Toward an Open and Interoperable e-Learning Portal: OEPortal

Kevin Chihcheng Hsu

Department of Information Management, National Central University, Taiwan // khsu@mgt.ncu.edu.tw

Fang-Chuan Ou Yang

Department of Information Management, National Central University, Taiwan // Department of Information Management, Ching Yun University, Taiwan // ouyang@cyu.edu.tw

ABSTRACT

With the rapid advance of stand-alone e-learning systems, we believe a sharable and interoperable portal platform capable of integrating various existing learning systems is critical for the future development of e-learning systems. We highlight two problems as the root causes for current ineffective sharing of learning resources: learning object interoperability and learning activity interoperability. In this paper, we propose an open and interoperable e-learning portal architecture to solve these problems: 1) With regard to learning object interoperability, we propose WSRP+ service as an enhancement to WSRP standard to send presentation content and intercept crucial information from LMSs in order to facilitate the presentation-level integration of learning objects and keep the learner's learning status in our e-learning portal. 2) With regard to learning activity interoperability, BPEL4WS is used for the task-based orchestration of course workflows among shared learning systems, in which value-added tasks such as grading, evaluation, recording, etc., can also be integrated as a whole. At the end of this paper, we present our prototype implementation to illustrate the feasibility of our approach.

Keywords

E-learning, WSRP (web service for remote portlets), BPEL4WS (business process execution language for web services), Workflow orchestration

Introduction

Electronic learning (e-learning) has become an important medium of interactive learning and knowledge sharing. There are more than 60 commercial or free stand-alone learning management system (LMS) products in the world. One of the key challenges of today's e-learning systems is to make these LMSs seamlessly integrated to provide an open and interoperable e-learning portal able to do the following:

➢ support interoperability for better collaboration of related LMSs to achieve learning task(s) of bigger scope
➢ support formal course composition methods for learning design to provide better adaptive learning activities across LMSs
➢ provide a learner-centric single and open portal for learners that is capable of incorporating learning materials from heterogeneous LMSs
➢ facilitate life-long learning in which a learner will access many LMSs during his/her learning career as well as the option of consistent scoring, tracking, accessing, and/or learning-assisting methodology

Several consortia contributed to the standards of reusing and sharing learning objects among LMSs, such as SCORM (sharable content object reference model), IMS (instructional management systems) and ULF (universal learning format), which defined the features of learning objects and facilitated the exchange of learning objects. Currently, many e-learning products support these specifications. However, the functionalities of learning systems hosted on decentralized platforms (such as learning assessment and complex interactive functions) still have difficulty communicating with others because each system's infrastructure is platform-dependent and non-interoperable. The interoperability is the ability of two or more systems or components to exchange information and use the information that has been exchanged (IEEE, 1990). In this study, we highlight two interoperability problems in the e-learning systems:

➢ *Learning object interoperability* is the ability to share and reuse the learning objects among heterogeneous LMSs in a standard way. Though the most popular SCORM and IMS specifications define learning objects for their interoperability, they still lack capability to achieve some intended complex interactions. For example, in the problem solving oriented intelligent tutoring systems (ITS), repeated prompts and sophisticated diagnoses are often required from learners to proceed in the direction in solving learners' problems. To implement these

complex interactions, the applications are often designed in service-oriented architecture. Several studies explore e-learning activities as services to provide a better solution among learning objects across heterogeneous e-learning systems (Kong et al., 2005; Hsu & Ou Yang, 2005; Liao et al., 2003; Lin et al., 2001).

➢ *Learning activity interoperability* is the formal definition of learning activity such as learning path, learning strategy, and navigational sequencing, which can be shared and reused among compatible LMSs. The recent SCORM 2004 SN (sequencing and navigation) model defines several simple sequencing rules to be employed by teachers in incorporating learning strategies into the learning objects (The SCORM 2004 specification, 2004). Carnegie Mellon Learning Systems Architecture Lab further developed ten templates to simplify these learning use cases to help SCORM users implement their learning strategy with ease and less cost. The IMS-LD model defines learning activities, support activities, roles, environments, learning objects, services, and their relationships to facilitate the designing and sharing of learning strategies (IMS Learning Design Information Model V1.0, 2003). Nevertheless, these specifications focus on describing learning strategies at the metadata level without providing concrete system architecture and external interfaces for e-learning systems to connect external learning resources and services. Workflow is also a popular approach to facilitate the learning activity interoperability (Koper, 2005).

Considering the aforementioned problems, previous research suggests service-oriented architecture combined with workflow orchestration as a better solution (Kong et al., 2005; Friesen & Mazloumi, 2004; Puustj¨arvi, 2004; Lin et al., 2001; The TimesTen Team, 2000). In that research, the common concept is to use distributed computing technologies and formal workflow definitions to solve the learning object interoperability and learning activity interoperability. However, the major problem in these studies is that the learning objects from different LMSs cannot be plugged into web pages as easily as components, which leads to high implementation costs to most users such as teachers and students. Able to easily incorporate learning resources from heterogeneous platforms into a web-based learning platform, as well as able to incorporate significant business logic during the course arrangement based on these incorporated heterogeneous learning resources, in our opinion, should be the ultimate goal for interoperability in the e-learning technology field. This is similar to the video files hosted on YouTube (www.youtube.com, a web platform that allows web users to upload and distribute their own movies) that can easily be reused and plugged into any website with only simple codes and configurations. The website can then reuse YouTube's video clip to produce related multimedia content and programs. As presentation content, these source video from YouTube are provided as services, which is what our proposed OEPortal wishes to achieve for heterogeneous learning resources with extra workflow orchestration capabilities.

In the early study, we present the novel concept of open and interoperable e-learning portal architecture (OEPortal) based on the technologies of service-oriented, workflow collaboration, and presentation level integration for e-learning systems to overcome the aforementioned shortcomings (Hsu & Ou Yang, 2005). In this study, we elaborate upon the pedagogical requirements and technical design of connecting WSRP with BPEL4WS within OEPortal to compose existing heterogeneous learning resources pedagogically.

The remainder of this paper is organized as follows: related works are discussed in Section 2. Section 3 presents our solution for enhancing WSRP (without changing the standard) in order to work together with BPEL4WS. Section 4 presents the implementation of our proposed OEPortal and Section 5 presents our conclusion.

Related works

In recent e-leaning development, some organizations undertook the standards of learning objects in order to identify the purposes of learning courses, and adhere to a formal and global definition to share and exchange learning objects by employing XML-based metadata tagging such as AICC, ARIADNE, IEEE LTSC, IMS, and ADL. Of the standards made by these organizations, the most popular standards were SCORM (sharable content object reference model), IMS-LD (learning design), and IMS-QTI (question and test interoperability). The last SCORM 2004 specification defines the models of CAM (content aggregation model), RTM (Run-TiMe environment), and SN (sequencing and navigation), which provides comprehensive descriptions for the interoperability among learning objects and makes learning objects capable of being packed to SCOs (sharable content objects) for sharing and reusing among compatible LMSs. The use of check in/out mechanisms of SCORM objects becomes the most popular interoperable approach of learning objects in the learning technology community (Liber, 2005). Many researchers have also decided to develop effective learning activities and theories into e-learning objects in order to increase

learning performance (Fjuk & Berge, 2006; Chang et al., 2004; Koper & Manderveld, 2004). A successful learning object design should combine learning design with learning theories (Roderick, 2005). Most researchers support the idea that learning materials can raise learners' performance by applying adaptive pedagogical approaches/theories. IMS-LD is a standard proposed to formally describe pedagogical learning design and facilitate the learning activity interoperability. For the interoperability of testing and assessment, IMS-QTI tackles the problem of exchanging, managing, and searching tests among different platforms.

In general, those e-learning standards provide XML-based metadata tagging to achieve different levels of interoperability while describing the organizing, accessing, retrieving, and traversing of learning objects. From the standpoint of the e-learning system's functionality, even though the SCORM standard provides fundamental metadata-level descriptions for learning objects, it still needs a solid system framework and external interface to coordinate heterogeneous learning resources and related functionalities among LMSs. An e-learning system architecture should provide more dynamic features than the basic abilities currently provided by SCORM-like standards, such as recording the learning processes of learners, the adaptive presentation of learning contents, interoperating with legacy e-learning systems, collaborating separated learning objects, etc. Hence an open and interoperable e-learning framework with an external interface suited to legacy e-learning resources and services is critical.

Moreover, it is difficult for these works to integrate learning objects efficiently to satisfy learning needs such as knowledge-based or skill-based courses in the rigid context of large-scale web-based education (Liu et al., 2002). Some complex knowledge management methods may be internally arranged within the e-learning system, such as traditional CBI (computer-based instruction) and ITS (intelligent tutoring system). A solid e-learning infrastructure or external interface is necessary in this domain to enhance these technical requirements.

Distributed technologies are well identified to integrate services at data level to aggregated courses and used in several studies to integrate e-learning systems. (Liao et al., 2005; Liao & Ou Yang, 2004; Reklaitis et al., 2003; Liao et al., 2003; Brusilovsky & Nijhavan, 2002; Fuji & Tanigawa, 2002; Reklaitis et al., 2002; Gaeta et al., 2002). In Virtual Campus (Nicosia, 2004) and Flex-eL (Lin, 2001), they collaborate the learning materials using formal workflow language defined by the WfMS (workflow management system). However, they only integrate local learning materials. In all these investigations, the issues of collaboration and management of learning objects were addressed mainly on the data level. We believe a well-collaborative LMS's infrastructure should also be collaborative at the presentation-level due to the presentation-centric nature of e-learning systems.

In the domain of service-based e-learning, there are several studies similar to our proposed e-learning framework (Kong et al., 2005; Friesen & Mazloumi, 2004; Puustj¨arvi, 2004; Lin et al., 2001; The TimesTen Team, 2000). In that research, the common concept is to use distributed computing technologies and formal workflow definitions or Petri net technology to solve the learning object interoperability and learning activity interoperability, respectively. However, the major difference between these works and our approach is that the learning objects from different LMSs cannot be plugged into web pages as easily as components, which results in high implementation cost to most users, such as teachers and students. A robust and common query interface is built on each sharable learning object repository to greatly enhance the global visibility and interoperability or the integrated e-learning systems in Ternier et al. (2005), but it lacks a formal workflow mechanism to coordinate the sharable and remote learning objects. A comparative description of these frameworks and our proposed OEPortal is given in Table 1.

According to the aforementioned problems, there are two central research issues, which our proposed open and interoperable e-learning portal (called OEPortal in this paper) wishes to address:

Presentation-level integration of learning objects for learning object interoperability: In the aforementioned service-based e-learning systems, most learning objects are delivered, shared, and managed among LMSs at the data level while they are being integrated. The source LMSs seldom handle the actual presentation of their hosted learning objects when the materials are shared and accessed by partner LMSs. In these data-oriented approaches, service consumers often need to additionally program and implement the presentation-level interfaces to display the retrieved data. In the real world, most users and learners often have difficulty finishing these missions by themselves because of the high technical threshold. We believe that in an integrated LMS environment, sharable learning objects residing in partner LMSs should be published and subscribed in a dynamic way at both function and presentation levels from the source LMS, which will make reusing the learning objects as a web component easy and convenient.

(This is similar to YouTube's video files, which can be easily reused and plugged into any website with only simple codes and configurations.)

Workflow management of learning objects for the learning activity interoperability: workflow is the basic technology in an e-learning system to implement a learning path and learning strategy by collaborating diverse learning objects. Workflow collaboration across multiple e-learning systems is impossible without an interoperable standard across LMSs. This standard needs to address the mechanical and semantic heterogeneity of distributed learning objects. Furthermore, for the e-learning workflow to succeed across multiple LMS, it is also important to provide better adaptive learning across various LMSs (e.g., a well-integrated sharable scoring or grading system to allow learners to proceed from LMS 1's material to the related materials of higher complexity hosted on LMS 2).

Table 1. Comparison of previous studies and our proposed e-learning framework

Compared item	SCORM and IMS based e-learning framework	Distributed technologies and service based e-learning framework	Our proposed e-learning framework in this paper (which was implemented in the OEPortal)
Learning object interoperability	SCORM, IMS standards: • focus on the navigational mappings of learning objects. • difficult to support knowledge-based or skill-based complex interactions	Distributed components or services focus on data-level integration of learning objects to support SCORM standard	Service-based infrastructure covering the presentation-level integration to support SCORM standard by Shareable LO Portlets service wrapper
Learning activity interoperability	IMS SS, IMS LD: • focus on integrating local learning objects • few implementation details provided	Workflow standards, WfMS, Petri net	Workflow standards extended to also support WSRP
Test and assessment interoperability	IMS QTI/RES: • only defines several basic item types • most are implemented by JavaScript	No support or non-proprietary	Supports service-oriented test and assessment interoperability (by LRCC and LRRT)
External functionality integration	N/A	Distributed components or services	Service-based infrastructure; external interfacing portal platform for providing more personalized features for learners

In this paper, we intend to solve these two issues by merging five predominant e-learning technologies/approaches: 1) Meta-level description: by employing e-learning standards such as SCORM to describe learning objects. 2) Distributed technologies: by employing web service technology to communicate among heterogeneous and separated e-learning platforms. 3) Workflow orchestration: by employing BPEL4WS (business process execution language for web services) to compose a lot of learning objects and construct complex learning activities. 4) Complex learning strategies and pedagogical theories: by conducting adaptive learning strategies or pedagogical theories to construct complex learning path for raising the learning or testing performance. 5) Presentation-level learning-object integration: by employing WSRP (web service for remote portlets) to facilitate the presentation-level learning object interoperability and customizations of GUIs. Figure 1 shows these five key elements on our purposed e-learning system.

The elements of SCORM-like standards and distributed technologies are introduced in the previous sections; we highlight the other three elements below:
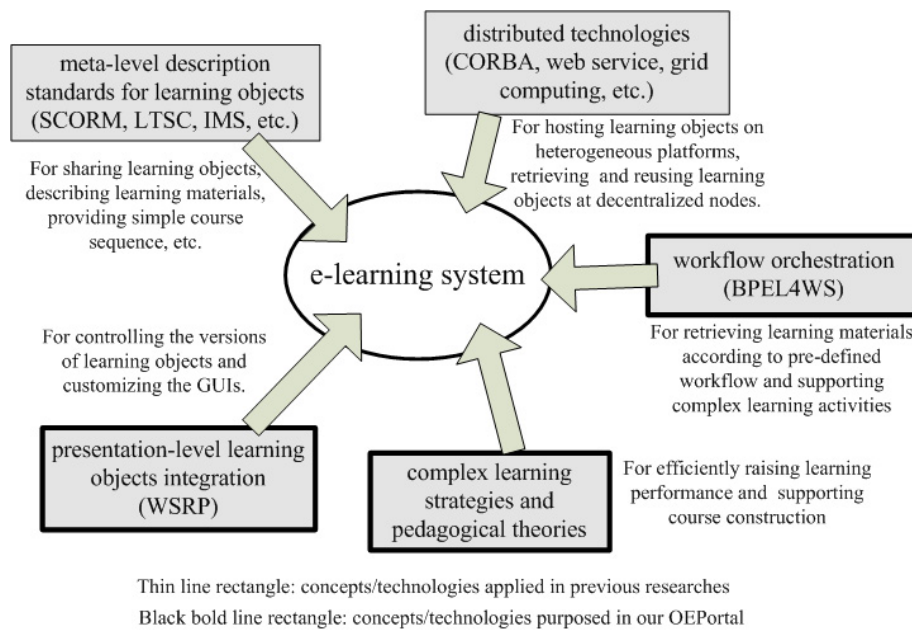
Thin line rectangle: concepts/technologies applied in previous researches
Black bold line rectangle: concepts/technologies purposed in our OEPortal

Figure 1. Five key elements in our purposed OEPortal

WSRP: for the presentation-level learning objects integration.

Traditional service-oriented architecture (Friesen & Mazloumi, 2004; Puustj¨arvi, 2004; The TimesTen Team, 2000) only provides the functional level integration and lacks the much-needed presentation level integration across LMSs. Even though application programs can invoke diverse external web services and get results from them, the programs have to handle the front-end presentation by themselves. Many organizations often employ portals to support presentation-level integration for their websites (Puschmann & Alt, 2004). In portal technology, portlet, as a web component for handling the HTTP requests and generating dynamic web content, is excellent in aggregating various web content. Portlets can work as pluggable GUI components and present dynamic web content through suitable configuration (The Portlets Specification, 2003). Portlets have two working models: 1) Portlets are hosted directly on local portal; 2) Portlets are hosted on remote endpoints based on their service types and invoked by portlet proxies of a local portal. In both models, the portlet services can be designed as sharable presentation-oriented web services, i.e., WSRP (Web Service for Remote Portlets) services. Different from data-oriented web services, presentation-oriented WSRP services provide the definition in the presentation level to make portlets capable of being invoked from remote sides, as well as pluggable into a local portal without the need of extra coding to handle front-end presentation during the integration. By employing WSRP, we can pack learning objects in a WSRP service type, and make learning objects sharable as well as manageable at the presentation level. However, WSRP, as a pure presentation standard, does not allow integrators to intercept sufficient content for properly managing workflow. In Section 3, we discuss this insufficiency and propose our solution, WSRP+, in order to enable seamless integration of presentation-preserved learning units.

BPEL4WS: for the workflow orchestration.

The chosen workflow standard should be sufficient to define how the learning objects from all LMSs are joined together to construct cohesive learning courses, as well as how value-added tasks can be added within these courses (e.g., grading, evaluation, recording, etc.). By carefully examining the most popular workflow standard, BPEL4WS, we find it is able to make learning objects and other external learning resources and functionalities work collaboratively across LMSs and provide the infrastructure of personalization of e-learning systems. However, BPEL4WS is designed to promote the service-based workflow in which the services it integrates are mainly data-oriented. To make our proposed OEPortal a practical and easy-to-use extension to SOA, we propose to add extra components/agents to BPEL4WS to allow BPEL4WS to invoke and integrate presentation-oriented web services.

Complex learning strategies and pedagogical theories.

From the pedagogical standpoint, we explain how these technological components in our purposed OEPortal are related to pedagogical elements in the Figure 2. The mapping from Figure 2 allows pedagogical theories to be incorporated into courses following adaptive pedagogical theories. The integrated materials should provide sufficient metadata for adaptive integration, the results of learning can be evaluated and obtained somewhere (mostly from the hosting LMS), and the integrator should be able to reorganize the different learning results in order to facilitate the pedagogical workflow for e-learning.
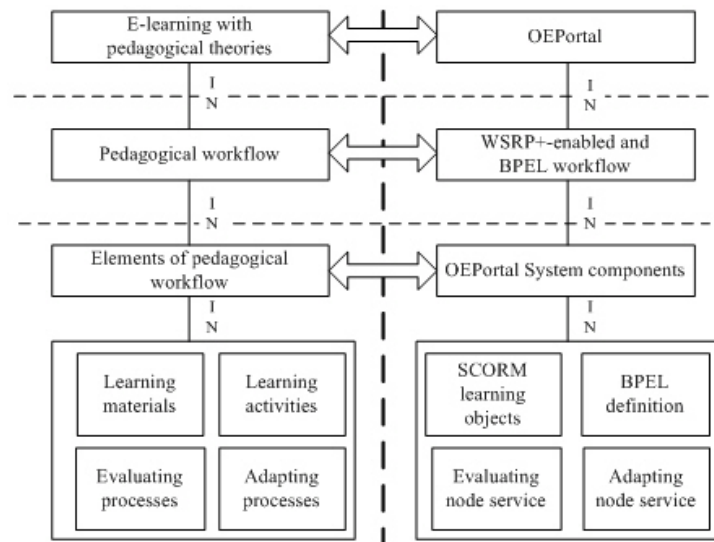

Figure 2. A relationship map of e-learning, pedagogical theories, and OEPortal

System architecture

In this section, we describe the collaboration diagram of OEPortal first, the procedure of wrapping legacy learning objects into WSRP services, and details of the principles of facilitating the workflow orchestration using BPEL4WS and WSRP.

The collaboration diagram of OEPortal

The collaboration diagram of OEPortal is depicted in Figure 3. It comprises two phases, the course preparation phase and the course execution phase. We describe these two phases below and in Figure 3:

Course preparation phase:

Learning materials sharing: learning materials are packed by content providers through SCORM packaging tool and stored in the LO Repository. These learning materials packed by SCORM can be viewed as sharable learning objects or SCOs (Sharable Content Objects). After that, LMSs will wrap the SCOs into sharable learning object portlet services (WSRP services) by the sharable LO portlets wrapper.

Content publishing: LO portlets services are published to the sharable LO portlets registry of OEPortal by the source LMSs. The sharable LO portlets registry will record related information and metadata regarding the LO portlets services and SCORM, respectively.

Course authoring: 1) Instructors and authors, with the help of metadata provided by LO, can choose diverse LO portlets services to compose a task-based and value-added learning course with the course authoring tool; 2) The

course authoring tool will interact with the sharable LO portlets registry to retrieve all the registered service information for the instructors and authors; 3) After finishing arranging the course, the course authoring tool will store related course information (e.g., necessary services, course objectives, etc.) in the OEPortal's databases (e.g., databases of personalization, learning courses, task-based course workflow choreography, navigation, interaction, user profiles, etc.).

Course execution phase:

Learning course subscription: 1) Learners use client user agent (e.g., browsers, mobile phones, PDAs, etc.) to enter the personalized course portal of OEPortal and subscribe to a course (some additional events will be triggered at first, such as registering for members, configuring personalized information, etc.). 2) All configurations and settings are stored in the related databases.

Learning a course: 1) Learners enter the personalized course portal of OEPortal and choose a course to learn; 2) The personalized course portal obtains enough information from the database of the task-based course workflow choreography and initializes the retrieval of course content; 3) Necessary information is loaded into Workflow Orchestration for to finish a workflow job; 4) The Workflow Orchestration working in the BPEL4WS runtime environment sequentially sends requests to the portlets-based content aggregation to invoke each LO portlets service; 5) The portlets-based content aggregation is responsible for invoking each LO portlets service until all LO portlets are gathered; 6) Workflow Orchestration obtains all gathered LO portlets and plugs them into the web page of the personalized course portal; 7) The personalized course portal incorporates personalized and adaptive configuration with LO portlets and responds to the client user agent used by learners.
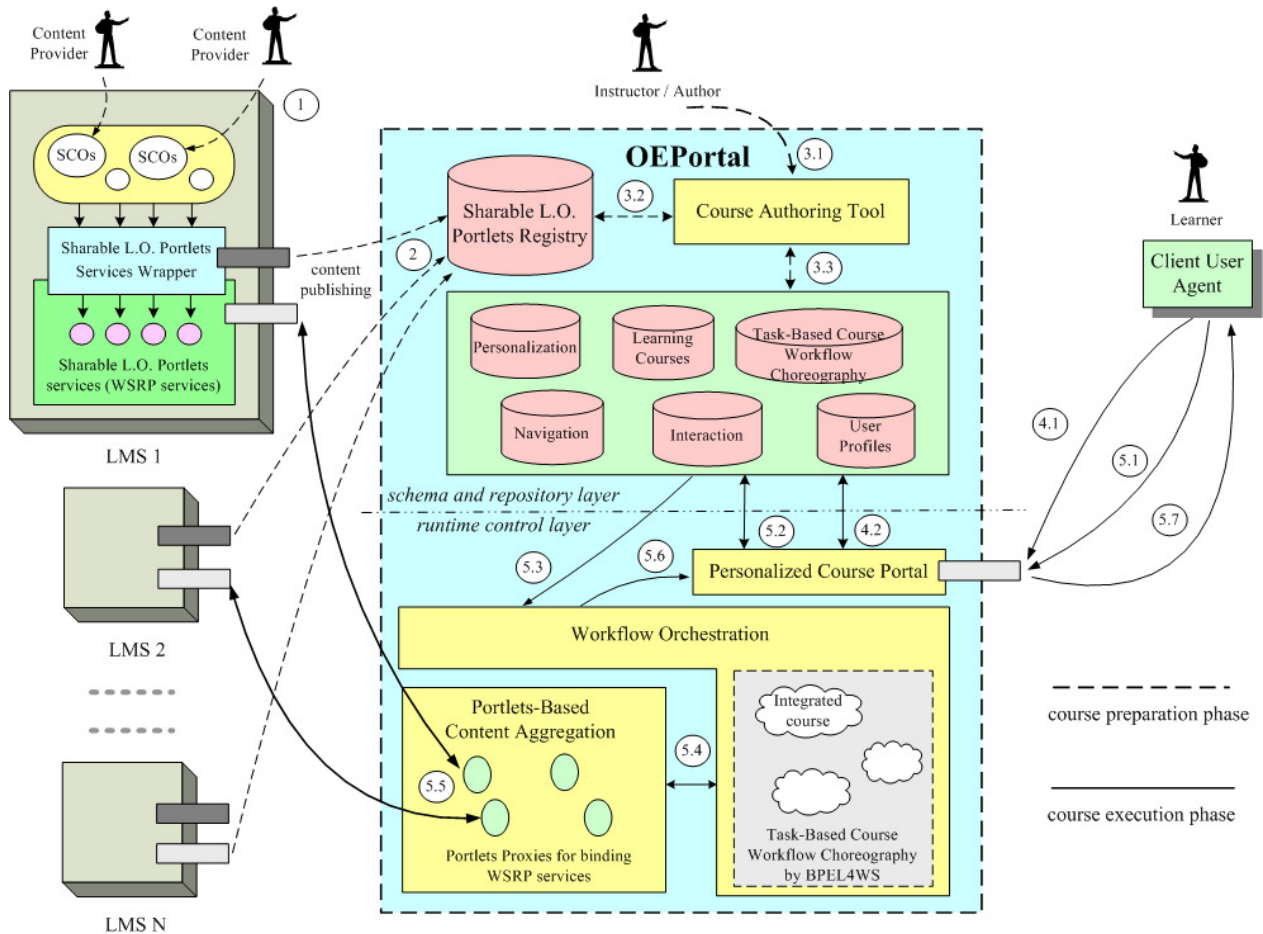

Figure 3. The collaboration diagram of OEPortal

Wrapping legacy learning objects into WSRP services

The procedure of wrapping legacy learning objects WSRP is critical to the success of our proposed system, since we believe presentation-preserving is an important requirement for our OEPortal. In our proposed OEPortal, the wrapping of legacy learning objects into sharable and portable WSRP services can be conducted in a two-stepped packaging process: 1) Legacy or existing learning objects (e.g., text files, video files, documents, etc.) already packaged by SCORM are converted to learning object portlets compatible with the portlet standard (The Portlets Specification, 2003); 2) Learning object portlets are then packaged under WSRP APIs to become the sharable learning object portlets services, namely the WSRP services. After this two-stepped packaging process, learning objects from participating LMSs can be published to our OEPortal and become sharable and reusable for other LMSs in service type.
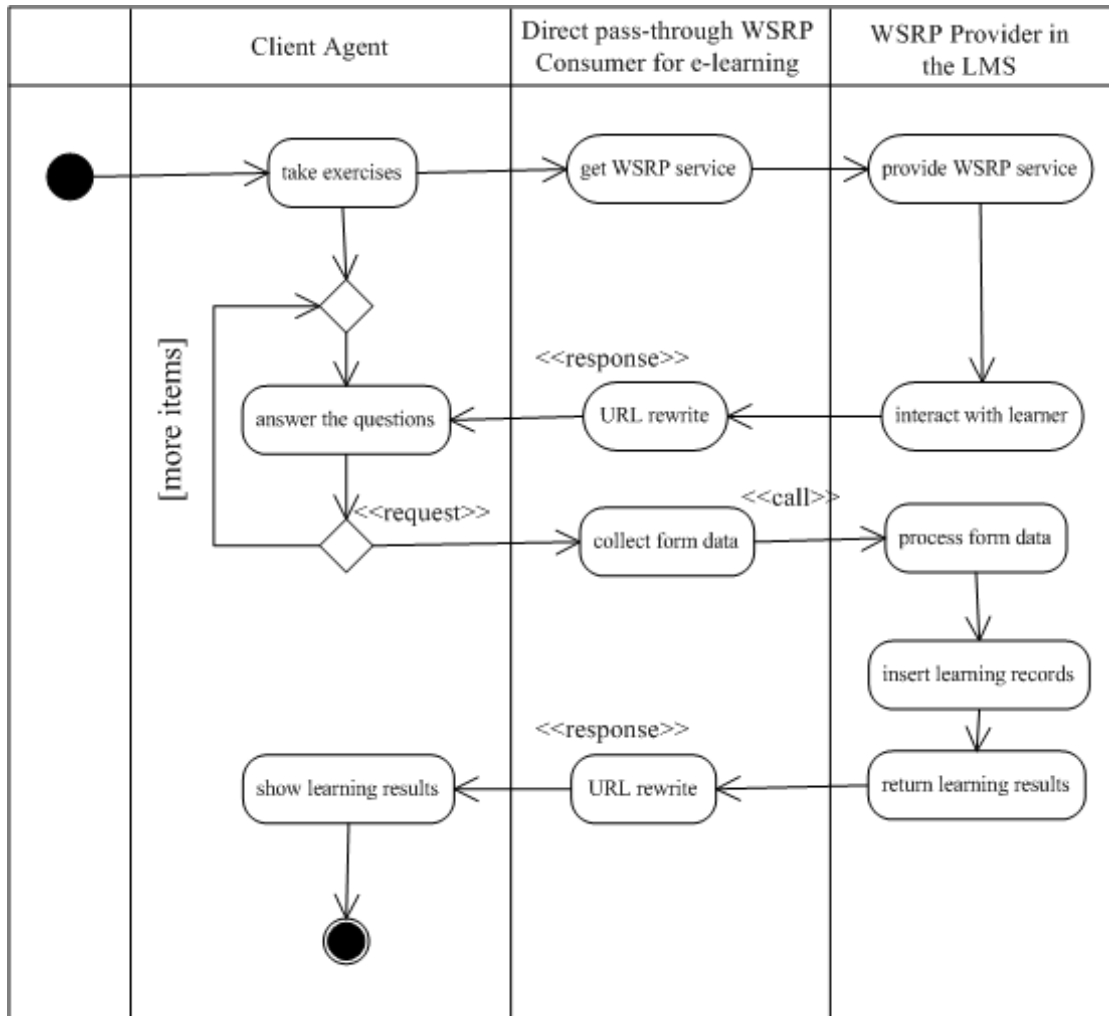

Figure 4. Swimlane diagram of a direct pass-through WSRP operation of course evaluation

Connecting WSRP with BPEL4WS

WSRP is a good presentation standard for passing viewable content through web services. However, WSRP, as a pure presentation standard, does not allow integrators to intercept sufficient content to properly manage workflow. Any information and data coming from the client agent (e.g., Internet Explorer) are only passed through the WSRP consumer and forwarded to the WSRP provider (even WSRP consumer may sometimes rewrite a very limited portion of the markup, such as the URL). However, it is very important for the mediated OEPortal to be aware of data such as learning status, navigational paths, learners' profiles, and personalized configurations in order to provide

the personalized features and construct the courses based upon the learner's learning status. If the WSRP is directly deployed to wrap LO, OEPortal can only play as a straight pass-through broker. We use Figure 4 to illustrate this important limitation of directly wrapping LO with WSRP. In Figure 4, the direct pass-through WSRP operation and LMSs for a scenario of course evaluation are shown as a swimlane diagram. Figure 4 shows a scenario of course evaluation. At first, the learner enters an integrated environment and takes exercises. The WSRP consumer gets WSRP service from the WSRP provider for the learner to learn. The URL of markup fragments provided by the WSRP provider always needs to be rewritten to the client agent. And then the WSRP consumer passes the form data from the client agent to the WSRP provider when the learner finishes choosing all the answers in the exercise. After that, the WSRP provider processes the form data and saves the learning results. Finally, the WSRP provider sends the fragment markup to the WSRP consumer with the appropriate URL rewriting and shows the learning results on the display of client agent. During this typical WSRP operational lifecycle, the WSRP consumer never preserves any information and data from the back-end business logic. It is a serious shortcoming of WSRP, especially for an integrated and collaborative e-learning environment in which presentation content should be shared remotely and coordinated and customized for different learners, courses, etc.

To remedy this shortcoming, we need to enhance both the WSRP provider and the consumer. In order not to change WSRP and BPEL4WS standards, we propose a new concept called WSRP+ session, which includes regular interaction between the WSRP provider and the consumer (i.e., regular WSRP session), two additional web services (called Message Receiver Web Service and Learning Record Web Service later in the text) to enclose the regular WSRP provider and consumer, and the necessary invocations of these additional web services at the right time. The proposed WSRP+ session, when used with WSRP agent for BPEL4WS (another concept to be proposed later in this section), enables the WSRP providers to participate in the BPEL4WS workflow orchestration similar to other web services. Incorporating WSRP into BPEL4WS enables presentation content to be included into workflow orchestration. This incorporation is necessary since presentation content could be crucial in defining certain workflows (e.g., in the integrated and collaborative e-learning system), and the capability is lacking in traditional workflow language (even for current BPEL4WS). The following design ideas outline how we enhanced the interaction between WSRP consumer and producer to provide WSRP+ session:

➢ The WSRP consumer does not need to know all the presentation content. It only needs to know whatever is important in order to control the workflow customization and orchestration. We call this critical information "content feature for workflow customization and orchestration". The "content feature for workflow customization and orchestration" can be specific data used for the customization, such as the learner's learning status and results preserved in the WSRP provider. With different customization, the workflow orchestration can adaptively organize the learning activity to match the learner's professional skill and ability.

➢ The WSRP consumer needs to be aware of the content feature for workflow customization and orchestration only at the end of a WSRP session. Various presentation-level data may pass between the WSRP consumer and provider. Only the resulting data available at the end of a regular WSRP session could be important and useful for the workflow customization and orchestration. The content feature for workflow customization and orchestration, in fact, is part of this resulting data. This resulting data certainly resides at the WSRP provider side since it is the content processor of WSRP session capable of knowing the final result of the conversation/session.

➢ The WSRP provider should provide the content feature for workflow customization and orchestration of a finished WSRP session as a standard web service for OEPortal (which hosts WSRP consumer) to retrieve this corresponding content feature for workflow customization and orchestration. This web service is called Learning Record Web Service.

➢ In order to keep to WSRP consumer simple and capable of getting content feature for workflow customization and orchestration at the right time, the following changes are needed for the interaction between WSRP consumer and provider. We call the whole interaction sequence of our proposed change between the provider and consumer side as a WSRP+ session.

    o Before the WSRP consumer calls WSRP provider, the WSRP consumer side will create a corresponding Message Receiver Web Service, which is a web service launched parallel to the regular WSRP consumer. The Message Receiver Web Service is a web service sitting at the consumer side to get the session completion notice from the provider side through web service invocation. This design insulates the WSRP consumer from any change.

    o Then, the interaction between regular WSRP consumer and provider is kept the same without any change. The regular WSRP provider detects that it is time to finish this WSRP session in response to either WSRP

consumer's input or its own computing result. Now, the ending procedure of our proposed WSRP+ session should be initiated at the provider side.

- o The ending procedure of our proposed WSRP+ session includes the WSRP provider side invoking the Message Receiver Web Service to signal the end of WSRP session to the WSRP consumer, and then the WSRP consumer side calling Learning Record Web Service on the WSRP provider side to get the corresponding content feature for workflow customization and orchestration for this WSRP session. This completes the WSRP+ session.
- o The Learning Record Web Service should be designed toward general WSRP consumers, instead of providing different content features for workflow customization and orchestration for different WSRP consumers. The consumer can incorporate the calling of these web services at the end of WSRP session into BPEL4WS workflow orchestration. This extension makes BPEL4WS capable of incorporating WSRP without changing standards and remedies the problem shown in the Figure 4 in which BPEL4WS now can intercept WSRP content.

On the other hand, BPEL4WS is designed to promote the service-based workflow, and the services it integrates are mainly data-oriented. WSRP intends to integrate presentation-oriented services. This difference demands some enhancements in the way BPEL4WS invokes WSRP. In order to make the BPEL4WS capable of driving the WSRP services, it is necessary to have an agent between BPEL4WS engine and the invoked WSRP consumer. This agent, called "WSRP agent for BPEL4WS," is applicable whenever BPEL4WS need to invoke WSRP. To not change the standard, the following design ideas should be kept in mind:

➢ The WSRP agent for BPEL4WS will be instantiated when the workflow process is initialized and destroyed when the workflow process is terminated. Using WSRP agent for BPEL4WS allows the OEPortal to connect WSRP services to the BPEL4WS engine.

➢ Learning activity (the sequence of the invoking of WSRP services) should be defined as the bpel file in order to stick to BPEL4WS, but the output/input SOAP messages should be directed to the WSRP agent for BPEL4WS to process because the BPEL4WS engine can't recognize the presentation-level data of WSRP standard.

➢ The WSRP agent for BPEL4WS has two main tasks: to process the SOAP messages from/to the BPEL4WS engine and to notify the WSRP consumer to invoke the WSRP services from the WSRP producer. At the beginning of a workflow process, the BPEL4WS engine sends a SOAP message to the WSRP agent for BPEL4WS, and the WSRP agent for BPEL4WS drives the WSRP consumer to invoke the WSRP service for the learners. At the end of the workflow process, the WSRP agent for BPEL4WS gets a token from the WSRP consumer and sends a SOAP message to the BPEL4WS engine to finish this workflow process and continue on to the next process.

In the implementation, we create a pair of adapters, one in the LMS and the other in OEPortal, to implement the design ideas 1 to 5. These adaptors employ hybrid approach of PUSH and PULL technologies to support the actions of course finished notification and data retrieval. On the provider side, we design a "Learning Record Common Connector" (called LRCC), and on the consumer side we design "Learning Record Retrieval Trigger" (called LRRT). To implement the design ideas 6 to 8, the WSRP agent for BPEL4WS is designed and implemented to control the workflow orchestration of WSRP services with BPEL4WS engine.

We show the interaction pattern of OEPortal and LMS in Figure 5 and describe the detail of its seven circled steps below:

The learner chooses a course to learn and the BPEL4WS engine prepares for it (Step 0).

Step 1: (1.a) The BPEL4WS engine initializes the process and sends a SOAP message to the WSRP agent for BPEL4WS. (1.b) The WSRP agent for BPEL4WS notifies the WSRP consumer to invoke the WSRP service from the WSRP provider.

Step 2: (2.a) The WSRP consumer invokes WSRP service from the WSRP provider and (2.b) presents these learning materials to the learner.

Step 3: A learning object calls the Learning Record Access Interface of the LRCC to insert the learning records when the learners have finished the exercises.

Step 4: The Learning Record Access Interface of the LRCC does the following: (4.a) inserts learning records to the learning repository of LMS; (4.b) returns a "course finished" notification token to the WSRP consumer. It then calls the Message Receiver Web Service with the course's and learner's ID in order to notify the LRRT of the OEPortal, which triggers the action of data retrieval (PUSH approach). By using the "course finished" notification token, the WSRP consumer becomes aware of terminating the process and sends the message to the WSRP agent for BPEL4WS.

Step 5: The Message Receiver Web Service calls the Learning Record Retrieval Interface for the data retrieval process.

Step 6: (6.a) The Learning Record Retrieval Interface tries to call the Learning Record Web Service of the LRCC with the course's and learner's ID. If the action fails, it will recall at set interval until it gets the learning records from the LMS (PULL approach). (6.b) Learning Record Web Service retrieves the data from the database of LMS and responds to the Learning Record Retrieval Interface with the learner's learning records. (6.c) The Learning Record Retrieval Interface inserts the learner's learning records retrieved from the LMS to the learning repository of OEPortal.

Step 7: (7.a) The WSRP consumer calls to the WSRP agent for BPEL4WS to prepare to end the process. (7.b) The WSRP agent for BPEL4WS sends a SOAP message to the BPEL4WS engine. Finally, the BPEL4WS engine terminates the process and finishes the whole interaction of OEPortal and LMS.
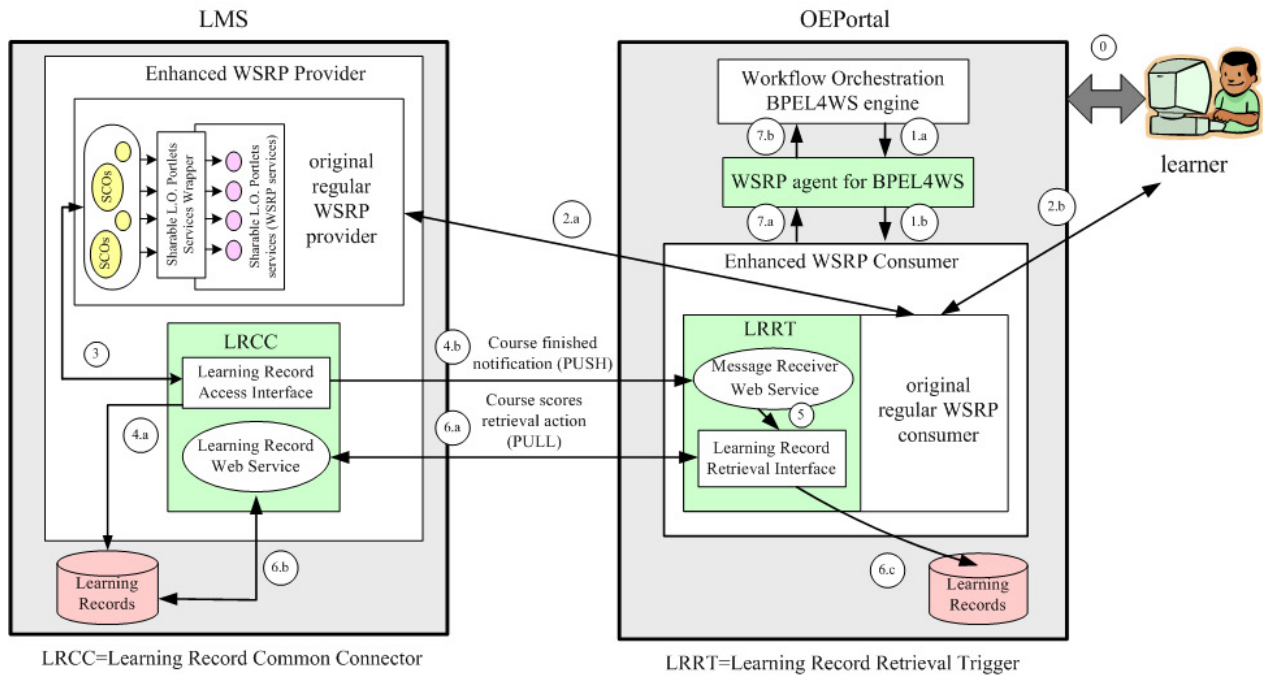

Figure 5. The interaction pattern of OEPortal and LMS

By using the LRCC and LRRT in our proposed e-learning portal, it makes the WSRP consumer aware of the useful content feature for workflow customization and orchestration. Moreover, LRCC and LRRT are based on web service in order to correspond with the open environment of LMS and OEPortal (i.e., [4.a] and [.b] are standard web service invocation). Besides, the hybrid approach of PUSH and PULL technologies applied to LRCC and LRRT can effectively pass the related learning information to the OEPortal and avoid causing failure during the data retrieval in case of unavailable service from LMS.

The WSRP agent for BPEL4WS is used to facilitate the WSRP orchestration with BPEL4WS. The most important feature of our extension is that the approach does not require any changes to either the standards of WSRP or BPEL4WS. Our proposed approach of using standard WSRP and BPEL4WS not only effectively combines both the

technologies but also facilitates a well self-constructed platform for e-learning capable of workflow orchestration of learning materials from various LMSs. A swimlane diagram of the whole interaction of OEPortal and LMS is shown in Figure 6.
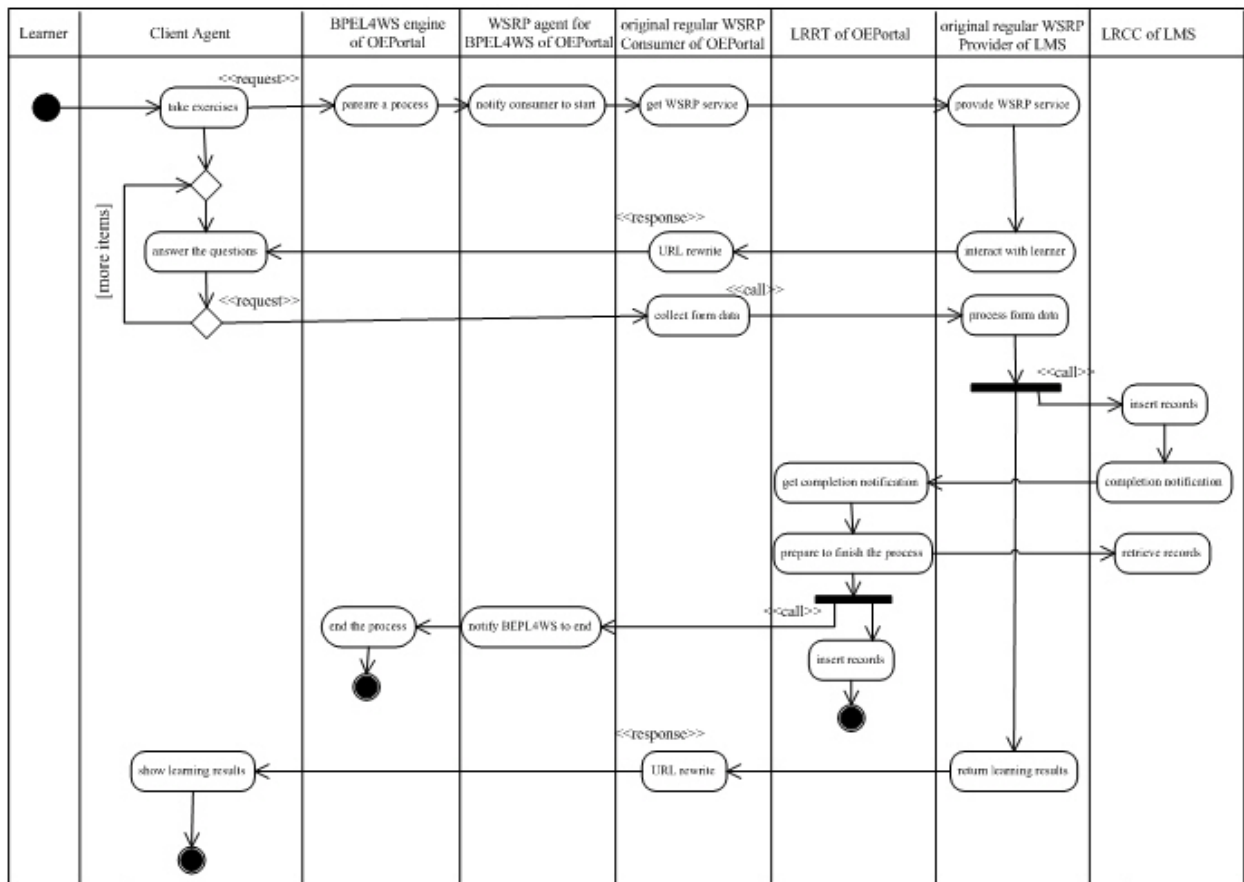

Figure 6. The proposed interaction of LRCC and LRRT

Implementation

In this section, we use a learning scenario to demonstrate our OEPortal prototype system. The prototype system is developed by the Liferay Portal v4.1.1 (Liferay portal, 2006) and activeBPEL engine v2.0 (activeBPEL, 2006). This learning scenario shows how the learner interacts with OEPortal during the preparation for the Java technology certification course. Based on the illustrated course design, the learner needs to take "Java Tutorial — Syntax & Test" first. Based on his or her test result, the learner will take either "Java Tutorial — General" or "JavaTutorial — Methods." Details are described below as well as showed in Figure 7.

Course preparation phase:

Registration of e-learning materials. In this scenario, each LMS provides one learning material and registers to the OEPortal. It is showed in the bottom of Figure 7.

Learning materials. There are three learning materials in the scenario: "JavaTutorial — Syntax and Test," "JavaTutorial — General," and "JavaTutorial — Methods." Each learning material is deployed at different nodes.

Categorization and composition of e-learning materials. From the viewpoint of lifelong e-learning, good e-learning materials should be suitable for the learner's age and matched to the learner's ability. For this reason, in order to

provide adaptive learning content for the learners, categorization and composition of e-learning materials are of important necessity. This should be done by course authors or experts as showed in the right part of Figure 7. Moreover, pedagogical theories and learning strategies could be delivered by methodologically reorganizing these learning materials.
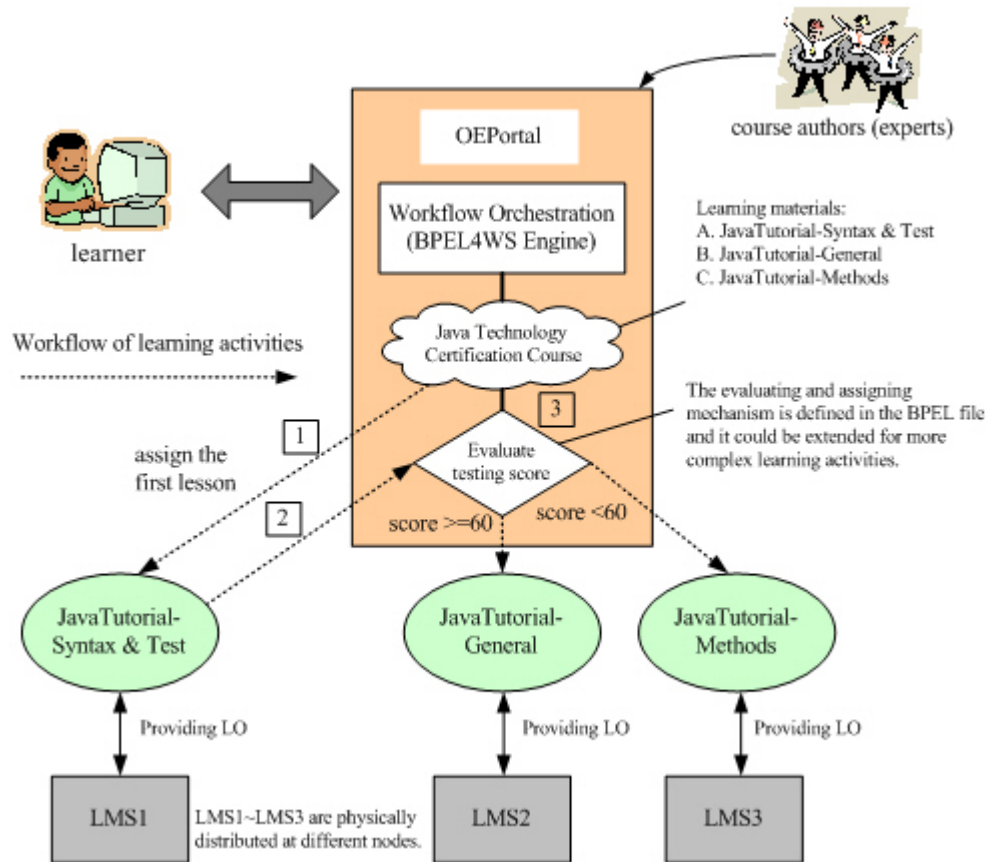

Figure 7. The scenario of Java technology certification course composition

Pedagogical theories and learning strategies. In e-learning, it is useful to improve learning performance by applying learning strategies rather than merely putting numerous learning materials to learners. In this study, several Java experts are asked to design Java learning materials, evaluate the difficult degree of each learning material, and arrange an adaptive learning sequence for learners. A simple learning strategy is applied in our OEPortal prototype system to guide the learning path for learners: an alternative lesson will be assigned to learners according to the testing score of the first lesson. The learners with testing scores higher than 60 will move on to the "JavaTutorial — General" lesson. If not, they take "JavaTutorial — Methods." Figure 8 shows the BPEL process of test evaluation and learning material assignment. The visual process definition shown at the top of Figure 8 is easy to understand and convenient for domain experts and course designers to revise during the workflow design phase. After the visual process definition is confirmed, the workflow engineer will transform it to the BPEL4WS specification via the activeBPEL workflow authoring tools, as shown in the bottom of Figure 8. Finally, the BPEL4WS specification will be uploaded to the BPEL4WS workflow engine for the runtime execution.

Course execution phase:

Learning activities. When all the tasks in the course preparation phase are done, the learners can choose the Java technology certification course from the course list of OEPortal and begin his learning tour. A runtime screenshot of whole the learning activities are showed in the Figure 9 and 10.
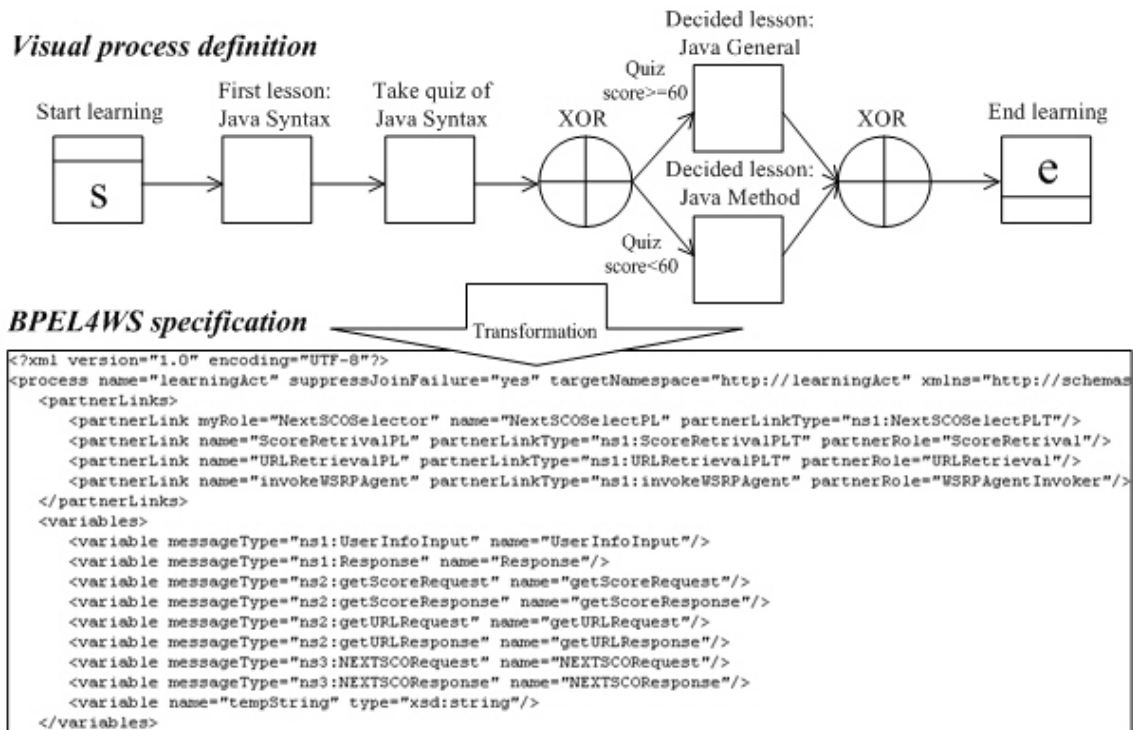
Visual process definition

Start learning | First lesson: Java Syntax | Take quiz of Java Syntax | XOR | Decided lesson: Java General (Quiz score>=60) / Decided lesson: Java Method (Quiz score<60) | XOR | End learning

BPEL4WS specification — Transformation

```
<?xml version="1.0" encoding="UTF-8"?>
<process name="learningAct" suppressJoinFailure="yes" targetNamespace="http://learningAct" xmlns="http://schemas
    <partnerLinks>
        <partnerLink myRole="NextSCOSelector" name="NextSCOSelectPL" partnerLinkType="ns1:NextSCOSelectPLT"/>
        <partnerLink name="ScoreRetrivalPL" partnerLinkType="ns1:ScoreRetrivalPLT" partnerRole="ScoreRetrival"/>
        <partnerLink name="URLRetrievalPL" partnerLinkType="ns1:URLRetrievalPLT" partnerRole="URLRetrieval"/>
        <partnerLink name="invokeWSRPAgent" partnerLinkType="ns1:invokeWSRPAgent" partnerRole="WSRPAgentInvoker"/>
    </partnerLinks>
    <variables>
        <variable messageType="ns1:UserInfoInput" name="UserInfoInput"/>
        <variable messageType="ns1:Response" name="Response"/>
        <variable messageType="ns2:getScoreRequest" name="getScoreRequest"/>
        <variable messageType="ns2:getScoreResponse" name="getScoreResponse"/>
        <variable messageType="ns2:getURLRequest" name="getURLRequest"/>
        <variable messageType="ns2:getURLResponse" name="getURLResponse"/>
        <variable messageType="ns3:NEXTSCORequest" name="NEXTSCORequest"/>
        <variable messageType="ns3:NEXTSCOResponse" name="NEXTSCOResponse"/>
        <variable name="tempString" type="xsd:string"/>
    </variables>
```

Figure 8. The process of test evaluation and learning material assignment
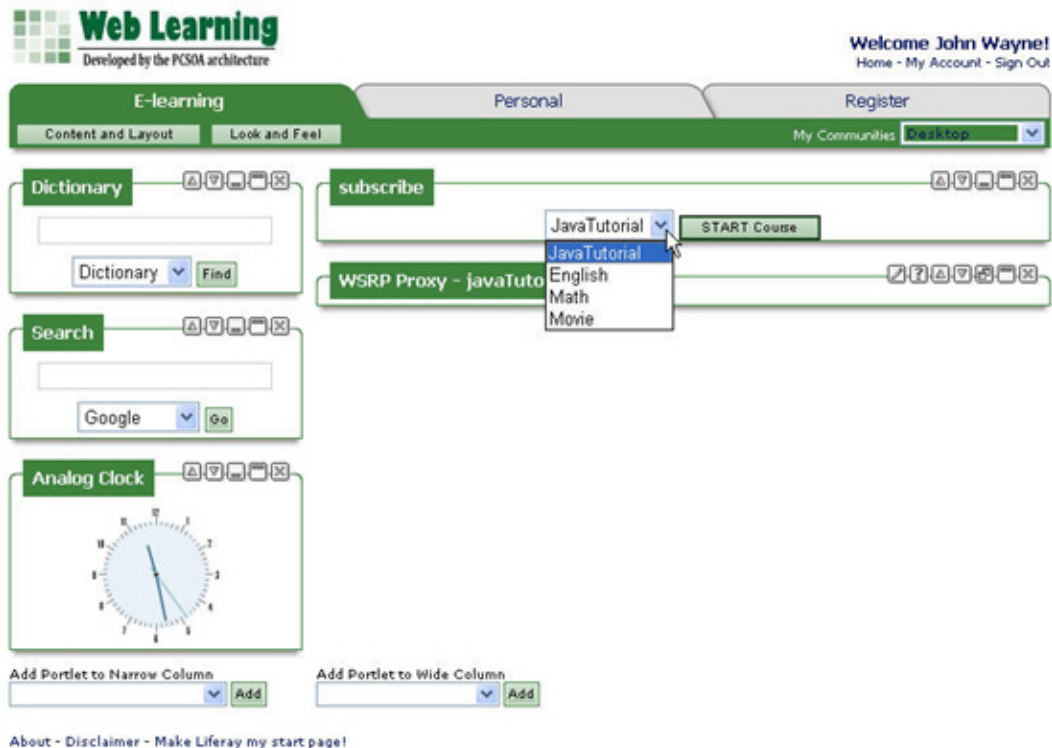

Fig. 9. Screenshot of choosing courses in OEPortal

A learner chooses the course (see Figure 9), then the BPEL4WS engine assigns to the learner the first e-learning material ("JavaTutorial — Syntax and Test," which includes one tutorial and one test) from the LMS1. In our OEPortal, each e-learning material provided for the learner is wrapped as a WSRP service, which enables an

interactive learning session between assigned LMSs and the learner. All related content will be displayed in the form of a portlet plugged within a web page during the learning activity (e.g., in Figure 9, there are five portlets in the learner's web page).

When the learner finishes the first lesson, OEPortal will intercept and save the test score through LRCC and LRRT. The test score will be used for determining the following lesson assigned to the learner.

After obtaining the test score, the BPEL4WS engine will assign next e-learning material for the learner according to the learning strategies defined in the BPEL file. That is, judging by our learning strategy, if the learner gets a score higher than 60, he or she will continue to take the "JavaTutorial — General" lesson. A sequence of screenshots for the demonstrated e-learning workflow is shown in Figure 10.
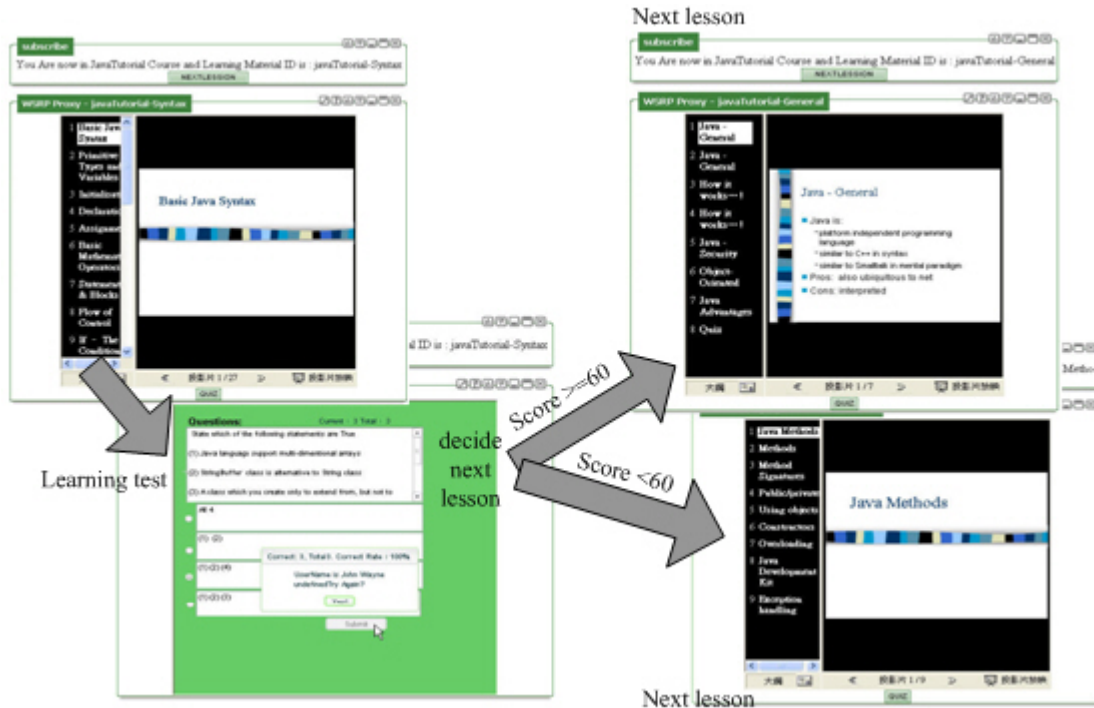

Figure 10. Screenshots of the demonstrated e-learning workflow

In this demonstration, BPEL4WS is used for the course organization and workflow process driving. By using BPEL4WS, all resources of the OEPortal unified Java Technology Preparation Course can be reorganized, rearranged, and reused effectively. BPEL4WS provides good process-oriented integration of services and makes the learning activity definition flexible and more efficient. On the other hand, WSRP is used for making the original e-learning material compatible with open standards and makes all WSRP-based e-learning material sharable among different LMSs by the same protocol. Through the combination of BPEL4WS and WSRP, OEPortal can effectively manage and organize the e-learning materials and provide an open protocol for the participant LMSs to share their heterogeneous e-learning materials. Moreover, a simple learning strategy is applied in the prototype system in order to justify the feasibility of conducting pedagogical theories or learning strategies in our proposed architecture. The experimental result shows that workflow orchestration can be efficiently used for collaborating and reorganizing decentralized learning objects to support the diversity of pedagogical theories and strategies in e-learning.

Conclusion and future works

In this study, we present an open, unified, and interoperable e-learning portal, OEPortal, in order to improve the integration and interoperation among e-learning systems. For integrating e-learning systems effectively, the service-oriented architecture is adopted by OEPortal to solve the problem of integration among heterogeneous platforms. For

the collaboration of learning objects, BPEL4WS is used to compose multi-WSRP services to a task-based learning course adaptively in order to provide abundant learning content to learners. For the presentation level integration of learning objects, WSRP is used to wrap learning objects into sharable learning object portlet services. These wrapped services adhere to portlet standard, which make their reuse and sharing as easy as web components at the presentation level. WSRP, as a pure presentation standard, cannot be effectively integrated via BPEL4WS. We propose a clean and effective modification to make WSRP and BPEL4WS work seamlessly together. Best of all, our modification changes neither WSRP nor BPEL4WS. We believe this is also one of the major contributions of this work. A simple pedagogical and learning strategy is shown in our example for Java Technology Learning via a sequence of reading and testing. Our workflow design can accommodate more complicated workflow, which implements more complicated real-life pedagogical theory. We believe our approach for workflow orchestration of learning objects will benefit learner by efficiently implementing these pedagogical theories. To conclude, OEPortal enhances the e-learning framework with respect to learning object interoperability, learning activity interoperability, and tests and assessments in the service-based e-learning environments, as well as supports the most popular SCORM standard. In practical implications, there are many possible business models for OEPortal to succeed via integrating e-learning systems as a business. Making materials in legacy LMS open and pluggable anywhere erases the technical difficulty and high cost of integrating e-learning systems. If needed, the authorization mechanism to be embedded in WSRP will allow the management functions of portal (such as billing) to be easily implemented for OEPortal. Due to the open nature of service-based workflow orchestration, we believe OEPortal can be extended to incorporate and support other mainstream e-learning standards such as IMS-LD and IMS-QTI in our future works.

References

ActiveBPEL (2006). , Retrieved December 20, 2005 from http://www.activebpel.org/.

ADL (2004). The SCORM 2004 (Sharable Content Object Reference Model) specification. Retrieved December 20, 2005 from http://www.adlnet.org/scorm.

BPEL4WS (2003). Business Process Execution Language for Web Services (BPEL4WS), version 1.1. Retrieved June 30, 2004, from: http://www6.software.ibm.com/software/developer/library/ws-bpel.pdf.

Brusilovsky, P., & Nijhavan H. (2002). A framework for adaptive e-learning based on distributed reusable learning activities. *Proceedings of E-Learn 2002, World Conference on E-Learning in Corporate, Government. Healthcare, & Higher Education* 1, 154–161, Oct. 2002, Montreal, Canada.

Carnegie Mellon Learning Systems Architecture Lab (2003). *SCORM Best Practices Guide for Content Developers,* (1st ed.).

Chang W.-C., Hsu H.-H., Smith T.K. & Wang C.-C. (2004) Enhancing SCORM metadata for assessment authoring in e-learning. *Journal of Computer Assisted Learning 20*, 305–316.

Fjuk A. & Berge O. (2006) Understanding the roles of online meetings in a net-based course. *Journal of Computer Assisted Learning 22*, 13–23.

Fuji, T., & Tanigawa, T. (2002). The methodology for reuse of e-learning resources, *Proceedings of E-Learn 2002, World Conference on E-Learning in Corporate, Government. Healthcare, & Higher Education, 1*, 305–310, Oct. 2002, Montreal, Canada.

Friesen, K., & Mazloumi, N. (2004). Integration of learning management systems and web applications using web services, *Advanced Technology for Learning, 1*(1), 2004.

Gaeta, M., Ritrovato, P., & Salerno, S. (2002, September). Implementing new advanced learning scenarios through GRID technologies. *Proceedings of First LeGE-WG International Workshop on Educational Models for GRID Based Services*, Lausanne, Switzerland.

Hsu, K. C. & Ou Yang, F. C. (2005, July 5–8). OEPortal: An open, unified, and interoperable presentation-preserving e-learning portal, *Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies (ICALT 2005)*, 628–632, Kaohsiung, Taiwan.

Institute of Electrical and Electronics Engineers (1990). A compilation of IEEE standard computer glossaries. *IEEE Standard Computer Dictionary*. New York: IEEE.

IMS Global Learning Consortium: IMS QTI/RES V1.1 (2000). Retrieved December 20, 2005 from http://www.imsproject.org.

IMS Global Learning Consortium: IMS Learning Design Information Model V1.0. Final Spec. (2003). Retrieved December 20, 2005 from http://www.imsglobal.org.

Koper, R. & Tattersall, C. (Eds.). (2005). *Learning design: A handbook on modelling and delivering networked education and training.* New York: Springer.

Koper R. & Manderveld J. (2004). Educational modelling language: Modelling reusable, interoperable, rich and personalised units of learning. *British Journal of Educational Technology, 35*, 537–551.

Kong W., Luo J. & Zhang T. (2005). A workflow based e-learning architecture in service environment. *Proceedings of the Fifth International Conference on Computer and Information Technology* (pp.1026-1032), Washington DC: IEEE Computer Society.

Liber O. (2005) Learning objects: Conditions for viability. *Journal of Computer Assisted Learning, 21*, 366–373.

Liao, C. J., Chi, Y. L., & Ou Yang, F. C. (2003, November 7–11). A grid service oriented platform for mobile learning of English. *Proceedings of E-Learn 2003: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, 2265–2268. Phoenix, Arizona, USA.

Liao, C. J., & Ou Yang, F. C. (2004, July 21–25). A workflow framework for mobile learning objects integration by employing grid services flow language. *Proceedings of International Conference on Education and Information Systems, Technologies and Applications (EISTA 2004), 1*, 155–160. Orlando, USA.

Liao, C. J., Ou Yang, F. C., & Hsu, K. C. (2005). A service-oriented approach for the pervasive learning grid, *Journal of Information Science and Engineering (JISE)* 1(5), 959–971.

Liferay Portal (2006). Retrieved February 16, 2008, from: http://www.liferay.com/web/guest/home.

Portlets Specification final release (2003), JSR168 of JCP, Retrieved September 20, 2004 from http://jcp.org/en/jsr/detail?id=168.

Lin J., Ho, C., Sdiq, W., & Orlowska, M. E. (2001). On workflow enabled e-learning services. *Proceedings of IEEE Advance Learning Technology Conference* (pp.345–352), Los Alamitos: IEEE Computer Society.

Liu, X. Q., Wu, M., Chen, J. X. (2002). Knowledge aggregation and navigation high-level Petri nets-based in e-learning. *Paper presented at the First International Conference on Machine Learning and Cybernetics*, November 4-5, 2002, Beijing, China.

Nicosia (2004). Carrying on the e-learning process with a workflow management engine. *Proceedings of the 2004 ACM Symposium on Applied Computing*, March 14 -17, 2004, Nicosia, Cyprus.

Puustj¨arvi, J. (2004). Using one-stop portal in integrating eLearning systems. *Advanced Technology for Learning, 1* (2), 99-106.

Puschmann, T., & Alt, R. (2004). Process portals: Architecture and integration. *Paper presented at the 37th Hawaii International Conference on System Sciences*, January 5–8, 2004, Hawaii.

Reklaitis, V., Baniulis, K., & Masevicius, A. (2002, December). Towards e-learning application architecture based on GLOBUS framework. *Paper presented at the Euroweb Conference*, December 17-18, 2002, Oxford, UK.

Reklaitis, V., Baniulis, K., & Okamoto, T. (2003). Shaping e-learning applications for a service oriented grid. *Proceedings of 2nd International LeGE-WG Workshop on e-Learning and Grid Technologies: A Fundamental Challenge for Europe*, March 3-4, 2003, Paris, France.

Farmer, R. A. and Hughes, B. (2005). A situated learning perspective on learning object design. *Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies* (pp.72-74), Los Alamitos: IEEE Computer Society.

Ternier, S., Olmedilla, D., & Duval, E. (2005, June/July). Peer-to-peer versus federated search: Towards more interoperable learning object repositories. *Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications*, June 27 – July 2, 2005, Montreal, Canada.

The TimesTen Team (2000). High performance and scalability through application-tier, in-memory data management. *Paper presented at the 26th International Conference on Very Large Databases*, September 10-14, 2000, Cairo, Egypt.

Web Services for Remote Portles (WSRP) specification of OASIS (2003) version 1.0. Retrieved October 2, 2004, from http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf.

Appendix

The BPEL source code for the process of test evaluation and learning material assignment.

```xml
<switch>
    <target linkName="L4"/>
    <source linkName="L3"/>
    <case condition="number( bpws:getVariableData('getScoreResponse',
'getScoreReturn') )&lt;0">
        <assign name="AssignSCOToJavaTutorial-Syntax">
            <copy>
                <from expression="'javaTutorial-Syntax'"/>
                <to part="SCO_ID" variable="getURLRequest"/>
            </copy>
        </assign>
    </case>
    <otherwise>
        <switch>
            <case condition="number( bpws:getVariableData('getScoreResponse',
'getScoreReturn') )&gt;=60">
                <assign name="AssignSCOToJavaTutorial-General">
                    <copy>
                        <from expression="'javaTutorial-General'"/>
                        <to part="SCO_ID" variable="getURLRequest"/>
                    </copy>
                </assign>
            </case>
            <otherwise>
                <assign name="AssignSCOToJavaTutorial-Method">
                    <copy>
                        <from expression="'javaTutorial-Method'"/>
                        <to part="SCO_ID" variable="getURLRequest"/>
                    </copy>
                </assign>
            </otherwise>
        </switch>
    </otherwise>
</switch>
```