# FUNCTIONAL SEMANTICS AWARE BROKER BASED ARCHITECTURE FOR E-LEARNING WEB SERVICES

Demian Antony D'Mello

Department of Computer Science & Engineering, St. Joseph Engineering College, Vamanjoor, Mangalore -575 028. INDIA
demian.antony@gmail.com

**ABSTRACT**

*E-learning enables the learner to gain diverse knowledge anytime, anywhere and on any device. Learning resources (objects) and resource providers play a very important role in e-learning applications/systems. The increasing demand for interoperability in existing heterogeneous e-learning systems to support accessibility and reusability is the most challenging research issue. Web services and SOA enables interoperability between heterogeneous applications over the Web. To adopt Web services technology towards the reusability and aggregation of e-learning services, the conceptual Web services architecture and its building blocks need to be augmented. In this paper, a well formed functional semantics approach is proposed to describe e-learning Web services providing variety of learning objects/resources. The paper presents an extendible functional knowledge to map the learner's or provider's versions of service descriptions into a standard form called Abstract Description. The authors propose a broker based e-learning Web service architecture which facilitates effective e-learning service publishing and discovery mechanisms. The paper explores a scheme to extend the WSDL 2.0 document in order to incorporate functional semantics of e-learning Web services and their operations. The paper presents an e-learning service knowledge called Learning Operation Tree (LOT) for the quick e-learning service discovery. The experimentation shows that, the proposed broker based architecture for e-learning Web services facilitates effective discovery with moderate performance in terms of recall and response.*

**KEYWORDS**

*E-Learning; Functional Semantics; Broker Architecture; Discovery; Learning Object; Learning Operation Tree*

## 1. INTRODUCTION

With the rapid growth of Internet technologies, knowledge can be shared, retrieved and distributed through the Web all over the world. Currently, the educational system has been evolved into electronic learning (or e-learning) that enriches the traditional learning system by providing the autonomous learning for learners and allowing learner to learn anywhere and anytime [1]. E-learning is a technology driven learning process based on Web technology. E-learning is also defined as an acquisition, application and dissemination of knowledge facilitated primarily by electronic means. The rapid growth of computers and access to Internet made it possible to bring the concepts like virtual university or twenty four hour learning in reality [2]. Presently, e-learning is a major kind of knowledge and information sharing method which allows the learner to access different kinds of learning resources such as lecture video, teaching audio, lecture slides/handouts, e-books, downloadable application tools and software [3]. E-learning is also a type of education that offers some interesting benefits over traditional learning in terms of independence. The learners can work anywhere and communicate with instructor or other learners via e-mail, electronic forums, chatting, video conferencing and other forms of computer and Web based communication [4]. In e-learning, the reuse of learning objects/resources and services is a key issue and standardizing e-learning technology is taking place in order to overcome interoperability problems [5]. Reusing of existing resources and infrastructures to implement an e-learning system can reduce the overall operational cost of the system which is ideal for learners to exploit various e-learning facilities.

In e-learning domain, the learning resources (Learning Objects) and their providers play a very important role in E-learning applications/systems [6]. Content consumed by learners and created by authors/educators is commonly handled, stored, and exchanged in units of learning objects (LOs). Basically, LOs are units of study, exercise or practice that can be consumed in a single seamless session. They represent reusable granules that can be authored independently of the delivery medium and be accessed dynamically over the network [4]. For example, a LO on the basics of C++ can be used in classes on OOPs and Programming Languages. As the number of resources and providers grow, meta-data on the resources becomes a critical factor. A meta-data is needed for an appropriate description of learning objects so that plug-and-play configuration of knowledge dissemination is enabled. To this end, several standardization efforts have been initiated which include Learning Object Metadata (LOM) and SCORM, which is a collection of specifications adapted from multiple learning sources to provide a comprehensive suite of e-learning capabilities that enable interoperability, accessibility, and reusability of Web based learning content [7].

Web services technology enables the reuse and interoperation among heterogeneous platforms. It provides mechanisms for the description and search/lookup of computational entities over the Internet. A Web service is an interface, which describes a collection of operations that are network accessible through standardized XML messaging [8]. Web service discovery is the mechanism, which facilitates the requester, to gain an access to Web service descriptions that satisfy his functional requirements. UDDI [9] is the early initiative towards discovery, which facilitates both keyword and category based matchmaking and discovery.

### 1.1. Motivation

The existing UDDI [9] and WSDL 2.0 [10] do not support e-learning service specific functional descriptions for the discovery of e-learning resources. In order to publish e-learning service description for the global access, the existing WSDL structure need to be augmented to incorporate necessary information required for the discovery. As a motivating example, consider the scenario of learner who is interested to download the study material related to J2EE. In order to get correct e-learning resource, the learner visits and evaluates learning content present in many Web pages using popular search engines like Google. The search engines provide the listing of Web links of related material to the learner which makes him to verify genuineness of the learning content. Moreover, the search engines do not distinguish Web services and Web pages which again make him to choose only e-learning Web services for the semi-dynamic binding requirements. Thus, the e-learning Web services need to be stored at central repository towards consistent updating and global access. In order to initiate effective lookup for the specific e-learning services, the learner needs to describe his learning requirements in a specific form. Moreover, the providers of e-learning services have to publish all e-learning services at the central repository with well accepted description formats and procedures. This well-formed description of learning Web services and learning requirements will improve the hit rate of the lookup/discovery mechanism for the static and dynamic binding of e-learning resources/objects. Therefore, the existing Web service architecture and its major building blocks (UDDI and WSDL) need to be augmented to enable effective e-learning service/resource discovery.

### 1.2. Related Works

E-learning has been a topic of increasing interest in recent years. In literature, the authors of the paper [1] propose the reference architecture for interoperating the existing e-learning system with the help of Web services. The authors also explore a metadata-UDDI model which is designed as a core component of the architecture. The paper [11] proposes an extensible SOA based platform that facilitates implementation of e-learning systems. The platform has applied a service-oriented framework and model driven architecture into the analysis, design, implementation and integration of e-learning applications. In literature, the researchers also explored the application of semantic Web technologies to e-learning domain. The paper [12] illustrates the use of semantic Web technologies such as RDF to e-learning system for helping learners to select suitable learning course or retrieving relevant information. In literature, annexed algorithm, called eLSDAUS, is proposed to improve the existing semantic-based matchmaking algorithm [13]. The proposed algorithm, Introduces a new factor called "User Satisfaction", which is the user's level of satisfaction about the result of service discovery. This algorithm allows users to take part in the process of e-Learning service discovery, and evaluate the result of service discovery. The authors in [14] present an SWS architecture which is based on Web Service Modeling Ontology (WSMO) and uses the Learning Object Metadata (LOM) Standard to describe the learning object/resource. The paper [15] analyses *three* of the most successful e-learning platforms (Blackboard, Moodle and Sakai), identifying their Web services, and comparing their readiness for the development of a virtual campus based on these services. The authors also provide a mechanism to facilitate the integration of these platforms in an information technology infrastructure.

The researchers also explored the way to store e-learning resources and their descriptions into repository. The authors of paper [6] illustrate the design and implementation of a distributed learning resource registry system. They define Distributed Learning Resource Registry and Discovery Model, which enable the developers and repository systems to register learning resources into the registry system and provide a discovery mechanism to find required learning resources. The paper [16] proposes a metadata model for indexing the learning services. The authors propose to describe and index learning services with three dimensions: as learning resources, as services that contribute and help researchers and as general services.

Current e-learning frameworks should take advantages of Web services and intelligent agents. The paper [17] proposes a novel architecture for E-learning systems based on Web services and intelligent agents. This architecture provides a flexible integration model in which all the learning components and applications are loosely connected and can be distributed on the Internet. In addition, through the use of agents, learning content can be intelligently customized to fit the context and the special learning needs of particular users. The authors "Guo W. and Chen D." present the main features of e-learning scenario and setup an e-learning scenario ontology with the training and application domains. The authors also present semantic querying and semantic mapping approach for the query containing learning requirements. The paper [18] discusses three aspects of E-learning system: theoretical framework, function components and technology architecture. The authors provide both theoretical grounding and practical advice for designing and implementing effective E-learning system in each of these areas. The authors of [19] propose a Web Services based solution to exchange learner's information among different e-learning systems described following a Web Ontology. The proposed solution makes different e-learning systems to cooperate with each other in order to reach a set of learner information richer than that currently found in standard e-learning systems.

E-learning system provides a set of personalization functionalities such as personalizing learning plans, learning materials, test and necessary instant messages etc., to online learners [20]. The problem in the existing system is the lack of personalization due to weak-semantic learning resources. The possibilities of personalized searching for information will be improved, with the advance of the semantic web and available web services. The authors of the paper [20] present an approach to e-learning personalization based on ontology and information exchange is maintained by web services based on Service-oriented architecture. With this mechanism, the learning process is enhanced by providing personalized learning content to the learners in an effective and dynamic intelligent way.

Adopting Web 2.0 technologies and techniques in modern e-learning systems guarantees a more interactive e-learning experience [21]. It leverages collaboration among learners and enhances accessibility to various learning resources. The paper [21] identifies recurrent Web 2.0 and Service-oriented architecture (SOA) design and architectural patterns that would provide reusable building blocks for any Web 2.0 based service-oriented e-learning system. The proposed design patterns share three elementary architecture types, client-server, peer-peer and SOA. The paper also builds on UML4SOA techniques in modeling requirements prior application of proposed patterns in the case study. The paper [22] tries to support a personalized strategy customized for programming course. The authors have designed a model for personalized learning. It included an information model and a process model which has a great adaptability for strategy and strategy combination. The proposed service can adapt to the change of strategies, not only to a rule's change, but also to the change of entire strategy plan. So far there is no effort in the literature from the researchers to apply restricted natural form of functional description to e-learning Web services/systems. Moreover, the concrete e-learning architectures with effective mechanisms for discovery and publishing are today's needs which are not addressed by the e-learning research community

### 1.3. Contributions

The authors provide effective solutions for the key issues with respect to e-learning Web service description and discovery. The key contributions of this paper are:

- Definition of functional semantics terminology and well-formed semantic rules for the description of e-learning services.
- The design of domain dependent, extendible e-learning functional knowledge for the effective e-learning service registration/publishing.
- The extension of WSDL 2.0 document structure to accommodate functional descriptions of e-learning services in WSDL documents.
- The design of e-learning Web service knowledge called Learning Operation Tree (LOT) to enable quick discovery.
- A broker based architecture for the e-learning services discovery and publishing.
- Matchmaking mechanism for the e-learning service discovery based on the functional semantic descriptions of learner's information/knowledge requirements.

### 1.4. Structure of the paper

The paper is structured as follows: In the next section (Section 2), the paper describes the functional semantics terminology for e-learning services. Section 3 defines the e-learning domain knowledge structure which stores e-learning service specific descriptions for matchmaking. Section 4, presents e-learning service knowledge which is a tree structure to store all published e-learning services and their operations. Section 5 presents the e-learning service publishing using augmented WSDL 2.0 involving functional semantics. In section 6, the authors propose the broker based architecture for effective e-learning Web services description, publishing and discovery. Section 7 presents implementation of the broker architecture in .NET environment, experimentation dataset followed by results. Section 8 draws the conclusions and provides potential areas for further work.

## 2. FUNCTIONAL SEMANTICS FOR E-LEARNING SERVICES

Web service i.e. service is the globally accessible software whose functionality can be embedded within another application. In order to promote e-learning service reusability, the e-learning services need to be described in a precise way. This section of the paper presents the e-learning service description semantics to describe learner's request and e-learning services in a precise way. E-learning service is a network accessible system interface having collection of operations/functionalities that aim at providing some resources or information to the learners.

Thus e-learning service operation is nothing but the execution of appropriate learning action on specific learning object to provide information to the learner. The functionality of any e-learning service operation can be characterized using learning action on the learning object. The following definitions help to frame the functional semantics to describe e-learning services.

## 2.1. Definition of Functional Semantics for E-Learning

The functional semantics approach uses the natural way of expressing the functionality of Web services and their operations. The functionality of an e-learning service/operation is described in terms of learning action, qualifier, learning object(s) and noun.

**Generic Action:** Generic action is an action used to perform an operation on learning object or to get information in terms of learning object. For example “check teacher availability” description involves the generic action “check” which is commonly used across multiple domains.

**Learning Action:** Learning action is an action performed on learning object in a learning domain to render service to the learner. The examples for learning action are: Learning action normally has a related action noun which can be used to describe the operation functionality. For example, the learning operation description “load question set” involves an action “load” which is learning action.

**Learning Object:** Learning object is an object of e-learning service for which the required action is sought by the e-learning service/operation. For example, teacher, author, student, question, lecture, test, book, reference are some objects found in the e-learning domain.

Learning objects are classified as *Main Objects* and *Related Objects* based on the association among them. The main objects are the entities (objects) that constitute the e-learning service domain. For example course, subject, topic, question, assignment, seminar, book, lecture are the few main objects. The related object is a logical or physical part of the major object in e-learning domain. For example set, bank, material etc. are the few related objects.

**Learning Noun:** Learning noun is a noun used to describe the e-learning Web service or its operation. For example conduction, issuing, sending, loading etc. are the learning nouns used in e-learning domain.

Learning nouns are classified as Action Noun and Simple Noun (generic noun) based on the action represented by the learning noun. An action noun is a learning noun which has a related learning action. A simple (generic) noun doesn’t represent any learning action on the learning object. For example, the noun “conduction” is an action noun as it has a related specific action “conduct” whereas the learning noun “availability” (in case of check course availability/check text book availability”) is a simple noun, since it does not represent any learning action.

**Qualifier:** Qualifier is a word which adds the value to the learning object i.e. qualifier specifies the feature or nature of the learning object. For example, the operation description “find text book” has a qualifier “text”. Similarly, the description “check crash course availability” contains the word “crash” which is a qualifier.

**Operation Description:** Operation description refers to the functionality description of an operation which is found in the WSDL document of e-learning Web service during e-learning service advertisement. The operation description represents one of the abstract learning operations supported by the advertised e-learning Web service.

**Abstract Operation:** Abstract operation is a single, compact and complete description for the multiple and similar descriptions of e-learning Web services or their operations. All service operation descriptions are transformed to their corresponding abstract operation(s) during e-learning service advertisement for the effective discovery. The important property of abstract operation description is that, it does not allow generic learning action together with an action noun to be present in its description.

## 2.2. Describing e-Learning Service Operations

The functional semantics approach facilitates the provider and learner of e-learning service to use natural form of describing the e-learning service functions i.e. operation descriptions. Thus functionality of an abstract e-learning operation can be described in the following *three* formats:

- Operation description = (Leaning Action) (Qualifier)$^*$ (Learning Object)$^+$ (Simple Noun)
- Operation description = (Learning Action) (Qualifier)$^*$ (Learning Object)$^+$
- Operation description = (Qualifier)$^*$ (Learning Object)$^+$ (Action Noun)
- Operation description = (Generic Action) (Qualifier)$^*$ (Learning Object)$^+$ (Action Noun)

The grammar rules to validate the abstract operation functionality are designed as follows. Let $A_S$ be the set of terminal symbols representing various learning actions i.e. $A_S$ = {load, display etc.}. Let $A_G$ be the set of terminal symbols representing various generic actions i.e. $A_G$ = {do, perform, get, check etc.}. Let O be the set of terminal symbols representing various learning objects i.e. O = {lecture, book, author, teacher, slide, handout, note, summary etc.}. Let Q be the set of qualifier symbols representing various qualifiers i.e. Q = {text, reference, crash etc.}. Let $N_A$ be the set of terminal symbols representing action nouns i.e. $N_A$ = {loading, conduction, teaching etc.}. Let $N_S$ be the set of terminal symbols representing simple nouns i.e. $N_S$ = {availability etc.}. Let $A_G$, $A_L$, $N_S$ and $N_A$ represent generic action, learning action, simple noun and action noun respectively. Let S be the start symbol and X, Y and Z be the non-terminal symbols. The production rules of the grammar to validate functional semantics of Web service operation are presented in Figure 1(a). An example of generating (leftmost derivation) operation description "load text book" using the grammar production rules is depicted in figure 1(b).

$S \rightarrow X\,Y\,Z \mid X\,Y \mid Y\,K$
$X \rightarrow A_G \mid A_L$
$Y \rightarrow O\,Y \mid Q\,Y \mid O \mid Q$
$Z \rightarrow N_A \mid N_S$
$K \rightarrow N_A$

(a) Grammar rules

$S \rightarrow X\,Y$
=> $A_L$ Y
=> load Y
=> load Q Y
=> load text Y
=> load text O
=> load text book

(b) Production sequence for "load text book"

Figure 1. Grammar to Validate Learning Operation Description

Consider the learning scenario; the following e-learning operation descriptions follow the rules of functional semantics.

a) check course availability
b) download reference material
c) upload course assignment
d) course assignment uploading
e) start reference material downloading

## 2.3. Pre-processing of E-Learning Operation Descriptions

All operation descriptions are preprocessed before being mapped into abstract e-learning operations. The preprocessing and mapping mechanism facilitates the effective publishing and discovery of e-learning services. The preprocessing of operation descriptions involves elimination of co-occurrence of generic action and learning action noun present in the operation description. The following rules guide the preprocessing of operation descriptions.

**Rule I.** If the learning action noun is present along with the generic action, then the generic action is replaced by the learning action which is related to the learning action noun and the action noun is eliminated from the description.

**Rule II.** If the learning action noun is found in the operation description without a generic action then the related learning action of the action noun is used, instead of the action noun.

As an illustration, consider the operation description “perform lecture downloading”. The description contains generic action and action noun. The generic action is now replaced by “download” which is the specific action of action noun “downloading” and the generic action is eliminated from the description as per Rule I. This results in abstract operation description “load lecture”. Similarly, the operation description “test conduction” is transformed into “conduct test” by Rule II.

## 3. E-LEARNING DOMAIN KNOWLEDGE STRUCTURE

To perform e-learning service discovery based on the functional description of Web service operation, we design an extendible functional knowledge which contains interdependent knowledge structures to represent the complete functional knowledge for all categories of Web services. The interdependent knowledge structures are: *Object List*, *Action List*, *Qualifier List* and *Noun List*.

**Object List:** Object list is a sorted list with finite elements where each element contains *four* fields i.e. information items. They are- *object name*, *object identifier*, *object type* and *a pointer* to the sorted related object list having similar/related names of a specific object. The object name refers to learning object for which learning action is to be sought, object identifier is a unique identification string and object type refers to either main (M) or related object (R). The object list and related object list can be implemented as dynamic array which is sorted based on the object name.

**Action List:** Action list is a sorted list with finite elements each containing *three* fields namely *action name*, *action identifier* and *a pointer* to the sorted related action list containing similar action words for a specific learning action. The action list and related action list can be implemented as a dynamic array.

**Qualifier List:** Qualifier list is a sorted list with finite elements each containing *three* fields namely *qualifier name*, *qualifier identifier* and *a pointer* to the sorted related qualifier list containing similar qualifier words for a specific qualifier. The qualifier list and related qualifier list can be implemented as a dynamic array.

**Noun List:** Noun list is a sorted list with finite elements each containing four fields namely *noun name*, *noun identifier*, *noun type*, *a pointer* to its corresponding action (if any) and a pointer to the sorted related noun list containing similar noun words used to describe a specific learning noun. The noun list and related noun list can be implemented as a dynamic array sorted based on the noun name. The noun type refers to noun categories, learning action noun (A) and simple noun (S).

Figure 2 depicts the partial e-learning domain knowledge structure showing interdependent structures or lists having information of e-learning domain. The noun list contains two learning action nouns and one simple noun. The object list has an object “set” which is categorized as a related learning object. The unique identifiers like action identifier, noun identifier, qualifier identifier and learning object identifiers are fixed length strings used to identify the learning actions, nouns, qualifiers and learning objects. E-learning domain knowledge is augmented by the e-learning service providers in order to improve the discovery rate of their advertised services. In order to transform the description of e-learning service operation to its equivalent abstract operation, a separate list is maintained called *Abstract Operation List (AOL)*. The structure of AOL is defined below.

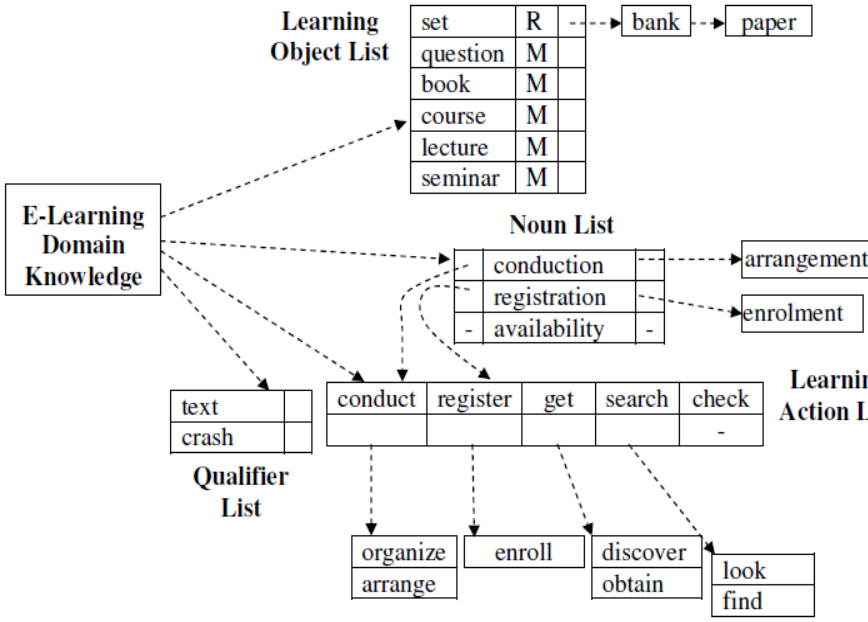


Figure 2. E-Learning Domain Knowledge Structure for Discovery

**Abstract Operation List:** Abstract operation list is a sorted dynamic array with finite elements each representing an abstract e-learning service operation. The element contains *operation identifier, operation pattern* and *e-learning service count;* where operation pattern is a string of finite length which contains fixed length identifiers of learning objects, noun qualifiers and learning actions. The e-learning service count refers to the number of e-learning Web services having description of operation which maps to an abstract operation.

Operation pattern is generated for each abstract operation defined in AOL. Let M be the fixed length for identifiers of learning actions, nouns, qualifiers and learning objects. The first M characters represent the action identifier. Next, the sets of M characters represent the qualifier identifiers (optional), finally the sets of M characters represent the learning object identifiers followed by the noun identifier (optional).

## 4. E-LEARNING SERVICE PUBLISHING WITH WSDL 2.0

The operations of e-learning services are described using the functional semantics as defined in section 2.2. Thus e-learning Web service can be described for publishing using functional semantics to facilitate the effective search.

### 4.1. Describing E-learning Web Services

Let Profile (WS) be the profile of the e-learning Web service to be published through the search agent into augmented UDDI registry.

Profile (WS) = {service-desc, binding-desc} where, service-desc refers to e-learning service specific descriptions like service name, provider name, operation descriptions etc and binding-desc refers to binding details like URL for the access. Thus service-desc = {service-name, provider-name, $OP_{List}$} where, $OP_{List}$ is the list of e-learning operations and their descriptions supported by the e-learning Web service.

The $OP_{List}$ = {$opr_1$, $opr_2$…$opr_N$} where, $opr_i$ is the description of an e-learning operation.

The description of each operation is: $opr_i$ = {opr-name, desc-list, info-list}; where desc-list is the functional semantics description of operation as defined in section 2.2 and info-list is additional information (optional) to update the extendible e-learning domain knowledge.
desc-list={learning action, qualifier(s), learning object(s) , noun} where, qualifiers and noun are optional.

info-list = {action-set, qualifier-set, learning object-set, noun-set} where, action-set contains similar learning action words, qualifier-set contains similar qualifier names for a given qualifier and object-set contains similar learning object names and noun-set contain similar noun names.

As an illustration, let us consider the description of “summer course registration” with an abstract operation “register course”.
Profile (Learning Service) = {service-desc, binding-desc}.
service-desc={“summer course registration”, “VTU”, $OP_{List}$}.
$OP_{List}$={$opr_1$} and $opr_1$={register course, $desc\_list_1$, info-list}.
$desc\_list_1$={action, qualifier, object, noun} and action={register}, object={course}, qualifier={summer} and noun={Φ}.
Info-list={action-set, object-set}.
action-set={enroll} and object-set={$object_1$} and $object_1$= {subject, topic}.
qualifier-set={$qualifier_1$} and $qualifier_1$={vacation}.
noun-set = {Φ}.

## 4.2. WSDL 2.0 Document Structure

WSDL 2.0 [10] separates the description of a Web service's abstract functionality from the concrete details of how and where that functionality is offered. This separation facilitates different levels of reusability and distribution of work in the lifecycle of a Web service and the WSDL 2.0 document that describes it. Figure 3 shows the abstract structure of WSDL 2.0 document.

```
<?xml version="1.0" encoding="utf-8" ?>
<description …>
    <documentation> ………</documentation>
    <types>
        <xs:schema …>……..</xs:schema>
    </types>
    <interface name =….>
        <fault name…../>
        <operation name =…….>
            <input messageLabel =…../>
            <output messageLabel=…../>
            <infault ……/>
            <outfault…./>
        </operation>
    </interface>
    <binding name =…….>
        <fault ref =……/>
        <operation ref = ……./>
    </binding>
    <service name = ……>
        <endpoint name =……/>
    </service>
</description>
```

Figure 3. Structure of WSDL 2.0 Document

Every WSDL 2.0 document has a *description* element as its top-most element. This merely acts as a container for the rest of WSDL 2.0 document, and is used to declare namespaces that will be used throughout the document. The *documentation* element is used to present the brief information about the document and the service. The documentation element allows the WSDL 2.0 author to include some human-readable documentation inside a WSDL 2.0 document. It can appear in number of places within the description element.

WSDL 2.0 allows message types to be defined directly inside the *types* element, which is a child of the description element. A WSDL 2.0 *interface* defines the abstract interface of a Web service as a set of abstract operations, each operation representing a simple interaction between the client and the service. Each operation specifies the types of messages that the service can send or receive as part of that operation. Each operation also specifies a message exchange pattern that indicates the sequence in which the associated messages are to be transmitted between the parties. The *binding name* specifies the concrete message format and transmission protocol details for an interface. A WSDL 2.0 *service* element specifies a single interface that the service will support and a list of endpoint locations where that service can be accessed.

## 4.3. Extension of WSDL 2.0 for E-Learning Web Services

WSDL 2.0 structure is extended to publish the e-learning Web services with functional semantics as follows. Table 1 provides the newly defined XML elements and their description.

Table 1. XML Elements for WSDL 2.0 Structure

| Element Name | Purpose |
|---|---|
| <operationDesc> | Container for functional description of learning operation |
| <operationList> | Container for all operation descriptions of learning Web service |
| <operation> | Container for descriptions of learning operation |
| <operationName> | To represent learning operation name |
| <semantics> | Container for functional description |
| <action> | To represent learning action |
| <object> | To represent learning object |
| <qualifier> | To represent qualifier |
| <noun> | To represent noun |
| <information> | Container for the information to augment functional knowledge |
| <related> | Represents related words for action, noun, qualifier and objects |

The *documentation* element is chosen to insert the information which is necessary for the effective service discovery into WSDL. A new tag called *operationDesc* is defined to insert the functional semantics of all abstract operations present in the e-learning service. The new elements *operationList*, *operation*, *action*, *qualifier*, *object* and *noun* are found within the element *operationDesc*. The new elements are defined in the XML schema which governs the structure of extended documentation element.

The functional semantics of an operation is defined within the element *semantics* and this element is placed within the element *operation*. The elements like *action*, *qualifier*, *object* and *noun* are used within *semantics* element which provides the functionality description of an abstract e-learning operation. The functional semantics for the illustration presented in section 4.1 is depicted in Figure 4.

# 5. E-LEARNING SERVICE KNOWLEDGE STRUCTURE

E-learning service can advertise multiple learning operations. To store such redundant e-learning service operations in an efficient way for the discovery, we define *two* data structures called Web service list (ESL) and Service Operation tree (LOT).

```
<?xml version="1.0" encoding="utf-8"?>
<description>
<documentation>
    <operationDesc xmlns:xf ="http://www.w3.org/2001/XMLSchema-Instance"
            xf:schemaLocation="http://www.sjec.ac.in/csedept/descSchema/desc.xsd"
            xmlns="http://www.sjec.ac.in/csedept/descSchema">
        <operationList>
            <operation>
                <operationName>Get Course Material </operationName>
                <semantics>
                    <action>get </action>
                    <object>course</object>
                    <object>material</object>
                </semantics>
            </operation>
            <operation>
                <operationName>Summer Course Registration</operationName>
                <semantics>
                    <qualifier>summer</qualifier>
                    <object>course</object>
                    <noun>registration</noun>
                </semantics>
            </operation>
        </operationList>
        <information>
            <action name="register">
                <related>enroll</related>
            </action>
            <noun name="registration">
                <related>enrollment</related>
            </noun>
            <object name="course">
                <related>topic</related>
                <related>subject</related>
            </object>
        </information>
    </operationDesc>
</documentation>
<types>…</types>
<interface>…</interface>
<binding url="www.elctroniceducation.com">…</binding>
<service name="Summer School"></service>
</description>
```

Figure 4. Extended WSDL for “Summer School” E-learning Service

**E-learning Service List (ESL):** E-learning service list is a sorted dynamic array having four fields namely, e-learning service key (es-key or ws-key), es-id (unique identifier generated by the broker), es-link and lot-link where es-id is a e-learning service identifier, es-link is a pointer to the e-learning service entry in ESL having same set of operations and lot-link is the pointer to the leaf node of Learning Operation Table (LOT) which corresponds to a link to its operations in the LOT or a pointer to the predecessor e-learning service in ESL having same operations.

**Learning Operation Tree (LOT):** A learning operation tree is a binary tree with each node consisting five fields. They are operation identifier (opr-id) which specifies the operation identifier of e-learning service operation (abstract operation); child pointer (child-link) which is a pointer to the remaining operations of a e-learning service; sibling pointer (sibling-link) is a pointer to the list operations which shares a common operation prefix; parent pointer (parent-link) is a pointer to its predecessor node; e-learning service link (esl-link) is a pointer to the ESL entry to which opr-id is the last learning operation in the sorted advertised operation list. The root node of LOT is labeled with T and has only child-link which points to Web various service operation sequences. The property of LOT is that at any node X of LOT, the opr-id at X will not be repeated at the child or sibling branch which is linked to X.

As an illustration consider *six* e-learning services having a total of *nine* learning operations to be advertised into repository. Let $Opr_1$ to $Opr_9$ be the operation identifiers of abstract operations obtained after mapping them into abstract operations. $ES_1 = \{Opr_1, Opr_2, Opr_3, Opr_4\}$, $ES_2 = \{Opr_1, Opr_2, Opr_3\}$, $ES_3 = \{Opr_5, Opr_6\}$, $ES_4 = \{Opr_7, Opr_8, Opr_9\}$, $ES_5 = \{Opr_5, Opr_6\}$ and $ES_6 = \{Opr_7, Opr_8, Opr_9\}$. Figure 5 shows the LOT and ESL after insertion of learning operations of e-learning services. The numbers within circles (node) indicate the operation identifiers ($Opr_1$ to $Opr_9$) in an ascending order. In ESL, $ES_1$ to $ES_6$ represents the e-learning service key which is same as Web service key.

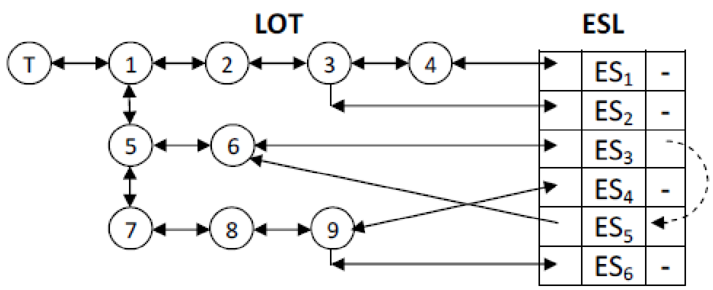


Figure 5. E-Learning Service Knowledge Structure

## 6. THE BROKER BASED ARCHITECTURE FOR E-LEARNING SERVICES DESCRIPTION AND DISCOVERY

The broker based architecture facilitates an effective discovery and publishing of e-learning Web services. Figure 6 depicts different roles and operations supported by the broker based e-learning service architecture.

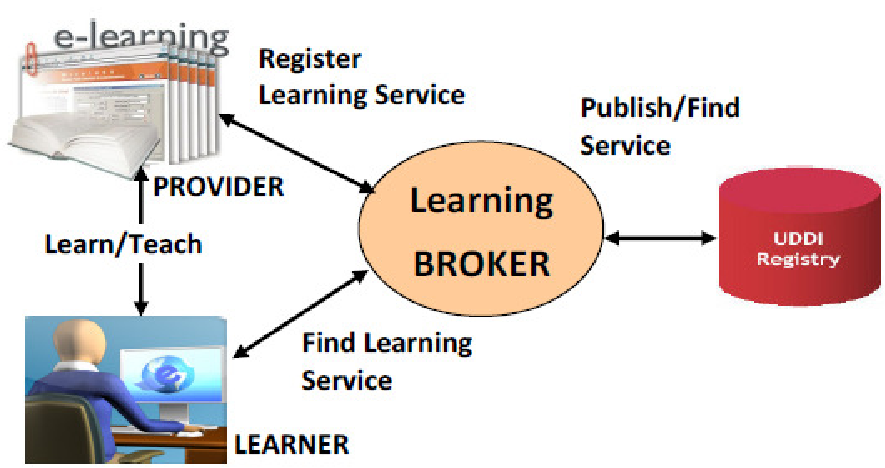


Figure 6. Learning Broker Based Architecture for E-learning Services

The architecture assumes that the e-learning service requesters and LO providers use functional semantics to describe learning operations and learning needs. An additional role is introduced by the authors to the conceptual Web service architecture [8] named *Learning Broker (broker)* and a new operations namely *Register Learning Service* and *Find Learning Service*. The broker is defined between Web service registry and learner (and provider) which facilitates the learner and provider to specify the needs and learning services in terms of functional semantics. The find learning service operation is defined between the broker and learner, which effectively explore the learning services from LOT. The register learning service operation is defined between the learning resource provider and broker for the e-learning service publishing.

## 6.1. Components of Learning Broker

The learning broker is designed with *four* internal components namely *Learning Service Publisher*, *Learning Service Finder*, *Learning Domain Knowledge* and *Learning Service Knowledge (Service Knowledge)*. Figure 7 depicts different components of a broker and the interactions among the components within the architectural boundary. The service publisher component facilitates the registration, updating and deletion of business and e-learning service related information. The main functionality of service finder is to discover the e-learning services which satisfy the learner's demands. The domain knowledge is an interlinked data structure which represents learning actions, qualifiers, learning objects and nouns of e-learning service domain. The service knowledge is an abstract representation of all published e-learning Web services and their learning operations.

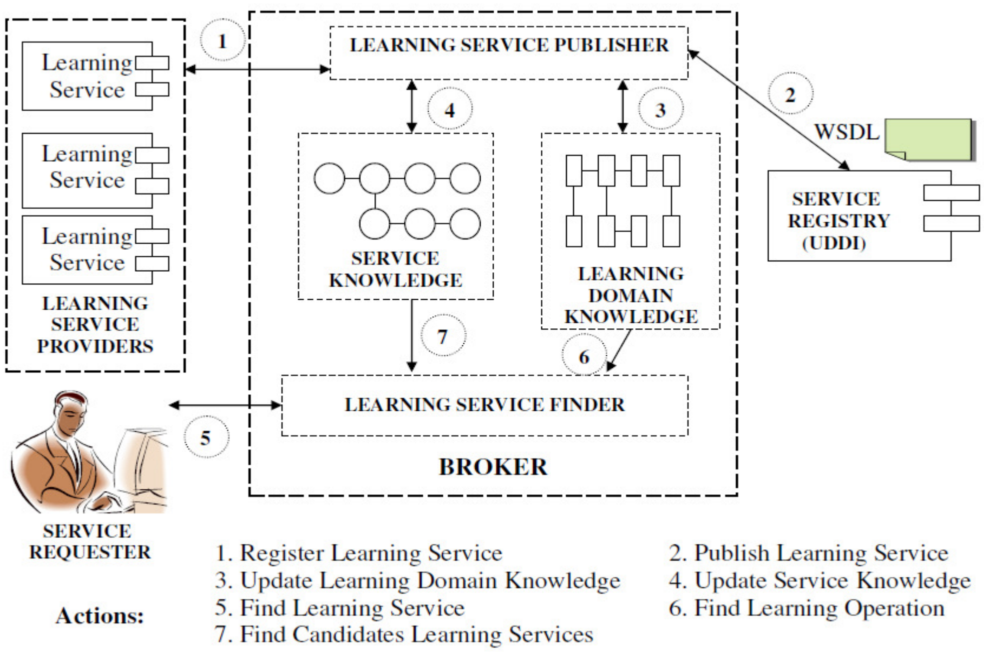


Figure 7. Components of E-Learning Broker and Component Interactions

The sequence of interactions among various architectural roles for e-learning service registration (publishing) is presented below.

1. The service publisher supplies the WSDL (with functional semantics information of learning operations) document to the broker along with other information pertaining to the e-learning service provider.
2. The broker publishes the service information into service registry and obtains the service key (es-key).

3. The broker now extracts functional description of all learning operations and updates the learning domain knowledge and service knowledge accordingly.
4. On successful service registration, the service key is returned to the e-learning service provider as an acknowledgment.

The sequence of architectural component interactions for learning service discovery is given below.

1. The learner sends the learning request as per the functional semantics format to the broker.
2. The learning service finder of the broker finds an abstract learning operation for the request by traversing the learning domain knowledge.
3. The broker now traverses the LOT to find possible learning services.
4. The broker now sends the service keys (and descriptions) of all discovered services to the learner.

Now the learner initiates a session with the provider towards execution of requested learning activity.

## 6.2. Publishing of E-learning Web Services

The provider of the e-learning service publishes the extended WSDL 2.0 into the UDDI (e-learning service registry) through the broker. The steps involved in the e-learning service publishing are presented below.

1. The provider registers the e-learning service by sending the WSDL and other necessary details.
2. The WSDL is processed by the broker to obtain the service name, binding details and the operation details like operation name and functional semantics.
3. The broker publishes the Web service into UDDI registry and obtains the service key.
4. The operation descriptions are preprocessed according to Rule 1 & 2 as defined in section 2.3.
5. The operation pattern is generated for the preprocessed operation description by obtaining the appropriate action, object, qualifier and noun identifiers from the e-learning domain knowledge. If the action/noun/qualifier and object is not present in the functional knowledge, then these are inserted by generating appropriate identifiers.
6. Search the operation pattern in ALOL. If found, return the operation identifier of the operation having the generated operation pattern otherwise insert the operation as a new abstract operation along with its pattern into ALOL.

The operation identifiers of all e-learning Web service operations along with service key are stored LOT to enable quick discovery.

## 6.3. E-Learning Web Service Discovery

E-learning service discovery for the learner's request and the matchmaking process is summarized below.

The learner sends the request, enriched with functional semantics to the broker for discovery.

1. The service finder of broker validates the functional semantics of learner's request (task to be carried out) using the rules as defined in section 2.2.
2. The learner's request is preprocessed according to Rule 1 & 2 (section 2.3) to retrieve the learning requirement.
3. The action list, qualifier list, object list and noun list of the e-learning domain knowledge are searched to get the corresponding identifiers. The unavailability of any identifier results in discovery failure.
4. The operation pattern for the request is formed using action, qualifier, object and noun identifiers.
5. After building the operation pattern, the pattern is searched in ALOL. If the pattern is found then the corresponding operation identifier is retrieved from the ALOL otherwise discovery failure is reported.

The LOT is traversed to search the requested learning operation identifier and all e-learning services with requested learning operation are returned to the learner as the suitable e-learning services/resources.

## 7. IMPLEMENTATION AND EXPERIMENTS

The proposed broker based e-learning service discovery mechanism is implemented on the Windows 7 platform using Microsoft Visual Studio .NET 2005 development environment and Microsoft visual C# as a programming language. The broker is designed and implemented as a Web service which is referenced in a standalone visual program. This visual program interacts with the e-learning resource/service provider and learner through different interface forms. The service repository is implemented as a Web service which in turn communicates with the SQL server 2005 database. The database table is created to store the information about the published e-learning Web services.

The e-learning service is published by activating the publish menu which pops out a window form where, the provider supplies the Web service information and attaches augmented WSDL of the e-learning Web service. The WSDL is processed by the agent program which stores and updates the necessary information in it. The broker also publishes e-learning Web service information into e-learning service repository. The service request processing is done by activating relevant interface form from the menu. The learner is expected to follow the functional semantic format while supplying learning request. The request is validated by the agent and the suitable Web service keys are obtained from its local store as the discovery result after successful matchmaking. Several experiments have been conducted to verify the effectiveness of functional semantics based matchmaking concept. We have also performed preliminary experiments to evaluate the system performance in terms of precision and recall.

Precision = (Relevant ∩Retrieved) / Retrieved
Recall = (Relevant ∩ Retrieved) / Relevant

The collection of 40 e-learning services having total of 65 distinct operations is used for experimentation. Thirty Three e-learning requests based on their short natural language descriptions were framed. From the experimentation it is observed that, the recall is less than 100% as the learner sometimes may not follow the functional semantics in a precise way. The mechanism also exhibit low precision if the published and requested e-learning operations are described with incorrect functional semantics. Figure 8 show the average recall values obtained for the different experiments.

Several experiments have been conducted to verify the effectiveness of functional semantics based matchmaking concept. The collection of 30 e-learning services having total of 45 distinct operations were used for experimentation. Fifty Three e-learning requests based on their short natural language descriptions were framed. From the experimentation it is observed that, the recall is less than 100% as the learner sometimes may not follow the functional semantics in a precise way. Figure 8 show the average recall values obtained for the different experiments.

The same set of e-learning services is represented using LOT at the broker. The LOT representation yields a compression ratio of 30%. The abstract learning operation of each learning service is stored in the main memory of the broker; the discovery mechanism need not load the entire page having the information of advertised services. This will substantially reduce the secondary memory access time which in turn improves the response time of the discovery mechanism. This is because the LOT acts as an index to the e-learning service descriptions present in the UDDI registry.

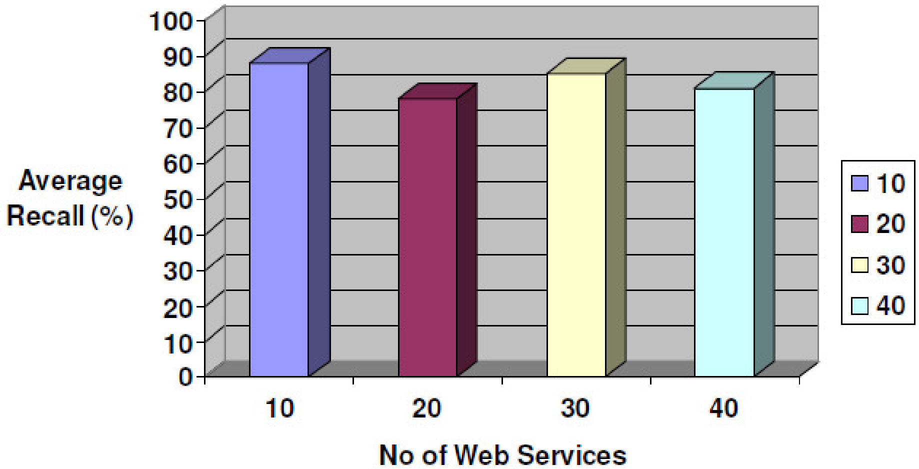


Figure 8. Performance Evaluation (Average Recall)

The empirical data reveals that, the proposed system has high Recall and 100% precision provided the learning requests are formed as per functional semantics rules. If the published or requested operation description does not follow the functional semantics (description is ill-formed) correctly, then the precision and recall of the proposed system suffers. In order to improve the recall of the e-learning discovery system, both the e-learning resource/service provider and the learner have to describe the published/requested e-learning operations as per the functional semantics rules.

## 8. CONCLUSION

E-learning service discovery is an important activity which explores multiple e-learning service services for the given learner's knowledge requirements. The authors propose a well-defined functional semantics to describe the e-learning services for publishing and lookup. The e-learning domain knowledge is designed which facilitates easy and effective service lookup and publishing. A compact index for service registry called *Learning Operation Tree (LOT)* is proposed which enables quick service lookup. The authors also propose broker based e-learning Web service architecture for discovery mechanism which finds the suitable e-learning services/resources for the given learning requirements. The broker architecture is implemented for experimentation and several experiments were carried out to deduce observations. The experimentation reveals that, the use of functional semantics in describing e-learning services and the use of compact service knowledge at the broker will improve the effectiveness (Recall, Precision and response) of e-learning service discovery.

## ACKNOWLEDGEMENTS

This research work is sponsored by "VTU Research Grants Scheme" of Visvesvaraya Technological University (VTU), Belgaum, Karnataka, INDIA (No. VTU/Aca./2009-10/A-9/11619, dated: 04/01/2010).

## REFERENCES

[1] Ngamnij Arch-int, Chidchanok Lursinsup and Peraphon Sophatsathit, "A Reference Architecture for interoperating existing e-Learning Systems using Metadata and Web Services Model", In Proceedings of the International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05), IEEE 2005.

[2] Afroza Sultana and Ishrat Sultana, "E-School: A Web-Service Oriented Resource Based E-learning System", In Proceedings of the International Conference on Networking and Information Technology, IEEE 2010.

[3] Jianhua Jiang, Shuxin Guo and Jie Zhang, "Architecture Design of Campus Information Convergence System for E-learning Based on Web Service Technology", In Proceedings of the 2nd International Conference on Future Computer and Communication, IEEE 2010.

[4] ZHU Zhen, "Design and Implementation of Web-services based E-Learning System", In Proceedings of the First International Workshop on Education Technology and Computer Science, IEEE 2009.

[5] Judith Rodríguez, Luis Anido and Manuel J. Fernández, "How can the Web Services Paradigm improve the E-learning?", In Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03), IEEE 2003.

[6] Zhao Gang, Yang ZongKai and Yan Kun, "Design and Implementation of a Distributed Learning Resource Registry System", In Proceedings of the Fourth International Conference on Computer and Information Technology (CIT'04), IEEE 2004.

[7] Gottfried Vossen and Peter Westerkamp, "E-Learning as a Web Service (Extended Abstract)", In Proceedings of the Seventh International Database Engineering and Applications Symposium (IDEAS'03), IEEE 2003.

[8] H. Kreger, "Web Services Conceptual Architecture (WSCA 1.0)", Published May 2001, [online] Available: www.ibm.com/software/ solutions/webservices/pdf/wsca.pdf, [visit: April 2007].

[9] Riegen, C.V. (Ed), 2002. UDDI Version 2.03 Data Structure Reference [online]. OASIS Open 2002-2003. Available from: http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.htm.

[10] David Booth and Canyang Kevin Liu, "Web Services Description Language (WSDL)" Version 2.0 Part 0: Primer, W3C Recommendation, Published: 26 June 2007, [online] Available: http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626.

[11] Zhifen Cheng, Tinglei Huang and Jia Nong, "An Extensible Development Platform for SOA-Based E-Learning System", In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, pp. 901-904, IEEE 2008.

[12] Wenying Guo and Deren Chen, "Semantic Approach for e-learning System", In Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06), IEEE 2006.

[13] ZHU Zheng-zhou, WU Zhong-fu and ZHOU Shang-bo, "An e-Learning Services Discovery Algorithm Based on User Satisfaction", In Proceedings of the International Conference on Networking, Architecture, and Storage, pp. 342-348, IEEE 2008.

[14] Mohamed Hanafy and Medhat Fakhry, "A proposed Architecture for Learning Object Selection and Discovery based on WSMO", In Proceedings of the 4th International Conference on Next Generation Web Services Practices, pp. 10-14, IEEE 2008.

[15] Huertas F. and Navarro A., "Web services availability in e-learning platforms", In Proceedings of the 7th International Conference on Next Generation Web Services Practices (NWeSP), Page(s): 170 – 175, IEEE 2011.

[16] Oussama Kassem Zein, Yvon Kermarrec, Serge Garlatti, Jean-louis Tetchueng and Sylvain Laube, "A Metadata Model for Web Services Applied to Index and Discover E-Learning Services", In Proceedings of the International Conference on Information and Communication Technologies (ICTTA '06), IEEE 2006.

[17] Xu Wei and Jun Yan, "An E-learning System Architecture Based on Web Services and Intelligent Agents", In Proceedings of the Ninth International Conference on Hybrid Intelligent Systems (HIS '09), Page(s): 173 – 177, IEEE 2009.

[18] Zhijun Rong, Kuisheng Chen, Min Zhou and Jinsong Xiao, "Designing and Implementing the E-learning system", In Proceedings of the 2008 International Conference on Computer Science and Software Engineering, pp. 797-800, IEEE 2008.

[19] Daniela Leal Musa, LydiaSilva Muñoz and Moreira de Oliveira J P, "Sharing Learner Profile through an Ontology and Web Services", In Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA'04), IEEE 2004.

[20] Rani, S.J., Ashok, M.S. and Palanivel, K., "Adaptive content for personalized E-learning using web service and semantic Web", In Proceedings of the International Conference on Intelligent Agent & Multi-Agent Systems (IAMA 2009), Pp. 1-4, IEEE 2009.

[21] Hussein D., Alaa G. and Hamad A, "Web 2.0 Based Service-Oriented E-Learning Systems: Recurrent Design and Architectural Patterns", In Proceedings of the 11th ACIS International Conference Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), Page(s): 227 – 234, IEEE 2010.

[22] Zhang jinghua, "A semantic web based personalized learning service for programming course in e-learning", In Proceedings of the International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), pp. 1563 – 1566, IEEE 2011

**Authors**

Demian Antony D'Mello received his Bachelor degree in Computer Engineering in 1999 from Mangalore University, India and his Master degree in Computer Science and Engineering in 2003 from National Institute of Technology Karnataka (NITK), Surathkal, India. He completed his PhD from the Department of Information Technology, NITK in the year 2010. Presently, he is working as a professor in the Department of Computer Science and Engineering, St. Joseph Engineering College, Mangalore, India. His research interests are in the areas of Web technologies, Web services, Distributed Computing and Cloud Computing.