An Extended E-Learning System Architecture: Integrating Software Tools within the E-Learning Portal

Zuhoor Al Khanjari, Swamy Kutti, and Muna Hatem
Department of Computer Science, Sultan Qaboos University, Sultanate of Oman

Abstract: *IT-based Higher education is found to be a feasible and economical solution in improving the traditional education model [2, 14]. Consequently the e-learning portals have been evolved as experimental tools in implementing online courses with the existing classroom education [3, 6, 13]. For an on-line course to be successful, it is essential to provide a necessary resource environment available in on-line. This expects designers to implement their courses standing alone without a third party dependency. An experimental study has been carried to verify the feasibility of implementing a system architecture that can support course-related resources in stand-alone mode. This paper explains the development of a prototype system architecture which results by extending the services of a portal by integrating software tools used in the related courses. The resulting architecture is simple, distributed and cost effective. The supporting experimental work has been carried out on the test bed around the e-learning portal WebCT that is currently available in Sultan Qaboos University (SQU).*

Keywords: *Workgroup learning, web-server, MyCoursesServer, application server, Unix network server, WebCT.*

Received June 13, 2004; accepted February 25, 2005

1. Introduction

Information Technology (IT) in relation to e-Education represents a set of facilities such as computer communication network (e. g., Internet), information systems (e. g., library database), WWW network, course-related software tools (e. g., compilers) and educational software tools (e. g., WebCT). The on-line education concept emerged as a result of the influence of IT on the current education particularly in distance mode. Consequently several e-learning portals have been prototyped for exploring the feasibility of developing an effective e-learning in the place of traditional education [7]. The success of on-line education depends upon the availability of resources on-line to support the functioning of courses. Most of the e-learning portals come with basic facilities allowing the course design based on the requirements of the traditional education. However, on-line education suffers some limitations due to some technical factors with portals as well as resource distribution strategies. For instance:

1. *Constrained access to system resources*: A difficulty is realized in linking software resources that are usually under the control of a central agency in educational institutions.
2. *Less powerful*: Portal architecture may not be able to support all types of tools.
3. *Need of system programming knowledge*: Instructors come from various disciplines. None of them is expected to possess either technical background or knowledge of system programming. While taking the role of course designers their expertise allows only designing the course contents, but not setting up required system architecture.

The solution to this problem is to provide a proper and complete on-line course environment readily available to the course designers. This should integrate all scattered elements (i. e., instructional materials, software tools and information systems) so that the portals provide the required resources in on-line along with each course access. The on-line education can then prove to be more efficient and cost-effective.

This paper explains a novel approach in extending the facilities provided by a portal to enable on-line access of course-related resources. This extended architecture allows the instructors to add a range of software tools including any customized ones as a part of their on-line courses. Section 2 of this paper gives a conceptual view over the features of e-learning portals. This section also analyzes the features of the WebCT portal that has been used in the experimental studies explained in this paper. Section 3 gives an overview of the existing e-learning architecture in SQU. Section 4 discusses various types of course tools and the limitations of supporting them directly in the existing e-learning Model. Section 5 describes the extended system architecture for plugging in different types of software tools with the existing e-learning portal WebCT. Section 6 explains the experimentations on

various configurations of the extended portal architecture. Finally, the section on conclusion discusses the advantages of the extended architecture and the issues with the implementation of this architecture. Further work in this line is also discussed.

2. Concept of E-Learning Portal

Portal as a terminology is used to mean a place to go. Concentrating on a specific theme or interest, portal can bring together people, services and products. Therefore, portals can be for Business, organization, Education, etc [12]. An e-learning portal represents an education-based service point providing various services related to students' on-line education. Its role can also be interpreted as an interface that enables the learners to locate course contents and perform a mixture of other course-related functions. The e-learning systems/portals have been developed to serve both virtual and traditional educational institutions [8].

A portal comes with a course template along with a control panel (or toolset) to allow users to design their on-line courses. When a course is instantiated, it comes with a starting home page with an access point to the control panel of the portal. The control panel provides necessary functions to design a complete course. It allows the users to add one or more service pages to the home page of a course. The service page types supported by a portal may include organizer page, tool page, simple contents page, URL page, etc. An organizer page similar to a directory in a file system can hold one or more pages of the available types. The home page of a created course is itself an organizer Page. By adding organizer Pages in a recursive manner, one can end up with a tree-structured course. A leaf-page can be a tool page, contents page or an URL page. The course template includes a file system to hold the data files (lecture notes, icons, images, etc.) that are linked to leaf-pages. The course template also includes the ownership label to grant design privilege to the nominated users.

The portals also come with a student database to manage student grades and progressive performances. Figure 1 shows a general overview of an e-learning portal. The portal has two distinct views: student view and designer's view. Instructors use the designer's view to design their courses and/ or from time to time update the designed course. The instructors can also control the students' access to specific course pages by using the designer's view. Students who are registered in a course are granted to access the students' view to read instructions, course materials and grades. They can also make submissions/resubmissions of their homework and communicate with their instructor as well as with their co-students. The portals allow both the students and designers to have their views simultaneously.

There are several e-learning portals that are used at present to support either a partially or completely the on-line education. Among them Blackboard [4, 5] and WebCT [9] come with a rich set of features that are suitable for implementing a standard e-learning model. These two have gained the interest of many researchers particularly in their prototyping experiments [16]. The experimental test bed explained in this paper is around the WebCT portal managed by the Center for Educational Technology (CET) at SQU.

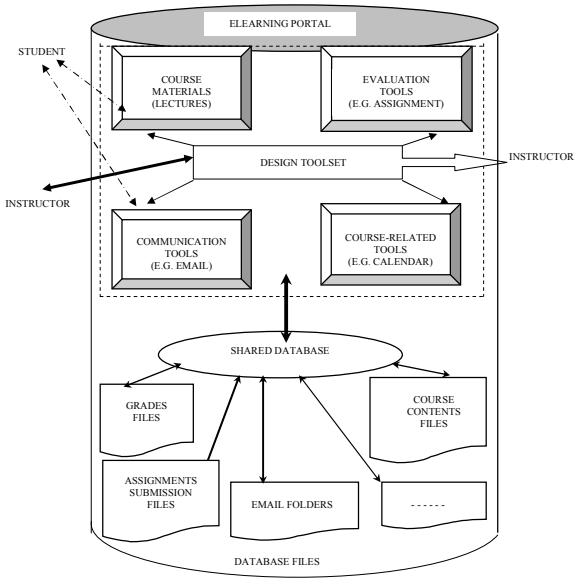

Figure 1. General overview of e-learning portal.

3. Existing E-Learning System Architecture

WebCT has been launched on a fiber-optics backbone network of one gigabit per second speed spanning from servers via switches to desktop networks of seven colleges and 100 megabit per second within each desktop network. There is a scope to upgrade the desktop network up to 1 gigabit per second. There are several servers including one for each college and a separate server for WebCT. Basically every college server serves its lab PCs as well as personal PCs within the college [15].

The Centre for Information Systems (CIS) at Sultan Qaboos University is responsible for maintaining the university network along with all academic-support-software resources. CET, a user of the CIS facility has been trusted with the responsibility of providing necessary university-wide educational services including the e-learning facility via the WebCT portal. Figure 2 explains the current network configuration of the current e-learning model in SQU.

The e-learning model in SQU is at its initial stage, and is allowing the community to use the basic facilities of WebCT. These basic facilities include only the course-related information on-line, but not the course-related software tools. Students use WebCT currently for reading the course-related information such as course outlines, lecture notes, tutorials,

assignments, etc., and submitting simply their homework.

WebCT provides two basic mechanisms to plug in software tools. They allow software tools plugged in through linking mechanisms provided in single and URL pages. The single page's linking mechanism can only allow to plug-in some simple software tools residing within a WebCT course, whereas the URL page allows us to link software tools residing over the network outside WebCT.

Currently, the configuration with the centralized resource distribution in SQU does not allow the URL link facility effectively to route the access of software tools via WebCT. Therefore, students access their software tools either from the centrally controlled network resources of CIS or from their local host systems. That means WebCT cannot be used for testing all types of students' programming tasks. In fact, this is a major obstacle in realizing an effective on-line education. This has obviously initiated an investigation to study the scope of existing facilities in WebCT to solve the identified problem.

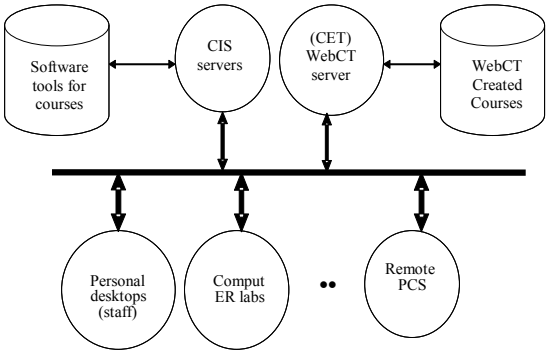

Figure 2. Existing e-learning system architecture.

4. Limitations of Existing E-Learning System Architecture

Software tools come with different sizes and configurations. For example, a computer simulator program (e. g., IBM-PC emulator) is small and single module program. On the other hand, compliers like Ada may come with a complete set of libraries requiring a specific file-residence environment. Similarly a database package needs not only a complex working environment but also facilities to work with various interfacing languages. An extreme case includes the software tools that can work efficiently in their own programming environment such as system software tools of Unix system. In general, the software tools are classified into the following categories in order to integrate them with suitable e-learning system configurations:

1. Stand-alone software tools e. g., simulators, assemblers, etc.
2. Large software tools e. g., Ada Language System, Mathematica, Matlab, SPSS, C++ compiler, movies, science demos, engineering packages, digital library, databases, etc.
3. Special Software Tools e. g., virtual labs science, multimedia software packages such as photoshop, movies, etc.
4. Software tools working in different system environments e. g., Unix software tools, MS-Office in Windows, etc.

The e-learning system explained in Figure 2 has several limitations in supporting all these types of software tools:

- First the e-learning model does not allow the instructors to add commonly needed used software tools such as language compilers.
- Secondly, some of the special tools such as virtual labs [10] and multimedia packages [1] need to be plugged in from where they are developed.
- Thirdly, plugging in a complete foreign host environment such as Unix or Unix network is somewhat complex task for any common course designer.
- Finally, there is no direct mechanism with WebCT available for researchers to access third party tools that are usually distributed in the remaining part of the world. All these limitations are due to the closed nature of the existing e-learning model.

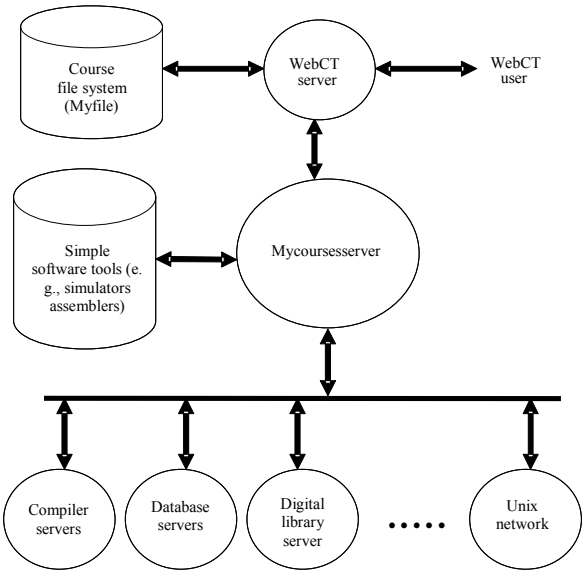

Figure 3. A logical view of the extended e-learning system architecture.

5. Extended E-Learning System Architecture

A logical view of the extended e-learning model is shown in Figure 3. This configuration provides three levels of servers i. e., WebCT server, MyCoursesServer and Application server (A-server) to support course-related software tools. The WebCT server can be used to host simple stand alone software tools. The course tools are uploaded into the MyFiles

folder of the course file system and used in a direct manner. The software tools that cannot be plugged in to WebCT can be plugged into a separate server called MyCoursesServer. The course tools are uploaded in the file system of the MyCoursesServer and used indirectly via WebCT. A powerful configuration can be realized to add one or more A-servers (i. e., application servers) to MyCoursesServer. MyCoursesServer can now play a double role. It can host some of the course tools and at the same time it can provide an interface between the WebCT server and A-servers. A-servers are useful particularly to home course tools requiring their own working environment. At the same time they can be used to share the load of MyCourseServer. For example, one A-server can be assigned to Database and another A-server to compilers. A-servers can also be configured to follow a distributed framework for achieving an improved performance and better reliability [11].

The working principle of this e-learning System model is as follows. A link is made between WebCT course and MyCoursesServer. Using this link a course page can access the services of MyCoursesServer. MyCoursesServer in turn maintains links to a set of A-servers and any available network server. The users' particularly students' can only view the software tools that are available in the system. That is, they can see only the WebCT server through their course access. The designers on the other hand have complete responsibility in managing and maintaining the server configuration as well as distributing the course tools. They can create new servers and add software tools to those servers. They can also remove servers or replace some with other servers as the situation arises. All these functions over the server network can be performed without disturbing the normal function of the WebCT server in a transparent manner.

In a fully distributed configuration there is one MyCoursesServer for each instructor. This is linked to a set of A-servers assigned to the courses managed by that instructor. MyCoursesServer is basically acting as a distributed resource scheduler controlling the tools availability via WebCT. It schedules a suitable A-server based on several factors including the load balancing among the available A-servers, real-time response required, etc. In its simple version it plays the role of routing the resource requests to appropriate A-servers.

MyCourseServer when invoked through a course page starts interacting with the user in an interpretive manner. Users can either request the server about various services currently maintained by the server or send a request for opening a software tool.

6. Experimentation on Various Configuration of Plugging Tools

A simple method is to use the single page tool to plug in a software tool in a WebCT course. The software is first uploaded in the MyFiles folder of the courses' file system. Then add a single page to a designated organizer page and make a link to the uploaded file. In this way the designers can add any number of simple software tools to their courses. This will cause a crowded organizer page. One way to avoid this situation is to create a directory on a separate Single Page and a directory document with a list of URL-tagged software tool identifiers that can be attached to it. Each URL-tagged identifier is linked to its corresponding uploaded file in the course file system. In this way the directory document page can accommodate any number of tool identifiers.

Another simple method to plug in software tools is a simple extension of the previous method. In this method the software tool is assigned to a URL-page. Note that a Single Page that was used in the previous method locates the software tool by using the file-path information. But an URL page locates the software tool by using the URL address. An URL page can be created in an organizer page of a course and it can be edited with the URL link address of the tool stored in the courses' MyFiles folder as shown in Figure 4. The figure explains how the macro assembler, MASM is accessed by a student user. The MASM assembler must be uploaded first in the courses' MyFiles folder before creating a URL page for the course tool.

The standalone tools chosen for testing this configuration are:

1. MASM: Macro Assembler.
2. Concurrent Language Interpreter (CLI) used in Real-Time programming course.
3. Unix's vi editor.

All these (object) programs are uploaded to the courses' MyFiles folder in WebCT. An organizer page for tools is created under the courses' Home Page. Within this tool organizer page URL pages are created and linked to the uploaded tools residing now in MyFiles folder. Students view allows users to invoke any one of these program tools for use. Figure 4 shows the MS-DOS prompt window and a history of commands while assembling a program. A sequence of actions starting from the action on the WebCT organizer page shows how the tool is finally brought to the user's screen.

In the third configuration a satellite server named MyCoursesServer is linked to the WebCT portal. This server is dedicated to home all types of software. It is basically a logical representation of a group of servers that includes language servers, database servers, network servers, digital library servers, etc. The WebCT's URL page is used to link MyCoursesServer.

MyCoursesServer is acting as an interface providing various functions over the management of software tools distributed among A-servers. A simple version of this software is implemented in a script form that is found to be flexible enough for prototyping the e-learning model.

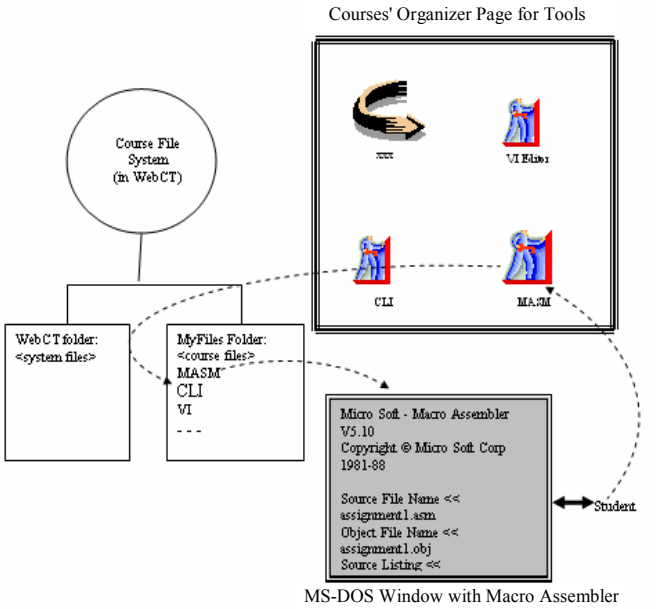

Figure 4. URL page and its reference to a local executable file (e. g., macro assembler).

Figure 5 shows the extended architecture with the satellite server, MyCoursesServer. Here, the satellite server provides the role of interfacing element between the WebCT server and all application servers as well as the role of an application server. The link, MyCourse Tools on WebCT's organizer page can only be seen on the screen. For a user the system is quite transparent except that by activating this link MyCoursesServer comes into action either in an interactive mode or simply responds by opening a menu page displaying a list of course-related software tools and other related services on the screen. The users have to simply activate a tool from the menu screen. MyCoursesServer immediately responds by opening the requested tool on the user's screen. A sequence of actions (1 to 6 and a to c) in Figure 5 explains how a course tool residing in an application server (in this case MyCoursesServer) is activated by a user.

Figure 5 also shows another sequence of actions (a1 to c1) invoked in bringing the Unix system to the user's screen. The Unix Network Server can display on the screen the Unix nodes that are currently available on the network. The users can activate a node to bring the Unix prompt on the screen.

7. Conclusions and Future Prospects

Experiments on software-tools integration have shown several configurations for plugging in the tools with the portal. Classification of tools is necessary to identify those tools that can directly be integrated with the Portal and other tools that can be indirectly integrated through local servers. Depending upon the type of tool and its usage one can select one or more configurations. An important observation is that all course-related software tools can be effectively integrated around a portal to bring in the entire resource environment for that course. This ultimate architecture gives a dynamic and reliable configuration to support course related programming tools or any virtual lab in a feasible manner. At the same time the implementation of this architecture can be made efficient and cost-effective depending upon the number of servers that the designers wish to add to the system.

Another significant advantage of this architecture is to avoid uploading the course tools into WebCT. First it saves uploading time and secondly it removes any constraint on the usage of the file space in WebCT. The extended architecture simply allows the designers to link the nodes that were used to develop the course tools. With the availability of several internet-based programming tools the tested configurations can be implemented with WebCT portal as a working e-learning System model.

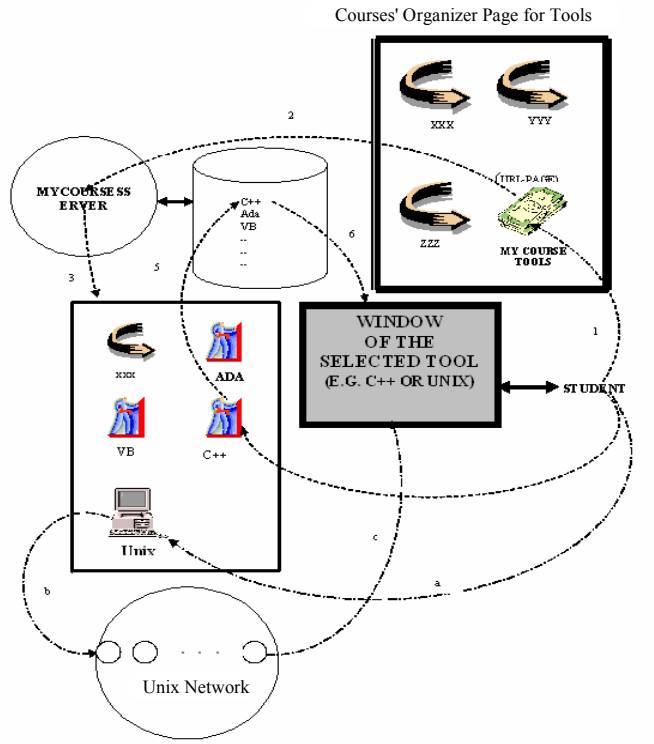

Figure 5. Extended model with MyCoursesServer as tool server.

The experiments in this phase of the project have shown only the feasibility of integrating software-tools with the portal. Real-time and reliability aspects of the integrated e-learning environment needs further experiments. The server-based configuration that has been evolved out of this experiment has a scope to incorporate the distributed approach. Our next phase of this project is progressing in integrating software tools using the distributed local servers around the portal.

Future work on this experiment has unlimited scope. This would include testing the use of web-based course tools, implementing a digital library server and integrating the software labs such as super computing facility, parallel processing labs, etc with this architecture.

References

[1] Al-Ahmed H. and All Al-Qayedi "lnnovatve Methods for Delivering Image Processing Courses," *in Proceedings of Millennium Dawn in Training and Continuing Education*, Bahrain, pp. 245-259, April 2001.

[2] Aldhafeeri F., "Analysis of the Impact of eLearning Technologies on Training and Learning Process," *in Proceedings of the Educational Technology Symposium (ETEX'2001)*, Sultan Qaboos University, Sultanate of Oman, September 2001.

[3] Bell M., Bush D., Nicholson P., O'Brian D., and Tran T., "Universities Online: A Survey of Online Educationand Services in Australia," *Canberra Occasional Paper Series*, Higher Education Group, Commonwealth Department of Education, Australia, Available at: http://www.dest.govt.au/highered/occpaper/02a/02_a.pdf, 2002.

[4] Blackboard Inc., *Blackboard Learning System*, http://www.blackboard.com/highered/academic/ls/index.htm, 2004.

[5] Blackboard Inc., *Educational Benefits of Online Learning*, http://www.blackboard.com, 2000.

[6] Burford S. and Cooper L., "Online Development Using WebCT: A Faculty Managed Process for Quality" *Australian Journal Educational Technology*, vol. 16, no. 3, 2000.

[7] Embong A. and Ismail I., "Transformation of Higher Education Through Information Technology," *in Proceedings of the Millennium Dawn in Training and Continuing Education*, Bahrain, April 2001.

[8] Fiaidhi J. A., Mohammed S. M., and Al-Khanjari Z. A., "Designing an On-Campus Intelligent Learning Portal, *Journal of Science and Technology (JST)*," University of Science and Technology, Yemen Republic, vol. 6, no. 1, pp. 9-18, 2001.

[9] Goldberg M. W. and Salari S. "An Update on WebCT (World Wide Web Course Tools): A Tool for the Creation of Sophisticated Web-Based Learning Environments," *in Proceedings of NAUWeb'97*, Available at: http://star.ucc.nau.edu/~nauweb97/papers/goldberg/goldberg.html, Arizona, June 1997.

[10] Raman S. U., Tukur N. M., and Kan A., "A Virtual Fluid Mechanics Laboratory for Chemical Engineering Students," *in Proceedings of Millennium Dawn in Training and Continuing Education*, Bahrain, pp. 245-259, April 2001.

[11] Razz M. R. and Mahdian K., "An Advanced System Architecture of a Distance Learning System for Collaborative Education," *in Proceedings of Millennium Dawn in Training and Continuing Education*, Bahrain, pp. 149-156, April 2001.

[12] Schelin E., *A Revealing Look at Learning Portals*, eLearning eZine, Available at: http://www.e-learning.com, February 2001.

[13] Shannon S. and Doube L., "Valuing and Using Web Supported Teaching: A Staff Development Role in Closing the Gaps," *Australian Journal of Educational Technology*, vol. 20, no. 1, pp. 114-136, 2004.

[14] Sheikh S. I. and Siddiqui J., "Internet/Intranet Based Higher Education," *in Proceedings of the 4th Workshop on Information and Computer Science (WICS'2002)*, KFUPM, SA, March 2002.

[15] SQU, An Interview with the Head of Network Section in Center for Information, Sultan Qaboos University, Sultanate of Oman, 2004.

[16] WWW-Source, "E-learning, the Internet is a Powerful Environment for Teaching and Learning," Available at: http://dir.yahoo.com /Business and Economy/Business to Business /Training and Development, May 2002.

Zuhoor Al-Khanjari is the assistant dean for Postgraduate Studies and Research, College of Science, Sultan Qaboos University. She is an assistant professor in software engineering, Department of Computer Science at Sultan Qaboos University, Sultanate of Oman. She received her BSc in mathematics and computing from Sultan Qaboos University, Sultanate of Oman, MSc and PhD in computer science from the University of Liverpool, UK. Her research interests include software engineering, database management, e-learning, human-computer interaction, intelligent search engines, and web data mining and development. Currently, she is the coordinator of the software engineering group in the Department of Computer Science, Sultan Qaboos University, Sultanate of Oman. Also she is coordinating elearning facilities in the same department. She is a member in the editorial board of the International Arab Journal of Information Technology (IAJIT) and a member in the steering committee of the International Arab Conference on Information Technology (ACIT).

Swamy Kutti is currently working as an associate professor in the Department of Computer Science, College of Science, Sultan Qaboos University. He received his BE in Electronics from University of Madras, India, ME in communications from Indian Institute of Science, India, MSc in computer science from Monash University, Australia and PhD in computer science from Deakin University, Australia. His research interests include computer languages, operating systems, real-time systems and e-learning.

Muna Hatem received her MSc degree in computer science from the University of Aston in Birmingham, UK in 1982. She worked as a programmer, system analyst, project leader and computer department manger in different governmental organizations. She was a lecturer in the Faculty of Engineering in Amman University for 5 years. She joined Sultan Qaboos University, Department of Computer Science in January 2002. Her main research interests are internet programming and database systems. Currently, she is doing PhD research in computer science at the University of Bradford, UK.