For The First International Conference on Web Services (ICWS'03)

Web Services Technology and Learning Technology

- A Web-services Model for Constructing Virtual Learning Environments

Lipeng Wen **Chris Jesshope**
L.Wen@dcs.hull.ac.uk **C.R.Jesshope@dcs.hull.ac.uk**

Department of Computer Science
University of Hull
HULL, UK HU6 7RX

Abstract:

Web Services Technology is a dazzling star in distributed computing technology, which provides a new way to integrate existing systems or applications, and the ability to access data in a heterogeneous environment. In this paper, we give a definition of the term "Web Services", describe and analyse its categories and characteristics. A Web-services Management Architecture is then defined based on Web Services Architecture, Workflow management and Sun ONE (Open Network Environment) Architecture. Design issues are represented and analyzed, such as how to design Web Services Metadata, to create Web Services Packages. The application of this work in the area of learning technology to make learning on the Web more efficient, we integrate Web Services Technology within learning technology, which offers a powerful solution in response to the current challenges in virtual learning environments. The result of this integration is a high-level Web-services model for both client/server and Peer-to-Peer frameworks for constructing decentralized, virtual-learning environments. The interoperable system components are defined and designed in this model.

Keywords: Web services, Web-services Management Architecture, Web services Metadata, Web-services Packages, Decentralized virtual-learning Environments

1. Introduction

Knowledge is becoming the main power to support the development of individuals, corporations and nations. From a technological point of view, it is essential to develop powerful virtual learning environments (VLEs) for anyone who needs to acquire that knowledge. In fact, there have already been a profusion of VLEs both commercial and open-source, which improve the learning effectiveness and efficiency, but they still can not solve some of the key technical and pedagogical issues that can be expressed as follows:

- The lack of an ability to easily integrate existing comprehensive software and Enterprise Resource Planning (ERP) [1] within Learning Management Systems (LMS) [2].
- A difficult in implementing a low-cost distribution of large learning objectives. With a few exceptions [3], Virtual learning environments are built on the client/server model and run entirely on the Web server, so it is impossible to allocate learning objectives to users using other distribution Channels.
- It is difficult to implement an organization's self-organizing requirements. So far, instructors can organize their teaching environments, i.e. create, run and maintain learning materials. But for administrators, enterprise workers, learners, evaluators and authors, the work is still ongoing.
- There is a lack of adaptability to new technologies such as the dynamic publication of units of learning according to a learner's preferences of knowledge. Educational modelling languages are now emerging [4] but are can not easily be incorporated into existing VLEs.
- There are platform-independence and heterogeneous network environment problems. Generally existing systems are heavily dependent on some certain operating systems and are built in

homogeneous network, such as IRI-h (Interactive Remote Instruction-heterogeneous) runs on UNIX system [5], so we need to think about how to build a platform-neutral system in a heterogeneous environment.

- It is difficult to search and locate instructional materials. We have methods to describe and search for them, such as the use of IEEE LOM, but there remain problems. For example, the metadata subsets allowed in IEEE LOM are many and different subsets can be used in different existing systems, which make it very difficult to maintain consistency in searching through different metadata subsets.

In order to deal with these problems, several methods, such as Workflow Management [6], have been adopted by researchers. Unfortunately, only a little progress was achieved. In this paper, we consider Web Services technology as the most efficient, practical and secure solution used to resolve the majority of the above issues. Based on Web-services technology, we have designed Web Services Management architecture (WSM) [7] and integrate it within learning technology [8]. A high-level Web-services model is created and the logical system components are defined and designed, which can be used to construct a Decentralized Virtual Learning Environment. A Decentralized Virtual Learning Environment can be defined as a Web-Services-based, virtual-learning environment, in which the structure is decentralized and activities and services are integrated in terms of the roles that require and/or provide services in the overall system. It is scalable, dynamic and self-organizing and suitable for the next generation of networked computing: Network of Embedded Things [9].

2. Web Services Technology

2.1. Web Services

So far, there is not a generally accepted definition about Web Services. In our opinion, it can be defined as a set of services that can be implemented individually or in batches on the Web to which a request is composed and sent, and an open, technology-neutral and service-neutral interface is created dynamically by a Web Services management system that defines users, systems and/or applications to access a specific or group of functions or data sources. Web Services provide a modular, well-defined and encapsulated service, based on standards such as XML, XSLT, SOAP, WSDL, RDF, DTD, UDDI [10] and WSRP [11], which are used for implementing loosely coupled integration between the systems or applications.

Web Services can be defined as three categories:

i. Web Services that can be provided and/or requested by a client application over the Web, which is always a Web browser. In order to efficiently implementing the functions provided by Web-Services Technology, it is necessary to extend Web browsers, so as integrate the Web-Services Technology within them. This is normally accomplished using Java Technology embedded in a Web browser.
ii. Web Services that can be provided and/or requested by an application running on a computer, which can communicate and interoperate with a Web Services Management system that can be embedded into existing software infrastructures, such as J2EE or CORBA.
iii. Web Services that can be provided and/or requested by a Java enabled client with Web Services enabled capability, e.g. PDAs or other mobile devices.

The characteristics of Web Services that learning technology lacks of can be described as follows:

- The ability to build systems or applications across program relationship models such as client/server [12] and Peer-to-Peer [13] models. In other words, a system or an application constructed by using Web-Services Technology can have two status based on these models which make them can neatly transform between two roles (client and server) to meet different requests from other systems or applications.
- The ability to provide an efficient and normative way to search and discover Web materials. A Web service is identified by a URI, which can describe a service in a common way and publish it over the Web.
- The ability to communicate and collaborate across systems or applications on the Web despite of programming languages, system platforms and network structures such as Heterogeneous Network and Virtual Private Network (VPN).
- Web Services technology is developed based on standards, so it is across-systems compatible, and applications can adapt to new technologies and future requirements.

2.2. Web Services Management Architecture

Web Services provide the ability to interact and communicate between unknown and unrelated systems or applications on the Web in a standard-based way that is easier and more efficient than the previously

ad-hoc interaction methods, Web Services Technology therefore, can be the key to a Services-Oriented Technology. In our opinion, Web Services Management (WSM) can be defined as the core building block of Web Services Technology, which enables the efficient implementation of the capabilities of Web Services such as deploying Web Services in various ways and on different nodes on the Internet. Currently, the OASIS Web Services Distributed Management (WSDM) Technical Committee is working on defining WSM based on Web Services Architecture [14].

According to our experiences and prior research, especially on Web Services Architecture, Workflow Management and Sun ONE Architecture, Web-services Management should be composed of three key architectural components:

The Role: this refers to the personalized description of the capability to perform a certain kind of function. Generally, there are three basic roles in Web-services Management:

i. *Web Services Requestor:* It can be a client application, an application running on a computer or any Java enabled client, which requests services on the Web.
ii. *Web Services Provider:* It can be a client application, an application running on a computer or any Java enabled client, which provisions and delivers services on the Web.
iii. *Web Services Mediator:* It is the core component of Web-services Management that provides a services engine that manages the performance of business processes and a set of interfaces, which are open, technology-independent and service-independent.

The Web Service: this is a software building block that is interoperable, modular and can be loosely coupled integrated and dynamically discovered and allocated to Web Services Requestor.

The Business Process: this is a time-limited procedure from requesting a service to receiving a response, which is always divided into some activities that are managed by Web Services Mediator.

We present a model of our definition of Web-services management in the following diagram:

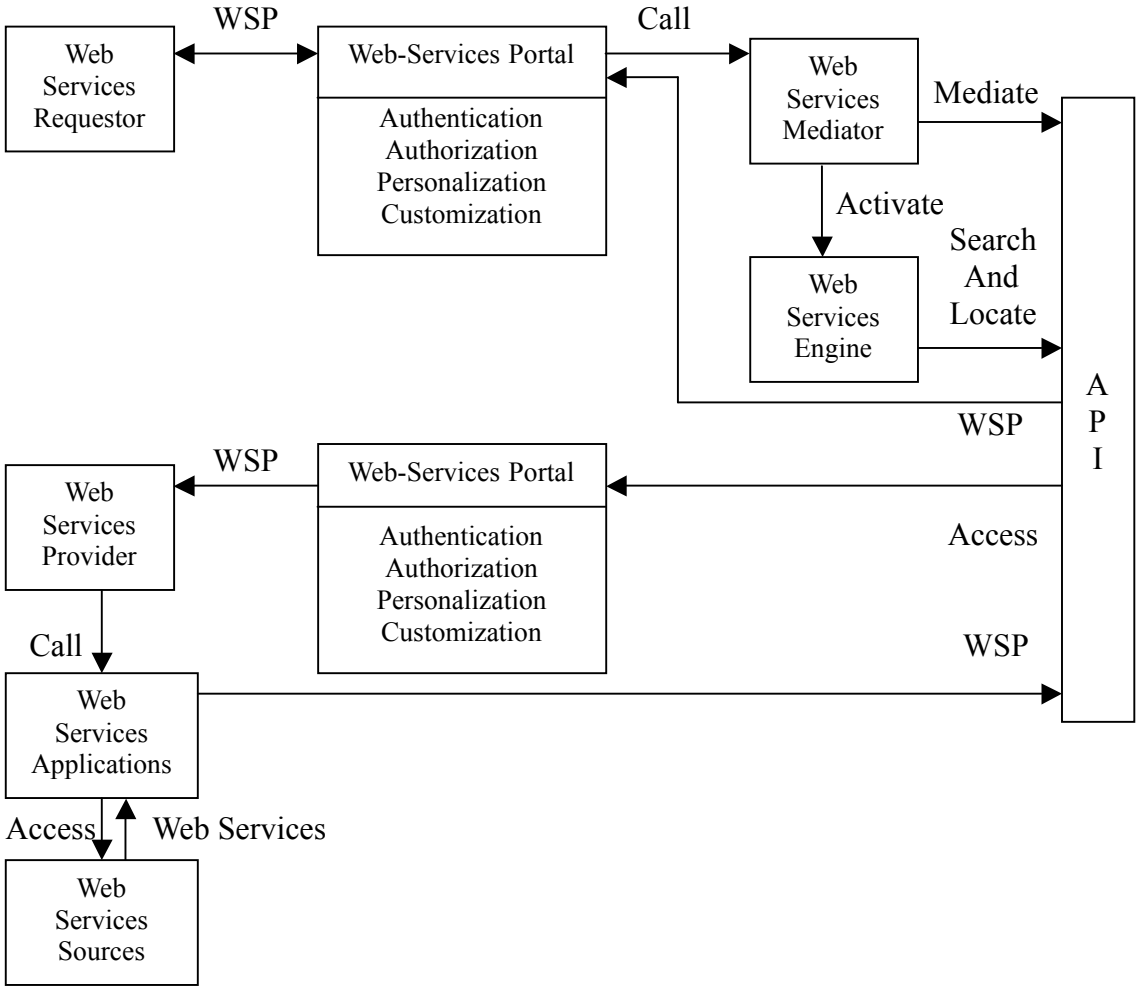

Figure 1 Web-services Management Model

Notes:

i. WSP is the acronym for Web Services Package, which contains all of information about the Role, Web Services and Business Process and is created based-on Web Services Metadata.

ii. The relevant standards for this model can be described as the following table:

WSP	XML, eb XML, XSLT, WSDL, RDF, DTD, ODRL, WSBPEL, BPSS, etc.
Call	JMS, IIOP, HTTP, FTP, SMTP, etc.
Mediate	JMS, IIOP, HTTP, FTP, SMTP, etc.
Activate	JMS, IIOP, HTTP, FTP, SMTP, etc.
Search and Locate	UDDI, RDDL, URL, URI, etc.
Access	SOAP, WSRP, ODRL, etc.
Web Services	XML, WSDL, RDF, DTD, ODRL, etc.

Table 1 Standards and Protocols for Web-services Management Model

iii. Most of these standards are well known, however references to the less well known are given as follows: ebXML [15], WSDL [16], ODRL [17], WSBPEL [18], BPSS [19], JMS [20], IIOP [21].

Currently, we pay more attention to the capability of Web-services Management on facilitating architectural components to communicate and interoperate with each other on a heterogeneous network. In brief, the main problems can be expressed by:

- Lack of standards on Web Services Metadata, which defines the structure of content of Web Services and can be the infrastructure of Web Services Description Language (WSDL).
- So far, Web-services technology has always been deployed based on a client/server model, which constricts the used and range of Web-services technology. For example, with the development of distributed computing technology, systems or applications can become more powerful if they are built on both peer-to-peer and client/server model.
- In order to dynamically allocate Web Services on heterogeneous network, we have to consider how to define a standard-based and Web-services-neutral API to deliver Web services across networks with various frameworks.

3. Integration between Web Services Technology and Learning Technology

As discussed above, Web-services technology has the inherent ability to solve various issues in distributed computing technology. In this paper, we focus on integrating Web-services technology within learning technology. The crucial work in this loosely coupled integration is to assemble Web-services management architecture into a Learning Management System (LMS) [22] to create a Web-services LMS, and to create a unified communication interface, which allows architectural components in an VLE to interoperate with each other and to different LMSs or VLEs.

By integrating Web-services management architecture into a VLE, it becomes easier and more efficient to accurately analyze a business process and decompose it into its constituent activities. The related roles, services and data are integrated to form system components in a loosely coupled manner. Each component is modular, encapsulated, can dynamically locate and interact with other components on the Internet to provide services. It is also possible for one component to be combined with other components to make a more complex application. Components are usually invoked not by a human user but by an application or a system. In a word, roles, activities, services and related information are integrated in a loosely coupled style to constitute components and allow them to interoperate with each other.

When an entity (see section 4) plays one or more roles, one or more System components are allocated to the entity. It appears to be cut from the Web-services LMS server (see figure 2,3) so we call them “Slices”. Some System components are divided into sub-systems that are responsible for specialized functions in the system. Each entity that works with one or more System components or sub System components is authorized to interoperate with other systems or applications such as existing authoring tools through a common tool interface.

3.1. A Web-Services Model

In consideration of the powerful superiority of Web-services technology, a Web-services model is used to design and construct decentralized virtual learning environments. The model is a new web services framework on both client/server and peer-to-peer program relationship models, which is created in accordance with the standards such as IEEE LOM, ADL SCORM, IMS, DCMI, ISO/IEC JTC1/SC36 [23]. It is a new style of loosely coupled integration between systems or applications. It focuses on dynamic, scalable and self-organizing system in which both pear-to-peer transactions and server-centric communications exist. System components are distributed to any communication node over the web according to the existing and potential services' requirements from entities, each communication node is given both server and client capabilities.

The model can be illustrated in two following diagrams. In the figure 2, four system components are defined as three categories: *Process Category* (which are the active components), e.g. Slices Repository including sub components such as learner slice consisting of learner and instructional space (see figure 3); *Store Category* (which are inactive components), e.g. Slices Information Repository; *Flow Category* (which are inactive components), e.g. Institutional Portal and Interfaces Repository.

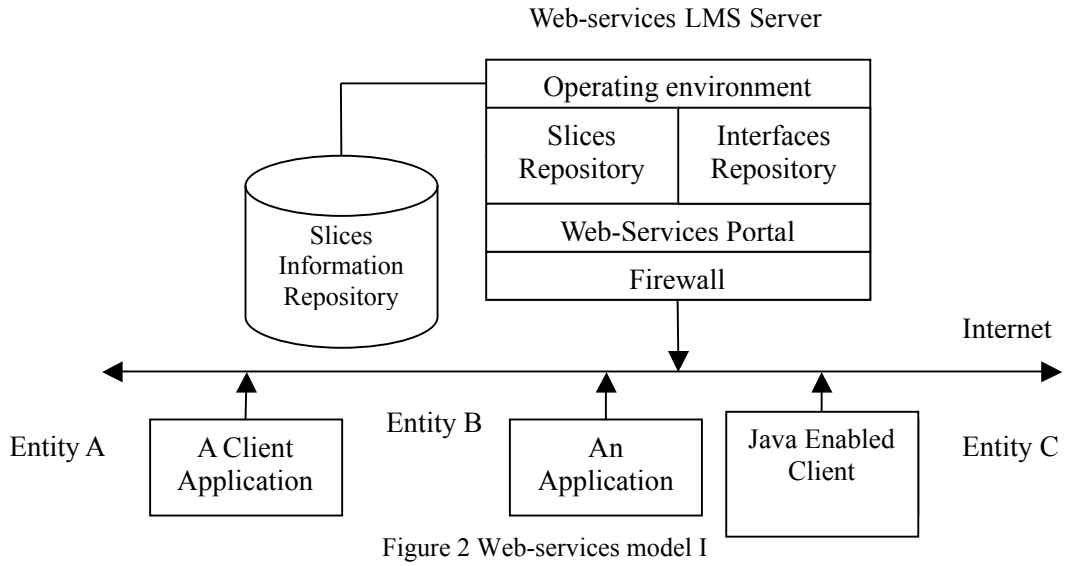

Figure 2 Web-services model I

In the above diagram, we can identify the following components:

- *An Entity:* which points to anyone who plays one or more roles in the system, such as learner, instructor, author, evaluator, organization and administrator. In the model, entities are defined as three categories according to the ways that services are delivered on the Web: Entity A is a client application; Entity B is an application running on a computer and Entity C is a Java enabled client (see section 2.1.: Web Services category)
- *A Web-Services Portal:* which is a special kind of gateway from which entities can locate all the Web resources and services required. It is customized to the role of a user [24, 25] and plays a role as a display system to search and access instructional materials by entities. Elements that might appear in portals include access to various kinds of data, instructional tools and learning objects (often arranged into "channels") such as support for collaborative learning.
- *A Slices Repository:* which is a collection of components of Process category (see figure 3) in which roles, services and relevant data are defined and integrated to form some System components. For example, a wizard program for system administrator is used to create a new component.
- *An Interfaces Repository:* which is a collection of system interfaces, such as user interface and application programming interface (API). In a decentralized virtual learning environment, there are two categories of system interfaces: One is used to communicate between slices and the LMS; the other is used to transfer information among slices.
- *A Slice-Information Repository:* which is a collection of components of Store category that comprises three components: role-service-activity database, instructional materials database and finance Database. The Slice-information database contains all information on certain roles and conglomerated services that are packaged into relevant components. There are many advantages a component approaches, such as they can be distributed, updated and synchronized in accordance with any entity's requirement anywhere and at any time, and are easy to be assembled and disassembled.

Figure 2 represents the initial status of the Web-services model that is built on client/server model, and its run-time state illustrated in figure 3 is established on both client/server and peer-to-peer models. From the initial status to the run-time status, the system is decomposed according to the different slices. That is, the slices, e.g. a learner slice, an instructor slice and an administrator slice, are dynamically allocated to entities on their roles (see figure 3), which comprise the following modules:

- *An Instructional space:* It is an interesting, efficient, comprehensive and powerful learning environment that mainly contains two parts: learner's space and instructor's space.
- *An Instructional Materials IDE:* It points to a tools package that authors use to create instructional materials, such as new books in each knowledge area, and improve existing learning objects. It provides an integrated working environment not only for an author to achieve some things individually but also for authors to work together.
- *An Assessment Studio:* It is an integrated environment for evaluators to assess instructional materials and the qualifications of instructors, authors and organizations.
- *Tools for Offering Credits:* It is a system to assess learners' achievements and award appropriate certificates to them.

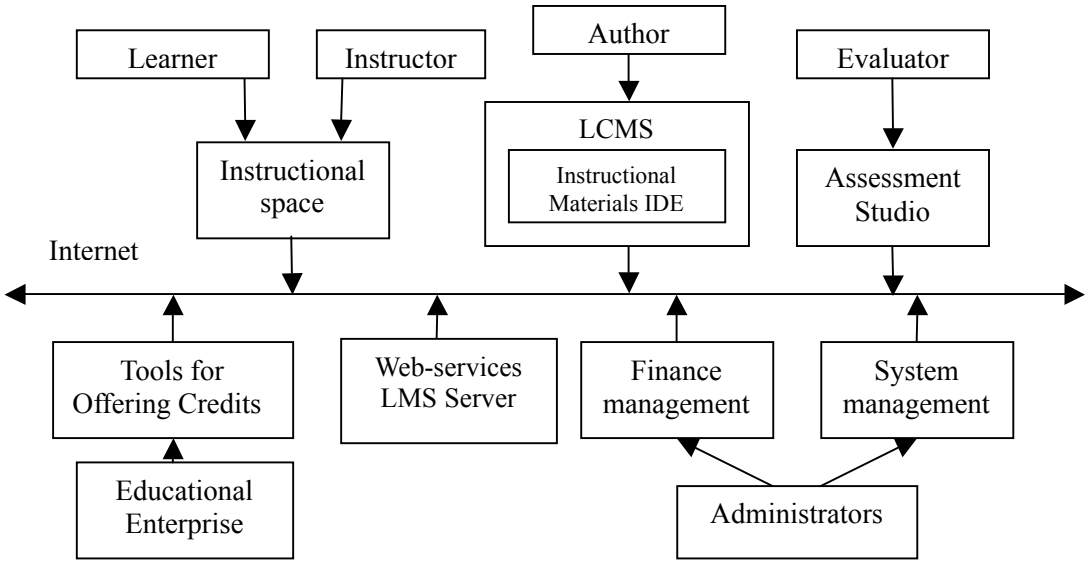

Figure 3 Web-services model II

4. Conclusion

In the discussion undertaken in the paper, we have defined a Web-services management architecture and integrated Web-services technology into a model for an on-line learning environment. A Web-Services model is created to construct decentralized virtual learning environments.

In our future work, we will focus on some key issues in perfecting Web-services management and its integration with learning technology. For example,

- How to design an open, technology-neutral and Web-services-neutral interface.
- How to build Web Services Packages based on Web Services Metadata.
- How to effectively support both client/server and Peer-to-Peer models simultaneously. In short, how to move smoothly between slices, or between each slice and LMS server in two system models, which enables much more flexibility in the system. It solves for example the issue of on-line/off-line browsing by allowing the learner to host their own Web services and to synchronise periodically with other slices.
- How to design a Web-services portal framework to be suitable for dynamic allocation of Web services to various entities involved in the on-line learning with personalized and customized presentation. We wish such a portal to hold all of the functions of a managed learning environment, e.g. an ability of self-organizing aggregation and integration of diverse source of information.

5. Reference

[1] S. V. Scott, E. L. Wagner, Available online 18 April, 2003, *Networks, negotiations, and new times: the implementation of enterprise resource planning into an academic administration,* Information and Organization, In Press, Corrected Proof.

[2] Camillan Huang, March-June 2003, *Changing learning with new interactive and media-rich instruction environments: virtual labs case study report,* Computerized Medical Imaging and Graphics, Volume 27, Issues 2-3, Pages 157-164.
[3] Jesshope, C.R., 2001, *Easy-to-use multimedia tools and scalable distributed architectures for Web-based teaching and learning*, Proc DCABES 2001, HubeiSci. And Tech. Press, Wuhan, China, ISBN 7-5352-2722-8, Pages 52-60.
(Keynote paper: http://www2.dcs.hull.ac.uk/people/csscrj/papers/dcabes.ptf)
[4] IMS Global Learning Consortium Inc, 2003, *IMS Learning Design Specification*, http://www.imsglobal.org/learningdesign/, accessed on 24 April, 2003.
[5] R. Maly, H. Abdel-Wahab, C. Wild, C. M. Overstreet, A. Gupta, A. Abdel-Hamid, S. Ghanem, A. Gonzalez, X. Zhu, Old Dominion University, VA, March 2001, *IRI-h, a Java-based distance education system: architecture and performance*, Journal of Educational Resources in Computing (JERIC)**,** Volume 1, No. 1.
[6] David Hollingsworth, January 1995, *The Workflow Reference Model, Workflow Management Coalition (WFMC)*, http://www.wfmc.org/standards/docs/tc003v11.pdf, accessed on 21 April, 2003.
[7] Rich Seeley, March 2003, *Web services management spec work underway*, http://www.adtmag.com/article.asp?id=7415, accessed on 24 April, 2003.
[8] Tak-Wai Chan, Chih-Wei Hue, Chih-Yueh Chou, Ovid J. L. Tzeng, September 2001, *Four spaces of network learning models*, Computers & Education, Volume 37, Issue 2, Pages 141-161**.**
[9] Sun Microsystems, Inc., 2002, *Sun ONE Architecture Guide*, http://wwws.sun.com/software/sunone/, accessed on 21 April, 2003.
[10] The World Wide Web Consortium (W3C), http://www.w3.org/, accessed on 24 April, 2003.
[11] OASIS Web Services Remote Portal TC, April 2003, *Web Services for Remote Portlets Specification 1.0 - Committee Specification 0.95*, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp, accessed on 25 April, 2003.
[12] Peter Duchessi, InduShobha Chengalur-Smith, May 1998, *Client/server benefits, problems, best practices*, Communications of the ACM, Volume 41, Issue 5, Pages 87-94.
[13] Philip E. Agre, February 2003, *Technical and social components of peer-to-peer computing: P2P and the promise of internet equality*, Communications of the ACM, Volume 46, Issue 2, pages 39-42.
[14] Web Services Architecture Working Group, November 2002, *Web Services Architecture*, http://www.w3.org/TR/2002/WD-ws-arch-20021114/, accessed on 22 April, 2003.
[15] Carol Geyer, March 2003, *OASIS and UN/CEFACT to Host ebXML Showcase at XML Europe 2003 in London,* http://www.ebxml.org/news/pr_20030327.htm, accessed on 22 April, 2003.
[16] Roberto Chinnici, Martin Gudgin, Jean-Jacques Moreau, Sanjiva Weerawarana, March 2003, *Web Services Description Language (WSDL) Version 1.2: working draft*, http://www.w3.org/TR/2003/WD-wsdl12-20030303/, accessed on 23 April, 2003.
[17] The Open Digital Rights Language (ODRL) Initiative, August 2002, *Open Digital Rights Language version 1.1*, http://odrl.net/, accessed on 24 April, 2003.
[18] OASIS Web Services Business Process Execution Language (WSBPEL) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel, accessed on 23 April, 2003.
[19] eBTWG BPSS Project Team, *Business Process Specification Schema Project Proposal*, http://www.ebtwg.org/projects/documentation/bpschema/ppbpschema.html, accessed on 24 April, 2003.
[20] Sun Microsystems, Inc., April 2003, *Java Message Service API*, http://java.sun.com/products/jms/, accessed on 23 April, 2003.
[21] Object Management Group (OMG), December 2002, *CORBA 3.0.2 specification*, http://www.omg.org/technology/documents/formal/corba_iiop.htm, accessed on 23 April, 2003.
[22] IBM Corporation, 2003, *Next generation Learning Management System debuts*, http://www-3.ibm.com/software/swnews/swnews.nsf/np/shoy5hztby, accessed on 24 April, 2003.
[23] Luis E. Anido, Manuel J. Fernández, Manuel Caeiro, Juan M. Santos, Judith S. Rodríguez, Martín Llamas, May 2002, *Educational metadata and brokerage for learning resources*, Computers & Education, Volume 38, Issue 4, Pages 351-374.
[24] Karen Calhoun, Summer 2002, *From information gateway to digital library management system: a case analysis*, Library Collections, Acquisitions, and Technical Services, Volume 26, Issue 2 , Pages 141-150.
[25] Volker Gruhn, Martin Mocker, Lothar Schöpe, February 2002, *Experience: Development of an electronic commerce portal system using a specific software development process*, Proceedings of the Fortieth International Conference on Tools Pacific: Objects for internet, mobile and embedded applications, Volume 10, Pages 93-101.