The design and delivery of hybrid PBL sessions in Moodle

Khulood Khalil Al-Dous, and Mohammed Samaka

***Abstract*—** in this study we provide a technical support for teachers to practice problem-based learning (PBL) in their real learning sessions at schools, universities, or any other learning environment. The provided support would enable teachers to dynamically instantiate PBL sessions based on a PBL meta-model. The PBL meta-model include the definition of all possible elements within a PBL process which are phase, activity, resource, tool, and artifact. It is originally built based on the best existing PBL practices and well-known models. The process of practicing a PBL session involves two steps, the first is designing a PBL session and the second is delivering the session into students to execute it in a learning environment. In this study we choose to provide the PBL technical support within an environment where both teachers and learners are familiar with, this is instead of a standalone environment which would be time and effort consuming for them to use. A Learning Management System (LMS) called Moodle was extended in this study to support PBL as it is commonly used in many universities and schools with a comfortable plugin mechanism for developer. We developed a complete Moodle plugin called (PBL lesson plan) that support the design and delivery of PBL sessions or PBL lesson plans with different view privilege for both teacher and student. The wide usage of Moodle would increase the usage of our developed plugin. The developed PBL lesson plan plugin was tested and evaluated positively by teachers and students in real sessions conducted at Qatar University.

***Keywords*—**Learning management system, Moodle, PBL lesson plan, PBL meta-model, Problem-based learning.

I. INTRODUCTION

RECENTLY, a wide range of changes occurred in the economic and technological fields which lead the world to experience a huge transition from the post-industrial economy to knowledge economy. This transition transformed professional life and increased the number of skills that they need to master which includes dealing with increasing internationalization, using information technology, working within groups and mastering the required expertise. This affected highly the training programs of employee and high education expectations. Graduate students are expected to have convinced knowledge-basis beside the skills of solving problems, analyzing, synthesizing, coaching, leading, presenting, and evaluating them. Hence, this is the expectation of the information community on the future [1].

Generally, the common way of teaching in Qatar and around the world is the traditional subject-based methodology. Hence, this methodology does not fit into the current expectations of students graduates and that raised the importance of integrating both knowledge and real-world problems together. The integrating can be done through developing and implementing instructional real-world practices. Normally, students gain huge amount of inert knowledge from the traditional subject-based learning methodology, however students have to learn facts passively without linking them into the right real-world context. As a result, they are not able to use such knowledge in solving real-world problems, since they do not experience a real use of its context. The true knowledge required these days have been defined by UNESCO's report [2] as a combination of four pillars which are "learn how to know, learn how to work, learn how to live, learn how to exist". Consequently, there is a rapid change in the concept of knowledge over time; hence applying the gained knowledge in real context is needed besides memorizing facts. This is according to the need of the current business activities which are more intellectualized and require universal labor that is creative [2].

Many attempts have been made to address the integration of instructional models together with the traditional knowledge transmission models [1]. One problem-driven approach that made inroads into different education fields, such as engineering and science, had been developed within the recent decade which is called problem-based learning (PBL). PBL is a learning-pedagogy that provides the students a guided experience in learning through solving complex, real-world problems [3].

Several definitions can be found to identify the process of PBL. One definition is "the learning which results from the process of working towards the understanding of, or resolution of, a problem" [4]. Another one is "the conception of knowledge, understanding and education that encourages open-minded, reflective, critical and active learning" [5]. Howard Barrows, one of the PBL inventors, defined PBL as "a total approach to education. In PBL there is a curriculum of carefully selected and designed problems. And there is a PBL process, which, among other things, replicates the commonly used systematic approach to resolve problems or meeting

This publication was made possible by NPRP grant # 5–051–1–015 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the author(s).

K. K. Al-dous is with the Computer Science and Engineering Department, Qatar University, Doha, Qatar (e-mail: khuloodk@qu.edu.qa).

M. Samaka is with the Computer Science and Engineering Department, Qatar University, Doha, Qatar (e-mail: samaka.m@qu.edu.qa).

challenges. Students and teachers roles are redefined. Students assume the responsibility for learning and teachers become facilitators: stimulating and guiding students in their problem solving and self-directed learning" [6]. The common definition of PBL is using the problem to drive the learning process and asses the outcomes.

PBL has shown high improvement in students learning process including the problem solving skills, group work, participation, and attendance [7]. Despite the successfulness of PBL in integrating knowledge with real-life problems, there are a few challenges facing the diffusion of PBL. Initially, with limited exceptions, teachers lack the expertise to transform a lecture driven course into a problem-driven course, as they are not cognitive or learning scientists. They are experienced in teaching and lecturing, thus they have difficulty in changing their role to that of a facilitator who guides students instead of providing or feeding them new knowledge [8]. Consequently, the task of redesigning a classroom and all students' activities can be a daunting and time-consuming task. Furthermore, students are not familiar with PBL-pedagogy and need time to adjust new kinds of demands including self-directed inquiry and complex problem solving. Some of these challenges have been addressed through applying a PBL learning model called Tutorial Based Learning (TBL) which incorporates real-problems with theoretical knowledge, hence TBL improved the students grades and participation [9]. Overcoming such challenges successfully would require various kinds of technical and theoretical support for both teachers and student to understand and change their traditional roles into PBL roles accordingly. As a result, in this study technological solutions were adopted to address these hurdles.

PBL has wide range of models with different specifications such as Wood's model [10], Maastricht "seven jump" model [11], and IMAS model [12]. A PBL model consists of steps that form a lesson plan for students to follow in order to solve a real-world problem. Applying PBL approach in learning environments raised the need for technical support that addresses the specification of PBL-pedagogy. According to this need the motivation of this study was to provide technical support for teachers to design and deliver PBL lesson plans for students. This study aims to investigate, design and develop an innovative PBL online system that supports teachers in designing and delivering PBL lesson plans for students. Also, aims to facilitate the design and execution of wide range of PBL models with their variations in flexible and reusable manner.

The main objective of this research study is to address the most important shortcoming of the existing online PBL systems which is that they are fixed into one specific PBL model without the flexibility to customize it. Also, they missed the support of interoperability and integration, besides that they are built based on traditional software development methods where the cost of both time and effort is relatively high. This study targets building a flexible online PBL system that enable representing a wide range of PBL models.

We found the most appropriate learning management systems which can be extended to implement the online PBL system. This involves contributing into the professional community of both developers and users since the used learning management systems is an open source. Also, we study and utilize existing PBL scripting language to be adopted within the PBL system in order to enable the representation of different PBL models and achieve our main objective. Besides that, we found the most appropriate way of evaluating the developed system based on previous evaluation methodology used to evaluate online PBL systems.

The scope of the study is limited to support PBL learning pedagogy in providing a flexible environment for teachers to dynamically instantiate PBL lesson plans in an easy, cost-effective, flexible, interoperable, and reusable manner. The PBL elements and relations within a PBL lesson plan are all identified based on the adopted PBL scripting language. Also, to stimulate the dynamic transformation of the designed PBL lesson plans into a run-time environment in which students can executed and teachers can monitor and handle the execution of the PBL lesson plan elements. The scope of implementing the PBL runtime environment involves implementing only one functional aspect for teachers to handle the execution of a PBL lesson plan. Also, some functional aspects are implemented for students to follow a PBL lesson plan.

The Significance of this study is to support teachers to change their roles into facilitators instead of information source. At the same time, students who are used to passive listening, note taking, and memorization also need help in transitioning to activities that situate learning in the need to solve real-world problems.

II. Background and Literature Review

Problem-based learning was first developed for face-to-face learning environment. However, as the computer technologies are growing rapidly, many attempts have been conducted to combine PBL with computer supported collaborative learning (CSCL). Currently, the high availability of Internet makes it possible to implement PBL in online environment that can be used in hybrid with face-to-face environment.

Over the past decade researchers in the area of CSCL have developed numerous computer supported online PBL environments. Five of these environments were reviewed in this study including Socio-Technical Environment for Learning and Learning-Activity Research (STELLAR) [13] [14], Computer Supported Intentional Learning Environments (CSILE) [15], Web-Scaffold Multi-user Integrated Learning Environment (Web-SMILE) [16], Collaborative Medical Tutor (COMET) [17], and electronic Problem-Based Learning (e-PBL) [18]. In summary, each of the reviewed online PBL environments relies on one specific PBL model. They support PBL processes by providing associated structures, resources, guidance, and tools. Using these PBL environments, teachers and students can easily generate, understand, and conduct PBL lessons. They all have a main common advantage for supporting a successful PBL process through providing proper and relatively complete environments. Nevertheless, such environments might be successful only in certain circumstances and might be inappropriate to other situations or domains. The practical problem of such environments is that they missed the support of interoperability and integration. That is users are limited to the functions and data structures provided in these environments. They can manually shift and transform data from one system/tool to another which is not easy for the users and is definitely a time-consuming task. Besides that, these PBL environments are built based on traditional software development methods where the cost of both time and effort is relatively high. Implementing such environments to change from one PBL model to another is not an easy task, hence the teachers have to follow the limited workflow in the software environment they are using. Consequently, teachers have less flexibility to customize existing PBL models and to apply them in their own desired PBL lesson plans.

PBL scripting language is an Educational Modeling Language (EML) developed by Miao et al to support PBL-pedagogy where PBL scripts are used to structure and support technology-enhanced, problem-oriented, collaborative learning processes [19]. The PBL scripting language adopted a domain specific modeling language paradigm, which supports higher abstraction level, requires less effort and fewer low-level details to specify a given system than general-purpose modeling languages. It is designed for teachers to represent PBL models. Additionally, it was developed according to the best PBL practices and the well-known PBL models. A teacher can use this language to create a PBL process which is represented as a sequence of phases and within each phase there is a sequence of relevance activities, resources, artifacts and collaboration tools. A phase could be problem-engagement, problem-analysis, aim-and-plan, research, problem-resolution, or evaluation. Some examples of activities are presenting, identifying, planning and investigating. A resource can be used as an input of an activity such as a problem source or real-world problem scenario. Artifacts are produced and used in activities, such as problem-statement and problem-solution. Furthermore, a collaboration tool could be chat room, wiki, or discussion-forum. In order to facilitate teachers in designing PBL lesson plans easily, a graphical PBL script editor was also developed based on the PBL scripting language [19].

A. State of the art Development

The formulation of PBL scripting language and the development of graphical PBL script editor addressed the limited aspects of the reviewed online PBL environments. However, using the graphical PBL script editor would require from teachers to first design the PBL process. Then to generate Unit of Learning (UoL) and use an IMS LD player to execute it, which is time and effort consuming. Eventually, a wide range of teachers are not familiar with IMS LD, as it is a research based specification.

Currently, the usage of Learning Management Systems (LMSs) is widespread over many colleges, universities and schools worldwide. Hence, most teachers and students are more familiar with such LMSs and they do not tend to experience or learn new systems. Teachers can use LMSs to dynamically update and manage their courses online [20]. In the same time, students can access their courses and collaborate with other students any time they prefer [20].In this study, Moodle LMS was selected to be customized for implementing a PBL design and runtime environment. Moodle is an open source system that customizing it is much easier than other LMSs. The wide international usage of Moodle and its continued growth during the last six years have made it the leading open source LMS solution. A previous study, which was conducted to show the effectiveness of teaching students using Moodle in different university levels, confirmed that Moodle “enabled the students to promote understanding and greater respect for digital technology” [21]. Beside that Moodle provides a comfortable plugin mechanism for functional extension and customization. That raised up the challenge of adopting the PBL scripting language to extend Moodle LMS in which many teachers and students can benefits from it. The successfulness of e-PBL to extend Moodle towards the support of PBL-pedagogy and its limitation to support only the Woods’ PBL model formed the basis of this study interest.

This study involves extending Moodle environment to dynamically instantiate a PBL lesson plan by adopting a PBL scripting language. This would enable ordinary teachers to design and deliver online (and hybrid) PBL lesson plans in an easy, cost-effective, flexible, interoperable, and reusable manner.

III. Methodology of Building the PBL Plugin

This study adopted a Model-Driven Approach (MDA) as the software development method to extend Moodle platform in order to design and implement a PBL design time environment based on PBL scripting language.

In this study, the MDA methodology supports customizing the design of PBL lesson plans for PBL teachers. For example, when they tend to create a new phase they will get a list of all possible phases and their associated activities, recourses, and tools based on the PBL scripting language. In this way they can easily create PBL lesson plans with less time cost. The key challenge of adopting the MDA methodology is the transformation of both PBL meta-model and PBL script of PBL lesson plans into Moodle platform-specific configuration. By applying the MDA methodology to the development involved in this study, the system high-level architecture is designed as depicted in Fig. 1. The system architecture consists of two main parts: design time environment and run-time environment. The design-time environment supports the design of PBL lesson plans. The PBL runtime environment used to execute the PBL lesson plans by both teacher and students. A basic workflow of the system architecture starts by an ordinary teacher or by any PBL designer that has a desired PBL lesson plan. The teacher would use the PBL script editor to design the desired PBL lesson plan which is built based on a PBL scripting language. After the teacher completed the design of his/her desired PBL lesson plan, he/she can reuse and share it through a PBL script. Also, a PBL lesson plan can be automatically transformed into a PBL runtime environment using a PBL open source player. In this study Moodle open source was used to implement both the PBL design time environment and the PBL run-time environment. Additionally, the teacher can instantiate multiple execution of a PBL lesson plan that can be used by different teachers and students.

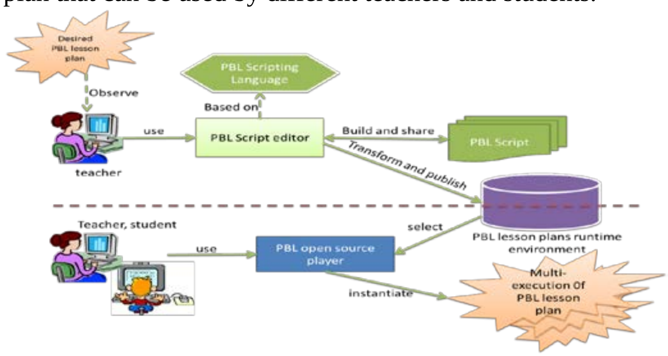

Fig. 1 system architecture

IV. Implementation

A PBL Moodle's plugin was developed to facilitate the design and delivery of PBL lesson plans. Normally, Moodle is independent from the used operating system within a machine since it is a web based platform in which end users can access it through a web browser. Moodle is compatible with most modern web browsers (e.g. Firefox, Internet explorer, and chrome). However, some functionality should be enabled within any used browser in order for Moodle to function correctly. These functions are Pop-ups, Cookies, and JavaScript. The implementation of the new Moodle's plugin involves both a design and a run-time environment of PBL lesson plans. The plugin is created according to the predefined structure of Moodle plugins.

A. Design-time Environment

The design time environment enables all the editing functions for teachers to design and create PBL lesson plans. Such editing functions include creating, editing, and deleting a PBL lesson plan and all its elements. PBL lesson plan elements are phase, activity, tool, resource, and artifact. A PBL lesson plan can have many phases and each phase consists of different activities. Resources and tools can be used by students in activity level to achieve the activity goals and submit artifacts as outputs. According to the used PBL meta-model, the structure of a PBL lesson plan is divided into two levels. As illustrated in figure 2, in level 1 the phases of the plan are defined while level 2 comprises the activities of each phase defined in level 1. According to the complexity of the desired PBL lesson plan, the designer can specify the number of phases and activities in each level.

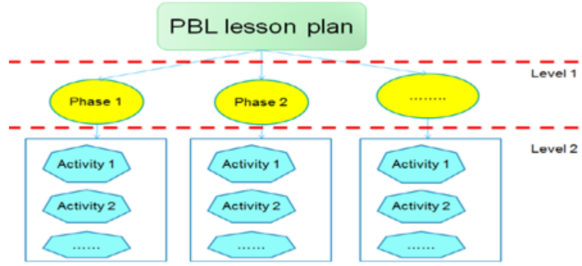

Fig. 2 the PBL lesson plan structure

The PBL design time environment read and processes the PBL meta-model which is a JASON script file. Then dynamically instantiate the list of all PBL elements types from the meta-model. This is beside the connections between the elements which restricts the relation between the elements. For instance, when the teacher creates a new phase, a list of phase types will be instantiated dynamically from the PBL meta-model as in Fig. 3. If the teacher chooses a specific phase like "problem definition", then he/she can create activities within this phase. Similarly, when creating a new activity, the associated list of activity types for the specified phase will be dynamically instantiated from the meta-model. Hence, the available activities for the "problem definition" phase are shown in Fig. 4. This list of activity types vary from one phase to another according to the PBL meta-model. In addition, the same procedure is performed when creating a new tool, artifact or resource.

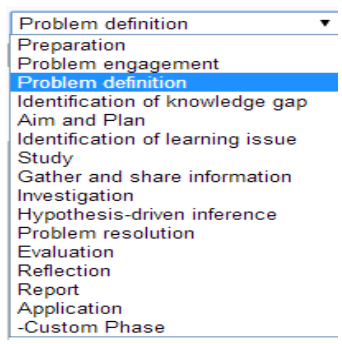

Fig. 3 list of phases types in a PBL lesson plan from the PBL meta-model

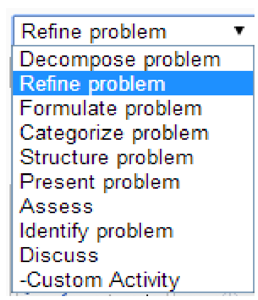

Fig. 4 list of activities types in "problem definition" phase from the PBL meta-model

The teacher can easily share any PBL lesson plan with other teachers through the implemented import and export functions within the design time environment. Hence, this function is built based on the language used within the PBL scripting language. In which the designed PBL lesson plan can be transformed into/from a PBL script. This involve an implementation of a PBL script reader and writer which read and write JSON file to be shared and reused by many teachers. Having such function would reduce the time and effort needed from the teacher to redesign similar plans for the same course for different problems.

Moreover, the teacher can specify the run structure of the activities within a phase whether to run in sequence, parallel or selective. For example, in case of sequence run structure, the teacher can drag and drop the activities of a phase within a PBL lesson plan to satisfy the desired sequence. Similarly, the running structure of phases can be applied. If the run structure is in sequence, then the activity or phase that appears first in the PBL lesson plan would run first. A screenshot of the implemented design time environment is shown in Fig. 5 in which a simple PBL lesson plan is designed. It consists of three phases each phase has two activities with the associated artifacts, tools, and resources.

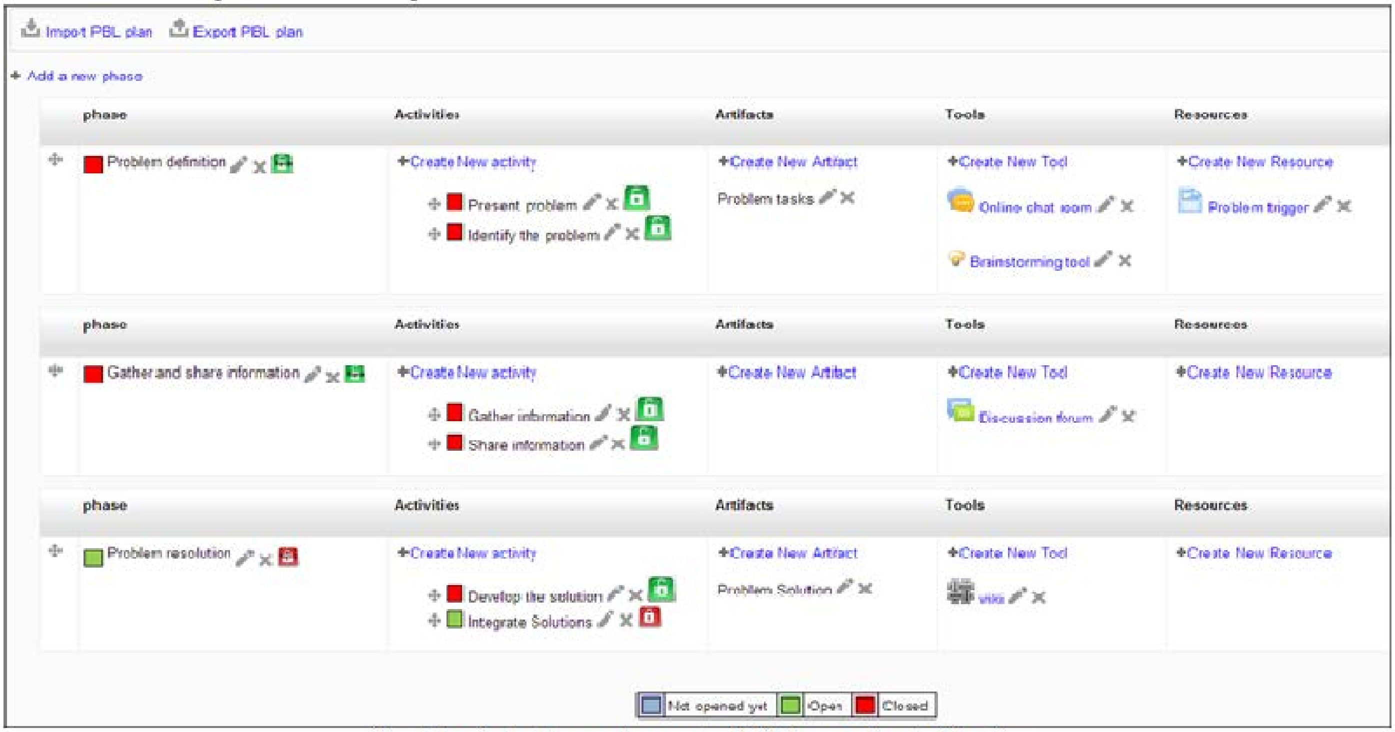

Fig. 5 the design time environment of PBL lesson plans in Moodle

B. Run-time Environment

The runtime environment is a player to manage the execution of the designed PBL lesson plans and make them ready for student to follow and for teacher to handle. It provides the technical support for both teachers and students. One functional aspect was implemented for teachers to handle the execution of a PBL lesson plan and some functional aspects were implemented for students to follow a PBL lesson plan. The teacher can open/close a phase/activity at any time. The students can follow the PBL lesson plan phase by phase and use the available tools and resources. Also, students can submit artifacts within activities.

The run structure specified in the design time of the PBL lesson plan was applied to handle the execution sequence for both activities and phases. In sequential run structure no time overlapping is allowed between the start and end time of activities or phases. However, the teacher can force open or close any phase or activity at any time. If a teacher forces closing an activity or phase at specific time, the end time of the activity or phase will be automatically changed to that specific time. Additionally, force closing any phase will force closing all its activities. On the other hand, force opening a closed phase would first close all the open phases and assign the force open time as the end time for them. After that, will force open the specified phase by changing its start time to the time the force open requested. Applying such constrains within the runtime environment in the sequential run structure ensure that only one phase or activity can be open in a time.

Furthermore, the teacher can specify the complete condition of activities and phases whether to be time limit or user control. In case of a time limited activity/phase, the teacher has to specify the start and end time for it. Though, the teacher can force open/close an activity/phase any time. In case an activity complete condition is set to “user control” then the teacher can simply choose to open/close an activity/phase as shown in Fig. 6. Opening an activity will change its status to open. If the run structure of activities inside a phase is sequential and there is already one open activity, then runtime environment will prevent opening new activity and warn the teacher to close previous activities first.

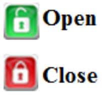

Fig. 6 open and close icons in runtime environment

The status of an activity/phase was categorized into three types which are not open yet, open, and closed. Each status type is identified by a different color as shown in Fig. 7. Having such colorful statuses help the teacher in tracking the execution of the PBL lesson plan elements at any time. Additionally, only activities/phases with open status are visible to the students and the others are hidden.

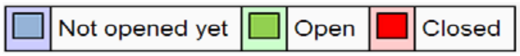

Fig. 7 activity/phase possible statuses

As the teacher makes a designed PBL lesson plan visible for students, they can work to solve the presented problem by following the plan as guidance for them. The students can view the current problem information in the home page of the plan including the title, description, grade, timing and resources. Also, if the students are required to work in groups then each student can see a list of his/her group members as shown in Fig. 8. This makes it easy for students to identify their group members with the ability to exchange direct messages. Sending a direct message from one student to another is an important feature of Moodle that was adopted in the developed runtime environment to ease the communication between students. Besides that, the home page of the PBL lesson plan states the phases involved in the PBL lesson plan with their status (open/close), time (start and end time) and grade.

Fig. 8 list of group members for students with the current logged in student highlighted

Moreover, another page called “current phase” can be viewed by students which shows the current open phase(s) that they need to work on. Hence, this page presents the detailed description, goals, and list of activities for the current open phase. The students can then navigate to the current open activities of that phase and see the details of each open activity including the activity tools that can be used, the activity resources and the required artifacts. For example, “identify the problem” is an activity type within “problem definition” phase which use two tools; the brainstorming and chat; use a resource “problem trigger” as a page; require a submission of “problem tasks” list artifact with online text submission medium type. Fig. 9 shows the page of an open activity that students can view for the illustrated example where students can choose to enter an online chat room tool to collaborate with their group members. This is besides viewing the available resources and submitting a desired artifact. An artifact submission page has been developed for student to be able to submit and edit their required artifact submission. Consequently, students can repeat the same work flow for every activity of each phase till all phases are completed to reach the end of the PBL lesson.

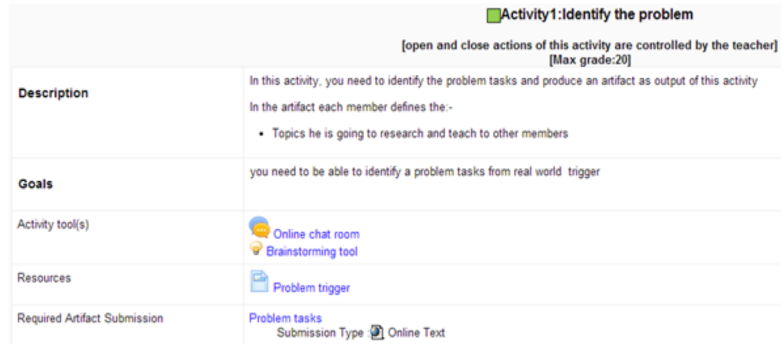

Fig. 9 an open activity workspace for students

V. Evaluation

This section describes the evaluation methodology applied to evaluate the usefulness of the developed PBL lesson plan plugin in Moodle.

A. *Evaluation of design-time environment by teachers*

This evaluation was applied to a PBL practical session conducted at Qatar University. The evaluation of the design time environment in the developed Moodle’s plugin comprised eight ordinary teachers who participated in the session. They had been chosen to represent a wide range of subjects including computer science, computer engineering, Arabic, industrial engineering, and civil engineering. Additionally, they also have come from different teaching levels such as teaching assistant, assistant professor, and lecturer. The session was self-controlled, as the teachers started using the PBL design time environment without any training. The target of this evaluation was to test the usefulness of the design time environment by teachers. The session lasting one hour was distributed in the following manner:

- The first ten minutes were used to give an introduction of the PBL learning methodology and the different models involved.
- The next 40 minutes were consumed to build a PBL lesson plan using Moodle. A predefined PBL lesson plan was given to each of them which comprised of four phases. Each phase had two activities with some interconnected tools, resources, and artifacts. The participating teachers were asked to create the PBL lesson plan even though no prior training was given to them on how to use the design time environment of PBL lesson plans.
- Following this, the teachers were given five minutes to fill a survey. The survey consisted of four sections to assess the teachers on; the background information, computer literacy, usefulness of the PBL scripting language, and the design time environment of PBL lesson plans, respectively. The last two sections of the survey evaluate the usefulness of the developed PBL design time environment, while the first two sections used to analyze the relation between the teacher’s evaluation and their prior background information and computer

literacy. The last two sections consist of 17 statements, for each, the participant has to choose his/her level of agreement. (1: strongly disagree, 2: disagree, 3: no opinion/unsure, 4: agree, and 5: strongly agree).

The collected data was analyzed using descriptive statistics such as mean and standard deviation. Table I shows the mean and standard deviation values of the analyzed 17 statements. The mean represents the average agreement score (1: strongly disagree, 2: disagree, 3: no opinion/unsure, 4: agree, and 5: strongly agree) of the eight participated teachers for each survey statement. The mean scores of all seventeen statements are larger than 3.25 and most are near 4.0. The first five survey statements were related to the ease of use PBL scripting language, while the rest of the statements represented the ease of use PBL design time environment. The PBL scripting language was part of the evaluation as it was used within the development of the PBL design time environment. Obviously, most of the participated teachers responded positively on all aspects of the evaluation statements for both; the used PBL scripting language and the PBL design time environment. Additionally, the Cronbach's alpha (α) value was calculated to measure the design time environment ease of use scale. Hence, the ease of use α scale is 0.835. Nunnally recommended that any instrument used in a basic research should have reliability of about 0.70 as a base or better [22]. According to Nunnally, the resulted α scale of Moodle's PBL design time environment demonstrates that the survey concerning ease of use is quite reliable comparing to the base value which is 0.70.

Table I Survey results for evaluating the PBL design time environment ease of use

	Survey statements	Mean	Std. Dev.
1	The two levels (phase-level and activity-level) structure of the PBL script editor.	3.875	0.35
2	The terms or vocabularies used to define the PBL elements types.	3.75	0.71
3	Find an appropriate term or vocabulary to represent a PBL lesson plan.	3.75	0.46
4	The activity structure that includes tools, resources, artifacts and their interconnections	4	0.00
5	Represent a narrative into a PBL lesson plan.	3.25	0.71
6	Define students groups.	4	0.53
7	Create/delete a PBL lesson plan in a Moodle course.	4.25	0.46
8	Create/delete a phase.	4.375	0.52
9	Create/delete an activity.	4.25	0.46
10	Create/delete a tool (e.g. chat, forum).	4.25	0.71
11	Create/delete an artifact.	4.25	0.71
12	Create/delete a resource.	4.25	0.89
13	Interconnect tools to be used in an activity.	4.25	0.71
14	Specify activities and phases running sequence.	4.125	0.35
15	Define the detailed information of the PBL lesson plan such as description and goals.	3.625	0.74
16	Specify the completion condition (e.g. user control or time limit) of an activity or a phase.	3.625	0.92
17	Export a PBL lesson Plan into PBL script and then import it again.	4	0.76

In order to validate our result from using the cronbach's alpha we provide evidence that the scale of the survey statements is unidimensional. As a result, we used exploratory factor analysis method in SPSS software in order to check the dimensionality. The factor analysis was settled to use the principle component method as the extraction method for dimensions. The eigenvalue of extraction was selected to be 2 as a minimum. By applying the factor analysis in the 17 statements the SPSS generate the table labeled "Total Variance Explained" as shown in Table II. From the table we observe that the eigenvalue of the first factor is quite a bit larger than the eigenvalue of the second factor (13.023 versus 1.811). This difference is shown clearer in the plot diagram shown in Fig. 10. Additionally, the first factor accumulate 76.608% of the total variance, hence this validate the unidimensionality of our concerned factor of the survey which is to evaluate the ease of use factor of our developed environment. This indicates that the statements of the survey are highly correlated to each other, thus the cronbach's alpha indicate a good internal reliability of the ease of use factor of our environment.

Table II total variance explained table generated from SPSS software for the 17 statements of the survey for the 8 teachers

Total Variance Explained						
Factor	Initial Eigenvalues			Extraction Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	13.023	76.608	76.608	13.02	76.608	76.608
2	1.81	10.650	87.258			
3	.840	4.940	92.198			
4	.631	3.710	95.908			
5	.348	2.046	97.954			
6	.348	2.046	100.00			
Extraction Method: Principal Component Analysis.						

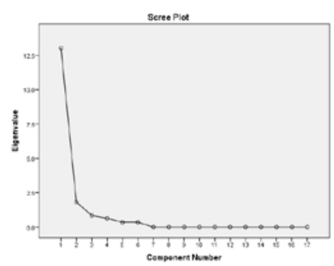

Fig. 10 Factor analysis plot diagram for the PBL design time environment evaluation data

Additionally, the results of the first two sections of the survey were analyzed. The first section involved information about the participated teachers' prior PBL knowledge. Fig. 11 shows the analysis of the prior PBL knowledge scale (1: Nothing, 2: a little, 3: basic knowledge, 4: knowledgeable, 5: expert) and the ease of use mean for each teacher where no significant difference was noticed regarding the different level of prior PBL knowledge. That is teachers with no strong prior PBL knowledge still thought that the Moodle's PBL design time environment is easy to use.

Moreover, section two of the survey involved information about the teachers' computer literacy levels. In Fig. 12, each point represents the relation between a teacher computer literacy level and his/her ease of use mean. The computer literacy level was calculated as the mean of different skills levels in different aspects significant for using Moodle's design time environment. The skills were regarding the teachers' use of generic computer tools (e.g. Word MS), communication tools (e.g. chat), teaching tools (e.g. digitalized whiteboards), and learning management systems (e.g. Moodle). On analyzing the collected data, we infer that there is slight positive influence of computer literacy level on the ease of use mean, though some deviations exist. Despite the fact, some participated teachers had little technical knowledge of computers; the majority of teachers still thought Moodle's design time environment is easy to use.

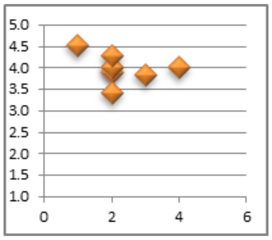

Fig. 11 scatter chart of teachers prior PBL knowledge(x-axis) and ease of use mean (y-axis)

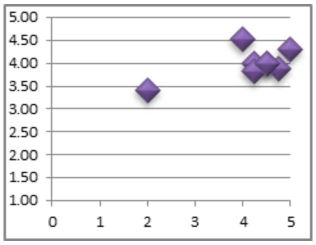

Fig. 12 scatter chart of teachers computer literacy mean(x-axis) and ease of use mean(y-axis)

B. *Evaluation of run-time environment by students*

The evaluation of Moodle's PBL run-time environment was conducted through another practical session conducted at Qatar University. The PBL run-time environment evaluation covered a sample of ten students who participated in the session. The students have come from different education programs. One student from the master program, two students from secondary school, and the other seven students were from bachelor program. The conducted evaluation organized in term of self-controlled that is the students started using the PBL run-time environment without getting trained beforehand. This evaluation targeted testing the usefulness of the developed Moodle's PBL run-time environment for students.

A complete PBL lesson plan with details about a real PBL problem was created for students to solve. The PBL lesson plan consisted of four phases; problem definition, identification of learning issues, gathering and sharing information, and problem resolution. The PBL lesson plan was designed for group working mode, so the participating students were divided into three groups.

The session lasting one hour was distributed in the following manner:

- The first ten minutes were used to give an introduction of the PBL learning methodology and the different models involved.
- The next 40 minutes, they worked in groups to solve a simple online PBL problem using Moodle's PBL run-time environment. They were asked to use Moodle's PBL run-time environment without prior training in how to use it.
- The last five minutes of the session they were asked to fill a survey according to their gained experience in using Moodle's PBL run-time environment. The survey consists of three sections; the first two sections are similar to the first two sections of the previously used survey in evaluating Moodle's PBL design time environment. While the last section involved 12 ease of use statements that were created to measure the usefulness of the developed Moodle's PBL run-time environment by students.

The collected survey data for evaluating Moodle's PBL run-time environment was analyzed the same way it was done for evaluating the PBL design time environment. Table III shows the twelve statements used in the survey together with their mean and standard deviation values for the ten participated students in the evaluation session. Eventually, the mean score for the twelve statements is near 4.0 and most are greater. Noticeably, most of the participated students responded positively on all aspects of Moodle's PBL run-time environment evaluation statements. In addition, Moodle's PBL run-time environment ease of use Cronbach's alpha (α) value is 0.78 which also demonstrates that the surveys' ease of use is quite reliable.

Table III results of evaluating the PBL run-time environment ease of use

	Survey statements	Mean	Std. Dev.
1	Recognize the two-layer (phase-level and activity-level) structure of the PBL lesson plan	4	0.47
2	Recognize the activity structure that includes tools, resources, and artifacts.	3.8	0.42
3	Work in group while solving a PBL problem.	4.7	0.48
4	Navigate into an existing PBL problem in a course	4.1	0.74
5	View the PBL lesson plan details and its phases	4.4	0.52
6	Identify the current status, grade, and time of each phase and activity.	4.4	0.52
7	Use existing communication tool (e.g. chat, forum) in activity level	4.1	0.32
8	Submit/edit a required artifact within an activity.	4.2	0.63
9	Navigate into an available resource within an activity.	4.2	0.42
10	Know the execution sequence of the PBL lesson plan phases and their corresponding activities.	3.8	0.63
11	View detailed information (e.g. description and goals) of activities and their corresponding tools, resources, and artifacts.	4.2	0.63
12	Specify the completion condition of an activity (e.g. time limited)	4.1	0.57

Similarly we validate the result by proving that the data of our survey are unidimensional through the exploratory factor analysis. By applying the factor analysis in the 12 statements, we generate the table "Total Variance Explained" using the SPSS software as shown in Table V. According to the eigenvalue shown in the table we perceive that the eigenvalue of the first factor is larger than the second factor (6.009 versus 1.366). Also, the first factor accumulates the highest percent 60.09% of the total variance which in turn prove the unidimensionality of our concerned factor in the survey. The gap between the factors is shown clearer in the plot diagram shown in Fig. 13.Through that we can conclude that the twelve statements of the survey are highly correlated to each other, thus the cronbach's alpha indicate a good internal reliability of the ease of use factor of the PBL run-time environment.

Table V total variance explained table generated from SPSS software for the 12 statements of the survey for the 10 students

Total Variance Explained						
factor	Initial Eigenvalues			Extraction Sums of Squared Loadings		
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	6.009	60.090	60.090	6.009	60.090	60.090
2	1.366	13.658	73.748			
3	.852	8.521	82.269			
4	.556	5.558	87.827			
5	.493	4.925	92.752			
6	.343	3.429	96.181			
7	.231	2.313	98.494			

8	.121	1.208	99.702			
9	.030	.298	100.000			

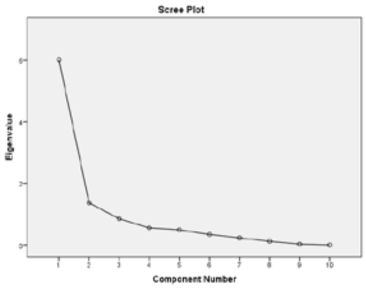

Fig. 13 factor analysis plot diagram for the PBL run-time environment evaluation data

Furthermore, the results of the first two sections of the survey were analyzed the same way as described in evaluating Moodle's PBL design time environment. This is because the first two sections are common between both used surveys. As shown in Fig. 14 the analysis of the students' prior PBL knowledge level and the ease of use mean have no significant difference. This means that the evaluations of both Moodle's PBL design time environment and PBL run-time environment were not affected by the level of prior PBL knowledge for the participated teachers and students. The scatter chart shown in Fig. 15, presents the relation between each student computer literacy level and his/her ease of use mean. The computer literacy level was calculated the same way done previously in evaluating the PBL design time environment. Though the computer literacy level showed a slight positive influence in evaluating the PBL design time environment, a slight negative influence appeared in evaluating the PBL run-time environment with some deviations. That is some students with little computer literacy knowledge evaluated the PBL run-time environment more positively than students with higher computer literacy knowledge. However, the majority of the students still thought Moodle's PBL run-time environment is easy to use.

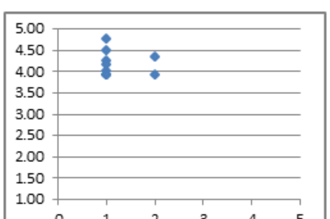

Fig. 14 scatter chart of students prior PBL knowledge (x-axis) and ease of use mean (y-axis)

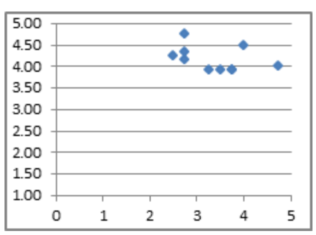

Fig. 15 scatter chart of students computer literacy mean(x-axis) and ease of use mean(y-axis)

VI. Conclusion

In conclusion, this study provides a technical support for teachers to apply PBL learning methodology in their courses. A PBL online system was proposed for designing and delivering PBL lesson plans. The PBL online system consists of a PBL design time environment to design PBL lesson plans and PBL run-time environment to simulate the execution of PBL lesson plans for students. Moodle's LMS was customized to implement both PBL design time environment and PBL run-time environment. This study used an existing PBL scripting language to enable the design and delivery of a wide range of PBL models. The design and implementation of Moodle's PBL design time environment facilitate teachers in building PBL lesson plans in flexible and reusable manner. Students would use Moodle's PBL run-time environment to follow a PBL lesson plan in order to solve a real world problem constructed by the teacher.

According to our evaluation of this study, both Moodle's PBL design time environment and Moodle's PBL run-time environment are easy to use with Cronbach's alpha (α) value greater than 0.7. To the best of our knowledge, no other study has built a PBL design time environment and PBL run-time environment within an existing LMS to support PBL-pedagogy which is making this study a unique international contribution. We believe that the developed Moodle's PBL design time environment and PBL run-time environment would inroad the PBL learning methodology into education here in Qatar and around the world.

VII. Future Work

The future work of this study involves different aspects. One is that enhancing Moodle's PBL run-time environment to support different functional aspects for teachers, as within the scope of this study only one functional aspect was supported. Other functional aspects including feedback function to continually give feedback for students for each PBL lesson plan activity. Also, providing peer and rubric evaluation to evaluate the students work in solving a PBL problem. Another functional aspect is a notification function that would notify the teacher when the students submit PBL artifacts.

Additionally, build a repository of PBL lesson plans for the teacher within Moodle's PBL design time environment to use and customize PBL lesson plans. Also, a search engine that aids the teacher to search the repository of PBL lesson plans for the best lesson plan that fit his/her needs from the repository. This would save the teacher's time and effort needed to design a new PBL lesson plan.

The interface of Moodle's PBL run-time environment will be improved in the future according to the students' feedback on the evaluation. Finally, this work inroads the development of PBL online system in other LMSs rather than Moodle.

References

[1] Dochy, F., Segers, M., Van Den Bossche, P., & Struyven, K. (2005). Students' perceptions of a problem-based learning environment. Learning environments research, 8(1), 41-66.

[2] Mandic, D., Kamuka, E., Lalic, N., Dejic, M., & Parezanovic, D. (2014). Computer Supported Changes in Education. RECENT ADVANCES in EDUCATIONAL TECHNOLOGIES and EDUCATION, 61.

[3] Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn?. Educational Psychology Review, 16(3), 235-266.

[4] Barrows, H. S. (1980). Problem-based learning: An approach to medical education. Springer Publishing Company.

[5] Johnston, A. K., & Tinning, R. S. (2001). Meeting the challenge of problem-based learning: developing the facilitators. Nurse Education Today, 21(3), 161-169.

[6] Barrows, H. S., & Kelson, A. M. (1993). Problem-based learning: A total approach to education. Monograph. Southern Illinois University School of Medicine, Springfield, Illinois.

[7] Klegeris, A., & Hurren, H. (2011). Impact of problem-based learning in a large classroom setting: student perception and problem-solving skills. Advances in physiology education, 35(4), 408-415.

[8] Dolmans, D. H., De Grave, W., Wolfhagen, I. H., & Van Der Vleuten, C. P. (2005). Problem-based learning: Future challenges for educational practice and research. Medical education, 39(7), 732-741.

[9] Chowdhry, B. S., Shah, S. M. Z. S., & Shah, S. M. A. (2014). A Tutorial Based Learning Model (TBL) for Engineering Students: the best they can get. RECENT ADVANCES in EDUCATIONAL TECHNOLOGIES and EDUCATION, 102.

[10] Woods, D. R. (1994). Problem-based learning: How to gain the most from PBL. Waterdown, Ontario: DR Woods.

[11] Wood, D. F. (2003). ABC of learning and teaching in medicine: Problem based learning. BMJ: British Medical Journal, 326(7384), 328.

[12] IMSA's PBL teaching and learning Template. Retrieved March 15, 2014, from http://pbln.imsa.edu/model/template/

[13] Derry, S. J., & Hmelo-Silver, C. E. (2005). Reconceptualizing teacher education: Supporting case-based instructional problem solving on the World Wide Web In L. PytlikZillig, M. Bodvarsson & R. Bruning (Eds.), Technology-based education: Bringing researchers and practitioners together (pp. 21-38). Greenwich, CT: Information Age Publishing.

[14] Hmelo-Silver, C. E., Derry, S. J., Bitterman, A., & Hatrak, N. (2009). Targeting Transfer in a STELLAR PBL Course for Pre-Service Teachers. Interdisciplinary Journal of Problem-based Learning, 3(2).

[15] Scardamalia, M., Bereiter, C., Mclean, R. S., Swallow, J., & Woodruff, E. (1989). Computer-supported intentional learning environments. Journal of educational computing research, 5(1), 51-68.

[16] Guzdial, M., Hmelo, C., Hübscher, R., Nagel, K., Newstetter, W., Puntambekar, S., ... & Kolodner, J. L. (1997, December). Integrating and guiding collaboration: Lessons learned in computer-supported collaborative learning research at Georgia Tech. In Proceedings of the 2nd international conference on Computer support for collaborative learning (pp. 95-105). International Society of the Learning Sciences.

[17] Suebnukarn, S., & Haddawy, P. (2007). COMET: A collaborative tutoring system for medical problem-based learning. Intelligent Systems, IEEE, 22(4), 70-77.

[18] Ali, Z., & Samaka, M. (2013, March). ePBL: Design and implementation of a Problem-based Learning environment. In Global Engineering Education Conference (EDUCON), 2013 IEEE (pp. 1209-1216). IEEE.

[19] Miao, Y., Samaka, M., Impagliazzo, J., Wang, Y., Wang, D., & Hoppe, U. (2013). A Model-driven Approach to the development of a PBL Script Editor. Global Chinese Conference on Computers in Education (GCCCE).

[20] Cápay, M., & Tomanová, J. (2010). Enhancing the quality of administration, teaching and testing of computer science using learning management system. WSEAS Transactions on Information Science and Applications, 7(9), 1126-1136.

[21] Hajjar, S. T. E. (2014) An Empirical Study about the Influence of Moodle on the Teaching–Learning Process at Higher Institutions. Advances in Educational Technologies, 182.

[22] Nunnally, J.C. (1978). Psychometric Theory, (2nd ed.). New York: McGraw-Hill.