Extending the Functionality of LMS to Support Computer Science Education Using Plug-in Tools

Zuhoor A. Al-Khanjari and Yusra M. Al-Roshdi
Department of Computer Science, College of Science, Sultan Qaboos University
Muscat, Oman
Email: zuhoor@squ.edu.om and m095653@student.squ.edu.om

Abstract—**E-Learning is the environment that supports the education field and provides the knowledge of a specific area in one place. Computer Science Education (CSE) is one area that uses the aspects of e-Learning to educate the learners and to distribute the knowledge to them. Several Learning Management Systems (LMSs) support many software tools to manage the theoretical materials of the CSE. However, the practical software tools that are needed by the Computer Science courses are ignored or simply not considered in most of the LMSs. To overcome the limitations of the LMSs towards the practical aspects of the CSE, this paper shows how to prepare the tools and make them available for use anytime through the LMS. For this to work, the authors of this paper take a step forward in discussing how to extend the LMSs functionality in general and Moodle as a special case. This extension involves adding the needed software tool as a component to the architectural structure of Moodle. Moreover, this paper introduces some literature that supports and implements these kinds of extensions.**

Index Terms—**E-Learning, Computer Science Education, Moodle, Software Tools, VPL**

I. Introduction

E-Learning makes use of the computer software to support students during their learning process to understand new concepts [1]. It gives new resources to the teachers to teach their courses with availability of different tools in LMS [2]. Also, It can offer a great deal of flexibility in when, where, and how educational resources are distributed [3]. Nowadays, many LMSs make their source code available for modification under the various displays of open source licenses and for customization to fit the users' needs [4]. In this research paper, the Moodle will be used since it is open source e-Learning software and used in Sultan Qaboos University (SQU) in the Sultanate of Oman.

Many researchers are trying to generally optimize and expand the functionality of the LMS in the field of education and specifically in the Computer Science Education. CSE needs extra effort to prepare practical sessions for the learners in e-Learning environments. Therefore, the needed software tools should be made available in LMS to be used as a support of, or as a complement to the traditional approaches. This area needs more graphical application tools that could encourage learning by sharing knowledge and resources. For this purpose, the needed software tools could be built and plugged-in within the LMS system as a new extended component [5, 6, 7, 8].

This paper explains how to improve and extend the functionality of LMS (e.g. Moodle) to support CSE according to some literature that has been done in this area. Also, it provides a way to develop software tools for CSE and integrate them into Moodle. This paper also provides a solution to overcome the limitations regarding the software tools that are needed for the practical sessions of online CSE. Section II, presents some related work of tool integration into LMS. Section III, introduces an overview of the Learning Management Systems (LMS) in general and Moodle as a special case. Section IV, shows the requirements of the software tools in the Computer Science Education and the limitations of those tools in the LMS. Section V, presents the proposed solution to that problem regarding the limitation of practical software tools of CSE in the LMS. Section VI, provides conclusion and the future work.

II. Literature Review

Many universities and educational institutions need to improve their Learning Management Systems to include practices from educational standards and provide a flexible integrated environment [5]. There are several ways to implement and support CSE by software tools and integrate them into the LMS. This idea was presented in 2005 by Doberkat and colleagues [1]. They introduced the concept “e-Learning Software Engineering” which is the novel e-Learning concepts techniques and tools to support education in the Software Engineering. It is used to support the practical tools available in the e-Learning systems for the Software Engineering course by proposing e-Learning portal called “MuSoft”. In this portal, they used video techniques and animation and developed tools for Unified Modelling Language (UML). Also, they integrated other tools in this portal.

In 2008, Corbera and colleagues [9] described a new Moodle Module CTPractical which was developed to support the practical content of a basic computer organization course. This Module carried a wide range of activity and assessment in computer courses. Also, it provides a feedback for student work and it works as a material repository.

doi:10.4304/jetwi.6.2.220-225

In 2009, Rößling and Vellaramkalayil [10] proposed integration approaches of a hypertext book as a Moodle Module for algorithm animations into the learning materials. In the same year, Rößling and Kothe [6] improved previous tools to be integrated into Moodle. These tools were prepared using AJAX technology and used as activity plugins and as resource tools in Moodle. Furthermore, Prieto-Blazquez and colleagues [11] proposed an integrated structure for virtual laboratories of programming in Computer Engineering and Software Engineering. They discussed the issues regarding the use of that laboratory in teaching and practical work.

In 2010, Riabov and Higgs [12] designed a lab to teach various Computer Science courses by using state-of the art free licenses software tools for conducting online virtual labs. The labs were designed to help students explore modern sophisticated techniques of computer system analysis and design, programming, networks, simulation and modelling, image processing, multimedia, web development and database. All the online courses use the lab-based project to support learning of students. In the same year, Bochicchio and Longo [13] proposed an integration approach between Moodle and MicroNet. MicroNet is a collaborate web-enabled electron microscope that can be remotely operated by small group of users between 2 to 20. They added the MicroNet as new plugins by rebuilding and redesigning it to adapt it with Moodle's style and techniques.

In 2011, Huan and colleagues [3] proposed practices towards teaching online CS courses such as Programming Compiler. A real time video on this compiler discusses how to write, compile and correct the code of C++. Huan and colleagues mentioned the commonly used formats to deliver the content of courses such as Microsoft PowerPoint (.ppt), Microsoft Word (.docx) and Adobe Portable File Format (.pdf). These researchers also introduced some course tools which include built-in menu that lists all available resources in Blackboard. Also, they introduced a good courseware like myitlab (http://www.myitlab.com). In the same year, Fest [14] proposed integration of architecture of the Dynamic Geometry Software (DGS) that was developed by Java applet then integrated as a component into interactive learning activities.

In 2012, Lavrishcheva and colleagues [7] researched how to support the Software Engineering with software tools. They developed a web site that contains different types of tools such as programming language compiler, Integrated Development Environment (IDE) such as (Eclipse, MS.Net, ..etc.) and others. They stated that the website could support the learning process of Software Engineering and make the tools available to public use.

In 2013, Palumbo and Verga [5] proposed integration approaches to integrate Moodle with Office suite to provide a multi-purpose and a flexible functionality of the OpenOffice.org such as: calculation, plots and diagrams by means of spread sheets or graphical application. They selected Moodle and OpenOffice.org to integrate them because of the simple and effective manner of coding style. They used Java and AJAX with XML to create a java applet which was embedded in the web page of new Moodle modules. By using these tools, students could save their work on one or more temporary files on the Moodle server. Also, they could export or import their work from a local system, edit and view their files and submit their assignments.

In addition, Ramos and colleagues [8] proposed architecture of Matlab assessment which depended on the work done by Corbera and colleagues [9]. They explained how to extend the functionality of Moodle to facilitate the assessment of programming languages of Matlab. They introduced a novel extension from the CTPractical [9]. They rebuilt a new module in Moodle as a component of it which used the activity module and control block to support the new module features. Basically, the students could submit their work either by (.m) or (.zip) files and find information from their lab work after evaluating it by the tester. As a result, the automatic verification process could provide information on the specification, deadline, feedback and results of the assignments.

III. Overview of the Learning Management System (LMS)

Learning Management Systems (LMSs) are software systems, which manage virtual learning environments (VLEs) to be used by learners and instructors to support online education. They provide a suite of tools which support the creation, maintenance and the delivery of the online courses [4]. Many LMSs are commonly implemented as web applications following the Client/Server approach [15].

There are two main categories of LMS: 1) Open source initiatives such as Moodle, A tutor, Sakai, etc. and 2) Commercial or property solution platforms such as: Blackboard, WebCT, Desire2learn, SharePoint, etc. Open source LMSs are typically built on extendable frameworks allowing developers to adjust and modify the LMS to suit their specific needs.

A. Moodle

Moodle is currently the most successful LMS as an open source. It is a Learning Management System that stands for Modular Object-Oriented Dynamic Learning Environment. It is open source system under General Public Licenses (GNU). The features of being open source will facilitate the development and maintenance of it according to the customers' needs. Also, this provides a research area to extend the software tools of CSE [16].

Moodle is a modular system which organizes the content as units that refer to the courses and as sections which contain the activities and resources of the course materials [5]. There are hundreds of general plugins in Moodle that are developed by many developers to improve the functionality of Moodle. These tools include: forum, chat, quiz tools, assignment tools, wiki and much more. Nevertheless, Moodle's administrator can install a wide range of additional plugins to extend the set of activities of Moodle installation. Regarding the technical side, Moodle is developed in PHP: Hypertext Preprocessor programming language and as most LMSs, it uses the common three tier client/server architecture [16].

IV. The Software Tools in Computer Science Education

CSE is related to a collection of scientific disciplines concerned with the education of computational and technology knowledge and its applications. Computer Science Education involves two main parts for teaching its courses as follows:

1) *Theoretical part:* is concerned with the theory content and knowledge that are needed for the courses. To organize the theoretical content, many LMSs provide a set of software tools such as:
   - Office suite (e.g. Presentations, e-books, worksheets, PDF document, Word, etc.),
   - Multimedia (video, audio),
   - Wikis and others.
2) *Practical part:* is concerned with the knowledge and practice that the learners gain through practical sessions in the lab using the computer applications. Many Computer Science courses need different software tools which might be classified into the following categories in order to integrate them with suitable e-Learning system configurations [17]:
   - Stand-alone Software Tools e.g. simulators, assemblers, etc.
   - Large Software Tools e.g. Ada Llanguage System, Mathematica, SPSS, Programing languages Compiler, Engineering Packages, and Digital Library.
   - Special Software Tools e.g. Virtual Labs, Science Demos, Multimedia Software Packages (Such as Matlab and Photoshop), Movies, UML CASE tools
   - Software Tools working in different system environment e.g. UNIX software tools, MS-Office in Windows.

The limited availability of these software tools in the e-Learning application produce a big gap in the e-Learning against the traditional learning. These limitations could be solved by developing and making these tools as a part of the LMSs, so students could use them anytime and anywhere. This solution could provide a way of making the e-Learning more powerful and useful especially in Computer Science Education.

V. Proposed Solution to Extend Software Tools in e-Learning Applications

To overcome those limitations regarding the practical sessions, Moodle could be enhanced by integrating it with the needed software tools. The integration could be done by building the software tools as a component in the Moodle. According to Rößling and Kothe [6] indicated that there are missing features in e-Learning environment that CSE needs which include the incorporation of programming exercises those allow students to practice their programming knowledge. Figure 1 demonstrates the following phases required to develop and integrate the software tools as plugins in the Moodle LMS.

- *Phase 1:* The existing LMS (e.g. Moodle) should be observed and analyzed in order to explore all its functionality and its architecture.
- *Phase 2:* The suitable extended architecture for Moodle system should be proposed and the needed software tools should be explored and/or developed.
- *Phase 3:* The Moodle system should be redesigned and customized to be ready to accept building new component on it.
- *Phase 4:* The needed software tools should be implemented using the same programming languages and coding style of the Moodle system to facilitate the communication between new component and system core and different components of the system.

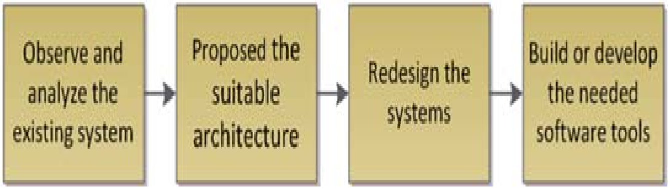
Observe and analyze the existing system
Proposed the suitable architecture
Redesign the systems
Build or develop the needed software tools

Figure 1. Development Phases of the Plugins Software Tools

A. Design and Analysis of the Plug-in Integration Approach

Figure 2 displays plugin tools integration approach. This figure shows the overall architecture of the integration between the users and the component of the Moodle. For this integration, the authors suggest the following three approaches to plug in the needed software tools to the Moodle:

1) Add new plugin tool as new features into the existing plugins tools such as: Courses and activities plugins. So the tools will be available in the activity resources as separate tools that the teacher can add them to course material.
2) Develop new plugins and create the API that communicates with the application core and other components in the system. So, it will be available in the Moodle as a new module.
3) Build a whole block in the system core to adopt more functionality to the software tools especially and course material in general. According to [18] there are many integration approaches of the tools to be used in the e-Learning platforms. One approach of this integration is by inclusion of Application Programming Interfaces (APIs) to generally make the functionality of the e-Learning platforms in terms of code written in the same language in which the e-Learning platform has been built. Figure 2 displays this approach.

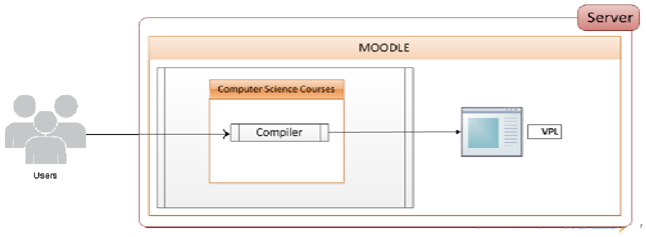

Figure 2. Plugin Tools Integration Approach

B. Experimentation on Plugin Tool

This paper suggested the implementation of the needed tools within the Moodle as new components. Different researchers indicated the importance of this approach [1, 5, 6, 7]. To prepare the Moodle with the needed tools for the CSE such as a compiler, the developers have to use the same programing language and technologies (i.e. PHP) that Moodle uses. In this case the developers could develop a compiler and make it as one activity plugins that they could add to the course of CS whenever needed. Figure 3 presents how the course developer can prepare and add the new tool to Moodle environment. It also shows how the user (i.e. student) can access and use the tool in the same Moodle environment.

In the Moodle there are hundreds of plugins that developers create since the Moodle is open source LMS. For example, the Virtual Programming Lab (VPL) [19] is one of the plugins that is related to Computer Science Education. It is a compiler that allows students to compile different programming languages and submit the assignments through it. This plugin software tool is part of the activity module and it is built with the same programming language of Moodle.

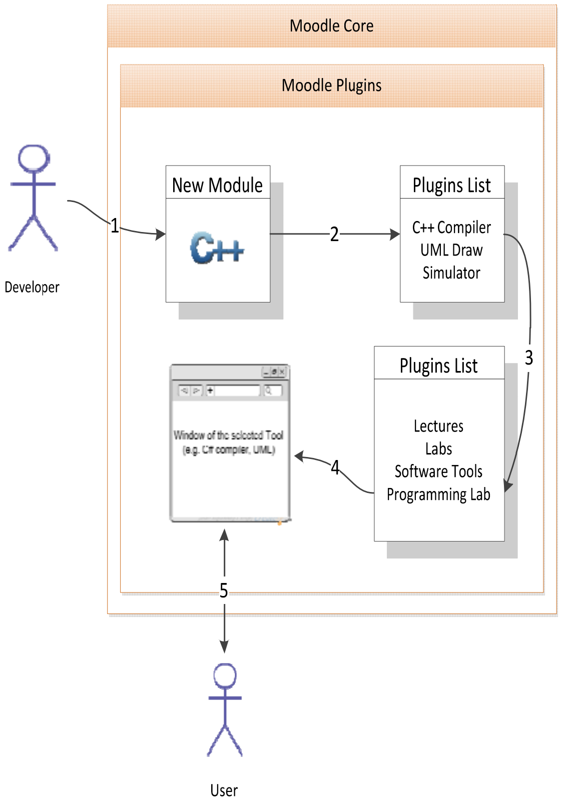

Figure 3. Development Process of New Module

The methodology of developing the needed tools is to use the coding style of Moodle to develop new plugins or sub plugins. First, develop the new module with the needed function using PHP programming language and test whether the new module works well. This development needs to access and customize a set of files that communicate with Moodle core system through its APIs to provide the needed functionality. First, the version of the system should be modified to detect the new version of the system during installation or upgrade in the files version.php. Second the new module should be introduced and added to the admin_tree and navigation list of the system, so the designer of the course can add the new software tools to their course. Then, the event handler must be identified for new module to communicate between core and plugins and plugins themselves in the file event.php. Furthermore, the database tables could be edited if there is information needed to be stored on it. However, it is not recommended to do that modification manually without change in the files install.xml and install.php. All these customization work could differ among versions of Moodle. After the preparation of the new modules and during the installation or upgrades of the Moodle the new module will be active in the plugins list. So, the course designers can add it to their courses and user could easily access its functions as shown in Figures 3 and 4.

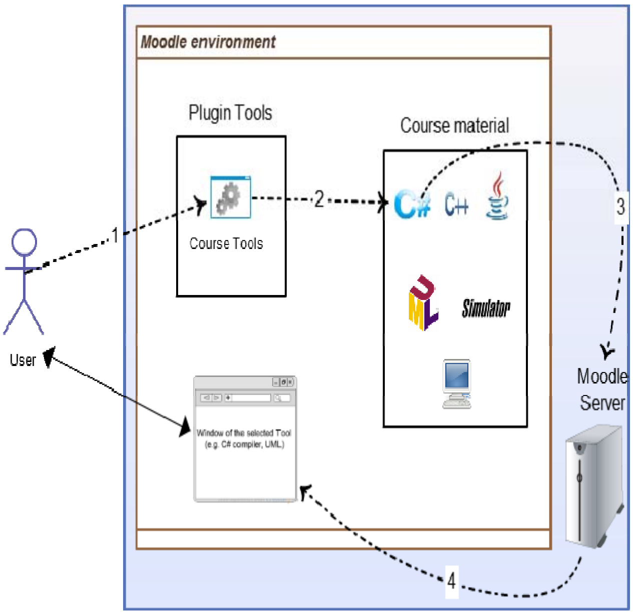

Figure 4. User Interaction with the New Component

VI. CONCLUSION AND FUTURE WORK

The study is aimed to extend the main idea behind the development of e-Learning platforms tools to be useful in the academic life. The contribution from this research work is to study the related work of the tools that were developed for the CSE. Another contribution is to explore the functionality of LMS to support software tools that are helpful in Computer Science area.

This paper shows that many Learning Management Systems have a limitation in the availability of Computer Science software tools. These software tools could help learners in CSE to gain knowledge through practical sessions. There are many LMSs categorized as open source which enable the developers to customize them according to the users' needs. The current authors preferred to use Moodle for two reasons: 1) Moodle's property as an open source and 2) Moodle is used in the Sultan Qaboos University.

In the literature, several researches introduced some attempts in extending the software tools for CSE. In this direction, this paper suggests building and integrating the needed software tools in the Moodle as a new component.

The proposed work in this paper helps the students to access the software tools of the CSE online from any place and at any time [5]. This will help in improving the students' knowledge about different practical concepts of the CSE. However, the proposed solution has several limitations including:

- Plugging software tools could produce a bottleneck on the Moodle because the students will remain logged-in doing their work in the same environment of Moodle.
- The new added component will consume the resources of the Moodle and need a huge capacity and performance facilities [3].
- The new added component could reduce the performance of the Moodle, since it will be utilized by the students who are running the software tools.
- This approach could increase the development effort when they integrate the tools in the Moodle. So, it needs to redesign and recode the needed software tools.

Future work of this paper is to look for alternative techniques or approaches to prepare the software tools and link them to Moodle. One of the solutions could be using the Service Oriented approach through which the student could use the service that represents the required software tool. This solution involves that the student should virtually be sent from Moodle to the specified location in the WWW depending on the required services (i.e. software tools).

ACKNOWLEDGMENT

The authors wish to thank The Sultan Qaboos University, College of Science and the Department of Computer Science. This work was supported in part by internal grant from the Sultan Qaboos University.

REFERENCES

[1] E.-E., Doberkat, G., Engels, J. H., Hausmann, M., Lohmann, J., Pleumann & J., Schröder (2005). Software Engineering and eLearning: The MuSofT Project. eleed, Vol. 2005. [online] <http://eleed.campussource.de/archive/2/201> [Retrieved on: 10/2/2014].

[2] K., Palanivel, & S., Kuppuswami (2011). "Service-Oriented Reference Architecture for Personalized E-learning Systems (SORAPES)". *International Journal of Computer Applications*, Volume 24, No.5, pp. 35-44.

[3] X., Huan, R., Shehane & A., Ali (2011). "Teaching computer science courses in distance learning". *Journal of Instructional Pedagogies*, Vol. 6, pp.47-60.

[4] D., Dagger, A., O'Connor, S., Lawless, E., Walsh, & Wade, V. P. (2007). "Service-Oriented E-Learning Platforms: From Monolithic Systems to Flexible Services". *Internet Computing*, IEEE, Vol. 11, No. 3, pp. 28-35.

[5] M., Palumbo & F., Verga (2013). "Creation of an integrated environment to supply e-learning platforms with Office Automation features". *Interactive learning Environment*. Taylor & Francis LTD, pp. 12, 2013, ISSN: 1049-4820.

[6] G., Rößling & A., Kothe (2009). "Extending Moodle to Better Support Computing Education". Proceeding of the 14th annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE'09), pp. 146-150, ACM New York, NY, USA.

[7] E., Lavrishcheva, A., Ostrovski & I., Radetshiy (2012). "Approach to E-Learning Fundamental Aspects of software Engineering". Institute of Software Systems NAS of Ukraine, ICTERI '12, pp.1-10.

[8] J., Ramos, M. A., Trenas, E., Gutiérrez & S., Romero (2013). "E-assessment of Matlab assignments in Moodle: Application to an introductory programming course for engineers". *Computer Applications in Engineering Education,* Vol. 21, No. 4, pp. 728-736.

[9] F., Corbera, E., Gutiérrez, J., Ramos, S., Romero & M. A., Trenas (2008). "Development of a new MOODLE module for a basic course on computer architecture". *ACM SIGCSE Bulletin - ITiCSE '08* Homepage Vol. 40, No. 3, pp. 349-349, ACM New York, NY, USA.

[10] G., Rößling & T., Vellaramkalayil (2009). "First steps towards a visualization-based computer science hypertextbook as a Moodle module". *Electronic Notes in Theoretical Computer Science*, Vol. 224, pp. 47-56.

[11] J., Prieto-Blazquez, J., Herrera-Joancomartí & A.-E., Guerrero-Roldán (2009). "A Virtual Laboratory Structure for Developing Programming Labs". *International Journal of Emerging Technologies in Learning (ijet)* . Vol. 4, No. 1, pp. 47-52.

[12] V. V., Riabov & B. J., Higgs (2010). "Software Tools and Virtual Labs in Online Computer-Science Classes". *Learning Management System Technologies and Software Solutions for Online Teaching: Tools and Applications,* edited by Yefim Kats, IGI Global, 2010, pp. 332-350.

[13] M. A., Bochicchio & A., Longo (2010). *Extending LMS with collaborative remote lab features.* Paper presented at the Advanced Learning Technologies (ICALT), 2010 IEEE 41st ASEE/IEEE Frontiers in Education Conference GOLC1, October 12 - 15, 2011, Rapid City, SD.

[14] A., Fest (2011). "Adding intelligent assessment: a Java framework for integrating dynamic mathematical software components into interactive learning activities". Journal: ZDM, Vol. 43, No. 3, pp. 413-423.

[15] G., Vossen & P., Westerkamp (2006, 5-7 July 2006). "Towards the Next Generation of E-Learning Standards: SCORM for Service-Oriented Environments". *The* 6th International Conference on *Advanced Learning Technologies.* pp.184-186, July 05-07, 2006.

[16] C., Alario-Hoyos, M. L., Bote-Lorenzo, E., Gómez-Sánchez, J. I., Asensio-Pérez, (2012). "GLUE!: An Architecture for the Integration of External Tools in Virtual learning Environments". *Computers & Education*, Vol. 60, No. 1, pp.122-137.

[17] Z., Al-Khanjari, N., Kutti, & M., Hatam (2006). "An Extended E-Learning Architecture: Integrating Software Tools Within The E-Learning Portal", *International Arab Journal for Information Technology (IAJIT)*, Vol. 3, No. 1, January, 2006.

[18] F., Huertas & A., Navarro (2013). "Integration Mechanisms in e-learning Platforms"**.** *International Journal of Computer Information Systems and Industrial Management Applications*. Vol. 5, pp. 714-721.

[19] J. C., Rodriguez-del-Pino (2014). "New Virtual Programming Lab (VPL) module", [Online] http://www.moodlenews.com/2011/3rd-release-update-to-moodles-virtual-programming-lab-vpl-module/ [Retrieved on: 2/3/2014].