USING DSML IN MOODLE CONFIGURATION TO SUPPORT PBL-PEDAGOGY

By
KHULOOD KHALIL AL-DOUS

MASTER THESIS

Submitted to Qatar University in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE IN COMPUTING

Department of Computer Science and Engineering
College of Engineering

QATAR UNIVERSITY
Doha - Qatar
April 2014

Supervised by:
Dr. Mohammed Samaka
DECLARATION

“To the best of my knowledge, the thesis contains no material previously published or written by another person or institution, except where due reference is made in the text of the thesis. The thesis contains no material which has been accepted for the award of any other degree in any university or other institution.”

Student’s name: Khulood Khalil Al-Dous

Student’s signature:

Date:

“To the best of my/our knowledge, the thesis conforms the requirements of Qatar University, and I endorse this thesis for examination.”

Supervisor’s name: Dr. Mohammed Samaka

Supervisor’s signature:

Date:
# TABLE OF CONTENTS

DECLARATION .................................................................................................................. iii
TABLE OF CONTENTS ......................................................................................................... iv
LIST OF FIGURES .............................................................................................................. vi
LIST OF TABLES ................................................................................................................ vii
EXECUTIVE SUMMARY ....................................................................................................... ix
ACKNOWLEDGEMENTS ......................................................................................................... x

## Chapter 1: Introduction .................................................................................................. 1

1.1. Problem ......................................................................................................................... 1
1.2. Scope and objectives ...................................................................................................... 2
1.3. Significance .................................................................................................................... 2
1.4. Document Overview .................................................................................................... 2

## Chapter 2: Background and Literature Review ............................................................... 3

2.1. Problem-based Learning (PBL) .................................................................................... 3

  2.1.1. Origin ....................................................................................................................... 3

  2.1.2. PBL-Pedagogy ........................................................................................................ 3

  2.1.3. Scenario of PBL ....................................................................................................... 3

  2.1.4. Characteristics of PBL ............................................................................................ 4

  2.1.5. Differences between PBL and other task-centered instructional strategies ........ 5

  2.1.6. PBL models ............................................................................................................. 5

  2.1.7. Current Challenges of PBL .................................................................................... 8

2.2. Online PBL ................................................................................................................... 8

  2.2.1. What is online learning? .......................................................................................... 9

  2.2.2. Why use PBL online? ............................................................................................. 9

  2.2.3. Online PBL environments ..................................................................................... 9

2.3. PBL Scripting language and PBL scripts .................................................................. 12

  2.3.1. Graphical PBL Script Editor .................................................................................. 13

2.4. Learning Management Systems ................................................................................. 15

2.5. State Of The Art Development .................................................................................. 16

## Chapter 3: Methodology of Building the PBL script editor and PBL player .................. 17

3.1. Model-driven Approach ............................................................................................. 17

3.2. Design of PBL script editor in Moodle ...................................................................... 19

  3.2.1. Design a new PBL lesson plan ................................................................................ 19
3.2.2. Constructing the PBL script editor using Moodle existing functions .......... 20
3.2.3. Import a PBL script into the PBL script editor ........................................ 22
3.3. Design of PBL Player in Moodle ................................................................. 23
  3.3.1. Handling the execution of PBL lesson plan by Teacher ............................... 23
  3.3.2. Students work on a PBL problem ............................................................ 23
Chapter 4: Implementation .................................................................................. 24
  4.1. Implementation of Moodle’s PBL Script Editor .............................................. 24
    4.1.1. Moodle’s plugin structure ........................................................................ 24
    4.1.2. Implementing Moodle’s PBL lesson plan plugin ..................................... 26
    4.1.3. PBL meta-model ...................................................................................... 26
    4.1.4. Developing the PBL meta-meta model ..................................................... 28
    4.1.5. Creating a PBL lesson plan ..................................................................... 28
  4.2. Implementation of Moodle’s PBL Player ....................................................... 33
    4.2.1. Handling the execution of a lesson plan by Teacher ................................. 33
    4.2.2. Students work on a PBL problem ........................................................... 34
Chapter 5: Evaluation .......................................................................................... 38
  5.1. Evaluation of Moodle’s PBL script editor by teachers .................................... 38
    5.1.1. Evaluation Description ............................................................................ 38
    5.1.2. Evaluation Results .................................................................................... 39
  5.2. Evaluation of Moodle’s PBL Player by students .......................................... 42
    5.2.1. Evaluation Description ............................................................................ 42
    5.2.2. Evaluation Results .................................................................................... 43
Chapter 6: Conclusion and Future Work .............................................................. 47
  6.1. Conclusion .................................................................................................... 47
  6.2. Challenges .................................................................................................... 47
  6.3. Future Work .................................................................................................. 47
Appendix A: Creating a PBL lesson Plan using Moodle’s PBL script editor .......... 54
  A.1. Example of PBL lesson plan description ..................................................... 54
  A.2. User guide to create a PBL lesson plan using Moodle’s PBL script editor ...... 56
Appendix B: Evaluation surveys ......................................................................... 70
  B.1. Survey of evaluating Moodle’s PBL script editor by teachers ...................... 70
  B.2. Survey of evaluating Moodle’s PBL player by students .............................. 73
LIST OF FIGURES

Figure 2.1: Graphical PBL script editor .......................................................... 14
Figure3.1: System architecture ..................................................................... 19
Figure 3.2: Moodle Activities ..................................................................... 21
Figure 3.3: Mapping PBL tools into Moodle tools .................................... 21
Figure 3.4: Moodle Resources ................................................................. 22
Figure 4.1: File structure of Moodle’s plugin ........................................... 25
Figure 4.2: Example of Moodle capabilities defined in access.php file ........ 25
Figure 4.3: Part of language file of a “chat” plugin in Moodle .................... 25
Figure 4.4: Moodle activities list with the new PBL lesson plan plugin ........ 26
Figure 4.5: Part of the PBL meta-model phases JSON file .......................... 27
Figure 4.6: List of phases types in a PBL lesson plan from the PBL meta-model 27
Figure 4.7: List of activities types in “problem definition” phase from the PBL meta-model 27
Figure 4.8: PBL lesson plan structure ...................................................... 29
Figure 4.9: List of Tool Types for “Problem definition” phase ...................... 31
Figure 4.10: A PBL lesson plan created within the developed PBL script editor in Moodle 33
Figure 4.11: Force open and close icons in runtime environment ............... 34
Figure 4.12: Open and close icons in runtime environment ....................... 34
Figure 4.13: Activities and phases possible statuses ................................. 34
Figure 4.14: List of group members for students with the current logged in student highlighted ...... 35
Figure 4.15: An open activity workspace for students ................................ 36
Figure 4.16: Artifact submission status ................................................... 36
Figure 4.17: Edit my submission page ....................................................... 37
Figure 5.1: Pie chart of Male and Female teachers’ percentage ..................... 41
Figure 5.2: Scatter chart of teachers gender and ease of use mean .............. 41
Figure 5.3: Scatter chart of teachers prior PBL knowledge(x-axis) and ease of use mean (y-axis) .... 41
Figure 5.4: Scatter chart of teachers computer literacy mean(x-axis) and ease of use mean(y-axis) ..... 41
Figure 5.5: Pie chart of Male and Female students’ percentage ................. 45
Figure 5.6: Scatter chart of students gender and ease of use mean .............. 45
Figure 5.7: Scatter chart of students prior PBL knowledge (x-axis) and ease of use mean (y-axis) .... 45
Figure 5.8: Scatter chart of students computer literacy mean(x-axis) and ease of use mean(y-axis) ..... 45
Figure A.1: Moodle’s Home page for a teacher ........................................ 56
Figure A.2: Web development course page .............................................. 57
Figure A.3: Create a new PBL lesson plan .............................................. 57
Figure A.4: Form of adding new PBL lesson plan in topic 1 of the course .... 58
Figure A.5: PBL lesson plan added into topic 1 course web development .... 58
Figure A.6: Empty PBL lesson plan in PBL script editor ......................... 59
Figure A.7: Upload area to import an existing PBL lesson plan .................. 59
Figure A.8: Form of creating a new phase ............................................. 60
Figure A.9: Created 4 phases of targeted PBL lesson plan ....................... 61
Figure A.10: Form of adding new activity ............................................. 62
Figure A.11: The PBL lesson plan after creating all needed activities .......... 63
Figure A.12: Form of creating new artifact ............................................ 64
Figure A.13: The PBL lesson plan after creating needed two artifacts.................................65
Figure A.14: Form of creating new tool .............................................................................66
Figure A.15: The PBL lesson plan after creating needed tools.............................................67
Figure A.16: Form of creating new resource .....................................................................68
Figure A.17: Phase “problem definition” after adding “problem trigger” resource ...............68
Figure A.18: Complete PBL lesson plan of the previously defined PBL lesson plan ...........69
Figure A.19: Export a PBL lesson plan into a PBL script ......................................................69
LIST OF TABLES

Table 2.1: A PBL Scenario ........................................................................................................... 3
Table 2.2: Barrows PBL model .................................................................................................. 6
Table 2.3: Maastricht “seven jump” PBL model ....................................................................... 6
Table 2.4: Mills five-stage PBL model ....................................................................................... 7
Table 5.1: Survey results for evaluating the PBL script editor ease of use ............................ 39
Table 5.2: Results of evaluating the PBL player ease of use ..................................................... 43
EXECUTIVE SUMMARY

Currently, the changes in the economic and technological fields lead the world to experience a huge transition from the post-industrial economy to knowledge economy. This transition transformed professional life and increased their required skills including dealing with increasing internationalization, using information technology and working within groups. Such transition affected the expectations of high education graduates where they should have convinced knowledge base beside the skills of solving, analyzing, synthesizing, coaching, leading, presenting, and evaluating real-life problems.

Reaching such expectations in the 21st century has raised the importance of integrating the knowledge with real-life problems by developing and implementing instructional practices. Generally, students have huge amount of inert knowledge observed by the traditional subject-based learning approach in which students have to learn facts passively. However, they are not able to use such knowledge in solving real-life problems, since they do not experience real uses of its context.

In the recent decade one problem-driven approach that made inroads into engineering and science education is called problem-based learning (PBL). This approach provides students a guided experience in learning through solving complex, real-life problems. A successful implementation of online (and hybrid) PBL lesson requires at the first step a well-designed PBL lesson plan from teachers.

This study aim is to support teachers to design and deliver PBL lesson plans in an easy, cost-effective, flexible, interoperable, and reusable manner. The aim is achieved by extending a pedagogy-generic Learning Management System (LMS) called Moodle to support PBL-pedagogy. That is, a PBL script editor and player were developed inside Moodle to facilitate teachers in designing and delivering PBL lesson plans to students. The facilitation provided according to an adopted PBL scripting language which is a domain specific modeling language (DSML). This study applied the Model Driven Approach (MDA) as the development method to transform a DSML into Moodle platform. The designed PBL lesson plans are delivered for both students and teachers through a developed Moodle’s PBL player. Both Moodle’s PBL script editor and Moodle’s PBL player were evaluated and the results indicated the high usefulness of them.

**Keywords:** problem-based learning (PBL), Learning Management System (LMS), domain specific modeling language (DSML), PBL-pedagogy, Moodle, Model Driven Approach (MDA).
ACKNOWLEDGEMENTS

My sincerest appreciation goes to my supervisor Dr. Mohammed Samaka for his patience, advice, guidance throughout my study work.

I would like to thank prof. John Impagliazzo and Dr. Yongwu Miao for their constructive feedback.

Special thanks go to all teachers and students participated in the study evaluation sessions conducted at Qatar University for their valuable comments and feedback.

I am also deeply grateful to Prof. Ali Mohammed Jaoua for his invaluable help.

Finally, the deepest thanks are expressed to my husband, my parents and my family for their endless confidence, encouragement, and patience.

Khulood Khalil Al-Dous

April 24, 2014
Doha-Qatar
Chapter 1: 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 don’t experience a real use of its context.

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 [2].

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” [3]. Another one is “the conception of knowledge, understanding and education that encourages open-minded, reflective, critical and active learning” [4]. 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 resolving problems or meeting 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” [5]. The common definition of PBL is using the problem to drive the learning process and asses the outcomes.

1.1. Problem

PBL has wide range of models with different specifications. 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.

1.2. **Scope and objectives**
The scope and objectives of this study are as follows:
- Study and utilize existing PBL scripting language to enable the representation of a wide range of PBL models.
- Design and implement a PBL script editor to facilitate the design of PBL lesson plans for teachers.
- Design and implement a PBL player to simulate the execution of PBL lesson plans for students.
- Evaluating the usefulness of the PBL script editor for teachers.
- Evaluating the usefulness of the PBL player for students.

1.3. **Significance**
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.

1.4. **Document Overview**
This study consists of the following six chapters:

**Chapter 1: Introduction**
It contains the problem statement, scope and objectives of this study.

**Chapter 2: Background and Literature Review**
It presents the knowledge and ideas that have been established on the topic and their relative strengths and weaknesses. Also, highlights the state of the art development involved in this study.

**Chapter 3: Methodology of Building the PBL script editor and PBL player**
It explains the study approach to solve the research problem as posed in section 1.1, and the design of both PBL script editor and PBL player.

**Chapter 4: Implementation**
It shows the implementation specifications for both the PBL script editor and PBL player.

**Chapter 5: Evaluation**
Discuss the evaluation methodology applied to evaluate the usefulness of the developed PBL script editor and PBL player.

**Chapter 6: Conclusions and Future Work**
Present the research findings, challenges, and the future work that would enrich this study.
Chapter 2: Background and Literature Review

2.1. Problem-based Learning (PBL)

2.1.1. Origin
Problem-based learning was originally developed in the late 1960s at McMaster University medical school in Canada to help students integrate basic science and clinical knowledge, as well as to develop clinical reasoning and lifelong learning skills [6]. After that, many other medical schools adopted PBL such as Maastricht University in the Netherlands, the University of Newcastle in Australia, Harvard University and South Illinois University [7]. As PBL provides a structured framework of collaborative learning more and more educators of various levels and disciplines are interested in it. PBL has rapidly been adopted into other professional fields rather than medicine including business, nursing, education, and engineering [8,9].

2.1.2. PBL-Pedagogy
Generally, here in Qatar and around the world the most common way of teaching is the traditional teacher-centric methodology [10]. However, the education cognitive science research shows that students who construct their own knowledge by themselves can comprehend, retain and apply their knowledge better than those who receive knowledge directly by the teacher [11]. In some education fields, such as medical and engineering, problem solving skills are highly required where students need to apply their knowledge directly to solve real-world problems. However, the teacher-centric method does not provide student with the needed skills. The rapidly changing world in engineering and medical science raised the need of other learning methodology. Problem-based learning is a student-centric teaching methodology where students acquire knowledge and skills and assess their learning through solving a problem in small groups and the teacher works as a facilitator. Hence, problem-based learning promotes “active, constructive, contextual, co-operative, and goal-directed learning” [12]. A previous study conducted to analyze the effectiveness of PBL on students, concluded that there is a robust positive effect of PBL on both the learning process and the learning outcomes of students [13].

2.1.3. Scenario of PBL
The PBL complete process can be described in a basic scenario that helps more in understanding and analyzing PBL characteristics. A PBL scenario involves a class of students divided by the teacher into groups of five to six that they perform steps as described in Table 2.1.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Step 1** | Meeting and identifying the problem | 1. The teacher presents a scenario to the students in the classroom.  
2. All students in the class read additional materials for better understanding of the problem.  
3. All students in groups identify the problem and write an initial problem statement individually. |
<table>
<thead>
<tr>
<th>Step 2</th>
<th>Analyzing the problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>All students (the whole class) are guided by the teacher to define and formulate the problem statement collaboratively.</td>
</tr>
<tr>
<td>2.</td>
<td>All students in each group brainstorm and organize learning issues.</td>
</tr>
<tr>
<td>3.</td>
<td>All students in each group identify what is needed to know.</td>
</tr>
<tr>
<td>4.</td>
<td>All groups in the class discuss and define a list of need-to-know issues as the group learning need.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Making a work plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>All students in each group brainstorm and propose learning tasks according to the group learning need.</td>
</tr>
<tr>
<td>2.</td>
<td>All students in each group define a work plan and allocate the tasks.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Investigating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>All students in the class individually collect information according to the work plan.</td>
</tr>
<tr>
<td>2.</td>
<td>All groups in the class have a meeting once a week (e.g., Monday afternoon) to report the work status and the acquired knowledge.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Applying knowledge to solve the problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>All students in each group individually generate a solution to the problem.</td>
</tr>
<tr>
<td>2.</td>
<td>Every student comments on three solutions generated by their group members and give feedback to the owners of the solutions.</td>
</tr>
<tr>
<td>3.</td>
<td>Students in groups synthesize their solutions and each group creates a solution to the problem.</td>
</tr>
<tr>
<td>4.</td>
<td>The teacher guides all students in the class to comment on each group solution.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Evaluating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>All groups in the class improve the group's solution according to the comments.</td>
</tr>
<tr>
<td>2.</td>
<td>All groups in the class identify additional issues that they need to know for better solving the problem.</td>
</tr>
<tr>
<td>3.</td>
<td>Every student in the class writes a reflection about the acquired knowledge, the group, and the learning process.</td>
</tr>
<tr>
<td>4.</td>
<td>Every student in groups comments on the reflection of their group members and give feedback to each other.</td>
</tr>
</tbody>
</table>

2.1.4. Characteristics of PBL
Howard Barrows, one of the PBL inventors, has defined six characteristics of the PBL approach [14]. This include student-centered learning, learning occurring in small students groups, teachers guide students rather than teach them, a PBL problem forms the basis for the organized focus of groups that stimulate learning, the PBL problem becomes a development vehicle for
problem solving skills to stimulate the cognitive process, and new knowledge is obtained through self-directed learning (SDL).

PBL scaffolds students in being gradually responsible of their own learning by applying a student-centered approach. According to this, the PBL teacher play the role of facilitators, hence PBL promotes self-directed learning. PBL is problem focused where students begin learning by addressing ill-structured problem where all learning activities are anchored to a real-world problem that students might face in their future careers. As the problem is ill-structured, insufficient information are given to students where they have to identify what they need to learn in order to solve the problem. Moreover, PBL involves group collaboration where students are divided into small groups and that encourage them to share ideas.

According to Barrows PBL definition, part of it is that “there is PBL process which commonly used as a systematic approach to solve problems” [15]. That is the PBL problem becomes a development vehicle for problem solving skills to stimulate the PBL cognitive process. In other words, solving a PBL problem is directed by a PBL lesson plan which is created by the teacher and followed by students.

2.1.5. Differences between PBL and other task-centered instructional strategies

According to Jan Schwartz [16], the different characteristics of PBL make it distinct from other inquiry-based methods including project-based learning and case-based learning in different aspects. Jan Schwartz appealed that the terms problem and project are used interchangeably as a synonyms where the difference between them is blurred. In fact, both problem and project based learning apply the student-centered approach and use authentic, real world problems. However, Jan Schwartz claimed that project-based learning focuses more on the end products which drives and shape the learning process, while problem-based learning focuses more on inquiry rather than the end product, consequently having an end product is not mandatory [16]. That is an end product of a problem-based learning might be a simple group report. Also, he indicated that problem-based learning focuses more on solving a problem by “decision making” and having a decision as a goal. Conversely, project based learning focuses more on “product making” and having the product as a goal.

Furthermore, Jan Schwartz differentiate PBL from other case-based learning methods in that PBL problem is presented first before students receive any instruction where all the learning arises out of the problem and anchored to the problem. However, most case-based learning approaches present cases for students after the learning subject is covered as a testing of the students understanding or as an example. In other words, Barrows claimed that PBL affords greater focus on inquiry and developing the metacognitive skills associated with problem solving [6].

2.1.6. PBL models

The PBL scenario described in section 2.1.3 can be regarded as a specific model to conduct PBL process. In fact, many PBL models currently exist with variations in implementations; in this study three models are presented. Generally, the model developed by Howard Barrows for use in medical school setting is conceived as the most institutionalized or general form. The following (Table 2.2) is the description of PBL process in Barrow’s model [17]:
### Table 2.2: Barrows PBL model

<table>
<thead>
<tr>
<th>Steps</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Explore problems</td>
<td>To start with PBL process, students have to deal with a complex and real-world problem that is usually open-ended.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Identify learning issues</td>
<td>Students observe and try to describe the subject of their observation to identify learning issues.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Solve the problems with existing knowledge</td>
<td>Students practice in dealing with problems using their prior knowledge and experience in a logical, analytical and scientific manner.</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td>Identify the learning needs</td>
<td>Students may need to acquire some other knowledge and they identify the learning needs.</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Set learning goals and allocate tasks</td>
<td>After identifying the learning needs, students are encouraged to set learning goals and allocate tasks.</td>
</tr>
<tr>
<td><strong>Step 6</strong></td>
<td>Study individually</td>
<td>Students work independently to solve the formulated learning goals or problems. Students search for relevant information and transform the information in accordance to the problem-solving process.</td>
</tr>
<tr>
<td><strong>Step 7</strong></td>
<td>Share and teach</td>
<td>Students work together to exchange their experience in searching and transforming the information. Students discuss, compare and clarify the relevance of the discovered information in regard to the initial and related problem(s).</td>
</tr>
<tr>
<td><strong>Step 8</strong></td>
<td>Assess and reflect on the whole process</td>
<td>Students try to integrate the knowledge acquired into a comprehensive explanation and reflect on what has been learned and the process of learning.</td>
</tr>
</tbody>
</table>

Another popular PBL model is the Maastricht “seven jump” process, which consists of seven steps. The Maastricht “seven jump” process is clearly described by Woods in [18] as follows in Table 2.3:

### Table 2.3: Maastricht “seven jump” PBL model

<table>
<thead>
<tr>
<th>Steps</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Present a real-world scenario to the students</td>
<td>Students identify and clarify unfamiliar terms presented in the scenario. Scribe lists those that remain unexplained after discussion.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Define the problem(s) to be discussed</td>
<td>Students may have different views on the issues, but all should be considered; scribe records a list of agreed.</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Brainstorming session to discuss the problem(s)</td>
<td>Suggesting possible explanations on basis of prior knowledge; students draw on each other's knowledge and identify areas of incomplete knowledge; scribe</td>
</tr>
</tbody>
</table>
records all discussion.

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Review steps 2 and 3 and arrange explanations into tentative solutions</th>
<th>Scribe organizes the explanations and restructures if necessary.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 5</td>
<td>Formulate learning objectives</td>
<td>Group reaches consensus on the learning objectives; teacher ensures learning objectives are focused, achievable, comprehensive, and appropriate.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Private Study</td>
<td>All students gather information related to each learning objective</td>
</tr>
<tr>
<td>Step 7</td>
<td>Group shares results of private study</td>
<td>Students identify their learning resources and share their results; teacher checks learning and may assess the group.</td>
</tr>
</tbody>
</table>

Another PBL model was suggested by Mills [19] which consists of five-stage process described by Milles as in Table 2.4.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Definition</td>
<td>1. Appoint chairperson and note taker. Discuss first reactions to trigger provided by teacher&lt;br&gt;2. What sense does the group make of the trigger?&lt;br&gt;3. What possible research problems lead from the trigger? List them</td>
</tr>
<tr>
<td>Step 2</td>
<td>Analysis</td>
<td>1. Brainstorm these possible research problems&lt;br&gt;2. What explanations or interpretations are there in the group about these problems?&lt;br&gt;3. Which explanation/interpretations seem most useful and why?</td>
</tr>
<tr>
<td>Step 3</td>
<td>Research aims</td>
<td>1. Formulate the key research problem hypothesis for investigation.&lt;br&gt;2. What further knowledge does the group need to explore this problem?&lt;br&gt;3. Define three specific research tasks to be completed&lt;br&gt;4. Agree on how the group will work together during the week such as email contact</td>
</tr>
<tr>
<td>Step 4</td>
<td>Research</td>
<td>1. Acquire knowledge in relation to research questions&lt;br&gt;2. Group or individual research over the week, limited to 3 hours.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Synthesis</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>Review the newly acquired knowledge within the group</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Pool findings, do they help on understanding the research problem?</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Final group response to the trigger</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Reflections on the learning process</td>
<td></td>
</tr>
</tbody>
</table>

The models variations can be categorized into two levels: process level and phase level. As we observed, the process of PBL scenario consists of 6 steps. Woods' model is made up of 8 steps. Maastricht' model is, as the name indicated, “seven jumps”. Mills' model consists of 5 steps. In addition, the corresponding phases in different models are defined slightly different, so that they have different titles and different focus. The activities performed in corresponding phases of different PBL models may be different. Therefore, it is required to support different PBL models and their variations within a PBL online system. PBL models forms the structure of PBL lesson plans.

2.1.7. Current Challenges of PBL
Despite the successfuss of PBL in integrating knowledge with real-life problems, there are few numbers of 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 [20]. 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. Applying such transitions would require various kinds of technical support. As a result, in this study technological solutions were adopted to address these hurdles.

2.2. Online PBL
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. In this context, the term ‘blended PBL’ is used which indicate that students learn through the combination of both online and face-to-face instruction. For instance, students use web-based materials, such as text and videos, to learn. In blended PBL, the online PBL environment tend to guide students through different steps of a PBL process, while at one level the use of the online PBL is student led in which the material provided support the learning they undertake in face-to-face PBL environment [21].
2.2.1. What is online learning?
Online learning means students learn over the Internet anytime and anywhere using computer based devices. Students can use different services through the Internet including e-mail, electronic whiteboard, online chat rooms, and discussion board. Online learning helps in enhancing and facilitating the learning environment.

2.2.2. Why use PBL online?
There are many features of online environments that are favorable for PBL. One feature is computer-mediated communication (CMC), which supports online learning by providing students more time to analyze and reflect on online content [22]. Besides that, the permanence of online discourse fosters students’ reflection in online discussions [23]. CMC facilitates reflection and reasoning of students which is critical in PBL process. In consequence, CMC provides students more comfortable environment and discussion opportunities. Especially for students who are shy in participating within a face-to-face discussion. In fact, previous studies have shown that shy students tend to participate more in online environment rather than in face-to-face environment [23]. Regardless of the students’ personalities, online learning would enable each student to participate actively in discussions which may foster generating more ideas or multiple perspectives in problem solving and that match the PBL characteristics.

In addition, students can control their own learning in a more decentralized and constructivist environment within online learning. Hence, the role of the teacher becomes a facilitator instead of being the main information source [24]. This feature is consistent with the characteristic of PBL in which students should apply the student-centered approach. Promoting interaction and collaboration between teacher and students is another essential feature of online learning [25]. As well as engaging students in high thinking level through active and interactive learning [26]. Such feature is appropriate for PBL, since PBL depends on collaborative learning. Moreover, online environments are rich of different type of resources that students can access easily [23]. PBL highly benefits from rich resources online environment feature, in which students can search for needed knowledge through self-directed learning.

Generally, PBL is structured to be held within small groups of students. That is the process of scaling up PBL into classes with larger number of students, implies significant implementation challenges. As a result, online PBL environment would grants a hybrid model that mixes online and face-to-face PBL in order to facilitate scaling up PBL into classes with larger number of students.

2.2.3. Online PBL environments
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), Computer Supported Intentional Learning Environments (CSILE), Web-Scaffold Multi-user Integrated Learning Environment (Web-SMILE), Collaborative Medical Tutor (COMET), and electronic Problem-Based Learning (e-PBL).

2.2.3.1. STELLAR
STELLAR, which stands for Socio-Technical Environment for Learning and Learning-Activity Research, is a system that contains tools to build and manage courses that systematically integrate collaborative design with text and video resources. STELLAR can be adapted to
different contexts and teaching subjects where communication tools are provided to support students’ collaboration. It was developed to support online, hybrid PBL courses, and it allows the implementation of PBL in larger classes. According to the scaffolding capabilities of STELLAR, small number of facilitators can distribute their attention among multiple small groups. STELLAR facilitates video uploading and presentation in multiple instructional formats. Also, it supports the development of hypertext environments. In addition, STELLAR allows facilitators to access all students’ works and to communicate with students whether individually or in groups [27].

A previous study held by Cindy, et al found that in a hybrid PBL course using STELLAR, students learned more about the targeted course concepts rather than in a traditional face-to-face PBL course. They concluded that a hybrid CSCL approach can be used to scaffold problem-based learning and foster deep understanding [28].

The STELLAR system suffers the complexity of authoring activities structures. STELLAR authoring system was never built to be usable by non-technical users. In this case, the teacher needs to structure the content of a PBL course manually and then give it to a technical person to implement it in STELLAR. Further, this system supports only one PBL model. Many technical issues appeared within STELLAR system, as the platform that was built on become too old. Since the funding of this system ran out, the system is no longer supported.

2.2.3.2. CSILE

The Computer Supported Intentional Learning Environment (CSILE) is basically a discussion board where students can create new topics, comment on different topics and comment on each other’s comments. The key feature of this system is that students have to label their contribution to the problem. For instance, a set of available labels include what I need to know and what I don’t know. Such labels serve to mentor students within a problem in a way analogous to the facilitator in the PBL environment [23,29].

An interesting aspect of the CSILE system is that it has a close connection to the knowledge building community model of education and learning[30]. The structure of this model is constructed based on sociological descriptions of creating and refining knowledge within scientific communities. The learning is described as a process of collective knowledge construction, where topics are discussed, elaborated, and continuously refined by a group of students. The shared environment of the group individuals’ actions compels the adaptation between students. Adaptation is a positive contribution in rising up the knowledge building factor.

In addition, scaffolding is an important feature of CSILE in which the system has a built-in structure that guides students to focus on particular knowledge building aspects of their discussions. Some examples of these scaffolds are automatic system notification to encourage students to identify problem statements, develop theories, identify the difficulties in understanding certain issues, attach new information to a topic, and conclude what they have learned. The actual design of Scaffolds structure the students discourse in order to replicate the work of a scientific research community [31].

Although, CSILE is an important system to support computer based collaboration, it is limited to support only one PBL model, which limits the flexibility of creating PBL lesson plan. The negative impacts of this limitation occur at those disjuncture points between the model and the situation or problem. In fact, a PBL problem details should be respected and taken into account when adapting a model to a problem. Beside that only one collaboration tool is
supported in CSILE system which is a discussion board, while there are other important tools to build an online PBL environment, such as online chat room and brainstorming tool. Also, CSILE has limited productive use of the technology; hence more focus on community building and on the organization of learning activities is required [30]. Even though, few number of CSILE system copies are still "float around" it is probably a dead thing [32].

2.2.3.3. Web-SMILE
Web-SMILE system has similar design goal to CSILE which is scaffolding the leaning process. However, Web-SMILE focuses on the tools limitation in CSILE and integrates different synchronous and asynchronous tools into one environment which can advance students’ collaborations. The formulation of the process model used in Web-SMILE was based on one PBL model. The process model consists of different goals including analyze the situation, generate learning issues, and carry out research. The goals are annotated with description, pre-conditions to state when they should be started, post-conditions to state when they are achieved, and the tools needed to achieve them. The used PBL model guides the students through proposing goals to be accomplished, actions to be done, tools to be used, and how to use them [33].

Even though, Web-SMILE addressed the limitation of tools support in CSILE system, it still suffers from another crucial problem. This is the lack of supporting different PBL models which limit the support of creating PBL lessons plans. In certain circumstances, CSILE system might be useful, while in other situations or problems it might not be applicable.

2.2.3.4. COMET
COMET is a learning system that supports only medical PBL environment. The design goal of COMET is to deliver an online session with an experience that emulates the live human-tutored medical PBL lesson [34]. In the same time COMET permits students collaborations among disparate locations. The system provides the students a chat pane, medical image pane, hypothesis board and hierarchical medical concept repository. The PBL model constructed in this system supports students in clinical-reasoning skills around practical patient problems. The constructed PBL model consists of three phases which are problem analysis, self-directed study, synthesis and application of newly acquired information [34]. The problem analysis phase includes group discussion, evaluating the patient problem presented, determining the possible underlying dysfunctions and enumerating all possible causal paths that would explain the progression of the patient’s problems. In the second phase self-directed study, students work individually to address all open issues identified in the first phase using different resources including books, laboratories and specialists. In the last phase, which is synthesis and application of newly acquired information, the students work collaboratively to analyze data and wrap up the problem.

In addition, COMET learning system fit only to medical domain, which makes it inapplicable for other domains. It supports only one PBL model that most fit to medical circumstances which limit the flexibility of designing PBL lesson plans.

2.2.3.5. e-PBL
A Learning Management System called Moodle have been extended to support PBL-pedagogy based on Woods’s PBL model, in an experimental environment for PBL called e-PBL [35]. E-PBL environment provides a new plugin in Moodle that present the workflows and activities of
Wood’s model. The development of e-PBL demonstrated the technical feasibility to incorporate PBL-specific support into a pedagogy-generic platform which is Moodle. According to the evaluation of e-PBL, the implementation was successful as a first attempt to support one PBL model in Moodle[35].

E-PBL environment is restricted only to Woods’ PBL model, in which the model is hardcoded within the new added plugin in Moodle. That means changing the supported PBL model would require a lot of hardcoding to be done by a technical person. As a result, e-PBL provides insufficient support for ordinary teacher to develop and execute different PBL models.

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.

2.3. PBL Scripting language and PBL scripts
The syntax of script was described well by Dillenbourg in the following way: “A script is a story or scenario that the students and teachers have to play, as actors play a movie script. Most scripts are sequential: students go through a linear sequence of phases. Some scripts are defined in an iterative way, but from the student’s point of view, they are run as a linear sequence” [36]. A script phase identifies how students should collaborate to solve a problem.

A collaboration script consists of an instructions set that is designed to structure the tasks within a collaborative process. Also, it handles the communication between team members by specifying team composition, roles differentiation, activity steps and modes of communication [37]. The necessity of collaboration script exists according to the need of designing productive teamwork interactions. Moreover, it was used within the design of both face-to-face collaboration [38] and computer supported collaboration learning (CSCL) [36].

These scripts guide the human thinking; however a computer can be used to coordinate human script by using computer supported collaboration learning scripts (CSCL scripts). CSCL scripts can be supported using two technical approaches either by using scripting languages or hardcoding scripts in a learning environment. Using scripting languages, the computer can understand and interpret the scripts and arrange the learning environment at runtime according to the scripts.
Instructional Management Standards Learning Design (IMS LD) can be used to support CSCL scripts. IMS LD is an open specification that used to describe learning scenarios [39]. The goal of IMS LD is to represent the “learning design” as “units of learning (UoL)” in a semantic, formal and machine interpretable way [40]. The learning design is defined as “the sequence of learning activities that students undertake to attain some learning objectives, including the resources and support mechanisms required to help students to complete these activities” [41]. A wide variety of the current modern pedagogical approaches are supported by IMS LD including active learning, collaborative learning and adaptive learning. IMS LD was developed based on the “Educational Modeling Language” (EML) which was developed by the Open University Nederland (OUNL). The final specification of the first IMS LD version was released on 2003 by the IMS Global Learning Consortium [40]. As claimed in [42], it is possible to describe various PBL scenarios by using abstract and pedagogy-generic terms such as roles, activities, environments, and learning objects. However, an IMS LD authoring tool provides ordinary teachers insufficient means for developing and understanding a PBL script. This is because IMS LD is pedagogy-neutral and its editing tools lack the support for specific PBL scripts modeling components, as they were not originally designed to serve PBL-pedagogy.

PBL scripting language is an 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 [42]. 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 processes. Additionally, it was developed according to the best PBL practices and the well-known PBL models including Barrows model, Mills model and Seven Jumps model. 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 in [42] based on the PBL scripting language.

2.3.1. Graphical PBL Script Editor
The graphical PBL script editor affords PBL practitioners the structure and guidance to create, communicate, customize, and reuse a PBL script. PBL practitioners can transform a PBL script into IMS LD UoL in order to run the designed PBL script. A UoL is executable where a practitioner can play it to scaffold PBL practices in an IMS LD player such as Service Based Learning Design System (SLeD) [43]. The existent Graphical PBL Script Editor consists of four parts as shown in Figure 2.1. Menus of basic function are provided in the top bar, while functions
to edit text are provided in bottom bar. The central area is divided into two parts which are the edit-space on the left, the tool-space on the right. The teacher can specify how many students in the class and divides them into groups via using the organizational role tool from the tool-space. Teachers can create a PBL script by dragging and dropping nodes from the tool-space into the edit-space. First teachers have to define the phases-nodes of the PBL process. Then inside each phase node they can define activities-nodes, artifacts-nodes, resources-nodes and actors-nodes. The sequence of running phases or activities can be identified by drawing arrow-edge between related nodes. When the teacher defines a new phase, the system provide a list of recommended phases types based on the PBL scripting language. Also, the same recommended type list is provided when creating other nodes including resource, activity, or artifact. For example, in the phase problem-definition, the teacher can choose one of recommended activities type such as identify-problem, formulate-problem and commenting-on, which are usually performed in this phase.

![Figure 2.1: Graphical PBL script editor](image)

The Graphical PBL script editor and PBL scripting language addressed the limited aspects of the reviewed online PBL environments. This include facilitating teachers in designing a PBL process based on variety number of PBL models and best practices. The Graphical PBL script editor and PBL scripting language was evaluated by Miao, et al where they concluded that “most of the 35 participants involved in the evaluation responded positively on all aspects of the PBL scripting
language and the PBL script editor” [44]. However, using this graphical PBL script editor would require from teachers to first design the PBL process then to generate UoL and to use an IMS LD runtime environment to execute it which is a time and effort consuming for teachers. 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 LMSs and they do not tend to experience or learn new systems. That raised up the challenge of adopting the PBL scripting language to extend existing LMSs in which many teachers and students can benefits from it.

2.4. Learning Management Systems

A Learning Management System (LMS) is a web-based software package used to administer one or more courses to one or more students [45]. The design and implementations of the current existing LMSs facilitate lecture-based learning and none of them efficiently adapted to the PBL context. However, most of the PBL required features and tools such as grouping, computer-mediated communications, and assessments are available within most of the LMSs. Also, these features and tools are frequently improved within each system upgrade.

The development of a comprehensive independent PBL online system separated from an existing LMS includes costly challenges in both the development and maintenance. Additionally, for learning courses that have both PBL and lecture-based activities, this would require from users including students and teachers to run two different systems. Thus, this is insufficient and time consuming for users. As a result, the PBL online system, which includes both the design and run time environment of PBL lesson plans, developed in this study was implemented inside an existing LMS and made use of its existing features and tools.

There are several LMSs existing currently in the market. In this study four platforms were investigated which are Blackboard, Desire2Learn, Moodle and Sakai. These four platforms were selected based on their feasibility and Maturity. They all demonstrate continued adoption by higher education institutions. Also, they have sustained product development and support by the parent company as in (Blackboard, Desire2Learn), or by community development like (Moodle, Sakai). Moreover, their proven track record of sustained and successful usage by similar sized universities is a baseline requirement for being considered in this investigation.

Approximately the four LMSs are having the basic required features needed to create a collaborative learning environment. Except that they are varying from being an open source (Moodle, Sakai) to commercial products (Blackboard, Desire2learn). Also, they differ in the percent of market share where Moodle has the largest market share percent over Sakai. In the other hand, Blackboard leads the market of commercial LMSs. Blackboard is extremely functional in terms of a course management system, however it lacks certain intuitive and flexible approaches that may be more apparent in a socially derived learning management systems such as Moodle [46]. Both Moodle and blackboard are pedagogy-neutral which mean
they were designed to support general collaboration learning environment and not PBL-pedagogy specific environment.

In this study Moodle LMS was selected to be customized for implementing a PBL design and runtime environment. Moodle is an open source system where customizing it is much easier than Blackboard, which is a closed commercial environment. Additionally, Moodle has nearly 67.5 thousand customers representing 60.04 million users, while Blackboard has 4.4 thousand customer representing 20 million users [47]. The wide international usage of Moodle and its continued growth during the last six years have made it the leading open source LMS solution. Moodle is an acronym and stands for (Modular Object-Oriented Dynamic Learning Environment) which is a classical web application, based on the technologies PHP, MySQL, HTML, and JavaScript. For functional extension, Moodle provides a comfortable plugin mechanism.

2.5. 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 this graphical PBL script editor would require from teachers to first design the PBL process. Then to generate UoL and use an IMS LD player to execute it, which is a 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. 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. 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. The PBL characteristics addressed in this study are student centered learning, learning occurring in small students groups and the problem becomes a development vehicle for problem solving skills to stimulate the cognitive process of PBL.
Chapter 3: Methodology of Building the PBL script editor and PBL player

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 script editor based on PBL scripting language.

3.1. Model-driven Approach
A model-driven approach (MDA) is a software development method that is defined by model-driven architecture for information technology system specification. It is defined as “the approach that separates the specification of system functionality from the specification of the implementation of that functionality on a specific technology platform” [48]. Generally, MDA methodology uses models like the primary source for documenting, analyzing, designing, constructing, deploying and maintaining a system [49,50]. Also, it is highly associated to modelling languages design and specification. Furthermore, it is built based on the four-layer modelling architecture [51]. The general modelling architecture of MDA comprises several layers; layer zero that involves the real world objects. At layer one several models of layer zero objects simplification and abstraction are defined. Defining these models based on concepts with higher level of abstraction form layer two meta-model. A meta-model determines the expressiveness level of its models. Accordingly, layer three consists of meta-meta-model which is a higher abstraction level of meta-models. For instance, in PBL circumstances layer zero corresponds to PBL general elements including phase, activity, tool, resource and artifact. While layer one represents a complete PBL model such as Wood’s model (see section 2.1.6). Layer two, however is being developed by the PBL scripting language (see section 2.3) that involves defining a PBL meta-model. Finally, Layer three of the MDA structure represents a meta-meta-model of all learning pedagogies meta-models including PBL-pedagogy.

The PBL scripting language used in this study applied the MDA by collecting the available PBL models and identifying their commonalities and differences. Based on that a PBL meta-model was developed through identifying the main concepts and their relations that are usually used to describe PBL-pedagogy. From this reason the decision was made to adopt the MDA methodology in this study. That is the PBL meta-model formed the base of developing Moodle’s PBL design and runtime environment within this study.

Implementing the MDA methodology has strengthened the development of PBL design and runtime environment in this study. A potential strength of implementing the MDA has increased the flexibility of creating a PBL lesson plan where a PBL lesson plan can be created and maintained based on different PBL models identified in the PBL scripting language. Another strong benefit of using MDA is the comprehensive interoperability framework for defining interconnected systems. For this regard, the MDA methodology used in this study to facilitate the creation of PBL lesson plan based on PBL meta-model, and to convert the PBL lesson plan into
PBL script. A PBL script contains all the design specification of a PBL lesson plan which is written by using the PBL scripting language. Hence, a PBL script facilitates the reusability and sharing of PBL lesson plans between many PBL teachers through sharing it.

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, resources, and tools based on the PBL meta-model. 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 Figure 3.1. 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 where its core is a PBL script editor. The PBL runtime environment used to execute the PBL lesson plans by both teacher and students. The PBL runtime environment core component is called the PBL player.

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 script editor and the PBL player. Additionally, the teacher can instantiate multiple execution of a PBL lesson plan that can be used by both teachers and students.
3.2. Design of PBL script editor in Moodle

In order to extend Moodle platform to design a PBL script editor, a hypothesis of how Moodle is actually designed and structured was considered.

3.2.1. Design a new PBL lesson plan

Designing a PBL lesson plan would require the designer first to think of the phases needed and their sequence. For instance, a PBL lesson plan designer might think of two phases: problem engagement and problem definition. The designer should then think of their running sequence, such that problem engagement phase should run first followed by the problem definition phase. Identifying the phases sequence in the design phase of the PBL lesson plan would be valuable at the runtime environment; hence a phase might not be visible to students unless they complete a previous phase. After thinking about the phases and their sequence, the designer need to think of what activities, artifacts, tools, roles and resources should be included in each phase. That is the designer should think about what activities should happen in each phase, which tool to be used within each activity, what resources -if any- should be reviewed by students to perform each activity, and whether an artifact should be submitted within each activity or not. For example, in problem engagement phase, the teacher may think of an activity, such as “meet the problem”, that has a “problem trigger” resource, and an online chat tool. This way of teacher thinking to create a PBL lesson plan constructed upon the “top-down” approach, in which the designer of the PBL lesson plan will start with top general thoughts (e.g. phase) and then goes down into more specification of these thoughts (e.g. artifact).

Additionally, the bottom-up approach can also be used to construct the thinking of PBL designer, which is the opposite of the top-down approach. In the bottom-up approach, a PBL lesson plan designer starts by the specification of the granular components and ends by the
general or abstract thoughts. Both approaches are valid, however according to the argue of Sodhi and others in [52], who stated that learning designers prefer the top-down approach, Moodle’s PBL script editor was designed based on the top-down approach. Applying the top-down approach supports designers of PBL lesson plans in visualizing the problems at higher level of view where the associated interconnection between PBL elements is clear from the main overview, rather than compartmentalized [53]. PBL lesson plans designers with even no experience in modeling PBL-pedagogy, benefit most from this approach as they might not understand the interconnection between elements until they start with a general overview of the PBL lesson plan [54,52]. Using top-down approach would gradually facilitate PBL designers in translating their thoughts into pedagogically-sound of PBL lesson plan.

3.2.2. Constructing the PBL script editor using Moodle existing functions
Moodle already has predefined roles including teachers and students. Also, Moodle has a full group management function; this function was used to design students’ groups of a PBL lesson plan. Through that, a PBL lesson plan designer can easily divide his existing students within a Moodle course into groups using the existing Moodle function. Hence, no need to worry about creating new roles and groups formulation function within the design and implementation of the PBL script editor.

The PBL meta-model defines all possible elements of a PBL lesson plan which are phases, activities, tools, resources and artifacts. The PBL meta-model predefines the needed tools to perform each activity. For instance, a “discussion” activity can have a tool “discussion board”. The predefined tools in the PBL meta-model then are mapped into the Moodle activities. It is important to distinguish here between PBL activities and Moodle activities, hence Moodle use the word “activity” to identify its available components including communication activities (e.g. chat), resources (e.g. URL) and other plugins (e.g. assignment). The PBL script editor was added as a new plugin into Moodle. Moodle activities can be added into a course as shown in Figure 3.2. Based on the used PBL meta-model, some Moodle activities (e.g. chat, forum) can be used as PBL tools within a PBL lesson plan. A Mapping between the PBL meta-model required tools and Moodle activities was formed as shown in Figure 3.3. Additionally, Moodle activities can be added at course level; however, in the design of the PBL script editor, Moodle activities were adopted within phase level of a PBL lesson plan.
Moreover, the PBL Meta-model defines the PBL resources needed for different PBL activities in a PBL lesson plan. The definition of a resource (e.g. problem trigger) within the PBL meta-model only specifies what the resource is and not the medium type of the resource such as URL or page. On the other hand, in Moodle adding a resource in a course involves defining its medium such as book, page or URL as shown in Figure 3.4. In the design of the Moodle’s PBL script editor both the PBL meta-model resources and Moodle resources were mapped to get use of Moodle predefined resources. The steps of mapping resources were afforded as follows:
Step 1: Creating the resource defined in the PBL meta-model which define “what is the resource”.
Step 2: The PBL script editor Provides PBL designer with options to choose the medium type of the resource according to Moodle predefined resources.

Figure 3.5: Moodle Resources

Extending Moodle to Map the PBL meta-model artifacts into the PBL script editor follows the same way applied in mapping the resources. This is because the PBL meta-model also identify “what is the artifact” (e.g. an artifact could be a “problem statement”) not the medium type of delivering an artifact (e.g. file or online text). It is important for students to know the medium type for submitting required artifacts in a PBL lesson plan. As a result, Moodle’s PBL script editor affords the PBL designer options to choose the artifact submission medium type such as file, online text and wiki.

3.2.3. Import a PBL script into the PBL script editor
Moodle’s PBL script editor was designed to support both interpretability and reusability of PBL lesson plans. This was done by allowing the PBL designer to import/export a PBL lesson plan as PBL script. Hence, a PBL script can be transformed into a PBL lesson plan in Moodle’s PBL script editor and in the same way a PBL lesson plan can be transformed into a PBL script. The PBL script contains a set of script files that include all the design description of a PBL lesson plan written using the PBL scripting language. Accordingly, a teacher can create a PBL lesson plan by either building it within Moodle’s PBL script editor or importing an existing PBL script into Moodle’s PBL script editor. In both cases full editing functionalities are provided to add, edit or delete PBL lesson plan elements within Moodle’s PBL script editor.

A PBL script of a PBL lesson plan is divided into different script files, one script file identifies all phases of the PBL lesson plan, and for each identified phase there is a separate script file that clarifies all its components including activities, artifacts, resources and tools with their relations. Transforming a PBL script into Moodle’s PBL script editor would require the following steps:
Step 1: Creating all the phases according to the PBL script.
Step 2: Creating the activities, resources, tools and artifacts of each phase.
Step 3: Identifying the sequence of phases and activities according to the PBL script.
Step 4: Identifying all the relations between the created PBL lesson plan elements according to the PBL script.

3.3. Design of PBL Player in Moodle
A basic PBL player has been designed in this study only to stimulate the execution of a PBL lesson plan and how a teacher can handle it, while students can follow the identified sequence to resolve a PBL problem. The Moodle’s PBL player was designed for both students and teachers. Teachers would use the PBL player to handle the execution of the designed PBL lesson plan. Students use the PBL player to follow the PBL lesson plan defined process in order to solve a PBL problem.

3.3.1. Handling the execution of PBL lesson plan by Teacher
In the runtime environment the role of the teacher is to facilitate the execution of the PBL lesson plan, give feedback, and comments for students. Moodle’s PBL player was designed for teachers to facilitate handling the status of each phase and activity. Hence, a teacher can open/close any activity or phase at any time. Only a phase with open status can be visible for students. Besides that, the teacher can monitor the students’ interactions within the used tools and also see their submitted artifacts.

3.3.2. Students work on a PBL problem
Moodle’s PBL player was designed for students to allow both group and individual working modes according to the PBL lesson plan specifications. As the teachers make a PBL lesson plan visible for students, they can work to solve the presented problem by following the PBL lesson plan as guidance for their learning process. The students can view the detailed information of a PBL problem as well as the information of only the current open phase beside its required activities to be performed. Additionally, the PBL player allows students to submit the required artifact within an activity using different submission medium types such as file and online text.
Chapter 4: Implementation

In this chapter, the implementations of Moodle’s PBL script editor and Moodle’s PBL player are elaborated.

4.1. Implementation of Moodle’s PBL Script Editor

This study aims at extending Moodle platform to dynamically instantiate a PBL lesson plan, it enables PBL designers and ordinary teachers to develop and deliver online (and hybrid) PBL lesson plans. In this study Moodle’s PBL script editor was developed, which facilitates the design and reusability of PBL lesson plans.

The implementation of Moodle’s PBL script editor first involved creating a new Moodle plugin according to the structure of Moodle plugins. Also, it required studying and analyzing the used PBL scripting language and being able to read and write PBL script files. Moodle’s PBL script editor enables all the editing functions for teachers to design and create PBL lesson plans. Such editing functions including 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.

4.1.1. Moodle’s plugin structure

Moodle was developed using the PHP, with a structured database of MySQL. Both PBL script editor and PBL player was developed within a new plugin in Moodle. Developing a new Moodle’s plugin requires structuring its contents and files following a predefined Moodle structure of plugins. The structure of Moodle’s plugin comprises of different files organized as shown in Figure 4.1. The “db” folder includes three files, the “access.php” file which defines the plugin capabilities. An example of Moodle’s capability is shown in Figure 4.2, where in a “certificate” Moodle’s plugin only Moodle’s users with roles of editing teacher and manager are allowed to create a new instance of this plugin. The “install.xml” file includes the definition of the database structure tables and fields. Besides that, the “upgrade.php” file handles all operations needed to upgrade from one version of Moodle’s plugin to another.

The second folder “lang” includes the definition of the plugin used language. That is a plugin can have different languages such as English, Arabic and French. Each language has to be defined in a separate sub-folder like “en” which presents the English language. The language file, which has the same name as the plugin name, includes a list of all the string needed in the plugin which helps in avoiding hardcoding the strings and make it easier for changes. An example of such file is shown in Figure 4.3 where these defined strings can be used within the plugin code by calling a Moodle’s built in function called “get_string()”. For instance, getting the string of the next session in a chat plugin to be displayed for Moodle’s
users, get_string(‘nextsession’, ‘chat’) should be called.

![Diagram](image)

Figure 4.6: File structure of Moodle’s plugin

```php
$capabilities = array(
    'mod/certificate:addinstance' => array(
        'riskbitmask' => RISK_XSS,
        'captype' => 'write',
        'contextlevel' => CONTEXT_COURSE,
        'archetypes' => array(
            'editingteacher' => CAP_ALLOW,
            'manager' => CAP_ALLOW,
        ),
        'clonepermissionsfrom' => 'moodle/course:manageactivities',
    ),
);
```

Figure 4.7: Example of Moodle capabilities defined in access.php file

```php
$string['modulename_link'] = 'mod/chat/view';
$string['modulenameplural'] = 'Chats';
$string['neverdeletemessages'] = 'Never delete messages';
$string['nextsession'] = 'Next scheduled session';
```

Figure 4.8: Part of language file of a "chat" plugin in Moodle

In addition, the “pix” folder contains all the icons used in a Moodle’s plugin. Beside that the “lib.php” file includes the main functions needed to create, update, or delete an instance of a Moodle’s plugin. Other supporting functions of a Moodle’s plugin can be defined in the “locallib.php” file. Furthermore, the user interface for creating and updating an instance of a Moodle’s plugin is defined in “mod_form.php” file. This includes defining the form elements such as text boxes, checkboxes and select options elements. A Moodle’s plugin can have different versions which should be defined in the “version.php” file. Each version is defined by a number that can be increased for each new version. Increasing the version number of a Moodle’s plugin will notify the user with the new update where the plugin can be upgraded into a new version according to the upgrade changes identified in the “upgrade.php” file.
After a new instance of a Moodle’s plugin is created, it can be viewed by Moodle’s users according to the view elements identified in the “view.php” file. A Moodle’s plugin instance can be viewed by users that have the capabilities to view such instance according to the view capabilities identified in the “access.php” file. For example, the view of a plugin instance can contain the description of the instance entered by the instance creator in the “mod_form.php”. This is the main structure of a Moodle’s plugin where other supporting files can be added to handle more complex plugins.

4.1.2. Implementing Moodle’s PBL lesson plan plugin

The same structure of Moodle’s plugin was followed in creating a new Moodle’s plugin, called PBL lesson plan. Hence, the new plugin contains the implantations of both the PBL script editor and PBL player. Additionally, the PBL lesson plan plugin was installed in Moodle and it appeared in the list of Moodle’s activities where a teacher can create a new instance of it within a Moodle course. Figure 4.4 shows Moodle’s activities list in which the PBL lesson plan plugin is highlighted.

![Figure 4.9: Moodle activities list with the new PBL lesson plan plugin](image)

4.1.3. PBL meta-model

This study used the PBL meta-model developed by Miao, et al [42] to facilitate the design of PBL lesson plans for teachers. The PBL meta-model is defined using JSON syntax and consists of a set of JSON files. One file defines possible phases of a PBL lesson plan and part of it is shown in Figure 4.5. Besides that, PBL elements of a phase are defined in a separate file including activities, tools, resources and artifacts.
Moodle’s PBL script editor accessed the PBL meta-model to get possible phases’ types in a PBL lesson plan as shown in Figure 4.6. The phases’ types list is dynamically instantiated from the PBL meta-model. This list is viewed for a teacher when he/she creates a new phase in the Moodle’s PBL script editor. In addition, Moodle’s PBL script editor provides the teacher with lists of possible types for activities, tools, resources and artifacts when designing a PBL lesson plan. These lists are also instantiated dynamically from the PBL meta-model, which enable the teacher to customize the design of a PBL lesson plan. For example, if the teacher chooses the “problem definition” phase, then he/she can create activities within the chosen phase by getting a list of possible activities types in the “problem definition” phase as shown in Figure 4.7.
4.1.4. Developing the PBL meta-meta model

The PBL meta-meta model was hardcoded in the Moodle’s PBL script editor. This is because the PBL meta-meta model remains constant as only one learning meta-model is supported, which is the PBL meta-model. Also, Moodle’s PBL script editor aimed to support teachers in designing PBL lesson plans, hence, only PBL-pedagogy is supported.

Developing the meta-meta model of the PBL-pedagogy comprises defining all PBL elements attributes and controls. Common defined attributes between PBL elements are type, title, description and goal. One developed element control was the execution sequence where the designer can define the execution order of the phases and activities. The identified execution sequence control was then used in the runtime environment to control the execution of the PBL lesson plan where the students at the beginning of the learning process can only see the first phase and the first activity of that phase. Such sequence control of phases and activities is very useful in Computer Support Learning practices. As students will not be confused by a list of all phases and all their activities, instead they can only view the current running process or phase at a time. This makes the PBL lesson plan much clearer and consistent with its submission time required. That is the runtime environment shows only the needed phase and activity details for students at a time, hence other unneeded phases details are hid from students.

After the PBL meta-meta model attributes and controls were identified based on the PBL meta-model, Moodle’s PBL script editor was developed to address both the PBL meta-meta model and the PBL meta-model.

4.1.5. Creating a PBL lesson plan

Moodle’s PBL script editor supports interpretability where a PBL designer can either develop a new PBL lesson plan and then export it into a PBL script or import an existing PBL script into an existing Moodle course. This would help the teacher in reusing and sharing PBL lesson plans. Also, if a PBL script is generated using the graphical PBL script editor (described in section 2.3.1.1, chapter 2), it can be imported into Moodle’s PBL script editor with the ability to edit. Hence, this satisfies the objectives stated in this study where ordinary teachers can develop and deliver online (and hybrid) PBL lessons plans in a flexible, interoperable, and reusable manner.

The structure of a PBL lesson plan consists of two levels this is according to the PBL scripting language as shown in Figure 4.8. The first level involves defining the phases of the targeted PBL lesson plan. On the other hand, the second level comprises outlining the required activities of each defined phase in the first level. The number of phases and activities vary from one PBL lesson plan to another depending on the complexity of the desired PBL problem to be solved and the teacher’s targeted PBL lesson plan.
According to the PBL lesson plan structure a teacher has to first create a new PBL lesson plan in the anticipated course in Moodle. Many attributes can be defined for the created PBL lesson plan including name, start date, end date, description, grade and the visibility whether to show/hide the created problem for/from students. Besides that, the teacher can choose an existing PBL lesson plan template provided by Moodle’s PBL script editor. One template has been provided for the designer which implements a model of a successful PBL practice provided by IMAS PBL Network [55]. The IMAS model has been created using the Moodle’s PBL script editor and then exported into a PBL script saved on the Moodle’s server as a template for the teachers to use. If the teacher chooses to use an existing PBL template, the PBL lesson plan will automatically be created within the Moodle’s PBL script editor with the ability to enhance and edit. Having such a template enhanced the flexibility of designing a PBL lesson plan using the developed Moodle’s PBL script editor.

In the first level, the designer needs to create the phases needed for his/her desired PBL lesson plan. This can be done simply in the PBL script editor by clicking on the link specified to create a new phase. When creating a new phase the designer can choose the phase type from a dropdown menu that is instantiated from the PBL meta-model such as “problem definition”, “identification of knowledge gap” and “problem resolution”. Hence, this list can be customized by the designer in case the desired phase does not exist in the list. In order to address the customization of the phases types list a “custom phase” option have been added to the list which does not originally appears in the used PBL meta-model. Adding such an option in the developed Moodle’s PBL script editor enhanced the customization of designing a PBL lesson plan.

In addition, the designer can assign a title for the created phase as well as add the description, add the goals, choose maximum grade, choose start and end dates, choose a group mode and choose a run structure for each phase. For the group mode the Moodle predefined
group modes were used which are (no groups, separate groups or visible groups). In no groups mode all the students will work as a whole, while in separate groups mode the students are divided into groups that work separately from each other and cannot see the work of others groups. However, in visible groups’ mode the students work in groups with the ability to see the works of other groups. The run structure options identify the sequence of executing level two activities that will be added inside the phase. There are three options provided for the designer in the run structure which are (sequential, parallel and selective). In sequential run structure, activities will run in sequence one after the other; hence no two activities can run at the same time and the sequence of running should be defined. Nevertheless, a parallel run structure allows more than one phase to run simultaneously. A selective run structure is similar to the sequential run structure in allowing one activity to run at a time; however it is not required to have a predefined sequence for the activities to run. That is any activity can be opened in a time while the others should be closed. The run structure options were provided as constrains to support the implementation of Moodle’s PBL player. Some constrains have been developed in Moodle’s PBL script editor for the phases grades which would also support Moodle’s PBL player. That is the total sum of grades for all identified phases in a PBL lesson plan should not exceed the maximum grade identified for the PBL lesson plan when it was created. However, Moodle’s PBL script editor provides the teacher an option to allow extending the PBL lesson plan maximum grade.

As long as the desired phases in the first level of a PBL lesson plan are created, the teacher needs to create the activities needed for each phase separately to build level two of the PBL lesson plan structure. The same attributes provided when creating a phase are also provided when creating an activity, but without the run structure attribute. Instead a completion condition attribute is provided when creating a new activity. In the completion condition attribute the designer can choose whether the complete condition is time limited or user control. Choosing a time limited option will force the open/close status of an activity to the specified start and end date of that activity. On the other hand, an activity with user control complete condition requires the teacher interaction to open/close the activity; no time will be forced in this case. This attribute is also provided to support the runtime environment of the PBL lesson plan. Alongside defining the activities, other PBL supportive elements can be defined which are tools, resources and artifacts.

Moodle’s PBL script editor facilitates the teacher in creating a communication or collaboration tool according to the recommended tools from the PBL meta-model for each phase. For instance, an “online chat room” tool is the recommended tool in phase “problem definition”. The recommended tools vary from one phase to another according to the PBL meta-model. There are four common tools that can be used within a PBL lesson plan which are chat room, discussion forum, wiki, and brainstorming tool. In order to facilitate the customization of the defined tools types list for a phase from the PBL meta-model, a “custom tool” option was provided within the list of tools types. As shown in Figure 4.9 a tool type with a “Recommended Tool” label is recommended from the PBL meta-model to be used within the
“Problem definition” phase, while other tools with “Custom Tool” label have been added to the tool types list as supportive choices for the teacher to customize the available options from the PBL meta-model. Any created tool in a phase can be mapped for one or many activities within the same phase. All the possible tools types within a PBL lesson plan already exist within Moodle, but they exist in course level. However, in PBL lesson plan circumstance, such tools are required to be in phase level. This was one of the biggest challenges resolved in the implementation of Moodle’s PBL script editor. Instead of changing the source code of these Moodle tools to be visible in phase level, a method was developed in the “locallib.php” file of the developed PBL lesson plan plugin in Moodle. In such method an instance of the desired tool has been created using its predefined attributes, where the course level visibility was set to hidden.

![List of Tool Types for “Problem definition” phase](image)

Furthermore, a resource can be created within a phase and assigned for one or many activities within the same phase. For example, in “Problem definition” phase the recommended resource is “Problem trigger/source” which can be assigned or mapped into activity “present problem”. According to the PBL meta-model some phases have no recommended resources types, as a result a custom resource option has been added to the list of possible resources types. By this the designer can understand that there are no recommended resources in a specific phase, at the same time the designer can choose to add a custom resource type. As a resource can have different medium type to be delivered for students such as a URL for YouTube video, book, or page. A transmission medium type attribute has been added in addition to the resource type attribute within Moodle’s PBL script editor. Three medium types have been provided for the teachers which are URL, book and page.

Additionally, a teacher can create an artifact within a desired phase and assign it into one activity or more. An artifact could be whether output or input artifact of an activity. An artifact has to be first assigned as an output artifact of an activity where student will submit. If an artifact is submitted by any previous activity then it can be used as an input artifact for the upcoming activities. Hence, an input artifact does not require a submission from students; instead it will be used within the activity as a resource. If another submission is required for the same artifact in different activity, the artifact will be an output artifact from one activity. For the next upcoming activity it can be used as both input and output artifact where another submission will be required from students. As an artifact require a submission from the student, it is important for student to
know the submission medium type, whether it is file to be uploaded, online text to be filled or both. Also, in PBL-pedagogy it’s common to have a wiki as an artifact. As a result, three submission medium types have been provided for the teacher to specify the desired submission medium type with the ability to choose more than one medium type for the same artifact. The provided submission medium types in Moodle’s PBL script editor are file, online text and wiki. Choosing one of these medium types is not mandatory as some artifact could be delivered face-to-face (e.g. oral presentation) especially in hybrid PBL lesson plans, in such case the teacher or the designer add the description for the face-to-face artifact.

The developed Moodle’s PBL script editor in this study enables the teacher to create, edit and delete any PBL element in the PBL lesson plan. Also, enables mapping one PBL element to another in easy manner by check boxes. Moreover, the designer can specify the activities execution sequence in case the chosen run structure is sequential. This can be done by dragging and dropping the activities of a phase in the PBL lesson plan to satisfy the desired sequence. Also, the same manner can be applied to specify the running sequence of all phases in a PBL lesson plan. Hence the activity or phase that appears first in the PBL lesson plan would run first.

Moodle’s PBL script editor facilitates the design of PBL lesson plans by allowing the teacher to construct the desired PBL lesson plan from PBL elements including phases, activities, resources, tools and artifacts. Additionally, facilitate understanding the PBL lesson plan by providing recommended elements types for each PBL lesson plan element. Also, by having a PBL lesson plan template that can be used to understand existing successful PBL practices. Besides that, facilitates the customization of the PBL elements types defined in the PBL meta-model by providing custom choices for the teacher. The reusability and interoperability of PBL lesson plans was facilitated via having the import and export functions in Moodle’s PBL script editor. Using the PBL scripting language highly supports the implementation of the export and import functions where the PBL lesson plan is exported into a PBL script by following the PBL scripting language rules. The exported PBL script can be imported again into the PBL script editor with the flexibility to add, edit or delete any PBL lesson plan element. Hence, the same PBL script can be shared by teachers and reused continuously for different courses to solve different problem situations. Sharing PBL lesson plans would in turn reduce the time needed to design new PBL lesson plans that are already designed previously. According to these implementation specifications of the developed Moodle’s PBL script editor the main objective was met by facilitating the design, understand, customization, and reusability of online (and hybrid) PBL lesson plans. Figure 4.10 shows a simple PBL lesson plan created within the developed Moodle’s PBL script editor, which consists of three phases each phase has two activities.
A complete example of a PBL lesson plan for a real-world problem is described in section A.1 in Appendix A, in which the steps of creating this PBL lesson plan are described in details in section A.2 in Appendix A.

4.2. Implementation of Moodle’s PBL Player

The second part of the targeted implementation in this study was implementing a simple PBL player by adopting the model driven approach and extending Moodle platform. Both teachers and students are supported within the PBL player to execute the designed PBL lesson plan. Moodle’s PBL player manage the execution of the designed PBL lesson plans and make them ready for student to follow and for teacher to handle. However, due to the time limit and scope of this study, only one functional aspect was implemented for teachers to handle the execution of a PBL lesson plan. Also, some functional aspects were implemented for students to follow a PBL lesson plan.

4.2.1. Handling the execution of a lesson plan by Teacher

The run-time structure was applied to handle the execution sequence for both activities and phases. In sequential run-time 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. Also, force closing a 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. The icons used for handling force open and force close functions are shown in Figure 4.11 which are used for both activities and phases when their complete condition are set to time limit. In case an activity complete condition is set to “user control” then the teacher can simply clicks the open or close icon for an activity as shown in Figure 4.12. Clicking on open icon will open the activity and its status will change to open. If the run structure of activities inside a phase is sequential and there is already one open activity, then Moodle’s PBL player will prevent opening new activity and warn the teacher to close previous activities first.

The status of an activity or 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 Figure 4.13. Having such statuses help the teacher to know to track which phase or activity is open at a time. Additionally, only activities or phases with open status will be shown to the students and other will be hidden.

![Force open and close icons in runtime environment](image1)

![Open and close icons in runtime environment](image2)

![Activities and phases possible statuses](image3)

According to the scope of this study the implementation of the PBL player provides only limited support for teachers to facilitate handling the execution of a PBL lesson plan. However, more functional aspects will be provided for teacher as for future work. For example, Feedback function or grading rubric could be provided for the teacher within the PBL player to be able to assign individual or group feedback or grade for students.

4.2.2. Students work on a PBL problem

As the teacher makes a PBL lesson plan visible for students, they can work to solve the presented problem by following the PBL lesson plan structure as guidance for them. In Moodle’s PBL player, a PBL lesson plan home page was created to show the information of the PBL problem 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 Figure 4.14. 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.

Figure 4.19: List of group members for students with the current logged in student highlighted

Another page was also created for students in which they can see the current open phase that they need to work in. In this page the students can see 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. Figure 4.15 shows the page of open activities that students can view for the illustrated example. In the open activities page a student can click to enter an activity tool to collaborate with his/her group members. Also, can view a resource or submit a desired artifact. The artifact submission page has been developed for student to be able to submit and edit their required submission. By clicking on the Problem tasks artifact link shown in Figure 4.15 the students will be directed into a page to view their current submission status (as shown in Figure 4.16) where they can edit their submissions. By clicking in the “edit my submission” button a student will be directed to a new page were he/she can edit the current submission as shown in Figure 4.17. In addition, 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.
Figure 4.20: An open activity workspace for students

![Activity Workspace](image)

Figure 4.21: Artifact submission status

![Artifact Submission Status](image)
Figure 4.22: Edit my submission page
Chapter 5: Evaluation

This chapter describes the evaluation methodology applied to evaluate the usefulness of both Moodle’s PBL script editor and Moodle’s PBL player.

5.1. Evaluation of Moodle’s PBL script editor by teachers

5.1.1. Evaluation Description
This evaluation was applied to a PBL practical session conducted at Qatar University. The Moodle’s PBL script editor evaluation comprised of 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 in the teachers started using the PBL script editor without any training. The target of this evaluation was to test the usefulness of the developed PBL script editor for teachers.

Moodle server was constructed locally in a normal computer with 8 GB RAM. In regard with the administrative work of Moodle LMS, an authentication account was created for each participating teacher. Also a course was created for each participating teacher where he/she played the role of a teacher in that course. A few predefined students were enrolled for each course as well.

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. They were also given an idea of the PBL problem structure as most of the teachers had no or a little prior knowledge about PBL.
- The next 40 minutes were consumed to build a PBL lesson plan (described in section A.1 in Appendix A) using Moodle’s PBL script editor. 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 PBL script editor. The usefulness of the developed PBL script editor was thus evaluated. All the created PBL lesson plans by the teachers were saved and accessed through administrative role of Moodle in order to evaluate the extent at which teachers are able to create the given PBL lesson plan.
- Following this, the teachers were given five minutes to fill a survey (see section B.1 in Appendix B) and answer two questions regarding their experience using Moodle’s PBL script editor. The survey consisted of four sections to assess the teachers on; the background information, computer literacy, usefulness of the PBL scripting language and the PBL script editor, respectively. The last two sections of the survey evaluate the
usefulness of the developed PBL script editor, 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). After filling the survey they were asked to answer two questions which were used to identify the strengths and weaknesses of Moodle’s PBL script editor according to the teacher’s experiences. One open question was “What is the main difficulty you experienced while using Moodle’s PBL script editor?” while the other question was “What is the best thing you liked about Moodle’s PBL script editor?”

5.1.2. Evaluation Results
On analyzing the PBL lesson plans created by the teachers, most of them were successful and consisted of all PBL lesson plan elements (phases, activities, tools, resources, and artifacts).

5.1.2.1. Results of Survey
The collected data was analyzed using descriptive statistics such as mean and standard deviation. The 17 statements of the last two sections in the survey (see Table 5.1) were used to measure the usefulness of Moodle’s PBL script editor. Table 5.1 shows the mean and standard deviation values of the analyzed 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 script editor. The PBL scripting language was part of the evaluation as it was used within the development of the PBL script editor. 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 script editor. Additionally, the Cronbach’s alpha (α) value was calculated to measure the PBL script editor ease of use scale. Hence, the PBL script editor ease of use α scale is 0.835. Nunnally [56], recommended that any instrument used in a basic research should has reliability of about 0.70 as a base or better. According to Nunnally, the resulted α scale of Moodle’s PBL script editor demonstrates that the survey concerning ease of use is quite reliable comparing to the base value which is 0.70.

Table 5.5: Survey results for evaluating the PBL script editor ease of use

<table>
<thead>
<tr>
<th>Survey statements</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 The two levels (phase-level and activity-level) structure of the PBL script editor.</td>
<td>3.875</td>
<td>0.35</td>
</tr>
<tr>
<td>2 The terms or vocabularies used to define the PBL elements types.</td>
<td>3.75</td>
<td>0.71</td>
</tr>
<tr>
<td></td>
<td>Activity Description</td>
<td>Ease of Use Mean</td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------------------------------------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>3</td>
<td>Find an appropriate term or vocabulary to represent a PBL lesson plan.</td>
<td>3.75</td>
</tr>
<tr>
<td>4</td>
<td>The activity structure that includes tools, resources, artifacts and their interconnections</td>
<td>4.00</td>
</tr>
<tr>
<td>5</td>
<td>Represent a narrative into a PBL lesson plan.</td>
<td>3.25</td>
</tr>
<tr>
<td>6</td>
<td>Define students groups.</td>
<td>4.00</td>
</tr>
<tr>
<td>7</td>
<td>Create/delete a PBL lesson plan in a Moodle course.</td>
<td>4.25</td>
</tr>
<tr>
<td>8</td>
<td>Create/delete a phase.</td>
<td>4.375</td>
</tr>
<tr>
<td>9</td>
<td>Create/delete an activity.</td>
<td>4.25</td>
</tr>
<tr>
<td>10</td>
<td>Create/delete a tool (e.g. chat, forum).</td>
<td>4.25</td>
</tr>
<tr>
<td>11</td>
<td>Create/delete an artifact.</td>
<td>4.25</td>
</tr>
<tr>
<td>12</td>
<td>Create/delete a resource.</td>
<td>4.25</td>
</tr>
<tr>
<td>13</td>
<td>Interconnect tools to be used in an activity.</td>
<td>4.25</td>
</tr>
<tr>
<td>14</td>
<td>Specify activities and phases running sequence.</td>
<td>4.125</td>
</tr>
<tr>
<td>15</td>
<td>Define the detailed information of the PBL lesson plan such as description and goals.</td>
<td>3.625</td>
</tr>
<tr>
<td>16</td>
<td>Specify the completion condition (e.g. user control or time limit) of an activity or a phase.</td>
<td>3.625</td>
</tr>
<tr>
<td>17</td>
<td>Export a PBL lesson Plan into PBL script and then import it again.</td>
<td>4.00</td>
</tr>
</tbody>
</table>

Additionally, the results of the first two sections of the survey were analyzed. The first section involved information about the participated teachers' gender and prior PBL knowledge. The pie chart in Figure 5.1 represents both male and female teachers’ percentages, as half of the participated teachers was male and the other half was female. The relation between the gender and the ease of use mean of each teacher was analyzed as shown in Figure 5.2. Each point represents a teacher’s mean of the seventeen survey statements scores according to their gender; hence the figure shows no significant difference between male and female. Similarly, Figure 5.3 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 script editor is easy to use.

Moreover, section two of the survey involved information about the teachers’ computer literacy levels. In Figure 5.4, 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 PBL script editor. 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 PBL script editor is easy to use.

Figure 5.23: Pie chart of Male and Female teachers’ percentage

Figure 5.24: Scatter chart of teachers gender and ease of use mean

Figure 5.25: Scatter chart of teachers prior PBL knowledge(x-axis) and ease of use mean (y-axis)

Figure 5.26: Scatter chart of teachers computer literacy mean(x-axis) and ease of use mean(y-axis)

5.1.2.2. Results of Questions
At the end of the evaluation session the participating teachers were asked to answer two questions:

1. What is the main difficulty you experienced while using Moodle’s PBL script editor?
2. What is the best thing you liked about the Moodle’s PBL script editor?

The answers of these two questions were analyzed and few shortcomings were identified. One was that there were no help options for the elements within the PBL script editor. Also, some teachers were concerned about the availability of a user manual. Both help buttons and user manual will be developed as an enhancement of Moodle’s PBL script editor. Another identified disadvantage was abundance of links in the PBL script editor. The PBL script editor provides a separate link for creating each PBL lesson plan element such as phase, activity, artifact, resource and tool. The teachers suggested having buttons instead of links. This
suggestion will be considered for future modifications.

Evaluating the answers of the second question, some strengths points were identified regarding Moodle’s PBL script editor. Most teachers indicated that Moodle’s PBL script editor is easy to use and the used learning management system, which is Moodle, has simpler interface than the one they are using which is Blackboard. Finally, one of the teachers who evaluated his prior PBL knowledge as (4: knowledgeable), had mentioned that the developed PBL script editor is aware of the complexity of PBL concept and presents it well for ordinary teachers that have no prior knowledge about PBL in general.

5.2. Evaluation of Moodle’s PBL Player by students

5.2.1. Evaluation Description
The evaluation of Moodle’s PBL player was conducted through another practical session conducted at Qatar University. The PBL player 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 player without getting trained beforehand. This evaluation targeted testing the usefulness of the developed Moodle’s PBL player for students.

The same previously used setting of Moodle server, which was used in evaluating Moodle’s PBL script editor, was applied in evaluating Moodle’s PBL player. As administrative work of Moodle LMS, an account was created for each participating student. Besides that, the students were enrolled in one existing Moodle course. A complete PBL lesson plan (described in section A.1 in Appendix A), 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 player. They were asked to use Moodle’s PBL player without prior training in how to use it. Thus the usefulness of Moodle’s PBL player was evaluated. All groups’ interactions while using the PBL tools and their submitted artifacts were saved in Moodle’s server, which were used in evaluating the extent to which students were able to solve a PBL problem using Moodle’s PBL player.
- The last five minutes of the session they were asked to fill a survey (see section B.2 in Appendix B) and answer two open questions according to their gained experience in
using Moodle’s PBL player. 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 script editor. While the last section involved 12 ease of use statements that were created to measure the usefulness of the developed Moodle’s PBL player by students.

Each phase of the created PBL lesson plan was given only ten minutes. The start and end time of each phase was fixed for the groups as they needed to finish each phase within its time period. The goal was not to solve the provided PBL problem correctly as this will not affect the evaluation results. It was important that students get a feel of PBL characteristics and how it can works in an online environment, since most of them have never experienced such learning methodology. As a result, the target of this evaluation session was to test the usefulness of the developed Moodle’s PBL player and how it can be improved in the future.

5.2.2. Evaluation Results
The students’ interactions within each group were analyzed by reviewing the communication done within each used tool (e.g. chat, brainstorming tool, and discussion forum) within every PBL activity of a phase. Also, by reviewing their groups submitted artifacts. Through the interactions analysis, it was clear that students were satisfied in working within groups and following the PBL lesson plan phase by phase to solve the problem. They tried to use all available tools, resources, and submitted the required artifact of each PBL activity within its time limit. Generally, the students got the feel of how a PBL lesson plan works and were satisfied in following the involved four phases one by one.

5.2.2.1. Results of Survey
The collected survey data for evaluating Moodle’s PBL player was analyzed the same way it was done for evaluating the PBL script editor. Table 5.2 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 player evaluation statements. In addition, Moodle’s PBL player ease of use Cronbach's alpha (α) value is 0.78 which also demonstrates that the surveys’ ease of use is quite reliable.

<table>
<thead>
<tr>
<th>Survey statements</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Recognize the two-layer (phase-level and activity-level) structure of the PBL lesson plan</td>
<td>4</td>
<td>0.47</td>
</tr>
</tbody>
</table>

Table 5.6: Results of evaluating the PBL player ease of use
Recognize the activity structure that includes tools, resources, and artifacts. | 3.8 | 0.42 |
---|---|---|
Work in group while solving a PBL problem. | 4.7 | 0.48 |
Navigate into an existing PBL problem in a course | 4.1 | 0.74 |
View the PBL lesson plan details and its phases | 4.4 | 0.52 |
Identify the current status, grade, and time of each phase and activity. | 4.4 | 0.52 |
Use existing communication tool (e.g. chat, forum) in activity level | 4.1 | 0.32 |
Submit/edit a required artifact within an activity. | 4.2 | 0.63 |
Navigate into an available resource within an activity. | 4.2 | 0.42 |
Know the execution sequence of the PBL lesson plan phases and their corresponding activities. | 3.8 | 0.63 |
View detailed information (e.g. description and goals) of activities and their corresponding tools, resources, and artifacts. | 4.2 | 0.63 |
Specify the completion condition of an activity (e.g. time limited) | 4.1 | 0.57 |

Furthermore, the results of the first two sections of the survey were analyzed the same way as described in evaluating Moodle’s PBL script editor. This is because the first two sections are common between both used surveys. The pie chart in Figure 5.5 represents the percentages of both male and female participated students; it shows that 60 percent of the students are male, while 40 percent are female. Three male students had the same ease of use mean 3.9, while the others had a mean of 4.00 and greater as shown in Figure 5.6. On the other hand, the ease of use means for female students were very close to each other. However, the relation between the students’ gender and the ease of use mean indicated no significant difference between male and female. Additionally, as shown in Figure 5.7 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 script editor and PBL player were not affected by the level of prior PBL knowledge for the participated teachers and students. The scatter chart shown in Figure 5.8, presents the relation between each teacher 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 script editor. Though the computer literacy level showed a slight positive influence in evaluating the PBL script editor, a slight negative influence appeared in evaluating the PBL player with some deviations. That is some students with little computer literacy knowledge evaluated the PBL player more positively than students with higher computer literacy knowledge. However, the majority of the students still thought Moodle’s PBL player is easy to use.
5.2.2.2. Results of Questions

At the end of the evaluation session the participated students were asked to answer two questions:

1. What is the main difficulty you experienced while using Moodle’s PBL player?
2. What is the best thing you liked about Moodle’s PBL player?

The answers of these two questions were analyzed and few problems were identified. One is that the students need more practice to understand PBL-pedagogy of solving problems in order to get use of the developed PBL player, as the majority of students have no prior knowledge about PBL-pedagogy. Though, they evaluated the PBL player positively, they were interested in having PBL practical sessions for understanding the PBL-pedagogy more and how it could be applied in solving real world problems. As a result, PBL practical sessions need to be delivered for students in the future when PBL learning pedagogy will officially be applied to be a part of the learning courses. These PBL practical sessions would help students in applying
PBL-pedagogy correctly using the developed Moodle’s PBL player.

Also, some students had problems with the user interface of the developed Moodle’s PBL player and suggested having a more colorful interface that would make the contents of the PBL lesson plan more organized. This suggestion will be considered in the future work. Besides that, some students were concerned regarding wiki and brainstorming tools, as only one student within a group could edit the content of a wiki page or brainstorming graph at a time. Both wiki and brainstorming Moodle tools are based on asynchronous communication to save the consistency of the edited content. For education tools, asynchronous communication would keep the teacher aware of the different content versions and who edited each version. The teacher awareness of the tools contents versions might help in the grading process of the students’ contributions within a group.

On the other hand, after analyzing the answers of the second question some positive points about the developed Moodle’s PBL player were acknowledged. Most of the students indicated that having different types of communication tools within a PBL lesson plan at activity-level assisted them more in expressing their opinions and correct each other thoughts. Also, the majority of students acknowledged the plus side of PBL learning pedagogy dividing the process of solving a problem into phases and activities. Accordingly, students have better understanding of what they were required to do by following the PBL lesson plan. Also, some students indicated that the PBL player provides a well-organized online learning environment especially for problems with group working mode where they can better handle solving big PBL problems collaboratively. Additionally, two students had acknowledged the different medium types for submitting artifacts especially having wiki as submission medium type which was totally new to them, as they used to submit files as an output for their normal (non PBL) assignments.
Chapter 6: Conclusion and Future Work

6.1. 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 script editor to design PBL lesson plans and PBL player to simulate the execution of PBL lesson plans for students. Moodle’s LMS was customized to implement both PBL script editor and PBL player. 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 script editor facilitate teachers in building PBL lesson plans in flexible and reusable manner. Students would use Moodle’s PBL player 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 script editor and Moodle’s PBL player are easy to use with Cronbach’s alpha (α) value greater than 0.7 (see section 5.1.2.1 and 5.2.2.1 above). To the best of our knowledge, no other study has built a PBL script editor and PBL player 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 script editor and PBL player would inroad the PBL learning methodology into education here in Qatar and around the world.

6.2. Challenges
The biggest challenge that we have faced is to understand specification of the existing PBL scripting language which is a DSML and also mapping the PBL elements into Moodle’s configuration to support PBL-pedagogy. Moreover, using Moodle’s activities and resources in the phase level of the PBL lesson plan instead of Moodle’s course level was not an easy task. Also, understanding and following the structure of Moodle’s plugins and their required specification was a big challenge.

6.3. Future Work
The future work of this study involves different aspects. One is that enhancing Moodle’s PBL player to support different functional aspects for teachers, as within the scope and time limit 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 script editor 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 player 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


Appendix A: Creating a PBL lesson Plan using Moodle’s PBL script editor

A.1. Example of PBL lesson plan description
Following is a description of a PBL lesson plan to be created using Moodle’s PBL script editor.

PBL lesson plan name: Website Crash problem
Description: You will help a web developer to overcome a website crash. In this problem you will identify the ways of detecting problem sources in a website.
Resource: You may use QU library bookstore to look for information needed.

Phases:
Phase1: Problem definition
   Description: Provide guidance to understand the problem (situation or scenario) and to formulate a solution.
   Goal: Construct the intended problem (situation or scenario) in a written form.
   Activities:
   1. Activity 1: present the problem
      a. Description: real scenario will be provided to you in a resource to read and analyze
      b. Goal: read a real scenario and identify the problem
   2. Activity 2:Identify the problem
      a. Description: You need to identify the problem here and produce an artifact of problem statement.
      b. Goal: you need to be able to identify a problem from real world scenario.
      c. Tool: online chat room

Phase 2: identification of learning issues
   Description: Decompose the problem into small issues and identify the issues involved in this problem. You have to use the mind map tool to formulate the problem issues
   Goal: You need to be able to divide a big problem into small pieces of issues.
   Activities:
   1. Activity 1: Identify learning issues
      a. Description: You need to identify all issues included in the problem.
      b. Goal: You need to be able to formulate a list of issues involved in a problem
      c. Tool: brainstorming tool
   2. Activity 2: Structure learning issue
a. **Description:** In this activity you need to order the issues you identified in term of their importance to the problem. Then you need to submit an artifact with the defined order of issues.
b. **Goal:** To be able to order a list of issues according to their importance.
c. **Artifact:** learning issues (online text)
d. **Tool:** brainstorming tool

Phase 3: Gather and share information  
**Description:** In this phase you need to look for information from different sources. Then filter your findings that best fit the solution of the problem. All findings should be shared by the group members.  
**Goal:** You need to gain search skills and share the search findings with the group.  
**Activities:**

1. Activity 1: Gather information  
   a. **Description:** Each student should search for information and gather useful information.  
   b. **Goal:** Searching for needed information to solve a problem

2. Activity 2: Share Information resources  
   a. **Description:** You have to share your search findings with the group and identify the important findings from these resources.  
   b. **Goal:** you need to work collaboratively with your group members.  
   c. **Tool:** discussion forum

Phase 4: problem resolution  
**Description:** Articulate a solution to the problem in a constructive and organized way that can sustain the change involved in the problem.  
**Goal:** Assemble a bone-fide resolution for the posed problem.  
**Activities:**

3. Activity 1: Discuss  
   a. **Description:** Discuss with your group members to generate the possible solutions of the problem. You can use the discussion forum to discuss with your group members.  
   b. **Goal:** Be able to generate solution of the problem  
   c. **Tool:** discussion forum

4. Activity 2: Synthesize solution  
   a. **Description:** Synthesize the best fit solution to the problem. Then submit only one file that contains your solution description.  
   b. **Goal:** You need to choose the best and most important solution of the problem among other solutions.  
   c. **Artifact:** problem solution (file)
A.2. User guide to create a PBL lesson plan using Moodle’s PBL script editor

Following are the steps of creating the described example of PBL lesson plan in section A.1.

1. The teacher login to his Moodle’s account
   The teacher has login using Moodle’s account and then get the home page with list of his/her current courses. Figure A.1 shows the home page of Mohammed Ali who is a teacher of a web development course.

   ![Image of Moodle’s Home page for a teacher]
   
   Figure A.31: Moodle’s Home page for a teacher

2. Create/delete an online PBL lesson plan activity in a Moodle course.
   The teacher clicks on the course link and he gets the content of the course as shown in Figure A.2. The course is divided into 4 topics. The teacher can add a PBL lesson plan into any topic by clicking on the “add an activity or resource” link. Which will then the Moodle’s activities list appears as shown in Figure A.3. The teacher chooses “PBL lesson plan” activity from the list. Then he/she will get the form of “adding new PBL lesson plan” into topic 1. The teacher has to fill required data (e.g. name, description) as shown in figure A.4. If the form is submitted and data validated successfully, the new PBL lesson plan will be added in to topic 1 of course web development as shown in figure A.5. Many icons are provided to the teacher to manage the PBL lesson plan. The teacher can update the data he entered when he created the plan by clicking on ✉ icon. Also, the teacher can delete the created PBL lesson plan through clicking on ✗ icon.
Figure A.32: Web development course page

Figure A.33: Create a new PBL lesson plan
Figure A.34: Form of adding new PBL lesson plan in topic 1 of the course

Figure A.35: PBL lesson plan added into topic 1 course web development
3. Create/delete a phase and choose type (e.g., Problem-engagement or Investigation)
The teacher can click on the created PBL lesson plan to start designing his plan in the PBL script editor. Initially the plan has no phases as shown in Figure A.6, in which the teacher can import a PBL script of a previously designed PBL lesson plan or add a new phase. If the teacher clicks on import PBL lesson plan link, then he/she will get upload area to upload an existing PBL script as shown in Figure A.7. The uploaded file should be a zip folder that contains the PBL script files of a PBL lesson plan which will be used to generate the plan automatically. Additionally, when the teacher chooses to create a new phase, he/she will get the phase creation form as shown in Figure A.8. After the form is submitted, the new created phases will be shown in the PBL script editor area in the same way is done to create the 4 identified phases of the previously described PBL lesson plan in section A.1. By that all needed phases are added into the PBL script editor work area as shown in Figure A.9. The teacher can delete or update any phase by clicking on the specified icons beside the phase name.

![Figure A.36: Empty PBL lesson plan in PBL script editor](image1)

![Figure A.37: Upload area to import an existing PBL lesson plan](image2)
Figure A.38: Form of creating a new phase
4. Create/delete an activity within a phase and choose activities type (e.g., Identifying problem). There is a “Create new activity” link associated with each phase. The teacher can click the link to add required activities for each phase where the teacher has to fill the form for adding new activity as shown in Figure A.10. After the teacher adds all the required activities of each phase of the PBL lesson plan described previously, he will have a similar view to Figure A.11 with the ability to update and delete any activity.
Figure A.40: Form of adding new activity
5. Create/delete an artifact and choose a type (e.g., Problem-statement or Learning-issue)

The teacher can navigate into “create new artifact” link in which he will get a form to fill with required data as shown in Figure A.12. Also, the teacher can choose the artifact submission medium type including file, online text or wiki. Additionally, the teacher can map the artifact to be submitted as output artifact from specific activity or/and to be used within specific activity as input artifact. According to the above described PBL lesson plan, phase “identification of learning issues” have an artifact “learning issues” which is mapped into activity “structure learning issue” as an output artifact. After creating the two required artifacts of the previously described PBL lesson plan the teacher get the view similar to Figure A.13.
Figure A.42: Form of creating new artifact
6. Create/delete a communication tool (e.g. chat, forum) and map it to be used in an activity. In the same way, the teacher can create a new tool in a phase within the PBL lesson plan by navigating into “create new tool” link. The teacher has to fill the creation form as shown in Figure A.14 in which he can map the tool into different activities within a phase. According to the previously described PBL lesson plan phase “identification of learning issues” has a brainstorming tool that need to be used for both activities “identify learning issues” and ”structure learning issues”. Thus, in the creation form activities list both activities should be checked as illustrated in Figure A.14. After all needed tools within each phase are successfully created, the teacher will get the PBL lesson plan similar to Figure A.15.

Figure A.43: The PBL lesson plan after creating needed two artifacts
Figure A.44: Form of creating new tool
7. Create/delete a learning resource (e.g. problem trigger/scenario) and specify its medium type (e.g. URL, page). Also, map a resource to be used in an activity.

Similarly, the teacher can create a new recourse by filling the resource creation form (shown in Figure A.16) in which he can map the resource to specific activity within a phase. The previously described PBL lesson plan required only one resource within “problem definition” phase which need to be mapped into “present problem” activity. After creating the “problem trigger” resource in phase “problem definition”, the teacher will get the resource within the specified phase in the PBL lesson plan as shown in Figure A.17.
8. Specify an activity and phases sequence by ordering them in the view by drag and drop. The teacher can rearrange the phases and activities sequence by dragging and dropping using icon, which appears in the left of the phase/activity name within the PBL script editor. The view order of phases/activities within the PBL lesson plan identifies the execution order of them in the runtime environment.
After the teacher has finished with designing the PBL lesson plan as described previously similar view to Figure A.18 will be displayed in the PBL script editor. The teacher then can share the created PBL lesson plan by exporting it into a PBL script file using the export link in the PBL script editor. By clicking the “Export PBL lesson plan” link the teacher will get a download link which will download the PBL script for the teacher on his machine as shown in Figure A.19.

Figure A.48: Complete PBL lesson plan of the previously defined PBL lesson plan.

Figure A.49: Export a PBL lesson plan into a PBL script
Appendix B: Evaluation surveys

B.1. Survey of evaluating Moodle’s PBL script editor by teachers

<table>
<thead>
<tr>
<th>SECTION 1: BACKGROUND INFORMATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1. Gender</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>1.2. Prior knowledge: To which extent you know problem-based learning before attending the session</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SECTION 2: COMPUTER LITERACY</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1. Are you skilled at using generic computer tools such as Word or PowerPoint?</td>
</tr>
<tr>
<td>2.2. Are you skilled at using communication tools such as chat rooms, discussion forums, or wikis?</td>
</tr>
<tr>
<td>2.3. Are you skilled at using education-specific tools such as digitalized whiteboards or online questionnaire authoring and responding tools?</td>
</tr>
<tr>
<td>2.4. Are you skilled at using learning management systems such as Moodle or Blackboard?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SECTION 3: EASY TO USE PBL SCRIPTING LANGUAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1. The two-layer (phase-level and activity-level) structure of the PBL script is Difficult to use.</td>
</tr>
<tr>
<td>3.2. The term or vocabulary used to define the phase or</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>3.3. It is difficult to find an appropriate term or vocabulary to represent lesson plan.</td>
</tr>
<tr>
<td>3.4. The activity structure that includes tools, resources, output artifacts and their relations is easy to understand.</td>
</tr>
<tr>
<td>3.5. It is difficult to represent a narrative into a lesson plan using the PBL authoring tool.</td>
</tr>
<tr>
<td><strong>SECTION 4: Easy to use Moodle’s PBL script editor</strong></td>
</tr>
<tr>
<td>4.1. It is difficult to define groups within the authoring tool.</td>
</tr>
<tr>
<td>4.2. It is easy to create/delete a PBL plan activity in a course</td>
</tr>
<tr>
<td>4.3. It is easy to create/delete a phase within the authoring tool.</td>
</tr>
<tr>
<td>4.4. It is easy to create/delete an activity within the authoring tool.</td>
</tr>
</tbody>
</table>
| 4.5. | It is easy to create/delete a communication tool (e.g. chat, forum) within the authoring tool. | 1: strongly disagree 
2: disagree 
3: no opinion/unsure 
4: agree 
5: strongly agree |
| 4.6. | It is easy to create/delete an artifact within the authoring tool. | 1: strongly disagree 
2: disagree 
3: no opinion/unsure 
4: agree 
5: strongly agree |
| 4.7. | It is easy to create/delete a resource within the authoring tool. | 1: strongly disagree 
2: disagree 
3: no opinion/unsure 
4: agree 
5: strongly agree |
| 4.8. | It is easy to map tools to be used in an activity. | 1: strongly disagree 
2: disagree 
3: no opinion/unsure 
4: agree 
5: strongly agree |
| 4.9. | It is easy to specify an activity or phase sequence within the authoring tool. | 1: strongly disagree 
2: disagree 
3: no opinion/unsure 
4: agree 
5: strongly agree |
| 4.10. | It is easy to define the detailed information of the plan elements, information such as description and group mode (e.g. no group, separate groups, visible groups). | 1: strongly disagree 
2: disagree 
3: no opinion/unsure 
4: agree 
5: strongly agree |
| 4.11. | It is difficult to specify the completion condition (e.g. user or time control) of an activity within the authoring tool. | 1: strongly disagree 
2: disagree 
3: no opinion/unsure 
4: agree 
5: strongly agree |
| 4.12. | It is easy to **export** your PBL Plan into script file, and then **import** it again. | 1: strongly disagree 
2: disagree 
3: no opinion/unsure 
4: agree 
5: strongly agree |
# B.2. Survey of evaluating Moodle’s PBL player by students

## SECTION 1: BACKGROUND INFORMATION

<table>
<thead>
<tr>
<th>1.1. Gender</th>
<th>Male</th>
<th>Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2. Prior knowledge: To which extent you know problem-based learning before attending the session</td>
<td>1: nothing</td>
<td>2: a little</td>
</tr>
</tbody>
</table>

## SECTION 2: COMPUTER LITERACY

| 2.1. Are you skilled at using generic computer tools such as Word or PowerPoint? | 1: not at all | 2: a little | 3: basic | 4: skilled | 5: very experienced |
| 2.2. Are you skilled at using communication tools such as chat rooms, discussion forums, or wikis? | 1: not at all | 2: a little | 3: basic | 4: skilled | 5: very experienced |
| 2.3. Are you skilled at using education-specific tools such as digitalized whiteboards or online questionnaire authoring and responding tools? | 1: not at all | 2: a little | 3: basic | 4: skilled | 5: very experienced |
| 2.4. Are you skilled at using learning management systems such as Moodle or Blackboard? | 1: not at all | 2: a little | 3: basic | 4: skilled | 5: very experienced |
# SECTION 3: EASY TO USE Moodle’s PBL player

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.</td>
<td>The two-layer (phase-level and activity-level) structure of the PBL problem is Difficult to understand.</td>
<td>1: strongly disagree</td>
<td>2: disagree</td>
<td>3: no opinion/unsure</td>
<td>4: agree</td>
</tr>
<tr>
<td>3.2.</td>
<td>The activity structure that includes tools, resources, output artifacts and their relations is easy to understand.</td>
<td>1: strongly disagree</td>
<td>2: disagree</td>
<td>3: no opinion/unsure</td>
<td>4: agree</td>
</tr>
<tr>
<td>3.3.</td>
<td>It is difficult to identify my group members while I’m working on the problem.</td>
<td>1: strongly disagree</td>
<td>2: disagree</td>
<td>3: no opinion/unsure</td>
<td>4: agree</td>
</tr>
<tr>
<td>3.4.</td>
<td>It is Difficult to navigate into an existing PBL problem in a course</td>
<td>1: strongly disagree</td>
<td>2: disagree</td>
<td>3: no opinion/unsure</td>
<td>4: agree</td>
</tr>
<tr>
<td>3.5.</td>
<td>It is easy view the problem details and its phases</td>
<td>1: strongly disagree</td>
<td>2: disagree</td>
<td>3: no opinion/unsure</td>
<td>4: agree</td>
</tr>
<tr>
<td>3.6.</td>
<td>Difficult to identify the current status, grade, and time of each phase and activity.</td>
<td>1: strongly disagree</td>
<td>2: disagree</td>
<td>3: no opinion/unsure</td>
<td>4: agree</td>
</tr>
<tr>
<td>3.7.</td>
<td>It is difficult to use existing communication tool (e.g. chat, forum) in activity level</td>
<td>1: strongly disagree</td>
<td>2: disagree</td>
<td>3: no opinion/unsure</td>
<td>4: agree</td>
</tr>
<tr>
<td>3.8.</td>
<td>It is easy to submit/edit a required artifact or deliverable within an activity.</td>
<td>1: strongly disagree</td>
<td>2: disagree</td>
<td>3: no opinion/unsure</td>
<td>4: agree</td>
</tr>
</tbody>
</table>
| 3.9. | It is easy to navigate into the available resource in an activity. | 1: strongly disagree  
2: disagree  
3: no opinion/unsure  
4: agree  
5: strongly agree |
| 3.10. | It is easy to know the execution sequence of the problem phases and activities. | 1: strongly disagree  
2: disagree  
3: no opinion/unsure  
4: agree  
5: strongly agree |
| 3.11. | It is easy to view all detailed information of the problem elements, information such as description and goals. | 1: strongly disagree  
2: disagree  
3: no opinion/unsure  
4: agree  
5: strongly agree |
| 3.12. | It is difficult to specify the completion condition of an activity (e.g. time limited) | 1: strongly disagree  
2: disagree  
3: no opinion/unsure  
4: agree  
5: strongly agree |