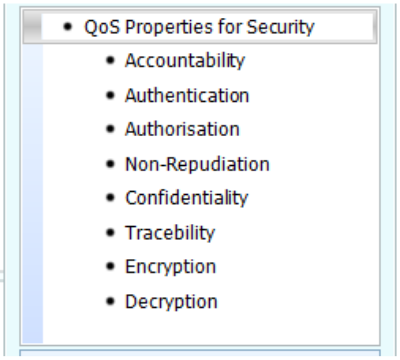

Fig. 3: Properties of QoS for book_information ontology

However, the shortest path algorithm is designed for graph usually defines cost on edges. For example in Bellman-Ford algorithm, we need to transform a composite service graph into a service path graph so that edges have weights (cost, value and delays).

Shortest path selection algorithm: Here paths are searched using the breadth-first approach. However, the composite service graph can be modified has shortest path algorithm to solve the problem. The main idea behind is that, first of all, sort all nodes in the graph to be in a topology, then.

Algorithm: shortest path selection algorithm:

Input: node, edges;
Output: selected service;
Sort each and every node and outgoing edges with delay;

```
sort each nodes in a graph to be in topology;
i = startnode;
for each node in a topological order n_i,
    for each outgoing edge ei, increasing delay d_i
    j = end node;
    delay = edge.delay;
    qutility = edge.qutility;
      if (i = = source_node)
      wait (i, j, delay, qutility);
      else
            for (each path p in list (i))
            delay = delay + p.delay;
            qutility = p.qutility;
      if (delay < = constraint)
        wait (i, j, delay, qutility);
             if new-path np is controlled by an
         old-path P
             return;
      if any old-path p is controlled by new-path
   np
      remove p from link;
      add np to link;
Suppose to connect service R1 with service R2;
  if R1 has two candidate services, like S1 and S2;
      if R2 has two candidate services S3 and S4.
```

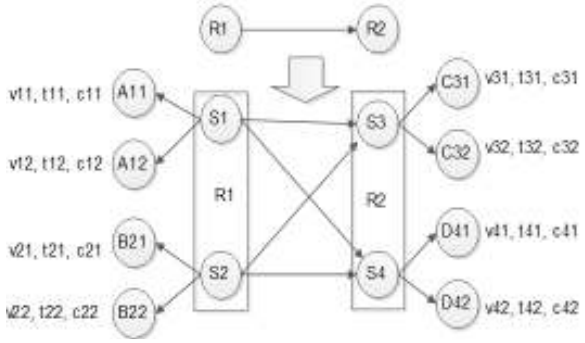

Fig. 4: Complex service connection

for Each service has two service levels Aij, Bij and Cij, Dij

where i = 1, 2, 3, 4 and j = 1, 2, 3, 4

each service level has service time tij and value vij,

where i = 1, 2, 3, 4; and j = 1, 2, 3, 4.

The network transmission delay can be defined as dij, where i = 1, 2; j = 1, 2

cost can be defined as cij where i = 1, 2; j = 1, 2

Figure 4 define how to interact with complex services based on shortest path algorithm, where communication of each and every node in a graph to be in the topological order. In each node, we maintain a list weights with delay constraint for different paths from the source node to response node.

EXPERIMENTAL RESEARCH

Here we proposed composition oriented service discovery, in which service matching is done with the help of OWL ontologies. The services in the repository are translated into tree structure, followed by a dependency graph construction from the atomic process, it matches the service request and the graph is checked for the satisfying the query. The issue of cross ontology for discovering web services is not addressed by them (Rui *et al*., 2011).

Case study:

Online books trading system: OWL is a standard for Web ontologies. OWL allows us to construct the semantics in a machine-accessible form.

OWL defines the types of relationships that can be expressed in RDF using an XML vocabulary to indicate the hierarchies and relationships between different resources. Figure 5 represents the Engineering Book search model ontology designed by using Protégé ontology development tool.

Protégé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with Ontologies.

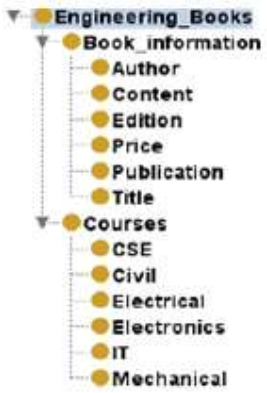

Fig. 5: Ontology of engineering book search model using protégé

E-book procurement system: A business process consists of a set of activities to provide a specified service. An Activity is automated action that indicates what is to be done at a particular step in the process. For example, consider the book procurement process of a book store, where a customer needs to order book (s) from a book store (provider). The service provider processes the order and delivers book (s) to the customer. Once the order is processed, payment is made by the customer using credit card or any other mode of payment. To carry out this process, various activities such as purchase order receiving, book availability checking, credit card validation checking, payment and delivery of books are involved.

Figure 6 represents the control flow diagram of E-Book procurement system. With the advancement of Internet, organizations express their functionalities in terms of services provided by them. A service requires individual and autonomous unit of activity to be performed. The size and scope of the functionality represented by a service varies. A process is composed of one or more services. Services are performed by a single organization or may interact with services performed by other organizations. For example, book availability checking is performed by book store, while credit card validation is conducted by a credit card company.

In an online Book dealer bill payment system, dealer will go for auction against bulk purchase of books. If auction price will matches to seller and dealer then only business process is get completed, if not once again new business process will start. Auction bill should contain whole information about what are the books purchased by dealer with price on unit wise and volume etc.

Collaboration diagram with weighted graph: The collaboration diagram of this system represents the entire control flow functions in this case study. The system has major live components like: GUI, Bill Trade, Bill verifier, Cash Trade, Cash Handler and Bill/ receipt.

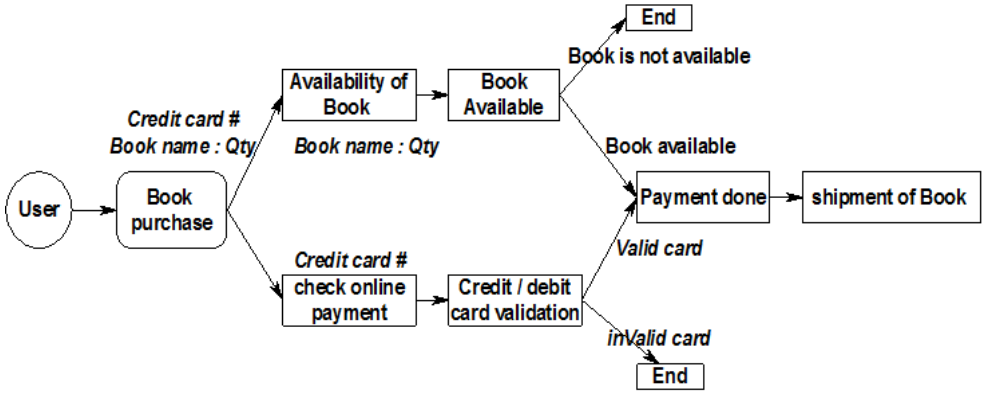

Fig. 6: E-book procurement system-a running example

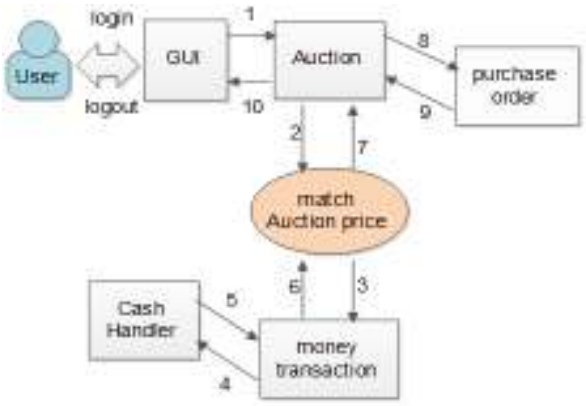

Fig. 7: Collaboration diagram with weighted graph in auction of books
1: Cash info; 2: Apply 4 auction; 3: Auction price matched; 4: Cash trading; 5: Cash verifier; 6: Confirm auction price; 7: Check balance; 8: Balance information; 9: Print purchase order; 10: Issue order and auction item info

The entire Book shop bill payment system has been represented in the form of a collaboration diagram.

After design and develop the collaboration diagram using Smart Draw software and save the file as XYZ.srd, then pass this file as input to the parser.

The parser collects all the information about objects and statements, which are represented as nodes and edges in a directed graph with weights are shown in Fig. 7.

When Dealer expects a bill in the bill counter, accountant will edit all the merchandised products through GUI. The dash boards of bill GUI will automatically generate the product price with respect to the number of units, that will take care of bill transact. Bill verifier will verify the bill information with price details.GUI will provide full information about the supplier, even though what are the facilities and products are available by the supplier as well as the forthcoming issues of Journal and merchandised product information along with sequence diagram shown in Fig. 8.

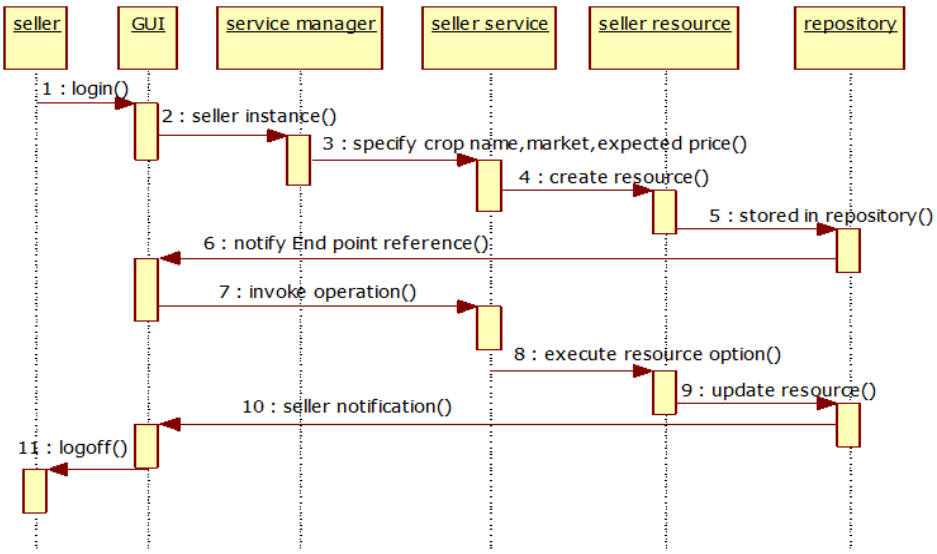

Fig. 8: Seller service sequence diagram of online trading system

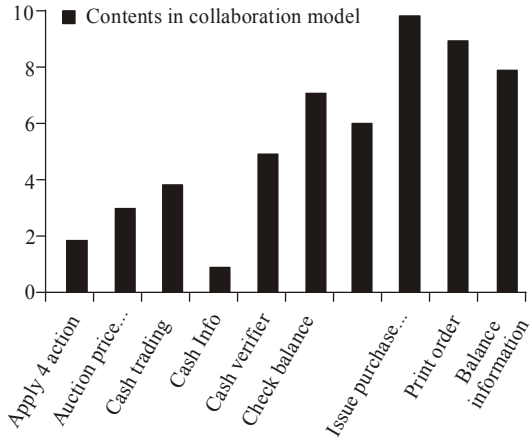

Fig. 9: Contents in collaboration model graph

Table 1: Mapping information of weight and contents

Contents in collaboration model	Weights
Cash info	1
Apply 4 auction	2
Auction price matched	3
Cash trading	4
Cash verifier	5
Confirm auction price	6
Check balance	7
balance information	8
Print purchase order	9
Issue order and auction item info	10

Table 1 and Fig. 9 represents the mapping information of collaboration model contents as well as weights in each edges and graph of collaboration model.

For example Fig. 8 represents the sequence diagram of Online trading system based on seller

```
<?xml version="1.0"?><!DOCTYPE rdf:RDF
[ <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
<!ENTITY
Ontology"http://www.semanticweb.org/ontologies/2013/Ontology.owl#" > ] >
<rdf:RDF xmlns="http://www.w3.org/2002/07/owl#"
xml:base="http://www.w3.org/2002/07/owl"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:Ontology="http://www.semanticweb.org/ontologies/2013/Ontology.owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<Ontology rdf:about="http://www.semanticweb.org/
ontologies/2013/Ontology.owl"/>
<!-- http://www.semanticweb.org/ontologies/2013/Ontology.owl#Engineering_Books -->
<Class rdf:about="&Ontology;Engineering_Books"/>
<!-- http://www.semanticweb.org/ontologies/2013/Ontology.owl#Book_information-->
<Class rdf:about="&Ontology;Book_information">
<rdfs:subClassOf rdf:resource="&Ontology;Engineering_Books"/>
</Class>
<!-- http://www.semanticweb.org/ontologies/2013/Ontology.owl#Content-->
<Class rdf:about="&Ontology;Content">
<rdfs:subClassOf rdf:resource="&Ontology;Book_information"/></Class>
<!-- http://www.semanticweb.org/ontologies/2013/Ontology.owl#Edition -->
<Class rdf:about="&Ontology;Edition">
<rdfs:subClassOf rdf:resource="&Ontology;Book_information"/></Class>
<!-- http://www.semanticweb.org/ontologies/2013/Ontology.owl#Price -->
<Class rdf:about="&Ontology;Price">
<rdfs:subClassOf rdf:resource="&Ontology;Book_information"/></Class>
<!-- http://www.semanticweb.org/ontologies/2013/Ontology.owl#Title -->
<Class rdf:about="&Ontology;Title">
<rdfs:subClassOf rdf:resource="&Ontology;Book_information"/></Class>
<!-- http://www.semanticweb.org/ontologies/2013/Ontology.owl #Publication -->
<Class rdf:about="&Ontology;Publication">
<rdfs:subClassOf rdf:resource="&Ontology;Book_information"/></Class>
<!-- http://www.semanticweb.org/ontologies/2013/Ontology.owl#Civil -->
<Class rdf:about="&Ontology;Civil">
<rdfs:subClassOf rdf:resource="&Ontology;Courses"/></Class>
<!-- http://www.semanticweb.org/ontologies/2013/Ontology.owl#Electrical -->
<Class rdf:about="&Ontology;Electrical">
<rdfs:subClassOf rdf:resource="&Ontology;Courses"/></Class>
<!-- http://www.semanticweb.org/ontologies/2013/Ontology.owl#Electronics -->
<Class rdf:about="&Ontology;Electronics">
<rdfs:subClassOf rdf:resource="&Ontology;Courses"/></Class>
<!-- http://www.semanticweb.org/ontologies/2013/Ontology.owl#IT -->
</rdf:RDF>
```

Fig. 10: RDF representation of book search information ontology using OWL-API

Table 2: Role and permission in trading system

Role	Create items	Read items	Update items	Delete items
Seller	√	√	√	√
Buyer	X	√	X	X
Broker	X	√	X	X
Admin	√	√	√	√

perspective, where the control flow of trading system can be defined as:

- Seller log in to portal
- Portal checks the Seller authentication and authorize it
- Portal creating, signing and forward information through endpoint
- Portal sending message to Web Service
- Web Services Handler accepting or denying request to Web Services based on original user's role
- Web Services Handler finally sending message to Web Services
- Execute the operation by specific Web Services
- Web Services send back response to portal

Role Based Asccess Control model will allocates privileges to each and every role and it is very easy to supervise all privileges to different roles, like Seller, Buyer, Broker and Admin. Our Online Trading System contains limited number of Web Services collection hierarchies, where we define least privileges to the junior roles like Buyer and Broker and the senior roles to Seller and Admin. Table 2 represents the privilege of different roles in trading system.

Some time Buyer will inherit the privileges of the Seller role, that moment we consider that Role delegation was happens between Seller and Buyer. It supports the fine authorization and easy management of role.

Figure 10 represents the respective RDF representation of Engineering Book search Information Ontology using OWL-API.

CONCLUSION

Due to the platform independent feature, web services will create the opportunity in enterprises to develop versatile business activities. The Web services application will combine the existing service with customized services based on referenced URI. Consider the semantic web services, the available services and the size of ontology models is too vast. So it is essential to reduce the search space during selection and composing of web services. Monitoring of selection and composition of web services activity leads better results. Generally the main problem in dynamic web service composition is to address the error handling mechanism at run time, else which may lead to failure in web service composition. In future work the degree of matching as well as selection of web services should also be considered as a salient feature in the composition of dynamic web services.

REFERENCES

Alamri, A., M. Eid and A. El Saddik, 2006. Classification of the state-of-the-art dynamic web services composition techniques. Int. J. Web Grid Serv., 2(2): 148-166.

ChengZhi, X., P. Liang, T. Wang, Q. Wang ad P.C.Y. Sheu, 2010. Semantic web services annotation and composition based on ER model. Proceeding of the IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC). Newport Beach, CA, pp: 413-420.

Deng, D., G. Zhang, Z. Gong, Z. Guo and P.C.Y. Sheu, 2008. Semantic programming of web-enabled database applications. Proceeding of the IEEE International Workshop on Semantic Computing and Applications (IWSCA, 2008), pp: 51-60.

Dustdar, S. and W. Schreiner, 2005. A survey on web services composition. Int. J. Web Grid Serv., 1(1): 1-30.

Frederico, G.A.D.O.J. and M.P.D.O. Jose, 2011. QoS-based approach for dynamic web service composition. J. Univers. Comput. Sci., 17: 71-85.

Laliwala, Z., R. Khosla, P. Majumdar and S. Chaudhary, 2006. Semantic and rules based event-driven dynamic web services composition for automation of business processes. Proceeding of the IEEE Services Computing Workshops (SCW'06). Chicago, IL, pp: 175-182.

Rui, W., G. Chaitanya, P. Maryam, Y. Haseeb, A.M. John, T.K. Eileen and C.K. Jessica, 2011. Web service composition using service suggestions. Proceeding of the IEEE World Congress on Web Services, pp: 482-489.