

Design and Composition of e-Learning and Research Resources using Service Oriented Architecture

Pradeep Gurunathan
Professor
Dept. of Information Technology
A.V.C College of Engineering
Mayiladuthurai, India

Dr. Seethalakshmi Pandian
Director
Centre for Academic Excellence
Anna University of Technology
Tiruchirappalli, India

Balamohan
Somasundaram
Software Architect
Wipro Technologies
Chennai, India

ABSTRACT

Service-oriented architectures (SOA) is an emerging approach that addresses the requirements of loosely coupled, standards-based, and protocol independent distributed computing. To build an SOA a highly distributable communications and integration backbone is required. This functionality is provided by the Enterprise Service Bus (ESB) that is an integration platform that utilizes Web services standards to support a wide variety of communications patterns over multiple transport protocols and deliver value-added capabilities for SOA applications. This paper posits a technology and approaches that unify the principles of SOA, mathematical definitions for individual services and how the functions are designed to offer a manageable, standards-based SOA backbone that extends middleware functionality throughout by connecting heterogeneous components and systems and offers integration of services for a e-learning and research resources.

Keywords: Service Oriented Architecture, Distributed Computing, Enterprise Service Bus, e-Learning

1. INTRODUCTION

The emergence of Web services developments and standards in support of automated business integration has driven major technological advances in the integration software space, most notably, the service-oriented architecture (SOA) ([30,33]). The purpose of this architecture is to address the requirements of loosely coupled, standards-based, and protocol-independent distributed computing, mapping enterprise information systems (EIS) appropriately to the overall business process flow.

Firstly, an SOA-based service is self-contained, i.e., the service maintains its own state. Secondly, services are platform independent, implying that the interface contract to the service is limited to platform independent assertions. Lastly, the SOA assumes that services can be dynamically located, invoked and (re-)combined.

Logically, a service in an SOA is a bound pair of a service interface and a service implementation. Service interface defines the identity of a service and its invocation logistics. Service implementation implements the work that the service is designated to do. Because interfaces are platform independent, a client from any communication device using any computational platform, operating system and any programming language can use the service. These two facets of the service are designed and maintained as distinct items, though their existence is highly interrelated.

The SOAs and Web services solutions support two key roles: a service requestor (client) and service provider, which communicate via service requests. A role thus reflects a type of a participant in an SOA ([33,34]). The interaction involves the publishing, finding and binding of services.

Service requests are messages formatted according to the Simple Object Access Protocol (SOAP) [31]. SOAP entails a light-weighted protocol allowing RPC-like calls over the Internet using a variety of transport protocols including HTTP, HTTP/S and SMTP. In principle, SOAP messages may be conveyed using any protocol as long as a binding is defined. The SOAP request is received by a run-time service (a SOAP “listener”) that accepts the SOAP message, extracts the XML message body, transforms the XML message into a native protocol, and delegates the request to the actual business process within an enterprise.

Requested operations of Web services are implemented using one or more Web service components [35]. Web service components may be hosted within a Web services container [36], serving as an interface between business services and low-level, infrastructure services.

Web services containers are providing facilities such as location, routing, service invocation, and management. In particular, a service container is the physical manifestation of the abstract service endpoint, and provides the implementation of the service interface. In addition, service containers provide facilities for lifecycle management such as start up, shutdown, and resource cleanup. A service container can host multiple services, even if they are not part of the same distributed process.

2. MATHEMATICAL REPRESENTATION OF SOA

A service orientation principle may be formulated as “a set of computing capabilities of a service-oriented environment for any given moment T, determined by the kind of the dynamically available (deployed) services”. Particular set of conventions for software designed for such environment makes up particular Service-Oriented Architectures.

From the above reasoning, we propose that:

$$SOA = Principle + Platform + Technology$$

The importance of this statement emerges in the context of Enterprise Application Integration in large organization: in fact it is practically impossible to provide a universal platform that would strike a perfect fit for all tasks. On the other hand, the service orientation principle enables different products to be designed independently but ensuring their potential integration viability.

2.1 Logical Service Definition

Service Oriented Architecture consists of n number of services and processes. A given logical service *i* is deployed in to an environment to provide the useful functionality *f_i*, expressed as a programmatic interface *l_i*. Important feature of a service is its capability to interact dynamically, in the given environment, with other services and non-service entities (such as end users). Logical service' implementation is thus a set of coordinated and interacting processes:

$$S_i = \langle P_1^i, P_2^i, \dots, P_n^i, \wedge \rangle$$

Where *S_i* — logical service instance, *P_{kⁱⁱ}* — *kth* process implementing the logical service functionality *f_i* through the programmatic interface *l_i*, and \wedge — network communication function between individual processes.

2.2 Environment Service Definition

Environment Service consists of a set of logical services implementation. The environment service is represented as

$$Env_T = \langle S_1; S_2; \dots; S_n \rangle;$$

where *n* — number of logical services deployed in the environment. Generally, service environments may be distinguished in various ways, for instance:

- On the basis of the communication protocol such as HTTP, HTTP/S and SMTP.
- On the basis of service' content. E.g., travel site service environments which offer logical services are different from eCommerce service environments.

2.3 Service-Oriented Application Definition

Overall functionality *F* of a service-oriented application *A* is determined by the logical services involved in the application and an environment

$$F_A = \langle S_1^A, S_2^A, \dots, S_n^A \rangle Env_T$$

Moreover, the *Application Functionality represented in a directing graph* is defined as:

$$V_A = (F_A, G)$$

having vertexes from the *F_A* set and verge set *G* formalizing the coordination between individual logical services of the *F_A* set. Finally, the Service-Oriented application *A* may be formalized as the set:

$$A = \langle F_A, V_A \rangle$$

3. ARCHITECTURE OF SERVICES FOR WEB-BASED EDUCATIONAL SYSTEM

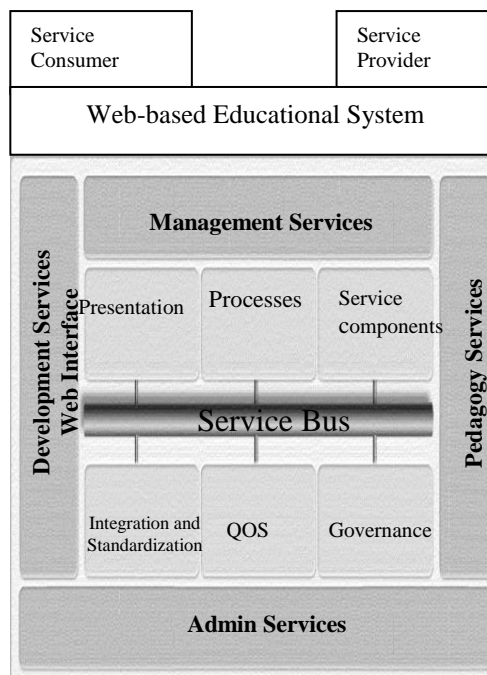


Figure 1: Architecture of Services for Web-based Educational System

In this section, we have created a SOA-based architecture to define the proposed Services for web-based educational system. Figure 1 introduces two views of Services for web-based educational system: Services Consumer view and Services Provider view. The users of web-based educational system include educators, students, and practitioners.

The major objective of creating this architecture is to make the Services Computing Curriculum more configurable, flexible, and extensible through a set of best practices in terms of processes, methods, and tools. The expected results are some reusable components in the Presentation Layer, Process Layer, Services Layer, Service Components Layer, Integration Layer, Quality of Services (QoS) Layer and Governance Layer. All those reusable modules will be used to compose a set of operational services for web-based educational system.

Presentation - The Presentation Layer includes various ways of representing knowledge in the field of Services Computing. The representations include (electronic) text books, presentation files, audio clips, video clips, web contents and contents in other formats.

Processes - The Process Layer includes all processes which are related to the course design, delivery, and management processes. In terms of course development, the course creators need to systematically form a set of activities to support the creation and refinement of courses. From the teaching process perspective, building and enhancing a set of well-documented processes can help delivering teaching services effectively for experienced educators as well as new educators. In the course development and teaching practices, we can also accumulate a set of processes to enhance the communications among all the role players in the education eco-system.

Service Components - The Service Components Layer refers to all supporting events and forums that help realize or produce reusable services defined in the Services Layer. Each service can be offered through single or multiple events and forums.

Each event may be able to help deliver multiple services as well.

Creating standards or leveraging standards to capture the best practices in the field of Services Computing is a very good addition to build a solid pool of service components in a service oriented web-based educational system. For example: we developed a tool for conference management system. The services can be accessed by others on demand.

Integration and Standardization - The Integration Layer is responsible for seamlessly integrating and standardizing knowledge representations from various sources based on certain policy and guidance. The executors of the integration service include individuals and organizations. All volunteers, including teachers, practitioners and other helpers, are part of the individual's community. The organization type of executors includes professional societies and industries.

Quality of Services - The parameters of Quality of Services (QoS) have to be associated with all the elements of web-based educational system architecture. The course time, assessment test level and measurable results should be clearly defined. For example, the number of hours for a specific course should be defined to make sure the necessary materials can be taught during the course teaching process.

Governance - The Governance Layer covers all governing policy and enablement framework including best practices, guiding principles, recommended text books and so forth.

4. PROGRAMME MANAGEMENT METHODOLOGY USING SERVICES

In this section, we present our findings in developing the services of web-based educational system based on the dimensions learning and research resources.

Dimension 1: Learning Resources

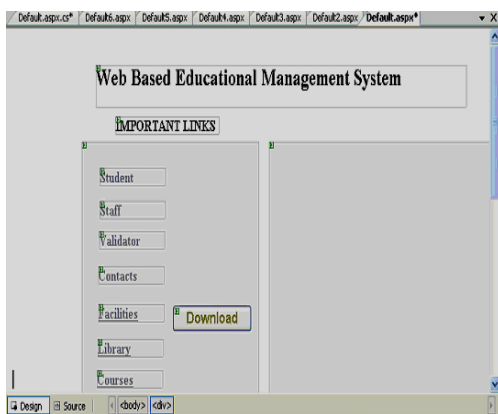


Figure 2: System Home Page

The Administrator system includes six modules: Login, Student, Instructor, Test, System, and Export modules. After logging into the system through the Administrator Login Module, a system administrator can manage (e.g. add, remove and update) instructor and student accounts in the Instructor and Student modules respectively. With these two modules, the system administrator also can query and sort instructors and students by name, course, department and college, created and updated time.

In the web-based educational system, each test is categorized into one of the three modes as Proposed, Active or Retired. A

new test is designated as Proposed by default. Using the Test Module, the administrator can configure the conditions to change a test mode, such as the satisfaction score threshold to change a proposed test to Active, and the inactive period to change an Active test to Retired. The satisfaction score of a test comes from surveys and questionnaires from both instructors and students. In practice, a test mode alteration is determined by several conditions working together which use logic operators. The Test Module allows the administrator to manage the test database, such as deleting Retired tests and querying tests by author, major, created time, mode, number of used times and satisfaction score.

The System Module is for the administrator to maintain the system and session management, in order to create log files and generate system configuration backup.

Student session Management - It will maintain the student timings. The timings are allocated based on the course. It will check at the time of course the login users records are update the student attendance record.

Staff Session Management - Based on the course the staffs sessions are allocated the staffs can asynchronous upload the materials for course. If some synchronous are needed they can participate by Interaction system with student i.e. video conferencing and instant messaging.

Validator session Management – Each and every online test completion, the test material is forwards to Validator from the administrator and suggested deadline time to validate the test papers. Finally, the Export Module is used to generate customized reports based on requests, such as daily (weekly, monthly) system usage reports and error records. The Instructor Subsystem also consists of five modules: Login, Help, Query, Test Creation, and Report modules.

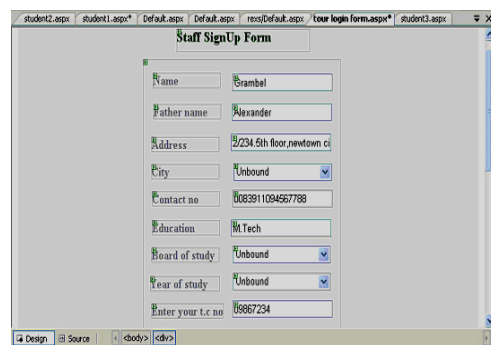


Figure 3: Instructor Sign Up

With the Instructor Login Module, an instructor can set up personal identification information including department, university, and contact information.

The Instructor Help Module provides two forms of guidance on system usage: animated demos and HTML files.

The Instructor Query Module allows an instructor to search questions by author, created time, Bloom's taxonomy level, course and topic, etc.

In the Test Creation Module, an instructor can create a test by either generating a set of questions or selecting existing questions from the system test database. Each question in the test database is categorized into one of the six levels of Bloom's taxonomy which can also be changed by other instructors. After a question creation, the Test Creation Module automatically recommends a Bloom's taxonomy level to the author based on

the question keywords. The author is allowed to set another level, and the question will be marked to future users. When other instructors scan this question, they can approve or disapprove the level set by the initial author. Therefore, each question also has a tag to indicate the percentage of approval or disapproval. The WBES system automatically notifies a targeted domain expert committee if the disapproval exceeds a threshold. The committee reviews the question to select the most appropriate level. During a test creation, an instructor can decide the total score, question weights, the test time, and the number of correctly or incorrectly answered questions before allowing students to go to the next higher level, etc.

The Instructor Report Module produces comprehensive reports of individual questions, tests, and student data as a response of the Instructor Query Module (e.g. question levels, test modes and student rankings). The Instructor Report Module can also generate reports comparing student performance in different questions and tests, which provides an efficient mechanism for teaching assessment. For example, if a majority of students failed a specific question or a test, their instructor can query the scores of students in other universities on this question or test. It would be helpful for the instructor to see if the question or test was too difficult for most students or the teaching methods were inadequate. With this information, the instructor could decide if s/he needed to improve the instructional materials or the teaching strategy. Finally, based on a test report, an instructor can submit a satisfaction score for the test using a predefined questionnaire/survey.

The Student Subsystem is comprised of nine modules: Login, Help, Query, Test Selection, Self-Assessment, Test Taking, Grading, and Report modules. Like the Instructor System, the Student Login Module allows a student to manage the account and the Student Help Module helps a student interact with the system using animated demos and static hypermedia files.

The Student Query Module allows a student to search previous tests by time, course and university. For a selected test, a student can view the questions and the test statistical data through the Student Report Module, including scores, the average and standard deviation of the test, ranking among all test takers, the correct answer percentage and the knowledge base for a question, etc. A student can select a requisite test from the Test Selection Module or an optional test from the Self-Assessment Module.

The Test Taking Module delivers the test after a test selection. If a requisite test is used for summative assessment, the test score is calculated toward student final grades. For a formative test, the score is used only by the Student Report Module to generate feedback about student performance on a test or individual questions. Through this feedback an instructor can monitor individual student learning outcomes and incorporate more materials and consultations on the topics that give students the most difficulty. The statistical data of all the required tests is saved in the system database for future analysis and reference. According to the grading policy created by the author, the Grading Module produces the score right after a test. After taking a test, a student can submit a satisfaction score through a questionnaire or a survey based on both subjective feelings (e.g. the sense of difficulty or ease as compared to other tests) and the objective test data (e.g. the statistical data related to the test and questions).

Dimension 2: Research Resources

Our investigation is to describe services that contribute and help researchers to collaborate, to work together and to exchange the information in their research activities. A service can be software (a program / application or a component), a document and so on that is provided by a researcher (the author) and can be used by other researchers to help them in their research activities. For this purpose, we must describe the context of the component precisely and we have identified three major fields: the research areas covered by the service, the researchers and the type of service (software, document) provided and so on.

A research area can be described by:

- **Indexes:** key words that describe the research area.
- **References:** a set of papers, documents and URLs that is useful in the research area.
- **Links:** which are towards other areas and topics which are linked to the research areas.
- **Description:** a short textual description of the research area.

In this section, we considered conference is one of the research resources and we present our findings in developing the services for conference management system (CMS).

Conference information relating to the conference is set by the administrator interface. It comprises of

- Conference Name – The name of the conference (e.g. **The International Conference on Computer Engineering and Technology 2009**)
- Conference Code Name – The abbreviation for the conference (e.g., ICCET 2009) Conference Date – The date(s) of the conference.
- Conference Location – The location of the conference (e.g., Reviewer Hotel, Singapore)
- Conference Hostname – The name of the sponsoring organization (e.g., Indian Pattern Recognition Society)
- Logos for Conference – An image file of the logo of the conference.

CMS is designed to perform the following tasks:

- Collection of expressions of interest and contact details
- Electronic submission of papers and author details
- Electronic bidding by reviewers for papers
- Collection of reviewer comments and recommendations
 - Emailing letters of acceptance and rejection
 - Allowing authors to submit revised papers after reading reviewers comments
 - Extracting the final papers to the web, CD-ROM, or print publication

4.1 Web Interface

The entire user Interface contains the login, signup and Functional pages.

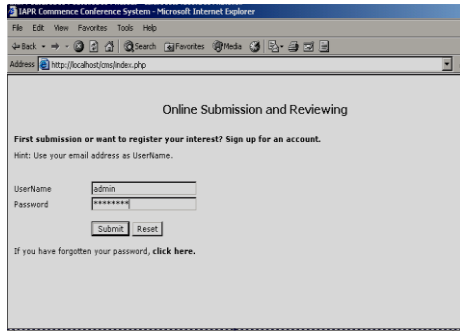


Figure 4: Online Submission and Reviewing

4.2 Paper Submission

Paper submission allows the registered users to submit their paper proposals. The users are able to submit and edit their papers during this phase.

Behavior of Paper Submission Phase:

- The user may submit a new paper or edit a submitted paper.
- The user may withdraw papers
- The reviewer accounts are not active in this phase although they can be set up.

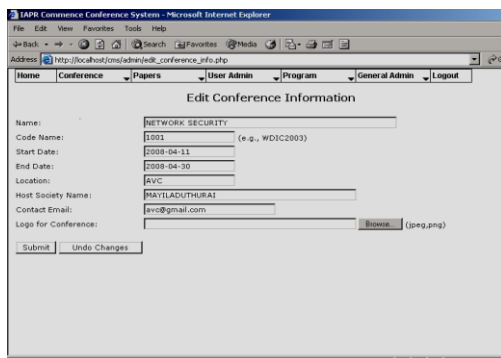


Figure 5: Submission and Reviewing

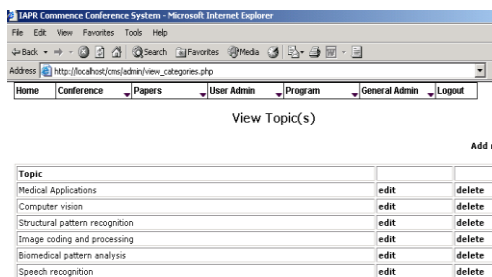


Figure 6: View Topic(s) of Interest

4.3 Final Paper Submission

Final paper submission allows the users to submit their final accepted papers for the conference.

- The users can see reviewer comments and the final decision on their papers.

- The users may submit a revised copy of their accepted papers for presentation at the conference and correct details such as paper title and author precedence.
- The reviewer accounts are inactive.

5. EXPERIMENT FINDINGS WITH PROGRAMME MANAGEMENT METHODOLOGY USING SERVICES

We analyzed the web services performance test for a sample of 10 users with its CPU load, network traffic and the band width. From the results, we found that the average click time for 10 users is 16ms. The time taken to initiate the URL with the help of protocol(s) is as shown in Figure 7.

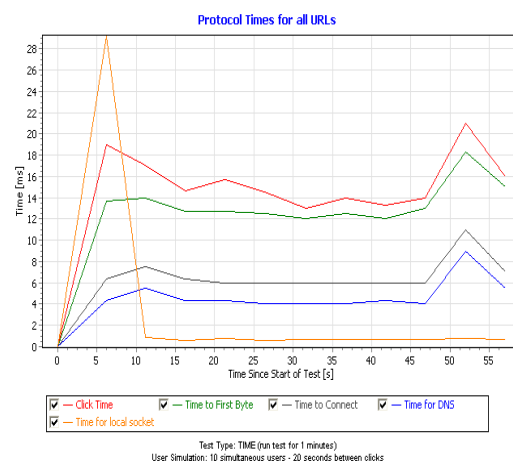


Figure 7: Protocol time for all URLs

The server and user bandwidth analysis is shown in Figure 8.

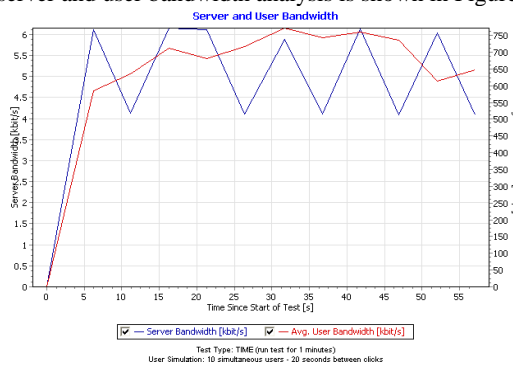


Figure 8: WebServer and User Bandwidth Analysis

This study is positioned as a preposition for a new program management methodology to be included in the curricula for educating information systems students on the evolving paradigm of SOA technology.

6. RELATED WORK

Several different approaches for describing and discovering services have been developed. For example, DAMLS[20] provides a set of characteristics that can be used as index or properties of a service. This description addresses only the static properties of a service. It does not describe the behavior of a service and its interface.

A variety of approaches [1,2,4,5,6,7,8,9,10,11,16,17] have been proposed for educational assessment in past decades. For example, learning-centered course portfolios [11] were employed to connect teaching assessment with learning assessment. A course portfolio is used not only to document student knowledge, but also to examine how different instructional practices improve student learning. This method enhances both teaching and learning in a complementary fashion. The researchers in [6-9] analyzed the relationships among assessment, teaching and learning. It was concluded that in addition to its constructive role in evaluating student learning outcomes and teaching quality, assessment also provides a powerful tool to improve teaching and learning effectiveness. In [4], Bergendahl and Tibell proposed a comprehensive assessment strategy which included input from written examinations, laboratory work, seminars, grant proposals and posters to enhance complex learning.

The assessment methods [3,4,10,12,13,14,15,17] based on Bloom's taxonomy theory have been used for concise analysis of cognition depth in educational assessment for many years. Oliver et al. [14] proposed a grading system to assign a Bloom Rating for a course test, which indicates the cognition difficulty of that specific test.

Many different approaches for service composition have been developed like [21], [22], [23] but the client must invoke all the service components of the composed service. In this context, the client must be familiar with the service composition.

Learning Blog (LBlog) [24] shares the experience and viewpoints of group members, and achieves learning goals together in the end. The LBlog also integrates Learning Management System (LMS) to manage the group members' profiles to evaluate the learners' learning efficiency.

The automatic composition of services [25] which is based on facilitator and it relies on the semantic description of service behavior. This includes the description of the interface through the inputs/outputs and conditions (post and pre-conditions) that control the operations.

7. CONCLUSION AND FUTURE WORK

In this paper, we have introduced an innovative way of creating services computing for e-learning and research resources to extend the education landscape from hardware, software and management of information system to the fast growing service area. The services are easily reusable, scalable and solution-oriented architecture. We have several technologies, research issues related to services and application servers that implement the backbone of an Enterprise Service Bus, which is of the critical importance to make the service oriented computing paradigm operational in a business perspective. These findings have to extend in further study of the methodology in the curricula with instructors and students and of the outcomes. Apart from this, our future work includes QoS aspects of SOA, Patterns of SOA. Such formulation provides a mature framework for designing service-oriented applications.

REFERENCES

[1] Lei He, Patricia Brandt, A Web-based Educational Assessment System", *ACMSE 2007*, March 23–24, 2007,127 -131.

- [2] Baker, E.L. and Mayer, R.E, "Computer-based assessment of problem solving. *Computers in Human Behavior*", 15 (1999), 269-282.
- [3] Baniulis, K and Reklaitis, V, "TestTool: web-based testing, assessment, learning." *International Journal of Informatics in Education*, 1 (2002), 17-30. 130
- [4] Bergendahl, C. and Tibell, L, "Boosting complex learning by strategic assessment and course design", *Journal of Chemical Education*, 82, 4 (April 2005), 645-651.
- [5] Vladan B. Deved'zic, "Key Issues in Next-Generation Web-Based Education", *IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews*, Vol. 33, No. 3, August 2003.
- [6] Bowden, J. and Marton, F: *The university of learning beyond quality and competence*. Kogan Page, London, 1998.
- [7] Brown, G., Bull, J., and Pendelbury, M, "Assessing student learning in higher education", Routledge, New York, 1997.
- [8] Brown, S. Institutional strategies for assessment. In Brown, S. and Glasner, A., editors, *Assessment Matters in Higher Education: Choosing and Using Diverse Approaches*, 3-13. St Edmundsbury Press Ltd, Suffolk, 1999.
- [9] Brown, S. and Knight, P. *Assessing learners in higher education*. Kogan Page, London, 1994.
- [10] Burgess, G.A. Introduction to programming: blooming in America. *Journal of Computing Sciences in Colleges*, 21, 1 (October 2005), 19-28.
- [11] Cerbin, W. The course portfolio as a tool for continuous improvement of teaching and learning. *Journal on Excellence in College Teaching*, 5, 1 (1994), 95-105.
- [12] Choren, R., Blois, M., and Fuks, H. Quest: an assessment tool for web-based learning. In *Proc. WebNet'98 – World Conference of the WWW, Internet and Intranet* (Orlando, FL, 1998).
- [13] Court, M.C., Tung, L., Shehab, R.L., Rhoads, T.R., and Ashford, T, "An adaptable learning environment that is centred on student learning and knowledge resolution", *World Transactions on Engineering and Technology Education*, 2, 1 (2003), 41-44.
- [14] Oliver, D., Dobebe, T., Greber, M., and Roberts, T. This course has a Bloom Rating of 3.9. In *Proc. ACM Sixth Conference on Australasian Computing Education*, 30 (Dunedin, New Zealand, 2004). 227-231.
- [15] Peat, M. Online assessment: the use of web based self assessment materials to support self directed learning. In Herrmann, A. and Kulski, M.M., editors, *Flexible Futures in Tertiary Teaching*, *Proc. 9th Annual Teaching Learning Forum* (Perth, Australia, February 2000). 2-4.
- [16] Roberts, G.H.B. and Verbyla, J.L.M. An online programming assessment tool. In *Proc. ACM Fifth Conference on Australasian Computing Education*, 20 (Adelaide, Australia, 2003). 69-75.
- [17] Scott, T. Bloom's taxonomy applied to testing in computer science classes. *Journal of Computing Sciences in Colleges*, 19, 1 (October 2003), 267-274.

- [18] Cheng Zheng, Weiming Shen, Qinghua Zheng, Feng Tian, “Design and Implementation of a Collaborative Conference Management System”, 2008, pp.5 – 10.
- [19] Zhang Qian, Liu Shi-jun, Meng Xiang-xu, “The Research and Implementation of Turning conference Management System into a Service”, IEEE Asia- Pacific Services Computing Conference, 2008, pp. 521 – 526.
- [20] <http://www.daml.org/services/>.
- [21] J. Yang, M. P. Papazoglou, and W. den Heuvel. Tackling the Challenges of Service Composition in E-Marketplaces. In the *12th International Workshop on Research Issues in Data Engineering : Engineering e-Commerce/e-Business Systems*. RIDE 2002.
- [22] M. Pistore et al.: Planning and monitoring web service composition. In *AIMS*, 2004.
- [23] D. Berardi et al: Automatic service composition based on behavioral description. In *IJCIS*, 2005.
- [24] Chun-Chia Wang,” The Development of Collaborative Learning Environment with Learning Blogs”, *Journal Of Software*, Vol. 4, No. 2, April 2009.
- [25] Oussama Kassem Zein and Yvon Kermarrec, “Description and composition of e-learning Services”, *Journal Of Software*, Vol. 2, No. 5, November 2007
- [26] Aliaksei Yanchuk, Alexander Ivanyukovich, Maurizio Marchese, “Towards a Mathematical Foundation for Service-Oriented Applications Design,” *Journal Of Software*, Vol. 1, No. 1, July 2006”
- [27] A. Malinowski and B. Wilamowski, “Paper Collection and Evaluation through the Internet”, *Proceedings of the 27th Annual Conference of the IEEE Industrial Electronics Society*, Denver, CO, Nov. 29 - Dec. 2, 2001, pp. 1868-1873.
- [28] Philippe Rigaux, "An Iterative Rating Method: Application to Web-based Conference Management", *Proceedings of the 2004 ACM symposium on Applied computing*, pp.1682-1687.
- [29] Mike P. Papazoglou · Willem-Jan van den Heuvel, “Service oriented architectures: approaches, technologies and research issues”, *The VLDB Journal* (2007) 16:389–415.
- [30] Booth, D., et al.: Web Service Architecture. <http://www.w3.org/tr/ws-arch/>, W3C, Working Notes, 2003/2004
- [31] Box, D., et al.: Simple Object Access Protocol (SOAP), Version 1.1. W3C Note, W3C, May 2000. <http://www.w3.org/TR/SOAP/>.
- [32] Bui, T., Gachet, A.: Web services for negotiation and bargaining in electronic markets: Design requirements and implementation framework. In: *Proceedings of the 38th Hawaii International Conference on System Sciences*, IEEE, 2005.
- [33] Burbeck, S.: The tao of e-business services: the evolution of Web applications into service-oriented components with Web services. IBM DeveloperWorks, 2000. <http://www-106.ibm.com/developerworks/WebServices/library/ws-tao/>
- [34] Krafzig, D., Banke, K., Slama, D.: *Enterprise SOA: Service Oriented Architecture Best Practices*. Prentice-Hall, Englewood Cliffs (2005).
- [35] Yang, J.: Web Service Componentization. *Commun. ACM* **46**(10), 35–40 (2003).
- [36] Dhesiaseelan, A., Ragunathan, V.: Web Services Container Reference Architecture (WSCRA). In: *Proceedings of the International Conference on Web Services*, IEEE, pp. 806–805, 2004.
- [37] Pradeep Gurunathan Dr. Seethalakshmi Pandian,” A New Tool for Web-based Educational System”, *Fourth International Conference on Natural Computation*, 978-0-7695-3304-9/08 \$25.00 © 2008 IEEE, DOI 112.
- [38] Pradeep Gurunathan Dr. Seethalakshmi Pandian, “A Novel Approach for Web-based Conference Management System”, *2009 International Conference on Computer Engineering and Technology*, 978-0-7695-3521-0/09 \$25.00 © 2009 IEEE DOI 10.1109/ICCET.2009.118

AUTHOR(S) BIOGRAPHY

1. **Pradeep Gurunathan** received the Master Degree in 1998 and obtained the Master of Technology in Information Technology in Manonmaniam Sundaranar University in 2004 and pursuing his Ph.D in Information and Communication Engineering in Anna University of Technology Tiruchirappalli, India. Currently he is working as a Professor in A.V.C College of Engineering, Mayiladuthurai, India.

2. **Dr. Seethalakshmi Pandian** received the Doctorate in Computer Science in Anna University, Chennai and currently she is working as a Director, Centre for Academic Excellence, Anna University of Technology, Tiruchirappalli, India.