

The Learning Service Bus Based on SOA

Xiaoqiang Liu and Yingpei Wu

Abstract--In this paper, the concept of the LSB (Learning Service Bus) is presented, and the SOA (Service-Oriented Architecture) is led to the architecture design of LSB, which includes LSDL (Learning Service Description Language, UDDI (Universal Description, Discovery, and Integration), SOAP (Simple Object Access Protocol) and QoS (Quality of Learning Service), in which the XML, a kind of transplantable data over the internet, was used in the LSB. The architecture of LSB has some characteristics such as standardization, extensibility, flexibility, reliability, plug and play, and customization.

Keywords: LSB, SOA, LSDL, UDDI, SOAP, QoS.

I . INTRODUCTION

With the rapid development of IT technology a lot of universities spread their educational activities to other countries, for example, distance education and e-learning.^[1-4] Accordingly, a lot of information technologies have been used in learning and support management,^[5-8] for instance, CSCL(Computer Supported Collaborative Learning)^[9] and middleware.^[10] Up to now, globalization of education has been given more attention. Predictability, the real learning activities and management must be built on the internet, in other words, it must be transformed into a series of web services. Meanwhile, SOA has become the main stream of software system design.^[11]

Xiaoqiang Liu is with the School of Software in Nanchang University, Ph.D and Professor.
E-mail:liu_xiaoqiang@hotmail.com

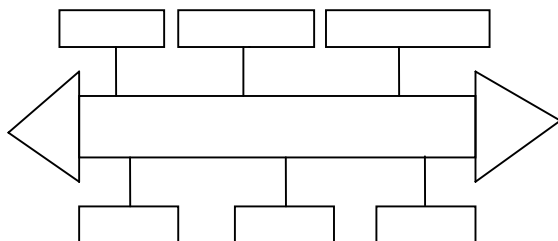
Yingpei Wu is with the School of Software in Nanchang University, software engineer and graduate. E-mail:
wyp962_2006@163.com

How can we combine SOA with learning service to build a new pattern of learning management? Consequently, seen from the view of SOA, the relevant standards and specification of the learning service description language, the naming and search, the message sending and the quality assessing should be considered.

Compared with computer bus architecture, the architecture of the Learning Service Bus (LSB) also has same characteristics such as standardization, extensibility, reliability, flexibility, plug and play. Additionally, service on demand is one important characteristics of LSB, which means that the architecture is alterable with the user's requirement, so called customization. The LSB is a logical architectural construct, which provides inter-connectivity services, and services interact with each other based on the quality of service requirements of the individual transactions.

- LSB connects and integrates an university's IT resource. In different locations, using different transports, across organizations.
- LSB mediates service requests and responses, performs transformation and routing, and enables connection type transparency.
- LSB enables the use of multiple protocols e.g. SOAP/HTTP can be converted to SOAP/JMS and vice versa.

II. THE ARCHITECTURE OF LSB



TM-----Teacher Management
 CM-----Course Management
 DM-----Distribution Management
 SM-----Student Management
 TPM-----Learning Plan Management
 Assess-----Evaluation of learning service
 Besides above building block, the LSB contains cost management, statistics & analysis and other component with extending of learning service.

III. THE LEARNING SERVICE DESCRIPTION LANGUAGE

Learning services are described using a standardized interface (metadata) that defines the service, its operations, and input and output parameters and how the service is reached and its location.

- Input and outputs may be simple parameters or XML documents
- An existing application can be given a service 'façade', e.g. an EJB, is then described as the service

LSDL is an XML document that describes a learning service using a number of key elements: A Port Type defines **what** the service does, and is described by a number of operations. For each operation the data that it receives and sends is described by a Message;

The Binding defines **how** the service (as described by the Port Type) is invoked e.g. SOAP/HTTP, SOAP/JMS;

The Port specifies the address **where** the service is located;

Learning services interact by exchanging data over the LSB. An example of LSDL on course selecting is following:

```
<?xml version="1.0"?>
<definitions name="CourseSelecting"
targetNamespace="http://example.com/
course.LSDL"
xmlns:tns="http://example.com/
course.LSDL"
xmlns:xsd1="http://example.com/
course.xsd"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns="http://schemas.xmlsoap.org/LSDL/">
  <types>
    <schema
targetNamespace="http://example.com/
course.xsd"
xmlns="http://www.w3.org/2000/10/XMLSchema"
  >
      <element name="courseTypeRequest">
        <complexType>
          <all>
            <element name="courseID"
type="string"/>
          </all>
        </complexType>
      </element>
      <element name="courseType">
        <complexType>
          <all>
            <element name="type"
type="float"/>
          </all>
        </complexType>
      </element>
    </schema>
```

```

</types>

<message name="GetLastCourseTypeInput">
  <part name="body" element="xsd1:
courseTypeRequest"/>
</message>

<message
name="GetLastCourseTypeOutput">
  <part name="body"
element="xsd1:TradePrice"/>
</message>

<portType
name="CourseSelectingPortType">
  <operation name="GetLastCourseType">
    <input
message="tns:GetLastCourseTypeInput"/>
    <output
message="tns:GetLastCourseTypeOutput"/>
  </operation>
</portType>

<binding
name="CourseSelectingSoapBinding"
type="tns:StockQuotePortType">
  <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"
"/>
  <operation name="GetLastCourseType">
    <soap:operation
soapAction="http://example.com/GetLastCourse
Type"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
</binding>

<service name="CourseSelectingService">

```

```

  <documentation>My first
service</documentation>
  <port name="CourseSelectingPort"
binding="tns: CourseSelectingSoapBinding">
    <soap:address
location="http://example.com/CourseSelecting"/
>
  </port>
</service>
</definitions>

```

IV. THE NAMING AND DISCOVERY OF LEARNING SERVICE

A Service Registry or Directory publishes service descriptions, finds business, services and service interfaces. UDDI is a standard for a registry web service. Service discovery may be at design time or run time.

IBM WebSphere Studio provides Web Services Explorer to publish and search UDDI. UDDI Service entries contain URL of its LSDL

- A UDDI directory may be private to an enterprise
- LSDL for services may be stored in local files and directories
- LSB mediations may be used to determine service's location at run time

Following is an example of UDDI on Learning Equipments:

```

<businessService>
  <name>
    Learning Equipments Catalog Service
  </name>
  <description xml:lang="en">
    Search Multimedia Products catalog on
line
  </description>
  <bindingTemplates>
    <bindingTemplate>
      <accessPoint content="endpoint">
        https:// Learning Equipment
//catalog.html

```

```

</accessPoint>
<tModelInstanceDetails>
  <tModelInstanceInfo
tModelKey="uddi:uddi.org:ubr:storage:http">
  </tModelInstanceInfo>
</tModelInstanceDetails>
</bindingTemplate>
</bindingTemplates>
<categoryBag>
  <keyedReference

tModelKey="uddi:uddi.org:categorization:ge
neral_keywords"

keyName="LearningEquipments:categorizati
on:multimedia"
  keyValue="f"/>
</keyedReference

tModelKey="uddi:uddi.org:ubr:categorization:u
nspsc"
  keyName="unspsc-org:UNSPSC:
Description of Learning equipment for
multimedia"
  keyValue="55-10-15-07"/>
</categoryBag>
</businessService>

```

V. THE MESSAGE SENDING OF LEARNING SERVICE

SOAP is a simple XML based protocol to let applications exchange information over HTTP, which provides a way to communicate between applications running on different operating systems, with different technologies and programming languages. SOAP forms the foundation layer of the Web services stack, providing a basic messaging framework. In the LSB, this protocol is adopted. One example of SOAP on goods is shown below:

The SOAP request:

```

POST /InGoods HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml;
charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-e
nvelope"
soap:encodingStyle="http://www.w3.org/2001/1
2/soap-encoding">
  <soap:Body
xmlns:m="http://www.LSB.org/goods">
    <m:GetCoursesName>

<m:CoursesName>IBM</m:CoursesName>
  </m:GetCoursesName>
</soap:Body>
</soap:Envelope>

```

A SOAP response:

```

HTTP/1.1 200 OK
Content-Type: application/soap+xml;
charset=utf-8
Content-Length: nnn
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-e
nvelope"
soap:encodingStyle="http://www.w3.org/2001/1
2/soap-encoding">
  <soap:Body
xmlns:m="http://www.LSB.org/goods">
    <m:GetCoursesNameResponse>
      <m:Name>Operating System</m:Name>
    </m:GetCoursesNameResponse>
  </soap:Body>
</soap:Envelope>

```

VI. THE QUALITY OF LEARNING SERVICE

With the widespread proliferation of learning services, quality of learning service (QoLS) will

become a significant factor in distinguishing the success of service providers. QoLS determines the service usability and utility, both of which influence the popularity of the service. The major requirements for supporting QoLS in the LSB are as follows:

Availability: Availability is the quality aspect of whether the learning service is present or ready for immediate use. Availability represents the probability that a service is available. Larger values represent that the service is always ready to use while smaller values indicate unpredictability of whether the service will be available at a particular time.

Accessibility: Accessibility is the quality aspect of a service that represents the degree it is capable of serving a learning service request. It may be expressed as a probability measure denoting the success rate or chance of a successful service instantiation at a point in time. There could be situations when a learning services available but not accessible. High accessibility of learning services can be achieved by building highly scalable systems. Scalability refers to the ability to consistently serve the requests despite variations in the volume of requests.

Integrity: Integrity is the quality aspect of how the learning service maintains the correctness of the interaction in respect to the source. Proper execution of learning service transactions will provide the correctness of interaction. A transaction refers to a sequence of activities to be treated as a single unit of work. All the activities have to be completed to make the transaction successful. When a transaction does not complete, all the changes made are rolled back.

Performance: Performance is the quality aspect of learning service, which is measured in terms of throughput and latency. Higher throughput

and lower latency values represent good performance of a learning service. Throughput represents the number of learning service requests served at a given time period. Latency is the round-trip time between sending a request and receiving the response.

Reliability: Reliability is the quality aspect of a learning service that represents the degree of being capable of maintaining the service and service quality. The number of failures per month or year represents a measure of reliability of a learning service. In another sense, reliability refers to the assured and ordered delivery for messages being sent and received by service requestors and service providers.

Regulatory: Regulatory is the quality aspect of the learning service in conformance with the rules, the law, compliance with standards, and the established service level agreement. Learning services use a lot of standards such as SOAP, UDDI, and LSDL. Strict adherence to correct versions of standards by service providers is necessary for proper invocation of learning services by service requestors.

Security: Security is the quality aspect of the learning service of providing confidentiality and non-repudiation by authenticating the parties involved, encrypting messages, and providing access control. Security has added importance because learning service invocation occurs over the public Internet. The service provider can have different approaches and levels of providing security depending on the service requestor.

VII. CONCLUSION

The core contents in this paper are to transform the learning activities into a series of web service, and to build one architecture of learning management based on SOA, named as the LSB. Designing LSDL, and using UDDI, SOAP and QoLS, one total solution is emerging. We try to

form a specification and standard of LSB in order to realize transaction among different teaching department. The further endeavor will be put on building one LSB system based on SOA. Perhaps the task will be very harder, but the goal will be reached if many people dive into it by the community of open source codes.

REFERENCES

- [1] Advanced Distributed Learning. <http://www.adlnet.org>
- [2]E-Learning from WBT Systems.
<http://www.wbt systems.com>
- [3] Advanced Distributed Learning Sharable Content Object Reference Model (Version 1.2):The SCORM Overview.
<http://galileo.enc.org/~library/scormover.doc>
- [4]IEEE Learning Technology Standards Committee.
<http://ltsc.ieee.org>
- [5]Learning Content Management Systems.
<http://lcmscouncil.org>
- [6]IBM Learning Solutions.
<http://www-3.ibm.com/services/learning/solutions/>
- [7]IMS Global Learning Consortium. <http://imglobal.org>
- [8]IEEE 1484 Learning Objects Metadata.
<http://www.ischool.washington.edu/sasutton/IEEE1484.html>
- [9]Kumar V. Computer-Supported Collaborative Learning Issues for Research. 2002.
- [10]Middleware Overview.
<http://www.eecs.wsu/~bakken/MiddlewareOverview.html>
- [11] White Paper, Software-Oriented Architecture at
<http://www.adobe.com/enterprise/>