City University of Hong Kong

Department of Computer Science

BSCS Final Year Project 2007-2008

07CS035

Web-Based Environment for Managing Undergraduates Project-Based Learning

(Volume 1 of 1)

Student Name : **Tsoi Wing Sze**

Student No. :

Programme Code : **BSCCS**

Supervisor : **Dr. KWOK, Lam For**

1st Reader : **Dr. HUANG, Chih-hao Scott**

2nd Reader : **Dr. YU, Yuen Tak**

For Official Use Only

Student Final Year Project Declaration

I have read the project guidelines and I understand the meaning of academic dishonesty, in particular plagiarism and collusion. I hereby declare that the work I submitted for my final year project, entitled:

Web-Based Environment for Managing Undergraduates Project-Based Learning

does not involve academic dishonesty. I give permission for my final year project work to be electronically scanned and if found to involve academic dishonesty, I am aware of the consequences as stated in the Project Guidelines.

Student Name:	TSOI Wing Sze	Signature:	
Student ID:		Date:	14 April 2008

Abstract

Undergraduates often involve in doing projects as a learning activity in universities. It is the Project-Based Learning (PBL) which is a popular teaching and learning method in this century. PBL is stated as a model that organizes the learning around projects. It often has a group of students engaged in cooperative learning and collaborative problem solving. In PBL, undergraduates often need to solve different problems. In addition, supervisors and undergraduates need searching, studying and sharing information in order to complete a project. Although undergraduates will have deeper understandings of learning contents and concepts, some issues would be raised during the learning process such as the difficulties in knowledge management and collaborative learning. Supervisors would also have the difficulties in supervising undergraduates to work on their projects. Furthermore, most of the existing learning systems and tools may not solve these problems effectively. Therefore, this project aims to develop a web-based environment to facilitate the Project-Based Learning of undergraduates. This environment is developed with the use of Ontology and Learning Object Metadata concepts and technologies in order to improve and support the knowledge management, collaborative learning and supervision in PBL. In addition, this environment provides some management functions which can help both supervisors and undergraduates to organize their projects and learning resources more efficiently and systematically.

Acknowledgements

I would like to give my special thanks to my supervisor Dr. Kwok, who gives me a great supports throughout this year. He is very keen on discussing and giving advices for my finial year project. With his guidance, my project can be finished and developed with more valuable features.

Also, I would like to thank for my classmate Jessie Yeung, who is another project student supervised by Dr. Kwok. Since she also needs to use Learning Object Metadata and Ontology concepts and technologies, we always share and discuss ideas in this year.

I would like to thank for my classmates, Carter Yu, Gelica Chan and Naomi Li, who have given me encouragement in doing this project.

Lastly, I would like to thank for my family, who give me a great supports during this year.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Project-Based Learning (PBL) of Undergraduates

In university study, undergraduates are often assigned projects to do by teachers from different courses. A project can be worked by an individual or a group to achieve specific objectives. The main purpose of doing projects in studying is to let undergraduates learn the knowledge and skills of specific topics in-depth. This is one of the most common teaching and learning activities used in universities, Project-Based Learning. According to Buck Institute for Education (BIE) (as cited in Markham, Larmer and Ravitz, 2003),

> "[PBL is defined as] *a systematic teaching method that engages students in learning knowledge and skills through an extended inquiry process structured around complex, authentic questions and carefully designed products and tasks* [italicized originally]" (p. 2).

In this learning approach, the learning process is not only led by teachers, but students have to organize and manage their own works. The focus is more student-directed.

1.1.1 Process of Doing a Project

To start a project, a supervisor should recognize the knowledge or topics students drive to learn. Hence, the supervisor may need to search and study some information related to the project. Then the supervisor could provide some issues and questions that can lead students to think in-depth. The supervisor could also

provide instructions and learning references to help students doing the project. Since PBL is more student-centered, students also need to investigate topics, solve problems, make decisions, etc by themselves during the learning process.

Students should identify the problems, the project aims and objectives at first. Then they need to plan various tasks such as the work allocations, learning outcomes, project activities, etc. They would follow their project plan to study and find the solutions to solve the problems. Moreover, they need to implement the solutions and produce some products to achieve the project goals. Finally, both students and the supervisor need to evaluate the project tasks, investigative activities, deliverables, etc to check whether the problems, the project aims and objectives could be solved and achieved.

Furthermore, during the learning process, students may need to investigate specific topics and collect further learning resources in order to solve the problems and perform the tasks. In addition, they may share and exchange ideas and learning resources with their supervisor and group members, thus, they could gain more knowledge and do the project efficiently. The supervisor may provide guidance and monitor the process to help that students are on the right track.

Doing a project, students would need to participate in solving the problems, planning different tasks and managing the whole process of the project. As PBL requires more involvement of students, it could enhance students' learning and motivation. Also, students could have deeper understandings of the knowledge and problems after finished the project.

1.2 Problems of Project-Based Learning

However, since a project may involve many complex and interactive activities, it may be difficult for supervisors and students to manage. In university, both supervisors and undergraduates often have more than one projects on-going at the same time. The working cycle of a project may not only complicate, but also may last for several months or even a year. It would be difficult for them to manage and organize several projects.

Knowledge Management

As mentioned before, during the process of a project, many different kinds of learning resources would be searched and studied. For example, the supervisors often need to search some information from the libraries or internet. After gathered all relevant materials, the supervisors could recognize the project domain and assign to students. This process would waste a lot of time and efforts. Also, it would be inefficient that the supervisors need to collect the materials again even if they assign another similar project to students.

In addition, students often need to search and study many learning references, in order to find the solutions and complete the project. It would be difficult for them to manage and organize large numbers of learning resources. If students are not able to centralize and classify different kinds of learning resources, it would be very hard and time-wasting for them to review the learning resources.

Collaborative Learning

Students may encounter difficulties in understanding and searching learning references. They may need to discuss and share learning resources with others in order to understand the knowledge. Moreover, as it would be hard for the project group members to have face-to face discussion throughout a project, they may not able to work on the project effectively.

Supervision of Students' Learning

Supervisors may need to provide guidance for a project group. They need to know the overview about how students manage and build up their projects, and the level of understandings about their learning references in order to provide suitable supports. Hence, it would be difficult for the supervisors to supervise project groups if students need to report the progress and meet with them face-to-face.

1.3 Approach to Solve the Problems

To facilitate PBL, a web-based environment is proposed to develop in my final year project which enhances supervisors and undergraduates to manage projects. As the environment is online, they could work on their projects with less limitation on time and location.

Knowledge Management

Moreover, this web-based environment is proposed to provide interfaces that help supervisors and students to organize and collect different kinds of learning resources. All the learning resources are centralized that supervisors and students could reuse for different projects.

As the supervisors could reuse the learning resources assigned for different project teams, it would reduce their times and efforts for searching information. The students could not only easier to handle different learning resources in a project. They could also review their learning resources effectively.

Collaborative Learning

This web-based environment would also support students and supervisors to share ideas and learning resources. It would reduce students' difficulties in finding and studying learning references. In addition, discussion among students and supervisors would able to enhance the knowledge exchange and learning.

Supervision of Students' Learning

This web-based environment would support supervisors to supervise different project groups. It would help organizing the project detail information that the supervisors could understand how each project group manages and builds up their projects. The supervisor could also know their level of understandings easily. Hence, the environment could improve the efficiency of supervision as the students need not to meet and report their progress face-to-face.

Most importantly, this web-based environment is proposed to allow centralizing learning resources and enhancing the collaborative learning and knowledge sharing to facilitate the Project-Based Learning of undergraduates.

1.4 Aims and Objectives

This project aims to develop a web-based environment for managing the Project-Based Learning of undergraduates. Using this environment, supervisors and students can manage and share their projects and learning resources easily.

Objectives

1. Study and review the related researches, systems and technologies, which can help in designing and developing a Web-Based Environment to facilitate Project-Based Learning of undergraduates.

2. Design and implement a basic project structure which can help supervisors and undergraduates to manage projects and learning objects.

3. Design and implement the system architecture and functions which enhance the environment for supervisors and undergraduates to manage projects and learning objects.

4. Illustrate how the environment facilitates the management of Project-Based Learning of undergraduates.

5. Write a report to evaluate and discuss the system.

1.5 Methodology

1. The researches, existing learning tools and technologies of the following topics will be studied:
   - Student-Centered Teaching and Learning
   - Project-Based Learning
   - Existing web learning systems and tools
   - Ontology
   - Learning Metadata Object

   The background and literature review of some researches and existing learning systems are studied and written.

2. Study existing learning systems and tools to investigate the overall system requirements which can facilitate Project-Based Learning of undergraduates.

3. Study the common learning process to investigate a project basic structure which can help supervisors and undergraduate to do a project systematically.

4. Study OWL Web Ontology Language and Ontology Applications to investigate and design an ontology model of the project basic structure to help managing projects and learning objects.

5. Study IEEE Learning Object Metadata (LOM) to a structure for organizing the Learning Objects, which can enhance the knowledge management.

1.6 Report Organization

In this report, the detail of this project will be introduced. The following are the summary of different chapters:

Chapter 1: Introduction

- Overview of this project

Chapter 2: Review of Project-Based Learning

- Background information related to Project-Based Learning
- Review of the studies and systems related to the Project-Based Learning
- Review of technologies

Chapter 3: Overview of Project-Based Learning Environment

- Overview of the environment

Chapter 4: Design of Project-Based Learning Environment

- Conceptual Design of the environment
- Functional Design of the environment

Chapter 5: Implementation and Technologies of Project-Based Learning Environment

- Implementation of the environment
- Technologies used in the environment

Chapter 6: Scenario

- Scenario of using the developed Project-Based Learning Environment
- Evaluation of using the developed Project-Based Learning Environment

Chapter 7: Conclusion

- Conclusion of the achievement and difficulties of this project
- Further development of this environment and project will be discussed

Chapter 2. Review on Project-Based Learning

2.1 Introduction

In this section, the background information and the review of studies and researches have been done based on the previous literatures. It includes reviews on Student-Centered Teaching and Learning, Project-Based Learning, different learning systems, Learning Object Metadata and Ontology used in learning environment.

2.2 Student-Centered Teaching and Learning

Student-Centered Teaching and Learning places students to be the center that they can learn actively rather than only receive information from teachers. Project-Based Learning, Problem-Based Learning and Inquiry-Based Learning are student-centered teaching and learning methods.

Problem-Based Learning and Project-Based Learning often used interchangeably (Markham, Larmer and Ravitz, 2003). However, there are some differences between them. The approach of Project-Based Learning focuses on the learning through a project performed by the students, while the approach of Problem-Based Learning focuses on the learning simulated by ill-structure and open-ended problems (Markham, Larmer and Ravitz, 2003; Starr, 2005; TREE, 2007).

According to Starr (2005) mentioned, there are some common features between Project-Based Learning and Problem-Based Learning. In these learning models, students involve in solving the authentic problems and situations. They are

student-centered that teachers facilitate students in learning. They also motivate and engage students in learning and solving the problems. Students need to find the information and resources in order to develop possible solutions. Moreover, they could improve students' problem-solving skills, research skills and collaborative learning.

Inquiry-Based Learning also has similar features that students need to investigate widely to build new understandings, meaning and knowledge to solve the questions and problems (Alberta Education, 2004). However, it is more student-centered. It requires students actively ask and formulate questions, while the questions or problems of the Project-Based Learning and the Problem-Based Learning may be given by teachers. In addition, students need to present the knowledge to support a point of view or develop the solutions.

In these three teaching and learning methods, students are required to actively engage in the learning process. They have to identify the questions or problems that they need to solve independently or with a group of students. The most important is that students have to investigate widely to find the information and learning resources in order to develop the possible solutions.

To conclude, Project-Based Learning, Problem-Based Learning and Inquiry-Based Learning are student-centered learning and teaching methods. Their learning approaches always place students in the real world problems that they can gain the real-life learning experience. And students can build up their own knowledge by investigating information.

2.3 Definition of Project-Based Learning

Markham, Larmer and Ravitz (2003) pointed out that there is no clear and accepted definition of PBL. Apart from the definition of PBL in Section 1.1.1, there are many definitions of PBL suggested by different educators and organizations. Since many researches about the features of PBL have been studied based on different theories and models, it would be difficult to review and identify the characteristics of a PBL project. Therefore, the following features of PBL and activities would be focused in this report.

2.3.1 Project-Based Learning Project

Thomas (2003) stated that Project-Based Learning is a model that organizes learning around projects. Therefore, the researcher concluded the criteria of a project that it can be considered as a PBL project. The following are the five criteria suggested:

1. The project is the central teaching strategy.
2. The project focuses on the questions or problems that encounter students to learn.
3. The project involves students in investigation that they need to design, make decisions, identify problems, solve problems, etc.
4. The project is more student-directed that students can manage the process of the project to some significant degree.
5. The project is more realistic that students involve in solving authentic problems and questions.

2.3.2 Benefits of Project-Based Learning

'Project-Based Learning (PBL) draws on the latest research on effective teaching pedagogies and learning approaches in the 21st century' (Wong et al 2006: 450). PBL requires active participation of students in the learning process. This does not only help students to learn knowledge and skills of specific topics in-depth, but also does benefit supervisors in teaching.

PBL would increase the motivation of students, as they would have more chances of engaging than the traditional education. In addition, since students would actively engage in doing rather than receiving knowledge from supervisors, they could able to construct their personal knowledge in PBL (ISTE, 1997).

Moreover, students would need to think of the real-life questions and problems that they need to solve in the project. This would enhance them to develop the knowledge and skills linked with the real world, and also enhance their problem solving skills (Markham, Larmer and Ravitz, 2003; Soloman, 2003; Mergendoller et al, 2005). In addition, PBL would increase the analytical skills of students because they would need to analyze, derive and synthesize the knowledge after gathered the learning resources and information (Soloman, 2003).

Since many learning resources and tasks would be involved in PBL, students would learn the skills of resource management. They would be more familiar with the management of a project such as plan the tasks, schedule the time, etc to complete the project (ISTE, 1997). They would also learn to organize the learning resources that they could review easier.

In PBL, students would work as a group that they need to share ideas and learning resources with their supervisors or project group members. This could enhance the collaboration of students. Also, their interpersonal and communication skills could be enhanced.

Furthermore, Wong et al (2006: 450) stated that "teachers facilitate students' project-based learning, often they also become their peers". Hence, supervisors would not only supply knowledge to students directly, but they would support and direct students to learn. Since they would involve in the learning of students in PBL, they could also expand their knowledge from different projects. Moreover, they would have more chances to communicate with students. This could not only improve their communication with students, but also increase their understandings of the students' needs.

Finally, PBL would be more motivational and interactive that both supervisors and students would actively work and communicate with others to complete a project. They would gain the real-life experience and knowledge by doing a project. Also, they could improve their problem solving skills, analytical skills, management skills, interpersonal and communication skills.

2.4 Project-Based Learning Problems

As mentioned, PBL benefits students in learning as they could learn knowledge in-depth. Students could also develop and improve their problem-solving skills, analytical skills and communication skills. In PBL approach, students are required to plan activities, decide what the information is needed, search and investigate the information (Land and Greene, 2000). Therefore, information management, information sharing among supervisors and project group members are the key factors affecting the learning effectiveness.

Knowledge

Knowledge is the facts, skills or condition of knowing something that acquired by a person through experience or education. Hence, knowledge is not gained from collecting different information. People need to understand and organize the information to gain different knowledge.

Figure 2.1 Showing how to gain knowledge

Information in a Project

In PBL, the information is not only gathered from the learning resources. Students have to learn around projects, so the whole process of doing projects has different information required to manage. The following is the list of common information in a project:

- Project information which helps supervisors and students to understand the purposes and subjects about the project.

- Information of project design process such as planning of project tasks and learning paths which shows how students proceed the project.

- Learning resources searched and studied in the project.

- Learning outcomes information such as the skills learned and deliverables finished in the project.

Since many different kinds of information in each project, the information has to organize and manage that allows supervisors and students to gain the knowledge.

Knowledge Management Problems

Students had the difficulties in understanding the central concepts of the projects (Thomas, 2000). Therefore, it is important to manage the projects' information that students can understand the projects' goal and domain subjects. Loose and unorganized information not only influences their understandings of the projects, but also fails to direct them to learn from the projects. Hence, supervisors and students have to organize the related information that can clearly describe the projects; otherwise, they may not effectively gain the knowledge and manage different projects.

In addition, students would encounter the problems in planning the effective paths to work on their projects (Thomas, 2000). Supervisors can provide learning paths to help them to study. Managing this information can also keep track to the learning process that supervisors and students can efficiently review and learn to manage projects. However, students should participate in project design in PBL. They can plan and make decisions on how to work on the projects; meanwhile supervisors act as facilitators who provide guidance to manage the learning process of students (Mergendoller and Thomas, 2000). Since PBL requires the great involvements and interactions among supervisors and students, the information of the learning process is difficult to manage collaboratively.

Large amount of learning resources would be searched and studied by supervisors and students from the internet or libraries. In addition, some deliverables like assignments would be finished in the projects. They can provide information of different subjects. The simple records of these learning resources may help searching and sharing information among project group members. However, Land and Greene (2000) stated that "providing access information does not guarantee that students will use it to solve problems or deepen understanding". Hence, it is difficult for supervisors to know the learning of students.

Collaborative Learning Problems

Supervisors and students need to work collaboratively in PBL. However, Thomas (2000) stated that students would have the difficulties in dividing the workload eventually. Also, the researcher mentioned that students would fail to provide feedback and understand others' work. Hence, students may able to work with

others, but they may not able to work collaboratively. The learning of projects could not be enhanced even if students work as a group.

Supervision of Students' Learning Problems

Supervisors play the important roles in students' learning that they support students in learning. They need to discuss the problems with students or facilitate their discussions, thus, they need to monitor students' working progress. However, it is difficult for them to decide when to provide instructions and know what is learned by students (Thomas, 2000). Hence, they may not give suitable feedback and suggestion to help students in doing projects.

2.5 Review on Existing Web-Based Learning Systems

There are many different kinds of web-based learning systems and tools developed. In the following, three learning systems and tools will be reviewed to learn how these systems and tools solve the problems mentioned in Section 2.4. The first one is a popular learning system used in academic institutions, Blackboard Learning System. The second one is a free inquiry-learning online tool, WebQuest. The third one is a paid project-based learning tool, Project Foundry.

2.5.1 Blackboard Learning System

Blackboard Learning System is a popular web-based learning system used among academic institutions. It provides a platform with different capabilities for helping supervisors and students in teaching and studying. This system includes three main areas of capabilities including Instruction, Communication and Assessment.

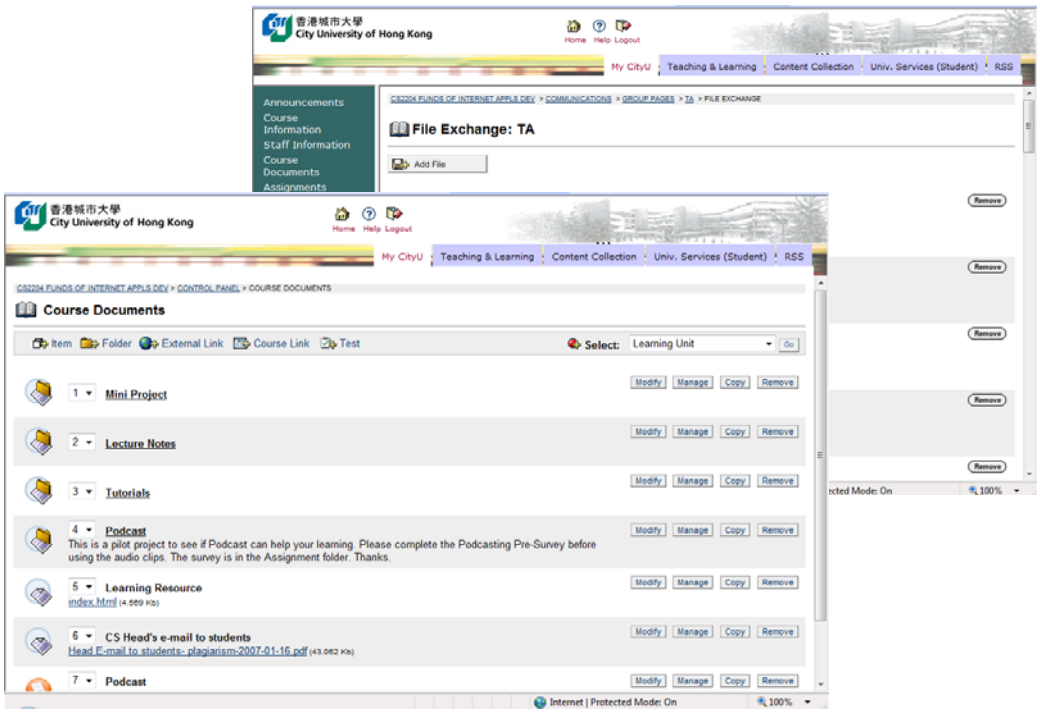


Figure 2.2 Blackboard Learning System

Knowledge Management

It provides interfaces for the supervisors to manage course materials such as course syllabus, contact information, etc. They can also upload the electronic materials used for different courses. It enables supervisors to share and deliver learning resources and references for students.

In addition, students can hand in their assignments to their supervisors through this system. Students can also share some files with other students who are in the same group.

Collaborative Learning

Besides sharing files among students, this system provides interfaces for students to discuss and even perform live chatting with the group members.

Supervision

Supervisors can construct learning paths for students that they follow the paths to study. In addition, supervisors are able to carry out auto-scored assessment and survey in this system. They can also give assignments and grades to students.

This system provides interfaces for supervisors to manage courses and materials and to provide tailor-made learning paths for students. This could benefit storing learning resources and instructing students to learn. Also, it can increase the efficiency of evaluating students' performances, as supervisors could assign and collect assignment online. In addition, this system allowing students to share ideas and files can enhance the collaborative learning.

However, this system may not provide sufficient environment to facilitate PBL. In knowledge management, it tends to help storing and sharing learning resources, but it does not help centralizing and classifying these learning resources. Hence, it cannot effectively organize the learning resources that students could not review easily. Although it enables supervisors to set the learning paths for students, those learning paths are not flexible that students could not design their own activities.

2.5.2 WebQuest

WebQuest is an inquiry-oriented online tool for learning. It helps to organize student-centered learning using internet resources (MacGregor and Lou, 2004). Teachers around the world would use it to design different online courses and lessons for their students in the websites. WebQuest has six main components: Introduction, Task, Information, Process, Guidance and Conclusion (Dodge, 1997).

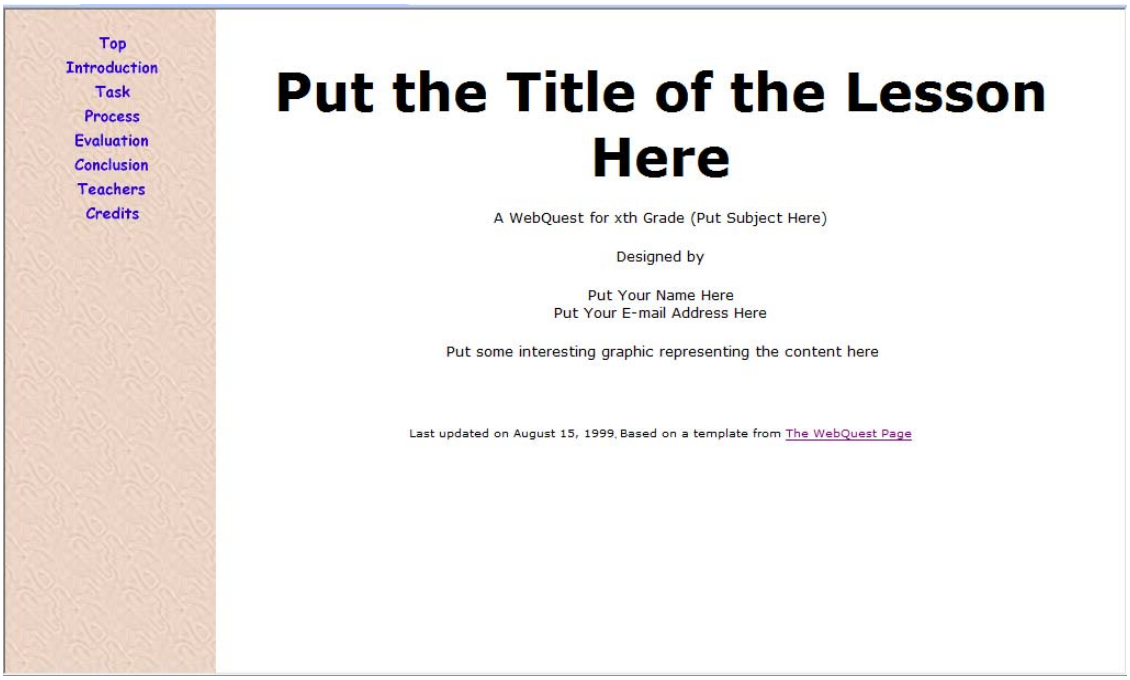


Figure 2.3 WebQuest Template (http://webquest.sdsu.edu/LessonTemplate.html)

Supervision

Using the WebQuest model, supervisors could provide some background information about the project, course or lesson. In addition, the supervisors could provide the description about the process how students should complete the tasks. They could also provide the guidance to teach students in organizing the information. At the end of the learning process, they could give the feedback about what students have learned (Dodge, 1997).

Knowledge Management

As supervisors would give tasks which are interesting for students, they could gather and provide some information sources to help students completing the tasks. In addition, students are required to perform information seeking, analyzing, and synthesizing strategies if they use WebQuest for inquiry-based learning (MacGregor and Lou 2004). After they have gathered the information, they could perform the tasks.

Collaborative Learning

Since the supervisors could provide some information sources such as real-time conferencing, email, etc, the supervisors and students are able to share ideas and communicate with others to finish the tasks collaboratively.

WebQuest is an effective learning tool to help supervisors designing an online course or lesson. Using the WebQuest model, the supervisors could create the websites that would guide students to investigate, search and analysis the information in the internet. In addition, as it enables the supervisors embedded

different internet resources such as concept map tool, email, etc, students could learn to organize the information and communicate with others.

However, the online courses and lessons developed with this tool may not be useful and effective in learning and learning resources management. This is because the quality of the courses and lessons would depend on whether the supervisors design a good learning environment. Moreover, the supervisors would only organize the internet information and sources for students to study, they could not know how students solve the problems and their level of understandings about the resources. In addition, the students may not able to organize the learning resources effectively as it does not provide a standard method to manage the learning resources.

2.5.3 Project Foundy

Project Foundry is a Project-Based Learning System, which enhances students in learning and streamlines the project-based learning process for supervisors and students. This learning system provides numbers of functions to facilitate them in organizing projects.

Knowledge Management

This system allows students to create projects by submitting the project proposal to supervisors. It provides interfaces to help students managing the entire project process such as submitting reports and assignments, writing progress log and daily journal. In addition, supervisors can search projects and students to review and approve their projects. They can also process and view assessments for different projects.

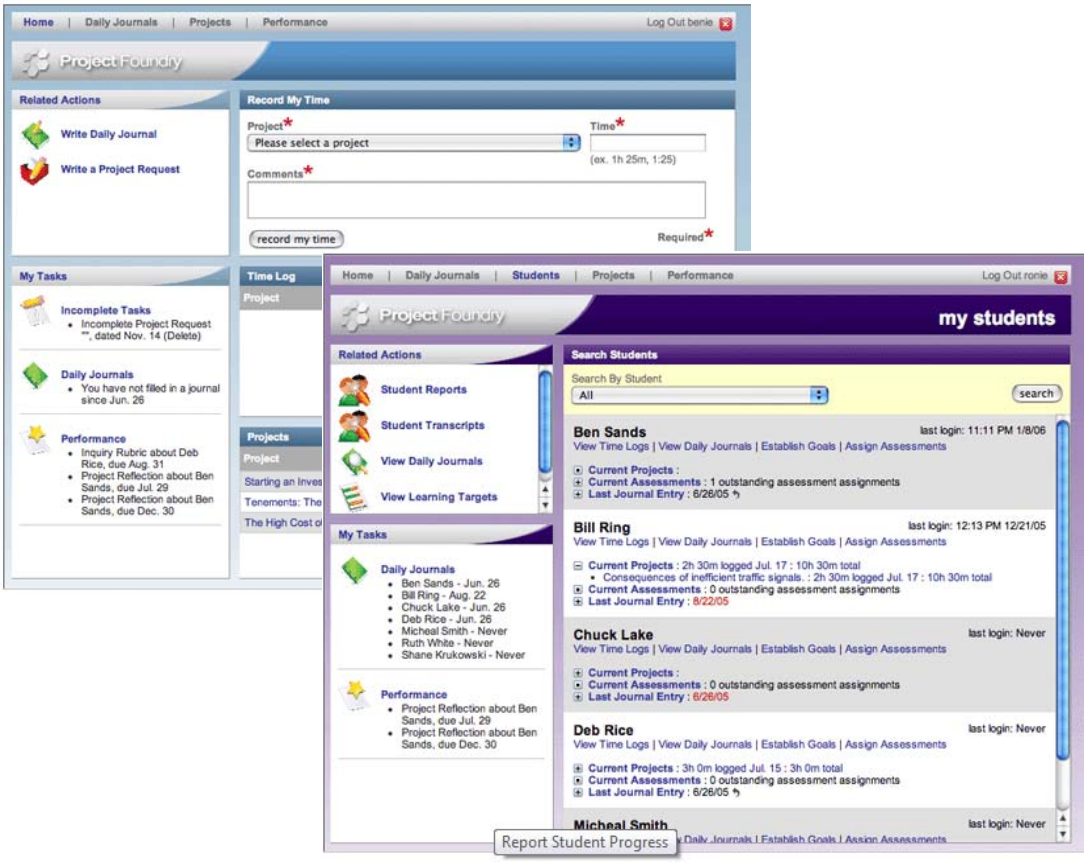


Figure 2.4 ProjectFoundry System (http://www.projectfoundry.org/)

Supervisions

Supervisors can view what a student is working on by the project logs, such that they can help students to achieve their goals. Moreover, the supervisors can communicate and understand the progress of students by viewing the daily journals students reported.

This system could enhance the management of projects as it streamlines the process for students to submit project proposal, reports or assignments. Also, since this system displays information about the progress and deadline of different tasks to remind students, it could help students to keep track of their learning.

However, this system lacks an environment for students to exchanges information and learning references. Hence, it may not facilitate the collaborative learning in PBL. In addition, students write project logs to report their progress, such logs do not provide a systematic indication of the exact progress, e.g. their level of understandings about the knowledge, etc. As a result, the supervisors may not provide suitable advice and supports for the students with this information.

2.5.4 Summary of Existing Web-Based Learning Systems Problems

Based on knowledge management, collaborative learning and supervision of students' learning, the web-based learning systems and tools are evaluated. It shows that different systems and tools have the advantages in facilitating teaching and learning. However, most of them can only solve some problems and difficulties in PBL.

In Section 2.4, it indicated that different information is needed to describe the projects. Since Blackboard is not a typical PBL or student-centered learning system, it provides functions to manage course information rather than project information. Although WebQuest enables supervisors to manage project information, it does not standardize how to describe the projects. In addition, it is more likely to help supervisors to manage different projects separately. It does not help supervisors to organize different projects in the same platform.

Moreover, managing the learning process is a kind of knowledge management of projects. These learning systems and tools can manage this information in different levels. Blackboard and WebQuest mainly provide functions for

supervisors to design the learning paths and to manage the learning process. Students cannot participate in the project design process. Involvement of students is one of the features of PBL projects, so these systems may not facilitate the PBL effectively.

Project Foundry provides interfaces for supervisors and students to manage different kinds of projects. Also, the learning process is designed and managed by students, while supervisors monitor their learning by viewing students' progress logs. However, it only provides platform for students to hand-in projects' assignments and reports. Supervisors and students cannot exchange other learning resources.

In opposite, Blackboard and WebQuest enable exchanging learning resources. Using WebQuest, supervisors can add different kinds of learning resources in the websites to share with students. Blackboard also allows both supervisors and students to upload learning references' files to share with others. Nevertheless, they do not facilitate organizing learning resources effectively as there is no standard format to classify and describe the resources. Supervisors can only know what learning resources have been found. As a result, supervisors may not efficiently monitor students' learning progress and provide suitable supports.

2.6 Ideal Project-Based Learning Environment

Project-Based Learning needs great interaction and participation of both supervisors and students. Therefore, one of the criteria of an ideal PBL environment is that both supervisors and students have the flexibility in managing the projects' information and learning process. In addition, knowledge management, collaborative learning and supervision of students' learning are the main problems that they always encounter in working on projects. Hence, an ideal environment should provide functions to solve these problems. Also, it should provide a good learning environment that supervisors and students can manage their projects efficiently and systematically.

2.7 Review on Related Technologies

The existing web-based learning systems and tools may not effectively solve the problems and facilitate PBL. To develop an ideal Project-Based Learning environment, Learning Object Metadata and Ontology are reviewed, which are used in some learning systems.

2.7.1 Learning Object Metadata (LOM)

Learning Object Metadata (LOM) is the information about the Learning Objects. Learning Object (LO) is defined as any entity, digital or non-digital that may be used for learning, education or training. By using LOM, it enables learning objects to be searched, evaluated, used and managed more efficiently (IEEE Learning Technology Standards Committee, 2002).

IEEE 1484.12.1-2002, Learning Object Metadata

In this standard, learning objects can be described with numbers of data elements which are grouped into nine categories.

1. General Category:

   The general information describes a learning object as a whole.

2. Life Cycle Category:

   The information about the history and current state of a learning object, and the data elements which have affected the learning object during its evolution are described.

3. Meta-Metadata Category:

   It describes this metadata record itself but not describes the learning

object.

4. Technical Category:

   It describes the technical requirement and characteristics of a learning object.

5. Educational Category:

   It describes the key educational or pedagogic characteristics of a learning object.

6. Right Category:

   It describes the intellectual property rights and conditions of use for a learning object.

7. Relation Category:

   It defines the relationship between a learning object and other learning objects.

8. Annotation Category:

   It provides the information about comments on the educational use of a learning object.

9. Classification Category:

   It describes the information about a learning object in a particular classification system.

Using this standard of LOM, different kinds of learning resources can be described with the same format. In e-learning system, there are large amount of learning resources. With using LOM, it is easier to organize different kinds of learning resources, which also increase the flexibility and possibility to share and reuse.

2.7.2 Ontology

Ontology is a conceptual model which describes a set of concepts (classes) in a domain and the properties and relationships between these concepts. It can be used in Semantic Web, software engineering, etc, to represent with a set of individual (instance) to form a knowledge base (Noy and McGuinness, 2001; Wikipedia). With Ontology support, the concepts and relationships of specific domain in computer systems can be defined and helped to present as reasonable as real world knowledge.

Ontology Used in Learning Environment

Semantic Web technologies play an important role in further development of learning systems (Devedzic, 2006). Ontologies are the base of Semantic Web that transform human understandable information and data into machine understandable knowledge (Maedche, 2002). The use of ontology has three main purposes: share and communicate between humans, interoperate data among computer systems, and knowledge sharing and reuse (Xie and Shen, 2006).

Few researches have been examined the use of Ontologies in enhancing the management of Learning Objects. Zhu, Li and Guo (2007) showed that ontology can support sharing and reusing of knowledge in e-learning by using a learning object ontology which manage and classify learning resources. The researchers also indicated that ontology could enrich the information content of the learning objects such that it could help developing a better search methodology and management of learning resources.

Heiyanthuduwage and Karunaratne (2006) showed that ontology could increase the effectiveness and usability of the e-learning system. The researchers used the ontology defining the learning object metadata and the learner profiles in a typical Learning Management System. The researchers found that the ontology could enhance locating and retrieving learning resources satisfying the needs of the learners.

These researches showed that ontology could be used to enhance the e-learning environment. By using different domain ontologies in managing learning objects, it could help to process, search, share and reuse the learning objects within the e-learning environment.

Chapter 3. Overview of Proposed Project-Based Learning Environment

3.1 System Environment Overview

In Project-Based Learning, undergraduates need to work on projects as a learning method. Supervisors and undergraduates need to manage their projects that many information and learning resources are required to organize and share during the learning process. In Figure 3.1, it shows the general overview of the environment that helps both supervisors and undergraduates to manage Project-Based Learning.

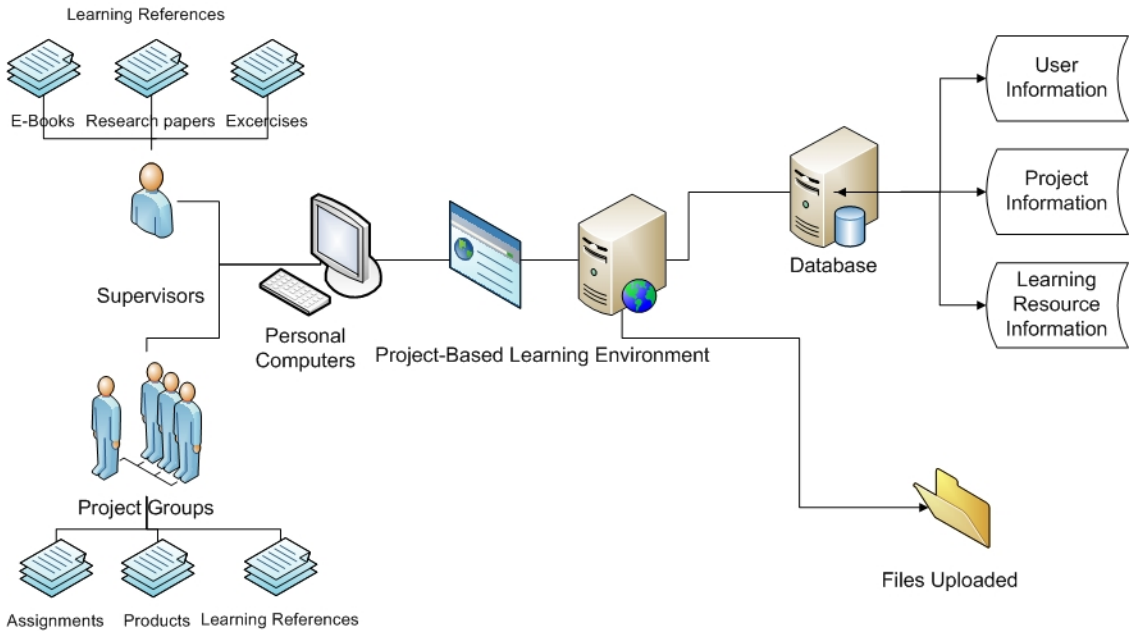


Figure 3.1 Overview of Project-Based Learning Environment

3.2 Proposed Project-Based Learning Environment

This proposed Project-Based Learning Environment aims to provide an ideal Project-Based Learning Environment for supervisors and students to manage projects and learning resources.

Knowledge Management

The environment facilitates them in managing the information that students can acquire different knowledge from the projects. Firstly, the information of project can be organized that they can effectively understand, review and manage different projects.

In addition, it allows supervisors to suggest the learning paths which drive students to learn and work on the projects. At the same time, students can also define the learning paths to manage their projects. This can let supervisors and project group members to understand others' learning process.

The environment helps them to manage different kinds of learning resources. They can upload or keep the records of the learning resources. Also, the learning resources can be described with some descriptive information. Supervisors and students can review and organize easily. And it is more efficient in sharing the knowledge among project group members.

Collaborative Learning

The environment not only facilitates knowledge management, but also helps sharing knowledge of the projects. Supervisors and students are allowed to organize the projects' information and learning resources collaboratively. They can discuss the learning resources to enhance their understandings of the concepts of different knowledge.

Supervision of Students' Learning

In the environment, students can describe and manage the information and learning resources. In addition, they can manage their learning outcomes such as project deliverables or skills learned. Hence, supervisors can easily assess the performances of students. And they can check whether students can achieve the projects' goals in order to provide feedback to enhance students learning.

3.3 Doing Projects with Project-Based Learning Environment

During the learning process of each project, supervisors and undergraduates always face the main problems of knowledge management, collaborative learning and supervision of students' learning. They need to manage the project information, activities, learning resources and etc. To facilitate the process of doing projects, this Web-Based Environment provides interfaces and functions for them to manage their projects and learning resources systematically.

3.3.1 Project Information Management

Supervisors or project group members need to set the project aims. Therefore, they have to know the project information before they can start a project. Also, they have to study and organize the project information so that they can plan the activities easier.

Figure 3.2 General Information of a Project

3.3.2 Learning Process Management

A good management of learning process can help supervisors and students planning and proceeding the tasks of a project. Additionally, the learning progress can reflect the actual performances of the students in the learning process of a project.

Supervisors

In the initial stage, supervisors need to manage some project information and guidance to help students to understand the projects. In addition, each project would have some goals that students have to achieve. Hence, supervisors need to define the intended learning outcomes (ILO), activities and assignment schedules as the learning paths, which can help students to do their projects. They also need to know and monitor the progress of the projects or even the learning progress of each student. Then they can realize the needs of students in order to give suitable supports.

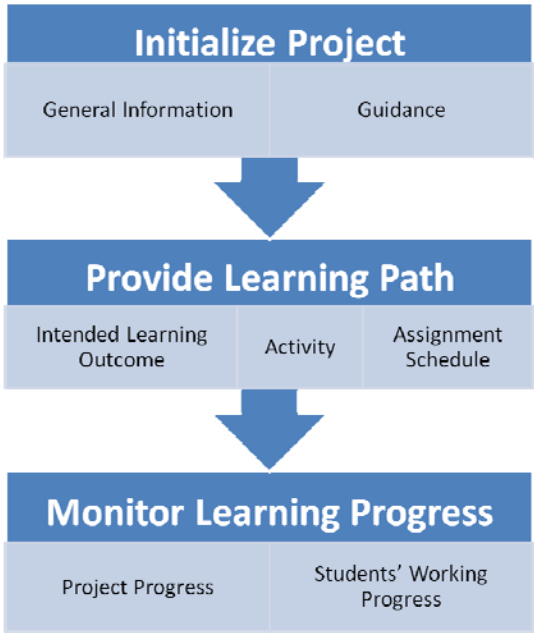


Figure 3.3 Supervisors managing project's learning process

Project Group Members

Project group members need to understand the projects' goals and follow the learning paths to proceed their projects; meanwhile they can manage their own learning path in each project. They can record their own activity, achievement and etc with the learning paths. Also, they need to manage the learning outcomes, activities and project log to record their current learning progress. With good management of the learning process, they can easily review the statuses and plan the tasks of the projects.

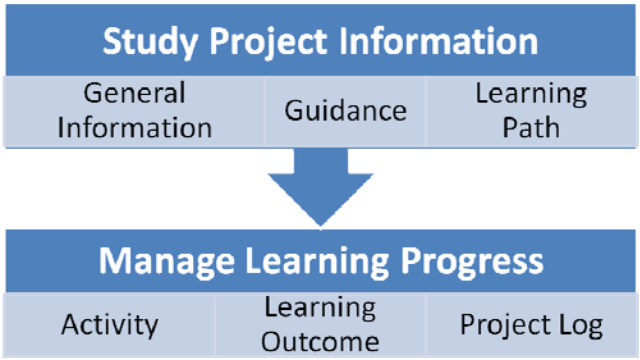


Figure 3.4 Learning Process of Project Group Members

3.3.3 Learning Resources Management

Throughout the whole learning process of PBL, both supervisors and project group members need to manage different kinds of learning resources. The environment does not only need to help them to collect learning resources. It is more important that students can gain the knowledge from the learning resources.

Supervisors

Supervisors need to gather learning resources and information to start the projects. They organize some relevant learning resources for students to use as

references in the projects. In addition, they need to review the learning resources gathered and defined by students. As they can know the progress and the levels of understandings of students from the organized learning resources, they are able to give feedback to supervise students' learning.

Project Group Members

In PBL, project group members not only study learning resources given by supervisors, but also search and study learning resources by themselves. They need to organize and define the learning resources such that they can efficiently review and share with supervisors and project group members. Also, they need to discuss and share ideas with others to gain deeper understandings about the learning resources.

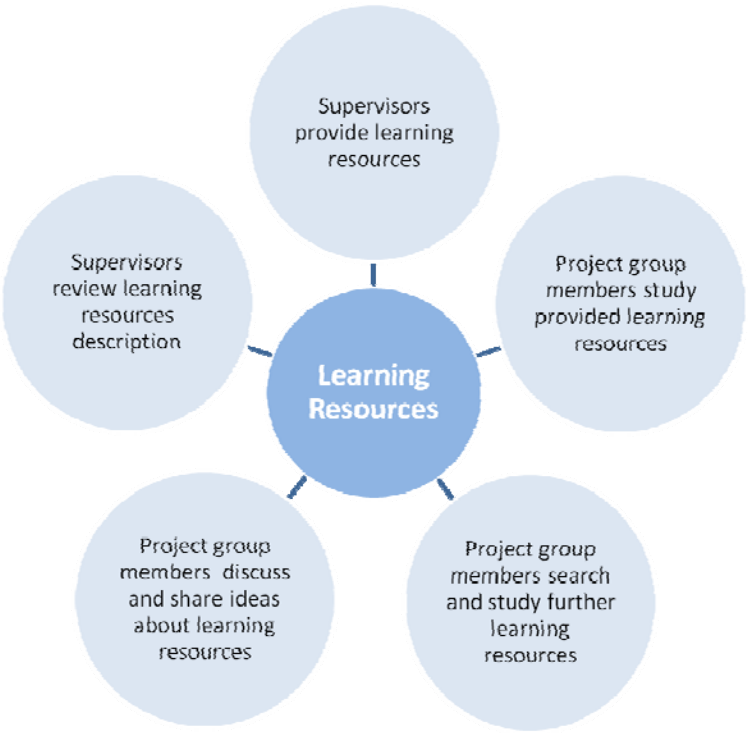


Figure 3.5 Learning Resources in a Project

3.3.4 Learning Outcomes Management

Learning Outcomes are the learning results of the projects which can be some actual deliverables or skills learned after the projects completed. They can reflect the performances and learning achievements of the project group members.

Supervisors

After the completion of projects, supervisors need to assess the results of the projects to check whether students can learn from doing projects. To evaluate the performances of each project group members, supervisors have to know their involvements and understandings of different learning resources. Also, they need to assess the assignments, deliverables and other learning outcomes to check whether students can accomplish the projects' goals and intended learning outcomes. Then they can give feedback to each project group member that the students can improve themselves.

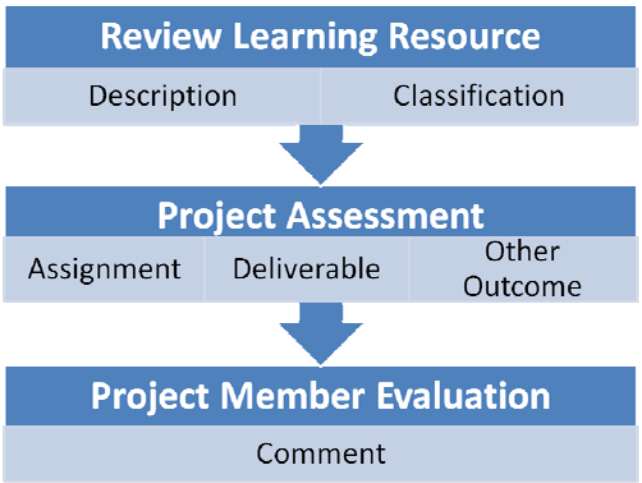


Figure 3.6 Supervisors evaluating project learning outcome

Project Group Members

Project group members have to search and study different learning resources in PBL. To manage their learning outcomes, they need to describe their level of understandings of the learning resources. They also need to submit the project assignments or other deliverables according to the projects' schedules. Since they can record their assignments, deliverables and skills learned with the learning paths during the learning process, this allows both supervisors and project group members to review and evaluate the learning outcomes efficiently.

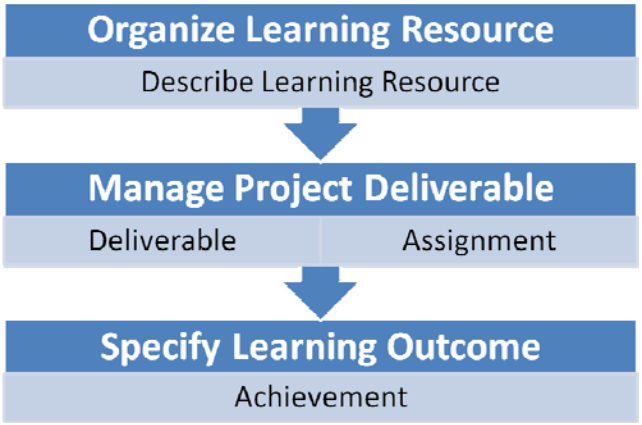


Figure 3.7 Project Group Members managing Project Learning Outcome

3.4 Conceptual Design of Project Structure

In Project-Based Learning Environment, large amount of learning objects such as project, projects' activities, learning resources and etc are studied or finished by supervisors and students. It is difficult to manage large amount of learning objects. In addition, these learning objects provide wide varieties of information for supervisors and students. They must be organized that supervisors and students can identify and review easily.

In PBL, many learning objects are needed to share among supervisors and students. Even the environment enables them to upload files or provide links of the learning resources, it may not enhance the knowledge sharing.

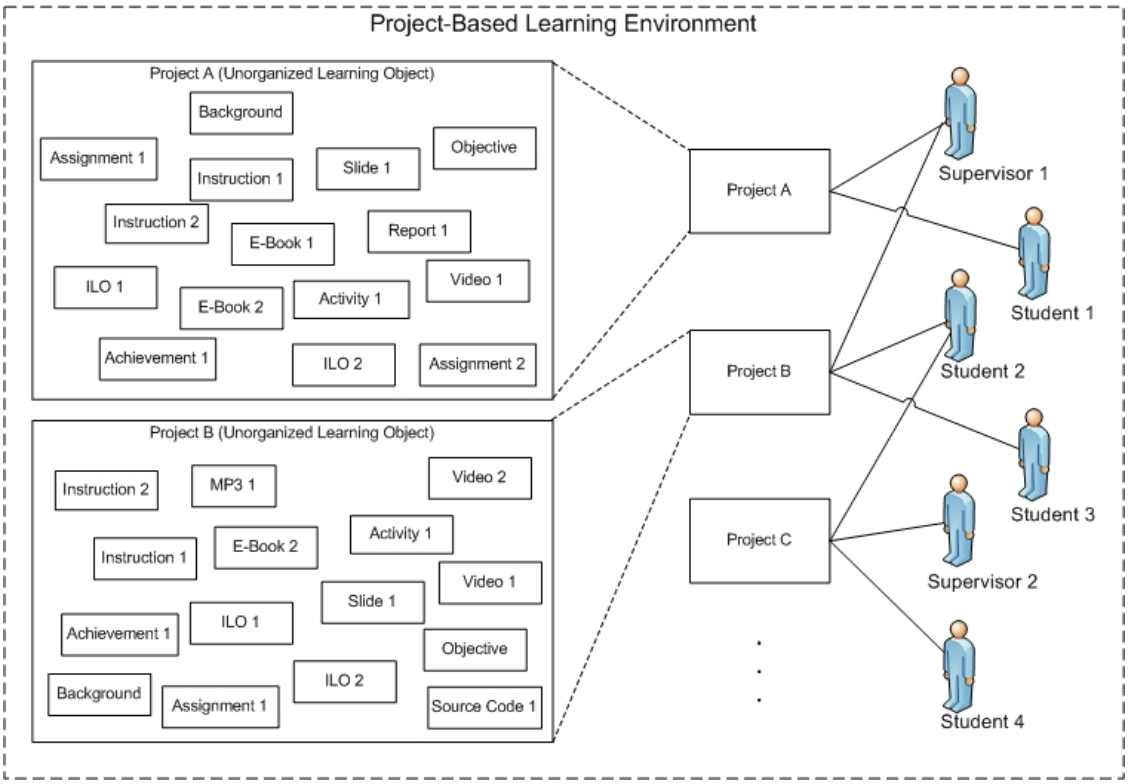


Figure 3.8 Projects with unorganized learning objects in Project-Based Learning Environment

In Figure 3.8, it shows that many kinds of learning objects in each project. It is difficult for both supervisors and students to identify and review those unorganized learning objects. It is also inefficient that they have to open the learning resources to review the contents. Therefore, it is necessary to manage and organize different kinds of learning objects which allow them to identify and review easier.

First of all, the learning objects should be managed with the same format. This allows centralizing all the learning objects in this environment. Since the learning objects can be centralized, supervisors and students can search and reuse them in different projects efficiently.

On the other hand, the learning objects should be defined with some descriptive information. Supervisors and students can remember what they have learned from the learning objects as they need to define the descriptive information. In addition, the information could provide the ideas of the contents of learning objects. This helps them to learn and share knowledge and ideas. Supervisors can also know the level of students' understandings. This can save the time on searching and studying learning object.

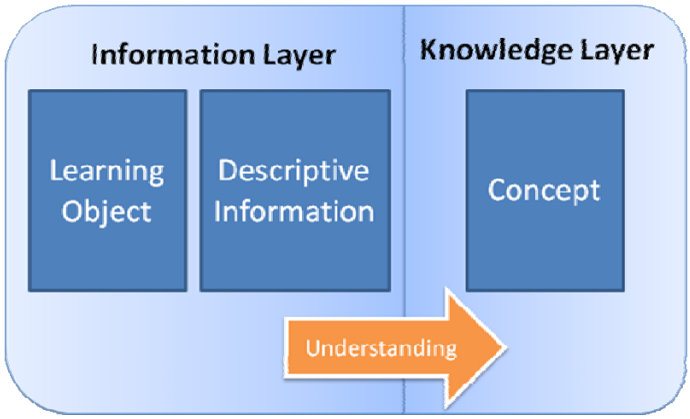


Figure 3.9 Process of gaining knowledge from learning objects

In Figure 3.9, it shows that learning objects with descriptive information can enrich the information layer. This helps students know more about the learning objects. However, students need to understand the subjects and concepts of learning objects that they can gain the domain knowledge from the learning objects.

Furthermore, supervisors and students would have different knowledge, subjects or topics that they supposed to learn from each project. Although adding descriptive information can enhance the management of learning objects, it may only help students to understand each learning object individually. It cannot organize different learning objects to represent some conceptual knowledge of the projects. Also, some learning objects would be used in more than one project, however, they may have different meaning and representation in the projects.

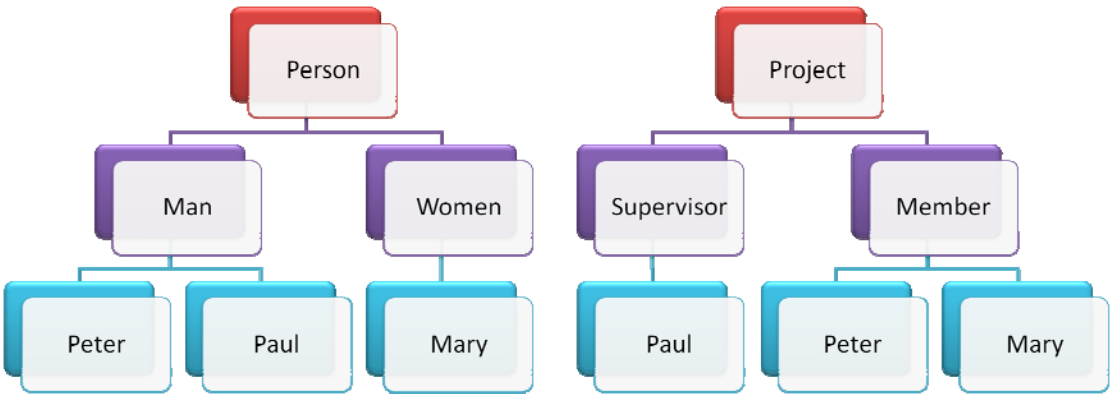


Figure 3.10 Same learning objects (instances) representing different conceptual knowledge

To enhance the learning object management of each project, supervisors and students should be able to classify the learning objects according to their understandings and conceptual knowledge in different projects.

Since the learning objects are organized with classification, supervisors and students can search and reuse them more efficient. Moreover, classification of learning objects can enhance students' knowledge management skills, while supervisors can know their understandings of the learning objects.

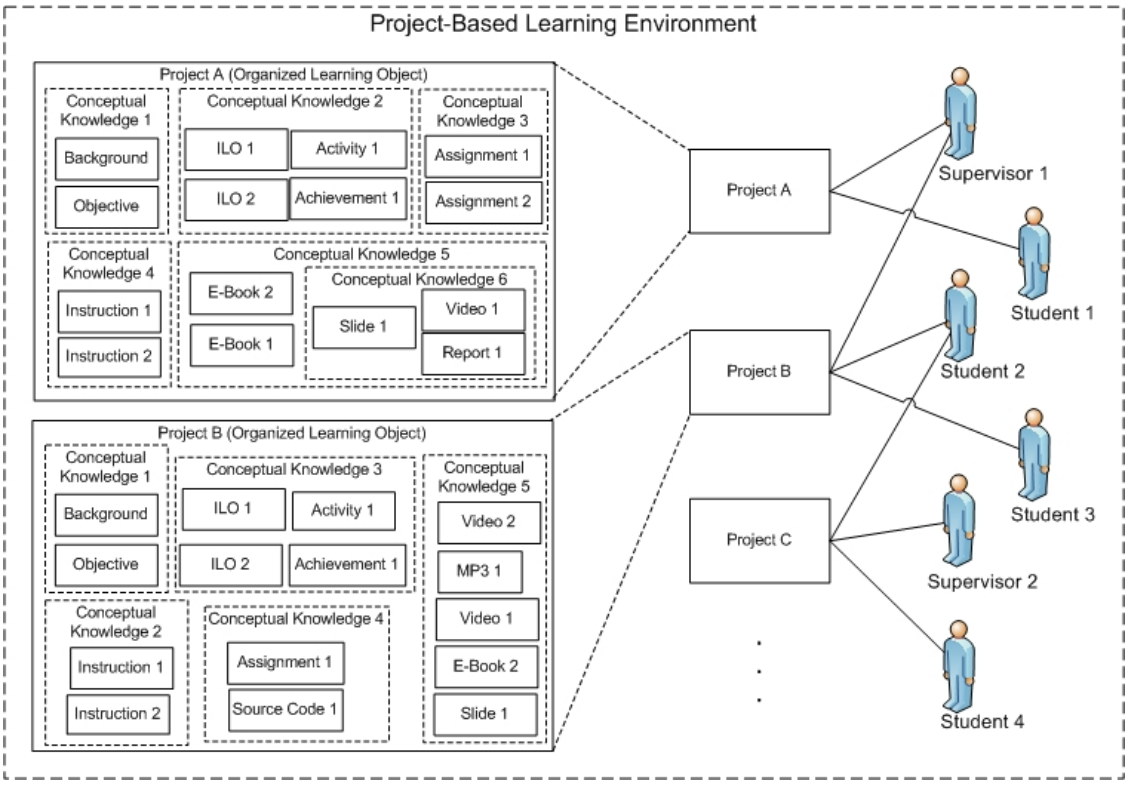


Figure 3.11 Projects with organized learning objects in Project-Based Learning Environment

On the other hand, the phase and process of doing projects are similar, thus, every project always constitutes some common types of learning objects as Figure 3.11. At the initial stage, the information of projects' background and objective are needed to study. Then the supervisors and students have to prepare the project plan which clearly states the expected learning outcomes and activities of the project learning process. During the learning process, different kinds of learning references are studied. At last, the students would finished some tasks or deliverables.

Therefore, to enhance the management of the projects and learning resources, some conceptual knowledge can be pre-defined that they could organize and classify the learning objects easily. The following main types of learning objects:

1. General Information
2. Instruction
3. Project Design Process (intended learning outcomes, activities, achievements)
4. Learning references
5. Deliverables (assignment)

If students could classify different learning objects based on the above conceptual knowledge, it could be more efficient to manage the projects and learning resources. With the descriptive information and classification of learning objects, it facilitates the information and knowledge sharing among project group members. They can also review the knowledge of project structure efficiently.

Chapter 4. Design of Project-Based Learning Environment

4.1 System Architecture

The following diagram shows the system architecture, which contains nine modules to handle different functions in this Project-Based Learning Environment.

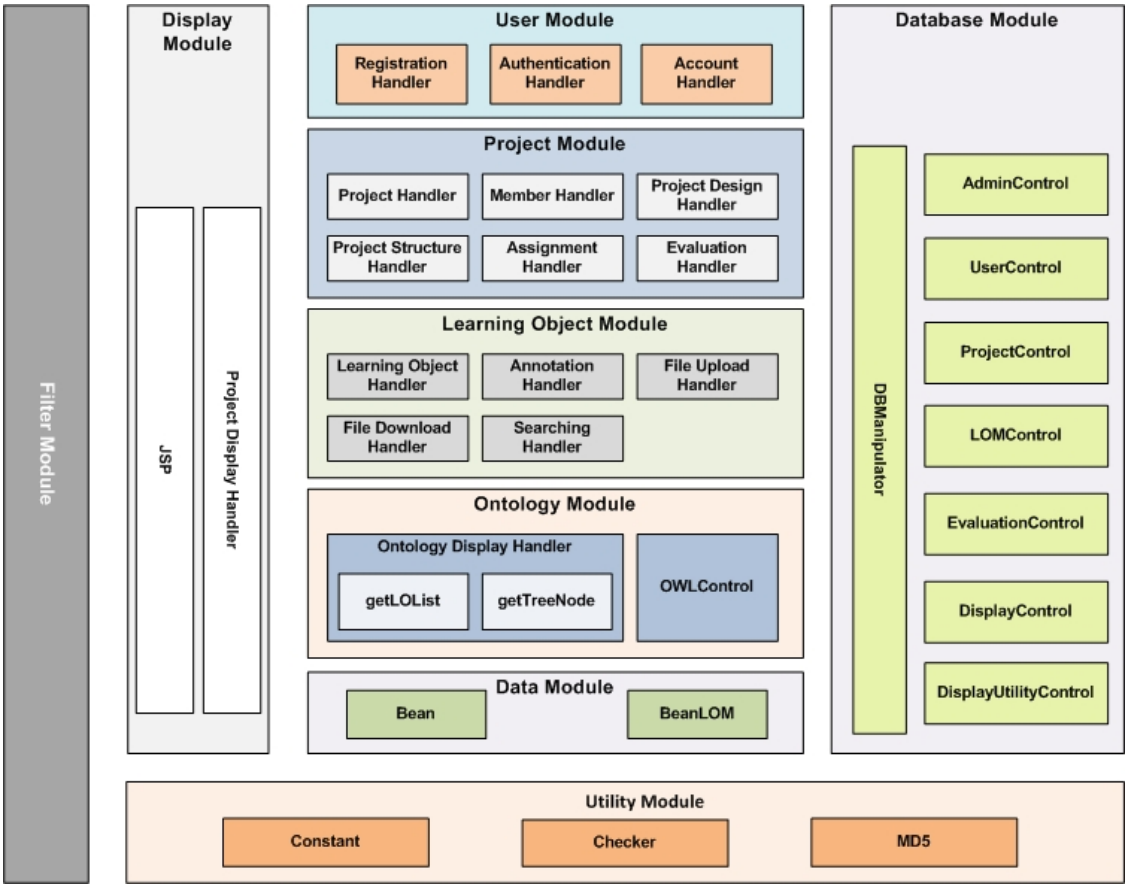


Figure 4.1 System Architecture of this Project-Based Learning Environment

User Module

In this environment, many users such as supervisors and undergraduates can manage their projects and learning resources. This module manages their user accounts and information. Also, it provides services and functions for other modules to access or process the user information.

Project Module

Many different kinds of projects would be managed in this environment. And each has different information like project group members, project design process and etc. This module manages projects and projects' information which provides services and functions to manage the projects effectively.

Learning Object Module

Large amount of learning objects are involved in each project. To enhance the management of learning objects, this module is designed to manage the learning objects in this environment.

Ontology Module

Each project has its own project ontology model. Also, the ontology models are needed to work with Jena API. Hence, a module is designed to provide services and functions to manipulate and to display the project ontology models.

Database Module

Since the information of users, projects, learning objects and etc is stored in the database, this module is designed to provide services for other modules to access

the system database records. This can enhance an efficient maintenance of this environment.

Utility Module

This module contains different Java classes providing the functions, which are used within the system. This can reduced the code duplication and improve the maintenances of the system.

Data Module

This module is used to handle different kinds of data for JSP to display the information. Two packages Bean and BeanLOM including all JavaBeans data models in this environment. One is used for handling users' information or different kinds of project information. Another is used for handling Learning Objects' LOM information.

Filter Module

This module is used to filter the request and response among client-sides and server-sides. In this system, the web-pages are organized into different folders according to their properties and authentication levels. When any request from client-sides, it would check the users' information before directing to other web-page or modules. In addition, as the data and JavaBeans passing within the system are stored in the Session, it is also used to manage the live time of different data and JavaBeans.

Display Module

This module contains all JSP for displaying interfaces for users in the web browsers. It interacts with other modules to display or retrieve information for providing services in this environment.

There are nine modules in the system, which works and interacts among modules to process and manipulate different kinds of data and functions. The detail information of the modules can refer to Appendix A3. The following Figure 4.2 shows the relationships between these modules.

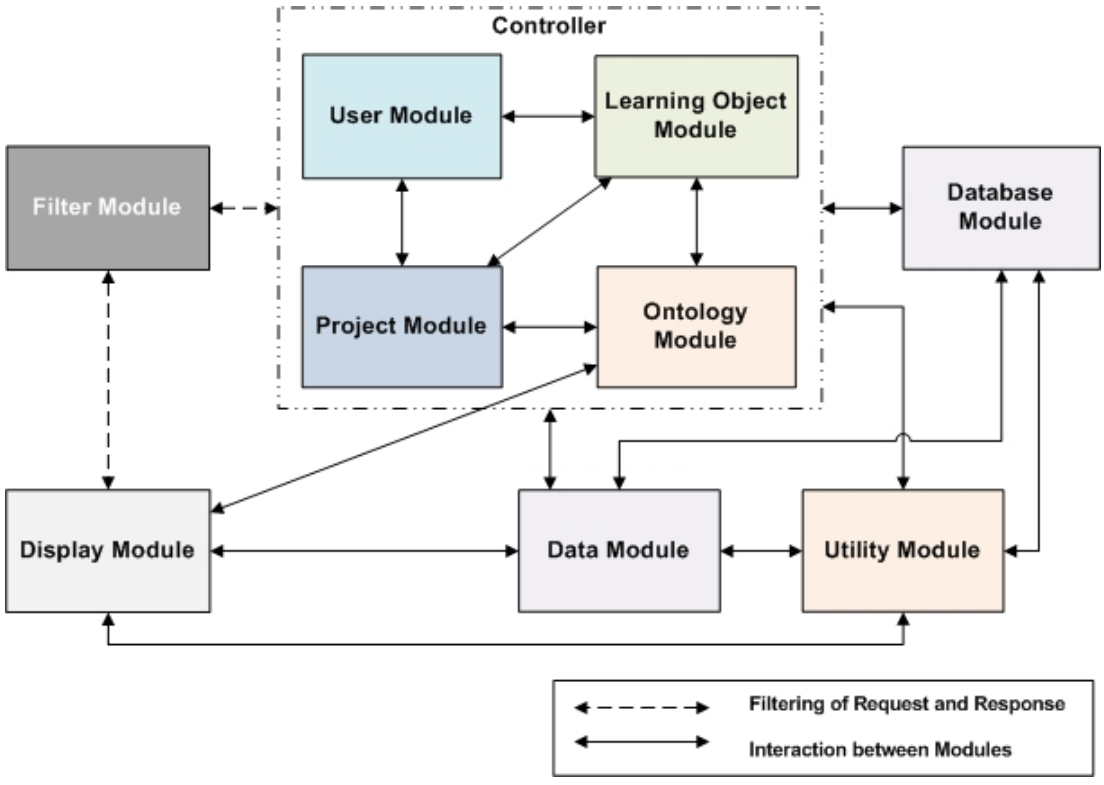


Figure 4.2 Showing the interactions between Modules

4.2 Representation of Learning Objects

4.2.1 Learning Object Metadata Structure

The original IEEE LOM standard has totally nine categories with more than seventy data elements. This may not efficient to describe all the learning objects involved in this learning environment. Although it could automate the process in filling some metadata, it would be hard for supervisors and undergraduates to learn to define all metadata by themselves.

Moreover, IEEE Learning Technology Standards Committee (2002) stated that a conforming LOM metadata instance may contain extended data elements or no value for any of the LOM data elements. Therefore, only the following five categories with eliminated data elements of IEEE LOM structure will be used in this environment.

1. General Category
2. Life Cycle Category
3. Technical Category
4. Educational Category
5. Annotation Category

In addition, two extended categories, Log and Object, are used to describe additional information of the learning objects. They facilitate the functionalities and management of different kinds of learning objects within this environment. The following is the table of the detail information of LOM structure.

Table 4.1 Explanation of the categories and the data elements of each category

Description of LOM		
General	**Identifier** *(automate)*	**LOM_ID** is used in this environment as a unique key to represent the learning object.
	Catalog *(automate)*	Designator of the identification for this entry. "URI" is used in this environment.
	Entry *(semi-automate)*	Actual URI of the learning object
	Title	Title of the learning object
	Language *(semi-automate)*	Language of the learning object
	Description	Description of the content of the learning object defined by its owner in this environment.
	Keyword *(semi-automate)*	Keywords or phrases describing the topic of learning object.
Life Cycle	**Version**	The edition of the learning object.
	Status	The completion status or condition of the learning object. (NA, draft, final, revised, unavailable)
	Role	Kind of contribution. (NA, author, publisher, editor)
	Entity	The name of the Role contributing to the learning object.
	Date	The date of the contribution.
Technical	**Format** *(automate)*	Technical data type (MIME) of the learning object.
	Size *(automate)*	The size of the digital learning object (upload file) in bytes.
	Location *(semi-automate)*	A string "URI" that is used to access the learning object.
	Installation Remark	Description of how to install the learning object.
Educational	**Learning Resource Type**	Specific kind of learning object. (NA, exercise, Simulation, questionnaire, diagram, figure, graph, index, slide, table, narrative text, exam, experiment, problem statement, self assessment, lecture)
	Typical Learning Time	Approximate or typical time it takes to work with or through the learning object for the typical intended target audience. (Hour unit)
	Language	The human language used by the typical intended user of this learning object.
Annotation	**AnnoID** *(self-define, automate)*	A unique identify of the annotation of the learning object in this environment.
	Type *(self-define, automate)*	Use to identify the annotation owner ("New" represents first comment given by owner; "Comment" represents comments given by others)
	Entity *(automate)*	The User Name and User ID of the user in this environment who give the annotation. (vCard format)
	Date *(automate)*	Date that the annotation created.
	Description	The content of the annotation.

Description of LOM		
***Log** (self-define. automate)*	**ProjectID**	Project ID of the project containing the learning object in this environment.
	UserID	User ID of the user created the learning object in this environment.
	SuperClass	Super class of the learning object belongs to in the project. (Pre-defined class)
	SubClass	Direct class of the learning object belongs to in the project. (Pre-defined or user-defined class)
	Type	Indicate whether exist in the project.
	Create	Date that the learning object created or removed.
***Object** (self-define)*	**Type** *(automate)*	Type of the learning object in this environment (link or file).
	Grade	Grade of the learning object given by supervisor.
	Path *(semi-automate)*	URI or file name of the learning object.
	Create *(automate)*	Last modified date of the learning object.

** automate – auto generate in this environment*
** semi-automate – auto generate in some kinds of learning object in this environment*
** self-define – data element or category define in this environment*

4.2.2 Management of Learning Objects with LOM

In Figure 4.3, it shows the learning objects which are described with two different forms of LOM in this environment. Some learning objects are only described with the General category; all assignments and other learning references are described with all LOM categories.

All the learning objects in the projects are defined as the above LOM structure. Supervisors and students can manage and describe the learning objects according to their levels of understandings with this format. It helps them to remember and summarize what they have learned from the learning objects.

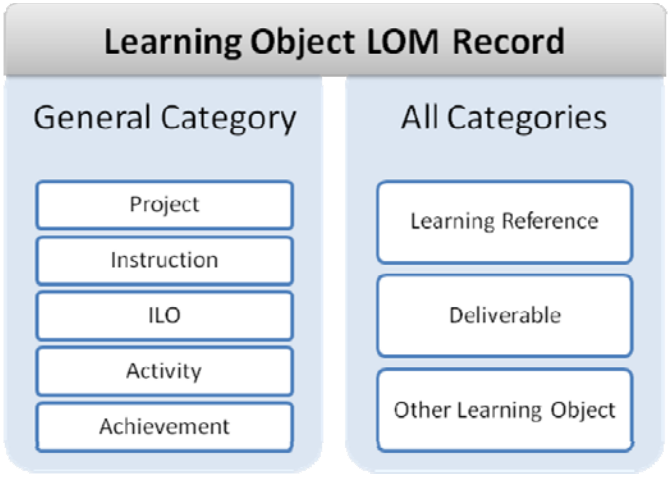


Figure 4.3 Two Forms of LOM describing Learning Objects

With the descriptive information, project group members can also review the learning objects effectively as they can be easily to know the general ideas about the content of each learning object. This can save their time in searching and studying large amount of learning resources during the projects' learning process.

Furthermore, supervisors and students can discuss about the learning objects by adding annotations. Since using LOM enables centralizing the learning objects in this environment, different project groups sharing the same learning objects can also exchange their ideas, which enhance the knowledge sharing. Figure 4.4 illustrates how ideas and knowledge sharing in this environment.

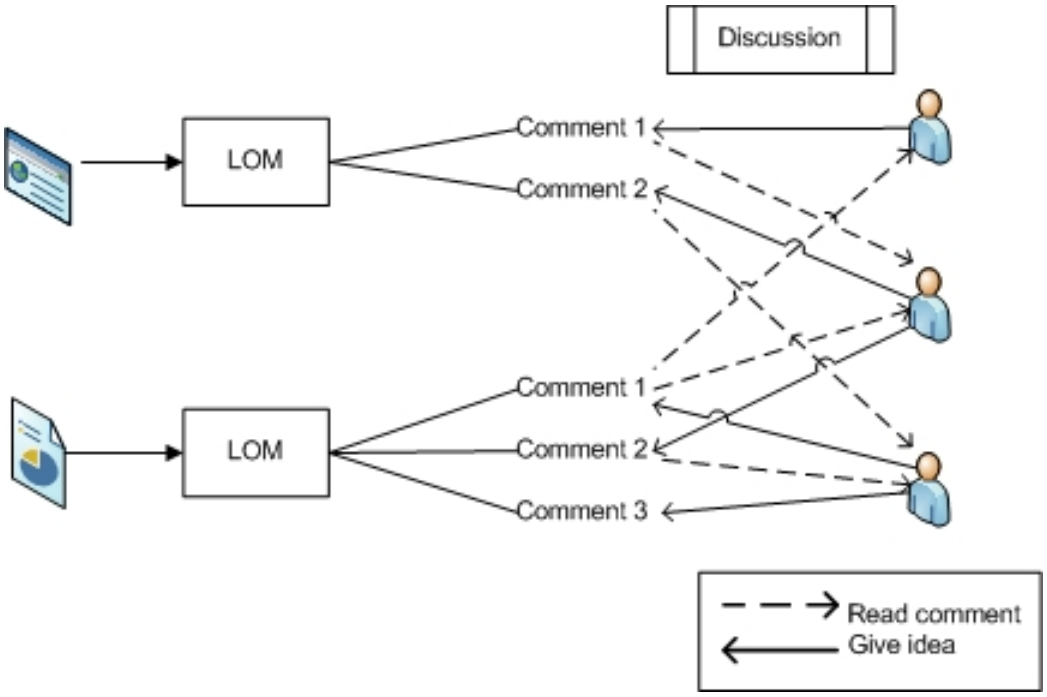


Figure 4.4 Supervisors and students discussing the learning objects with LOM Annotation

Discussion among supervisors and students can facilitate collaborative learning. They can share the ideas and help others to learn the concepts of domain knowledge through discussions. Hence, students can gain knowledge in-depth during the learning process of Project-Based Learning.

4.3 Representation of Projects

4.3.1 Knowledge of Project Basic Structure

As mentioned in Section 3, each project have different phases and processes. In this environment, the project basic structure is defined basing on the conceptual design of Project Structure mentioned in Section 3.4.

This Project Basic Structure is implemented as an ontology model. Different kinds of learning objects are organized with this model. And each learning object is referenced by its unique identifier, LOM ID, which is the actual data recorded in the model. They are classified as the instances of the following eleven main classes:

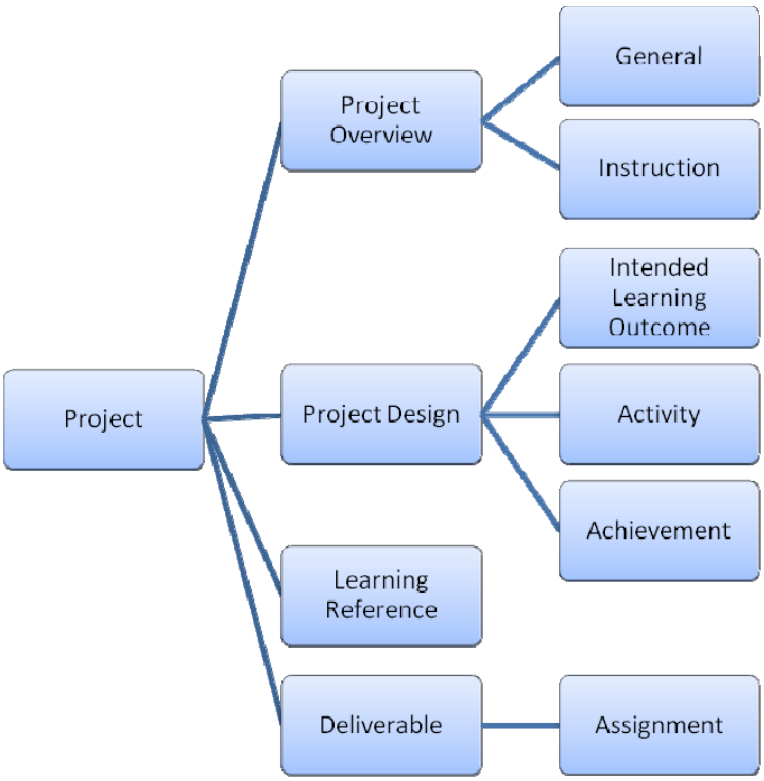


Figure 4.5 Project Basic Structure (Initial Project Ontology Model)

The following table shows the detail information of different classes, which represent different conceptual knowledge in each project.

Table 4.2 Explanation of the classes and sub-classes

Description of Project Basic Structure		
Project		Include the learning objects of Project Overview, Project Design, Learning Reference and Deliverable in the ontology model.
Project Overview		Provides project general information, which help project group members to work on and understand the project.
Sub-Class	**General**	Refer to the background, objective, general description and keywords to describe a project.
	Instruction	Defined by supervisors which can be any instructions or guiding questions for project group members to follow and think in-depth.
Project Design		Provides information about the design process, which gives clear working flow for project group members to manage and review a project.
Sub-Class	**Intended Learning Outcome (ILO)**	Descriptive text states the learning outcomes expected at the end of a project.
	Activity	Descriptive text states the tasks in the learning process. They can refer to some learning resources such as exercises, questionnaires and etc that project group members should be performed.
	Achievement	Descriptive text states the learning outcomes of a project. They can be some skills, knowledge, reports and etc which the project group have learned and achieved in a project.
	Relationships between Activity, Achievement and ILO instances indicate how project group members achieve the project expected learning outcome. *Relationships between Activity, Achievement and other instances indicate how project group members perform or finish those instances related to specific Activity or Achievement*	
Learning Reference		Learning resources such as reports templates, videos, electronic books and etc. They are the learning materials studied by project group members which help to work on the projects.
Deliverables		Learning resources of a project's products.
Sub-Class	**Assignment**	Learning resources related to the assignments given by supervisors. They are used to guide project group members to do their assignments. All sub-classes of Assignment are the assignments submitted by project group members.
User-defined Class		Learning resources are organized according to the knowledge management of the supervisors and project group members.

4.3.2 Classification of Learning Objects with Ontology

This Project Basic Structure defined in the ontology model provides a generic framework to manage different kinds of learning objects. This enhances the possibility and flexibility for managing different kinds of projects within this web-based environment.

Those pre-defined classes of learning objects such as instruction, intended learning outcome, activity, achievement and assignment, provide the basic ideas for supervisors and students to learn how to manage and organize their projects. In addition, supervisors and students can define new sub-classes to classify different kinds of learning resources. This facilitates the knowledge management of the projects. In Figure 4.6, the learning resources are organized with different classes (Conceptual Knowledge) defined by the project group members.

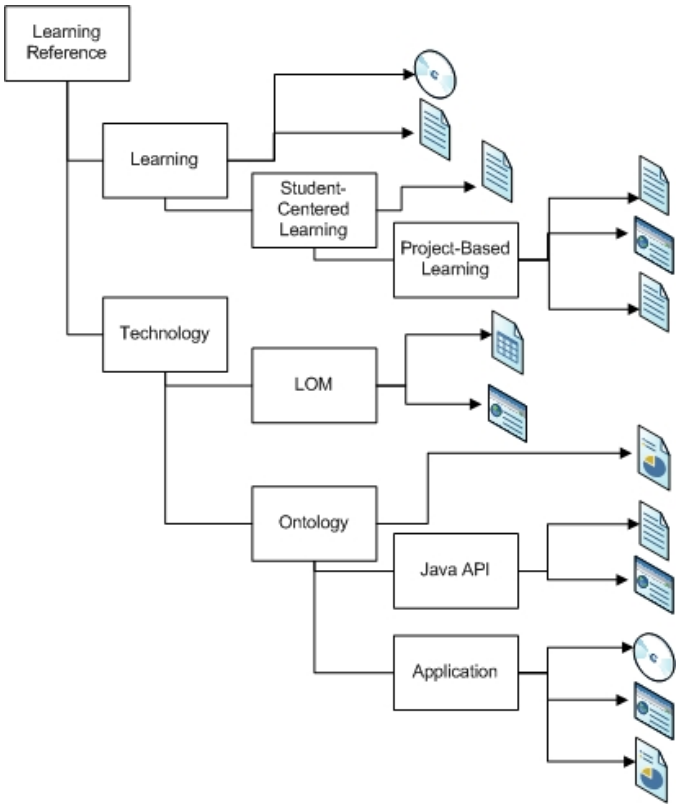


Figure 4.6 Classification of learning objects with ontology model

Since the learning objects are classified into different classes (Conceptual Knowledge), the project group members can easily study and review the learning objects by searching specific class of learning objects. This enhances the effectiveness of organizing and sharing different learning objects and ideas within a project group. Also, the project structure representing with ontology model can help showing the hierarchy of the project. This can help supervisors and students to review and understand the project knowledge.

4.4 Projects in Project-Based Learning Environment

Below diagram, it illustrates the representation of a project with using the Project Ontology Model and Learning Object Metadata.

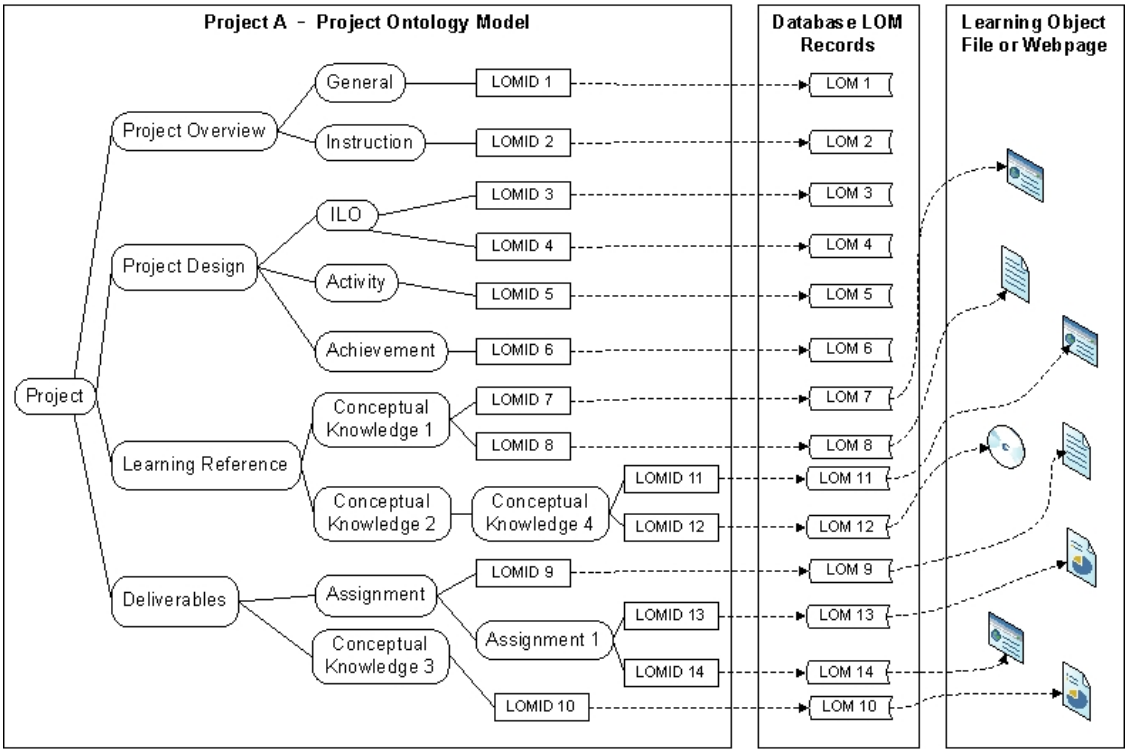


Figure 4.7 Illustration of the mapping between Project Ontology Model and LOM records

Each project has a unique project ontology model containing the initial project structure at the beginning. Supervisors and project group members can define new classes to represent different conceptual knowledge, thus, the project ontology model is able to organize different learning objects.

In the project ontology model, all learning objects of the project are managed with their LOM ID as in Figure 4.7. Also, they are classified into different conceptual knowledge. With the unique identifier of the learning objects, LOM ID, the project

ontology model can map with the learning objects' LOM records in database. Also, some learning objects are mapped with the file or webpage with their LOM records.

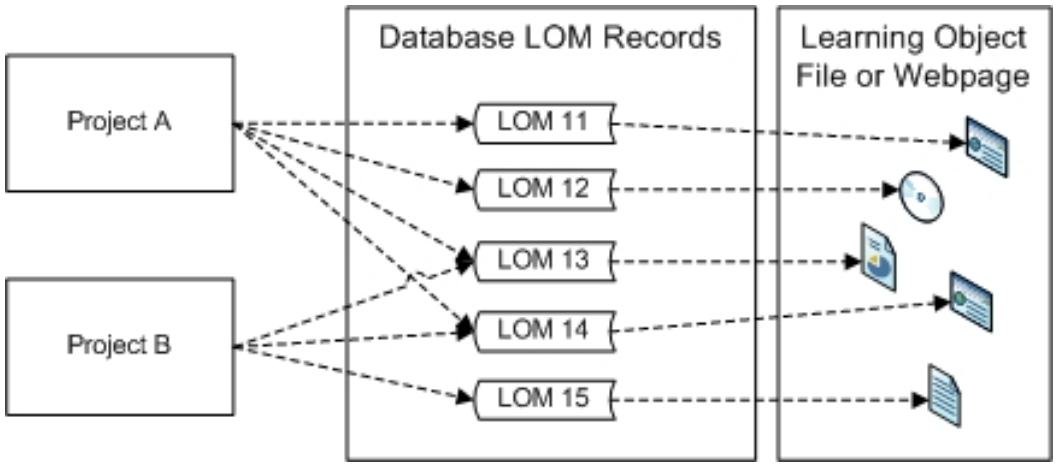


Figure 4.8 Illustrate two projects sharing some learning objects

Since there are numbers of projects in this environment, some of the learning objects are shared by different projects. If the LOM of learning objects are recorded in each project ontology model, it is inefficient that the data is required to modify in each model for any change. Hence, each project ontology model of a project only records the LOM ID of each learning object. This enhances the efficiency since it can reduce the duplications of defining LOM of some shared learning objects in different projects.

4.5 System Functionalities

In this section, the process of managing a project with the system functions is demonstrated. The following Figure is the Use Case diagram of the main functionalities introduced. The overview of all functions can refer to Appendix A1.

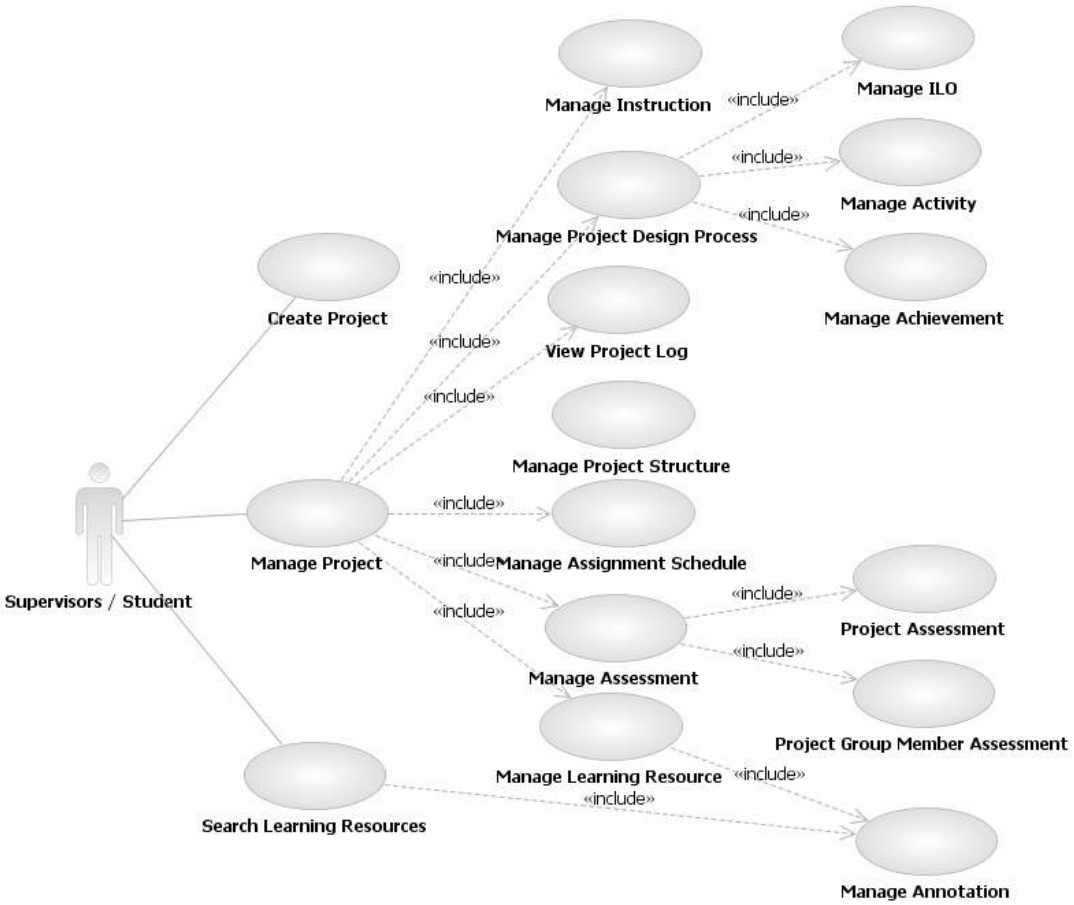


Figure 4.9 User Case Diagram of main functions

4.5.1 Project Information Management

Create Project

Supervisors can assign projects for undergraduates by creating a new project or duplicate from the existing one. They can define the project title, background information, objectives, deadline, general description and keywords to describe and initialize a new project. They can add registered users to form a project group to work on the project. The project and related information are stored into the database. Also, its initial project ontology model is constructed for organizing the project structure.

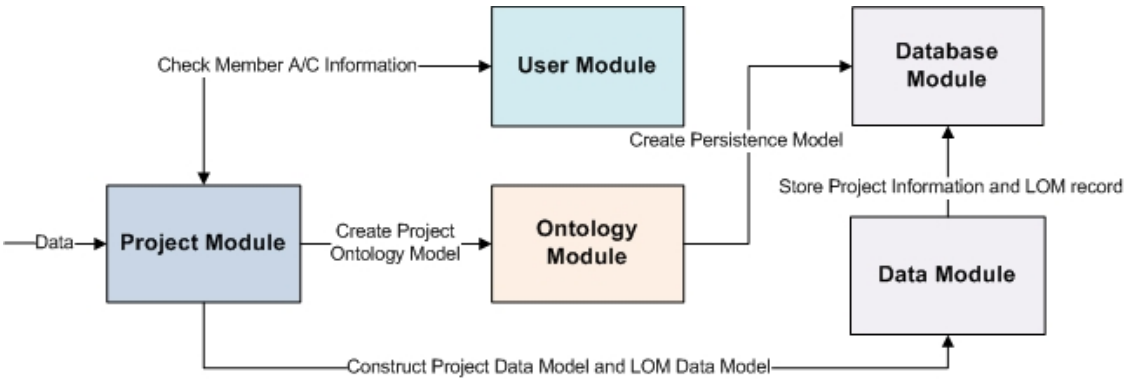


Figure 4.10 Interaction among modules to create new project

Manage Instruction

Supervisors can provide some instructions or guiding questions to the project group. Then the project group members can understand the direction of doing a project.

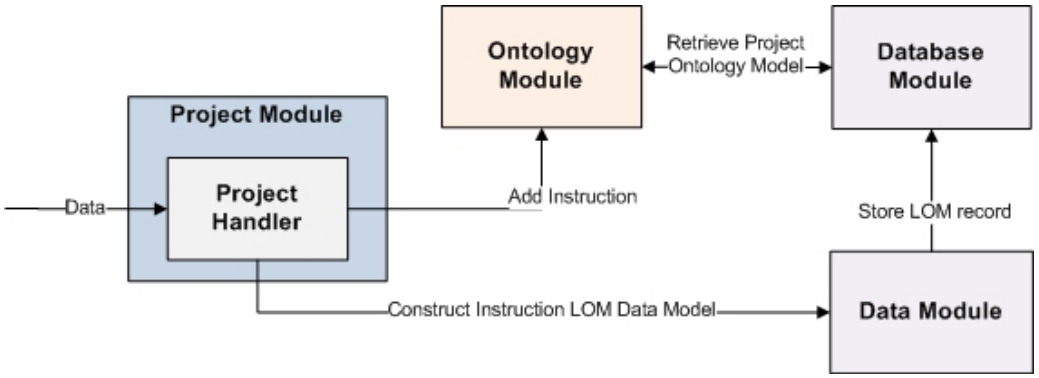


Figure 4.11 Interaction among modules and handlers to create an Instruction (learning object)

4.5.2 Learning Process Management

Manage Intended Learning Outcome (ILO)

Supervisors can manage ILO as some descriptive text, which state some projects' goals. Project group members can also manage ILO as their personal or additional goals. Each ILO is also a learning object which is managed with the project ontology model and LOM records as Figure 4.11.

Manage Activity

Supervisors and project groups can manage activities as some descriptive text, which state the learning tasks in a project; meanwhile, they can map the activities

with some ILO and learning resources. Those mappings indicate the learning resources needed to perform or study to achieve specified ILO.

Manage Achievement

Supervisors and project groups can manage achievements as some descriptive text, which state the learning outcomes in a project. Also, they can map the achievements with some ILO and learning resources. Those mappings indicate the learning resources have finished which achieve specified ILO.

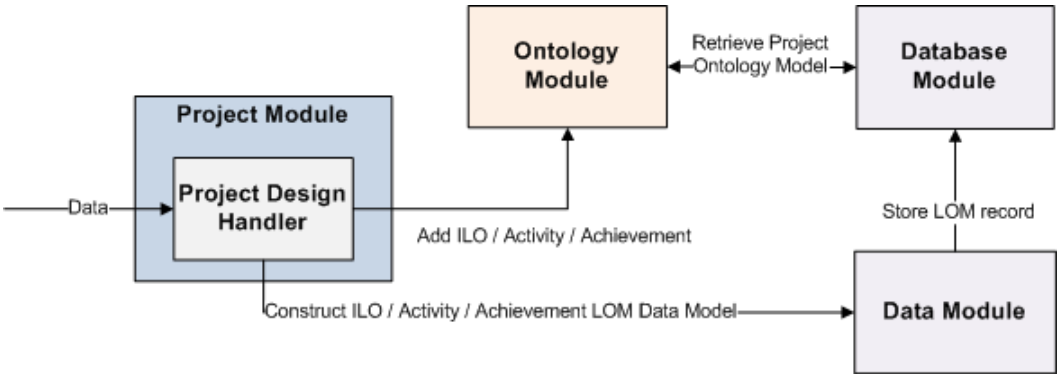


Figure 4.12 Interaction among modules and handlers to create an ILO, Activity and Achievement (learning object)

View Project Log

Project group members can record their working progress with the project logs. They can add, edit or retrieve their project logs to review the working progress. This can also let supervisors to understand their learning process such that they can give suitable supports.

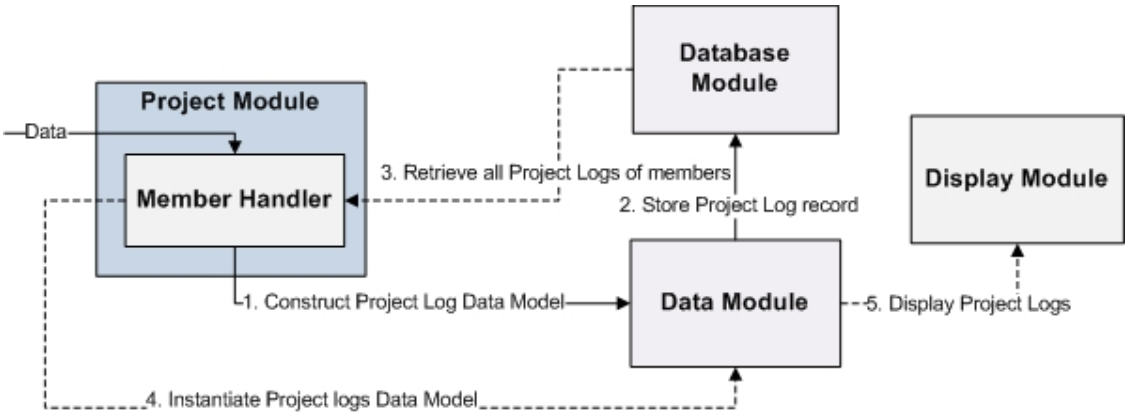


Figure 4.13 Illustrate the flow of the project log management

4.5.3 Learning Resources Management

Manage Project Structure

Supervisors and project group members can manage project structure based on the Project Basic Structure mentioned in Section 4.3.1. They can add some new classes representing the projects' conceptual knowledge with some descriptive information. Different kinds of learning objects can be classified and managed with the user-defined sub-classes.

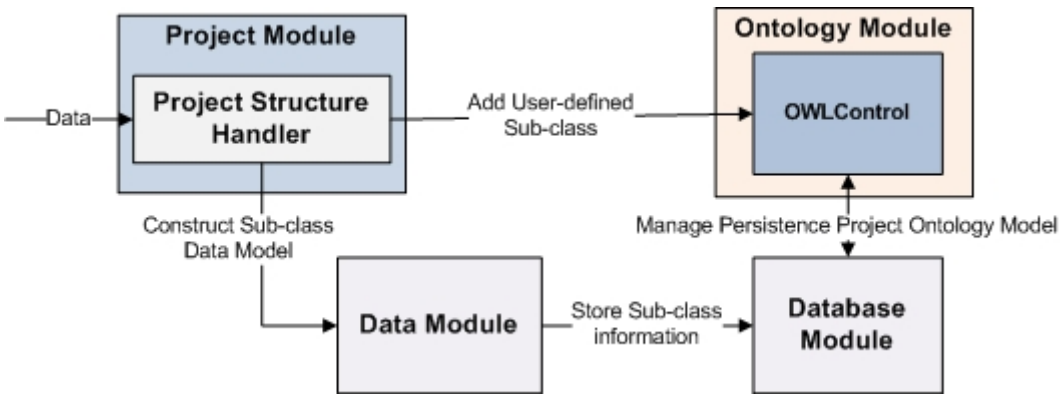


Figure 4.14 Illustration of a new sub-class adding into the project ontology model

Manage Learning Resource

Supervisors and project group members can upload files or provide URI of different learning resources in a project. They are required to define and manage the learning resources with LOM. It is easier for others to download and review the learning resources in this environment.

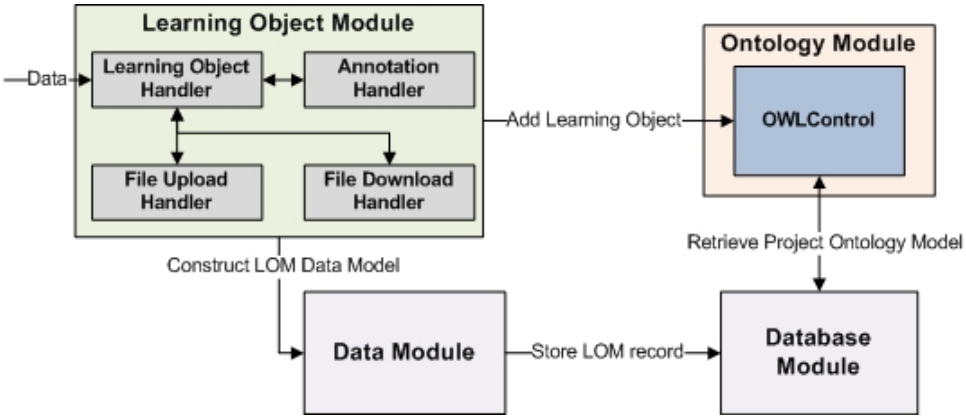


Figure 4.15 Illustration of learning resource adding into the project ontology model

Manage Annotation

Learning resources' owners can add their annotation to describe the contents according to their understandings about the learning resources. Supervisors and project group members can also give feedbacks or comments on different learning resources to discuss and share their knowledge and ideas.

Search Learning Resources

Supervisors and project group members can search learning resources which have been used in their projects. In addition, they can specify the keywords, classifications in different projects, type of learning resources to eliminate the searching results.

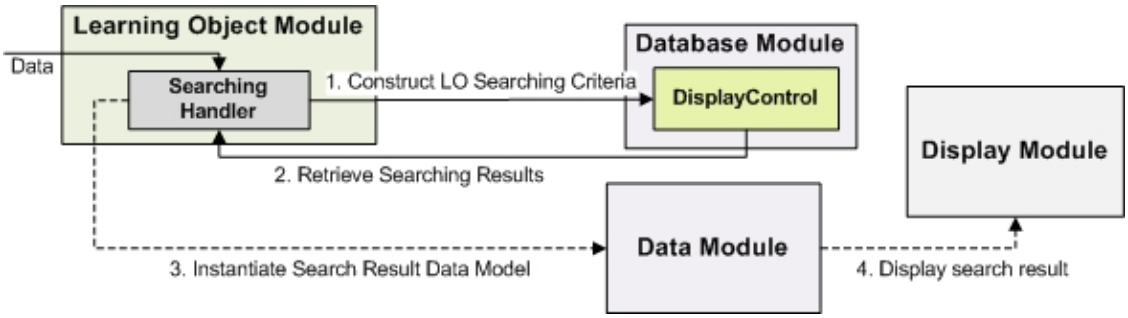


Figure 4.16 Interaction among modules for searching learning resources

4.5.4 Learning Outcomes Management

Manage Assignment Schedule

Supervisors can set the project assignment schedules, in which they can define assignment detail such as individual or group assignment, deadline and description. They can also provide some learning materials related to the assignment. Then the project group members can submit their assignment according to the schedule.

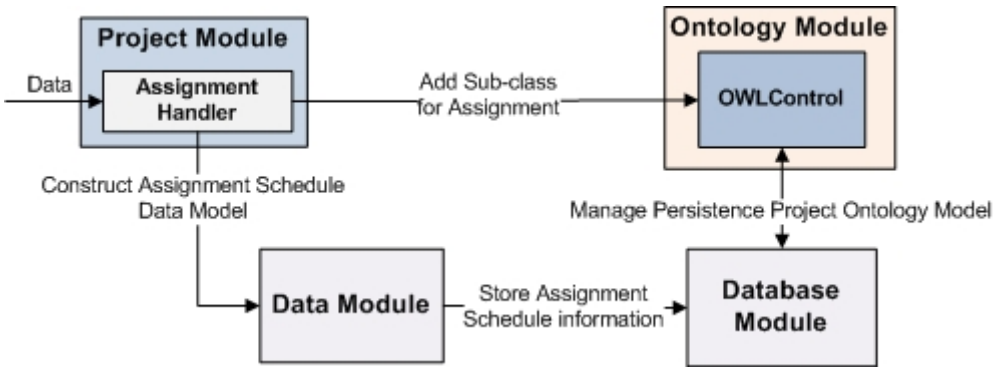


Figure 4.17 Illustration of new project assignment schedule adding in the project

Project Assessment

Supervisors can evaluate the learning outcomes of the projects. They can review the statistics about the number of learning objects studied. All assignments of the projects are organized that supervisors can easily give grade for each assignment.

Project Group Member Assessment

Supervisors can review the learning objects' statistics and assignments as above, but this indicates the performances of each project group member. Supervisors can give the grades and evaluation feedbacks.

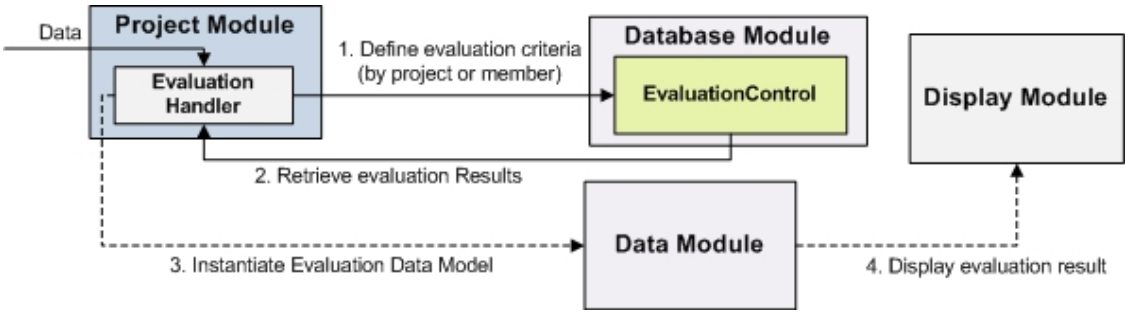


Figure 4.18 Interaction among modules for collecting evaluation results

Chapter 5. Implementation and Technologies of Project-Based Learning Environment

5.1 System Implementation

This whole system is self-developed with Java programming language. The detail configurations as follows:

1. Web-Server
   - Apache Tomcat 5.5.20 (xampp-win32-tomcat-addon-5.5.20-2.2.4 package)
2. Database
   - MySQL 5.0.45 (xampp-win32-tomcat-addon-5.5.20-2.2.4 package)
3. Programming
   - Java Servlet and JSP
   - Additional API: Jena 2.5.3 API, Ext 2.0 API, JSTL, mysql-connector-java-5.0.6, apache-commons-fileupload-1.2, apache-commons-io-1.3.2.

Java Servlet and JavaServer Pages (JSP) are used to implement this Project-Based Learning Environment. They are the free Java platform technologies for developing web-based applications. Java is a programming machine independent language. There are different kinds of free software and API provided for development. In addition, servlets are server-independent and platform-independent which enhance the flexibilities of the system development.

Apache Tomcat 5.5.20 is also an open source. This is a web-container and application server which provide environment for developing and deploying servlets and JSPs. Moreover, MySQL 5.0.45 used is an open source SQL database management system. The driver mysql-connector-java-5.0.6 is installed to enable the connecting of MySQL server and servlets.

Furthermore, MySQL supports several storage engines, which are the handlers used for different table types. Since large amount of data are needed to store and access, InnoDB Engine is used in this environment. It provides transaction-safe tables which has recovery capabilities. Hence, it can enhance multi-user concurrency and performance of this environment.

5.1.1 Use of Modal-View-Control (MVC) Framework

This system is implemented with the Model-View-Controller (MVC) framework. This enhances the maintainability of the system as the Interfaces, controller and data models are separated.

In Figure 5.1, when user requested to update the project information, the Controller interacts with the Data Model to update the project information in corresponding data sources. After the data is updated, the Controller and Data Model instantiate specific JavaBeans that the JSP can display the updated information.

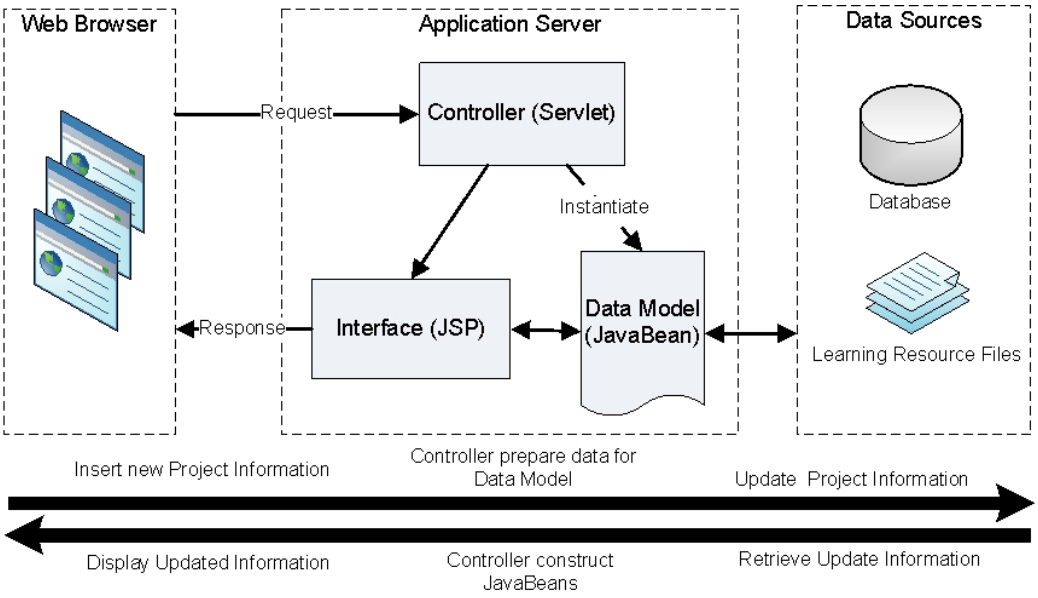


Figure 5.1 Data Flow in this system

User Interfaces

Since many different kinds of data such as user information, project information and learning resources information, are transferred in between the client-sides and server-sides, it is difficult and inefficient that the JSP directly access the data in database. Therefore, the data is manipulated with Data Model and represents as JavaBeans. The JSP just handles the instantiated JavaBeans to display the contents.

Controller

There are different Controllers in processing users and system requests, which are developed with servlets. The Controllers interact with the Data Model to process different kinds of data in database or files in the web server. They instantiate and prepare the JavaBeans for JSP to response the users' requests. Also, different Controllers work together to direct the flow of data or web-pages.

Data Model

Different Modules in Data Model are used to manipulate the data in the system or database. According to the requests, the Controllers interact with the Data Model to process the data.

5.1.2 System Security

System Authentication

In this Web-Based Environment, it can only allow registered users to use. Every user has to login with a correct User ID and Password. In addition, the password of each user account is encrypted with MD5 class before storing into the database. MD5 is an open source java class providing one-way encryption. Therefore, it can enforce better protection of user accounts.

For visiting each web-page of this environment, the Filter Module authenticates the user identities and roles. It ensures that the web-pages can only be accessed by authorized users.

Input Data Control

In this environment, users always have to fill in some forms to manage the information of projects and learning resources. It is necessary to prevent from entering invalid data that may cause errors. Therefore, the forms are validated by some JavaScript functions in client-sides before posting data to the server-sides.

Moreover, the data are validated again in server-sides such that it can prevent the failures of JavaScript functions in client-sides. For the input data using in SQL

queries, checking is also processed by the Database Module to prevent SQL injections.

Safeguard Learning Resource

Some learning resources are files uploaded by users. It is difficult to handle large amount of files storing in this environment. Also, it would cause errors if some uploaded files have the same filename. Hence, the File Upload Handler changes the filename to the MD5 encrypted LOM ID of each learning resource. This not only prevents duplication of filenames, but also prevents users to access the learning resources with the filenames directly.

Furthermore, some learning resources can only be shared among project group members that protect the projects' confidentialities. In addition, some of them such as individual assignments should be only accessed by the project supervisor or the owner. Hence, the File Download Handler authenticates users according to the type of learning resources. It can block unauthorized users to download the learning resources files.

5.2 Implementation of Learning Object Metadata (LOM)

As mentioned in Section 4.2, all the learning objects in this environment are described with the tailor-made LOM based on IEEE 1484.12.1-2002, Learning Object Metadata standard. Seven categories of LOM: General, Life Cycle, Educational, Technical, Annotation, Log and Object are implemented as seven separate tables in database. It can enhance the concurrency of the tables, since the LOM of learning objects may be only accessed or updated in some of the categories. The detail of these tables can refer to Appendix A2.

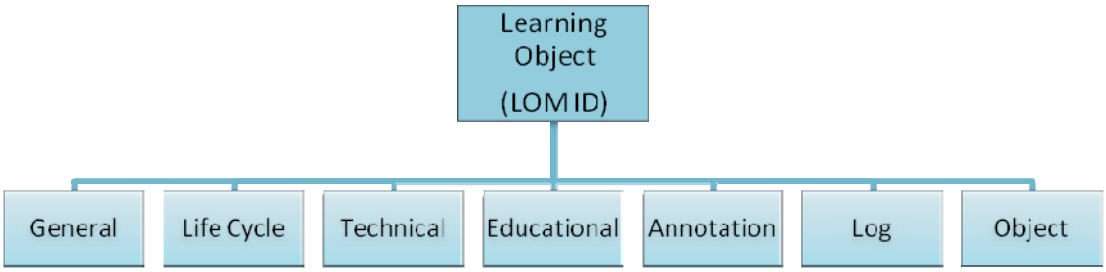


Figure 5.2 Learning Object Metadata using in Project-Based Learning Environment

Since learning objects can be shared among different projects, some additional information is required to recognize their information and statuses in different projects of the environment. The following is the categories added to manage learning objects which support the system functionalities.

1. Log Category
   - This mainly indicates the classification of a learning object in its belonging projects, its status (New /Remove), its creator and creation date.

2. Object Category

- This records the information of learning objects in this environment which mainly provide the information for download and project assessment.

In this environment, some of the LOM data is defined by users when the learning objects are added into a project; some of the LOM data is automatically defined by the system. Hence, the web interface for filling the LOM data is organized in one form of the webpage, which enables users to fill the information easily.

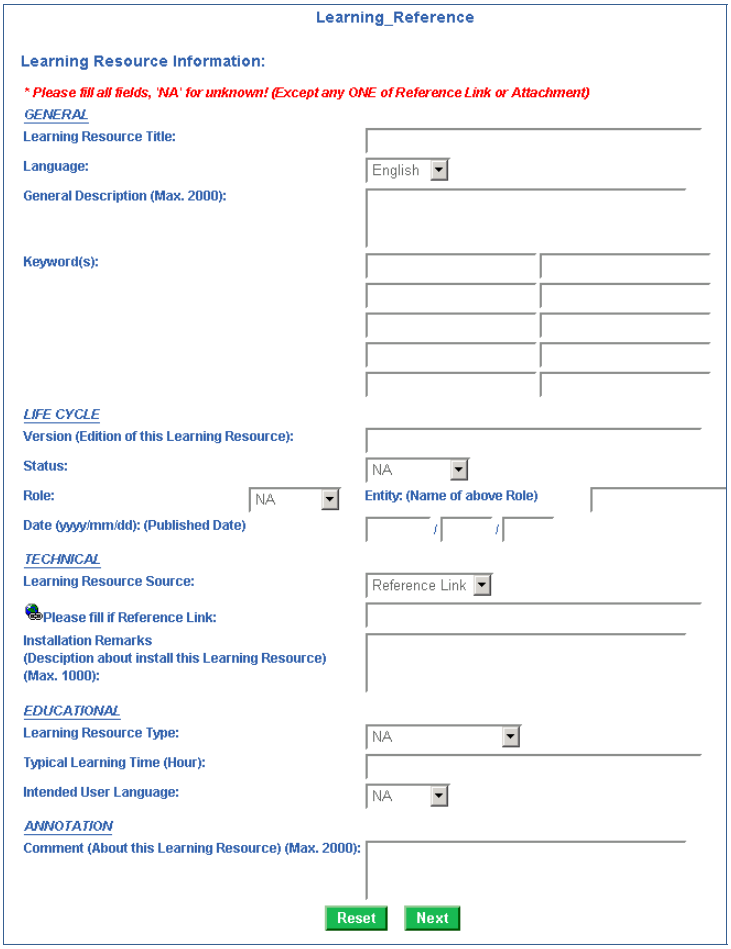


Figure 5.3 Web interface for filling the Learning Objects' LOM

On the other hand, as mentioned in Section 4.4.2, each learning object of Project, Instruction, ILO, Activity and Achievement are only organized with General Category. Also, they are the learning objects which may be only used in this environment. The LOM of these learning objects are defined as the following format:

Learning Object	LOM_ID	Catalog	Entry	Title	Language	Description	Keyword
Project	Auto-generate	"URI"	Host of this environment	Name of Project	"zh"	Project Description	Project Keyword
Instruction	Auto-generate	"URI"	Host of this environment	Title	"zh"	Content	"***Project ID***: Instruction"
ILO	Auto-generate	"URI"	Host of this environment	Title	"zh"	Content	"***Project ID***: ILO"
Activity	Auto-generate	"URI"	Host of this environment	Title	"zh"	Content	"***Project ID***: Activity"
Achievement	Auto-generate	"URI"	Host of this environment	Title	"zh"	Content	"***Project ID***: Achievement"

Table 5.1 LOM format of Project, Instruction, ILO, Activity and Achievement learning objects

General - Keyword

According to IEEE LOM standard, each learning object can have more than one keyword or phrase to describe its topic. However, it is difficult to manipulate the keyword as each learning object would have different number of keywords. Hence, maximum ten keywords can be defined in each learning object, which is the minimum requirement of IEEE LOM standard. Also, it would waste the storage space if the "General" table stores each keyword with a column. All the keywords of a learning object are stored in one column of the table, which is separated by "; ".

(e.g. "keyword 1; keyword 2; keyword 3; ")

Life Cycle - Entity and Annotation - Entity

"Life Cycle" and "Annotation" categories have a data element "Entity". The one in "Life Cycle" one records the information about the people or organization contributing to the learning object; the one under "Annotation" records the information about the people creating the annotation of learning objects, which is the information of the registered user in this environment. Since IEEE LOM standard requires this information to store in vCard format, the data is needed to be manipulated by the servlets before stored in the database tables. The following is the format of the data elements:

1. Life Cycle Entity
   - BEGIN:VCARD

     N:*"Name of Contributor(s)"*

     END:VCARD
2. Annotation Entity
   - BEGIN:VCARD

     N:*"Name of user"*

     NICKNAME:*"User ID of user*"

     END:VCARD

Technical Format and Size

Learning object can be uploaded or provide URI link. For those learning object with URI link, the system defines the Format: "text/html" and Size: "0" automatically; otherwise, the File Upload Handler in Learning Object Module detects the file's format and size when the learning object is being uploaded. This can reduce the difficulties and efforts of the users in finding this information of files.

5.3 Implementation of Project Ontology Model

OWL Web Ontology Language

OWL Web Ontology Language is used to define and instantiate web ontologies. It allows machine to process the information, in term of the meaning of vocabularies and relationships between the information (McGuinness and Harmelen, 2004; Smith, Welty and McGuinness, 2004).

Protégé-Frame Editor

Protégé is an open-source platform that provides tools to develop ontologies domain models or knowledge-based applications (Protégé, 2007). One of the freeware provided, Protégé-Frame editor 3.3.1, is used to construct the initial ontology model as a project basic structure. This freeware provides interfaces which facilitate defining the domain ontologies and relations among classes. And it has function to export the model to an OWL file. The detail OWL file of the initial Project Ontology Model can refer to Appendix A3.

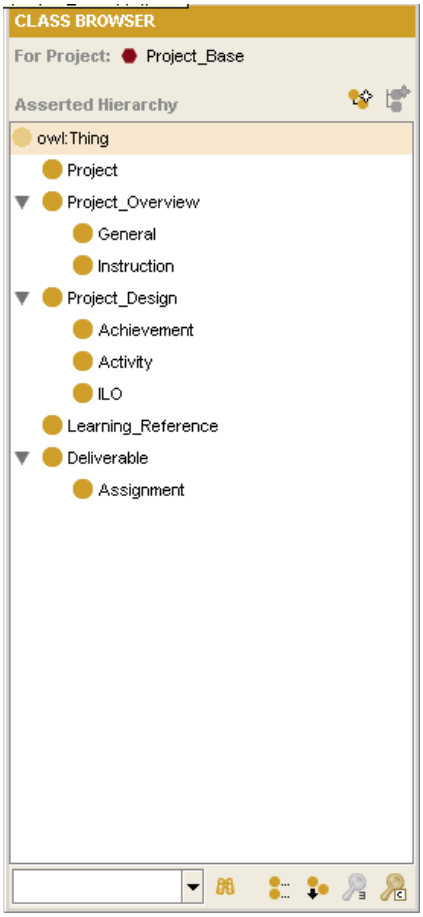


Figure 5.4 Initial Project Ontology Model constructing with Protégé-Frame Editor

Jena API

Jena is a Java framework for development of Semantic Web applications based on RDF and OWL (Jena, 2007). In this environment, Jena 2.5.3 API is used to manage the project ontology models. Using Jena API, the environment can construct the persistence OWL models in Jena repository database. It allows searching and manipulating classes and instances of different project models. Also, this API allows the ontology models importing from and exporting to OWL files.

Ext 2.0 API

Ext is a client-side JavaScript framework for developing web applications (Ext JS, 2007). Ext 2.0 API provides functions to visualize data in different formats. Using this API, the project ontology models are displayed as tree view. In addition, the nodes of the tree can interact with the Ontology Display Handler in this environment, which can retrieve the data in real-time.

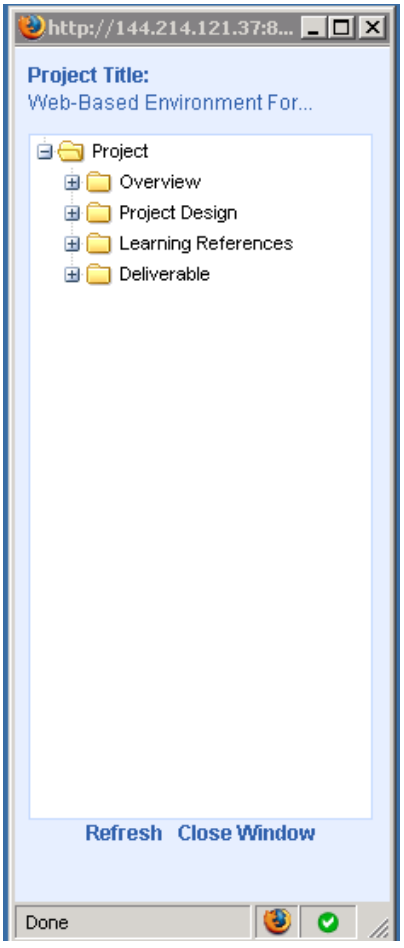


Figure 5.5 Project Ontology Model displaying as Tree with Ext 2.0 API

5.3.1 Technologies Selected for Project Ontology Model

Since the project basic structure is the same in each project, the initial Project Ontology Model can be pre-defined. Although the Jena API can be used for constructing the initial ontology model, some servlets and interfaces are required to implement as a handler to build the ontology model. Even if any change of the model, the handler must be amended for re-construct the ontology model.

It is not effective that the handler may be only used once in the system development stage. Therefore, Protégé-frame editor is chosen for constructing the initial model. It provides a user-friendly interface to build any type of domain ontologies. Also, it can visualize the ontology model that the structure of the model can be reviewed easier. This can reduce the time and efforts in designing the initial project ontology model.

Besides, Protégé provides Java API for ontology programming which include some OWL GUI plug-in for GUI display. However, this Protégé API needs to work closely with Jena API. Also, the visualized models are displayed with Java Applet which is a kind of Java program embedded in the webpage. In which, the users are required to download plug-in for the browser display. Using Java Applet may have the problems in managing the Session ID with other HTML or JSP. Also, Java Applet programs encounter problems as it cannot save the files into host server from client-sides.

Since this environment relies on the Session for passing and storing the parameters and JavaBeans, it is difficult to control the data consistency if the

Session ID of Java Applet renews every time in refreshing and directing to another webpage. In addition, the Jena API can solve the problems of manipulating ontology models, thus, it is chosen for this environment.

Furthermore, there are many open source Java class and JavaScript are provided for constructing the Tree in the webpage, which can be used to display the project structure. Nonetheless, most of them can only handle the fixed tree structure. This is unable to support the functions of this environment. Ext 2.0 API can solve this problem as it provides interfaces to generate the tree structure in real-time. It enhances the flexibilities to construct and display the free project structure in this environment.

Finally, Protégé-frame editor, Jena API and Ext 2.0 API are selected for implementing this system, based on the efficiencies, securities and flexibilities.

5.3.2 Project Ontology Model Framework

In Figure 5.6, there are three layers to manipulate the project ontology models. A Project Basic Structure is pre-defined as an ontology model in this environment. For each creation of project, Ontology Module imports the OWL file to construct a new persistence ontology model in database. To display the project structure, Ontology Module converts the model for Project Display Handlers. Then the project structure is displayed as a Tree in web browser.

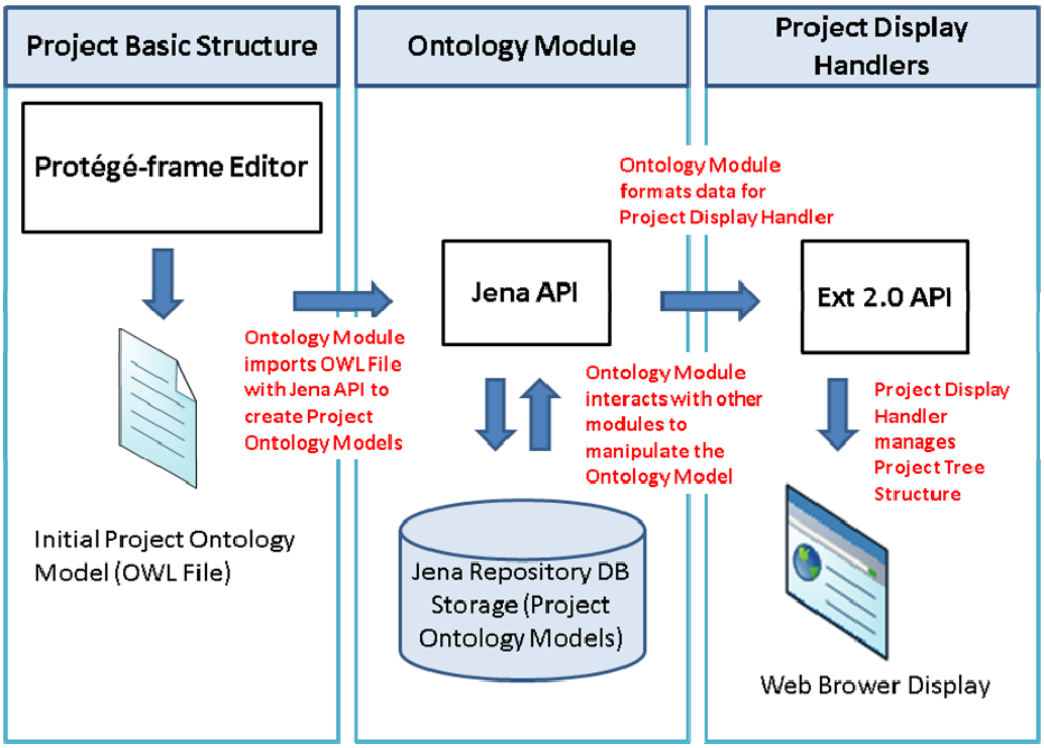


Figure 5.6 Interaction of Project Ontology Model

In the Ontology Module, it includes OWLControl and Ontology Display Handler. They control and provide the functions for other system modules to manipulate and to interact with the project ontology models.

5.3.3 Implementation of Ontology Module

OWLControl

Since Jena API mainly provides general functions to manipulate the models, classes, instances and relationships, OWLControl is implemented to handle and to construct the domain ontology for this environment. It is a java class which has numbers of functions providing for other system modules to manipulate the project ontology models. Hence, only some significant functions are introduced.

Create a Project Ontology Model

In this function, a unique Project ID and Project LOM ID are inputted. Then the pre-defined OWL file is imported to create a project ontology model. Its ontology model ID in the Jena repository database is set to its Project ID. After a project ontology model is created, a “Project” instance and “General” instance are added into it, which represent the project and the project information respectively.

Add User-defined Classes

To create a new class, the Project ID, name of the new class and name of its direct super-class, e.g. “A”, are passed into this function. The Project ID is used to identify that the new class is added into which project ontology model. By calling the Jena API function, a new class can be added as the sub-class of the direct class “A”.

Since it is difficult to handle the irregular project structure, there is some restriction on the user-defined classes. The user-defined classes can be created

as the sub-classes belonging to one of the pre-defined classes such as “Project_Overview”, “Project_Design”, “Learning_Reference” and “Deliverable”; or they can be created as the sub-classes belonging to another user-defined class.

As the user-defined classes inherent the properties of the pre-defined classes, it enables the system to handle the classes and project ontology model generically. In Figure 5.7, it demonstrates some parts of the OWL file of a project. The user-defined class “Ontology_Research” belongs to “Learning_Reference” and “Technology” belongs to “Ontology_Research”.

```
<owl:Class
    rdf:about="http://localhost:8080/ProjectManage/OntologyMod
el/Project_Base.owl#Ontology_Research">
    <rdfs:subClassOf
    rdf:resource="http://localhost:8080/ProjectManage/Ontology
Model/Project_Base.owl#Learning_Reference"/>
</owl:Class>
<owl:Class
    rdf:about="http://localhost:8080/ProjectManage/OntologyMod
el/Project_Base.owl#Technology">
    <rdfs:subClassOf
    rdf:resource="http://localhost:8080/ProjectManage/Ontology
Model/Project_Base.owl#Ontology_Research"/>
</owl:Class>
```

Figure 5.7 User-defined Classes added in a Project Ontology Model

Add Learning Objects

Furthermore, all the learning objects can be classified into different classes of the project ontology models. They are the instances of the specific classes in the model, which has the attributes of “LOMID” and “label” (title of learning object).

In this function, the title, LOM ID and direct class's name of a learning object are required to input. The learning object can be added as an instance of its direct class in a project ontology model. Also, the unique identifier of each instance is defined as the following format:

Host URI "/" *Initial OWL File Name* "#" *class of instance* "/" *LOM ID* (e.g. http://localhost:8080/ProjectManage/OntologyModel/Project_Base.owl#Ontology_Research/7)

The learning objects are added as the direct-instances or sub-instances of the pre-defined super-classes. Therefore, OWLControl provides four functions to add the instances according to their super-classes.

1. Add Project Overview Instance
   - The Project instance has relationship "hasOverview" with the instance
2. Add Project Design Instance
   - The Project instance has relationship "hasDesign" with the instance
3. Add Learning Reference Instance
   - The Project instance has relationship "hasReference" with the instance
4. Add Deliverable Instance
   - The Project instance has relationship "hasDeliverable" with the instance

In Figure 5.8, its shows a Project instance and an Ontology_Research instance, which have different relationships with the Project instance in a project ontology

model. And it shows that each learning object has the relationship “belongTo” with the “Project” instance.

```
<Project
    rdf:about="http://localhost:8080/ProjectManage/OntologyMod
el/Project_Base.owl#Project/1">
    <LOMID>1</LOMID>
    <hasReference
rdf:resource="http://localhost:8080/ProjectManage/OntologyModel
/Project_Base.owl#Ontology_Research/7"/>
    <rdfs:label xml:lang="zh">Web-Based Environment for Managing
Undergraduates Project-Based Learning</rdfs:label>
</Project>
<Ontology_Research
    rdf:about="http://localhost:8080/ProjectManage/OntologyMod
el/Project_Base.owl#Ontology_Research/7">
    <LOMID>7</LOMID>
    <belongTo
rdf:resource="http://localhost:8080/ProjectManage/OntologyMode
l/Project_Base.owl#Project/1"/>
    <rdfs:label xml:lang="zh">Interim Report</rdfs:label>
</Ontology_Research>
```

Figure 5.8 Learning Objects representing in a Project Ontology Model

OWLControl also provides a function for servlets which can check the super-class of the user-defined class. Hence, the servlets can add the learning objects according to their super-classes. With these relationships between learning objects and project, it enables the learning objects to be retrieved through the relationships instead of their classes. Most importantly, it can represent the project structure hierarchy rather than different separated instances in the project ontology model.

Add Relationships of Activity and Achievement Instances

In this environment, some activities of the project are performed to achieve some ILO. And they are performed with some learning resources. Meanwhile, some achievements of the project indicate the ILO achieved. And they finished

some learning resources. For that reason, there are relationships among the instances of ILO, Activity, Achievement and Learning Object in the project model. OWLControl provides different functions to manage these relationships.

1. Add, remove and retrieve the instances of the Learning Object related to the Activity instances.
2. Add, remove and retrieve the instances of ILO achieved by the Activity instances.
3. Add, remove and retrieve the instances of the Learning Object related to the Achievement instances.
4. Add, remove and retrieve the instances of ILO achieved by the Achievement instances.

```
<Activity
    rdf:about="http://localhost:8080/ProjectManage/OntologyMod
el/Project_Base.owl#Activity/5">
    <LOMID>5</LOMID>
    <achieveWith
    rdf:resource="http://localhost:8080/ProjectManage/Ontology
Model/Project_Base.owl#Ontology_Research/7" />
    <belongTo
    rdf:resource="http://localhost:8080/ProjectManage/Ontology
Model/Project_Base.owl#Project/1"/>
    <relateTo
    rdf:resource="http://localhost:8080/ProjectManage/Ontology
Model/Project_Base.owl#ILO/3"/>
    <rdfs:label xml:lang="zh">Ontology</rdfs:label>
</Activity>
```

Figure 5.9 Illustration of the relationships of an Activity instance in a Project Ontology Model

In these functions, the instances' LOM ID are needed to be entered for creating the relationships. With this information in the ontology model, the environment can manage the relationships among these instances without storing the extra

database records. It is more efficient as the numbers of ILO and Learning Object of each Activity and Achievement instance related are different.

Move Learning Objects

Learning objects may be classified into incorrect class. In this case, the learning objects are needed to move from its original class to another one. However, Jena API does not provide any functions to move the instances from one class to another. OWLControl is implemented with a function to solve this problem.

In this function, the learning object's LOM ID, the name of its original class and the name of the new class are inputted. Since a learning object instance may have different kinds of relationships with other instances, the unique identifiers of the related instances are recorded. Then the learning object instance is removed from the project ontology model using Jena function. The learning object is added as new instance of the new class and its relationships with other instances are added one by one with the records.

Export Project Ontology Model OWL File

Each project ontology model can be exported into an OWL file from the Jena repository. In this function, the directory of the destination and Project ID are inputted. The OWL file of the project is saved to the specified destination. In this environment, if the Project Handler checks that the project's status is changed from "Active" to "Close" as its deadline has been passed, the handler invokes this function of OWLControl to backup the project ontology model as an OWL file in the system.

Ontology Display Handler

In the following, the servlets, “getTreeNode” and “getLOList” in the Ontology Display Handler are introduced. These servlet are used to control the functions and display of the Project Structure in the web interfaces.

Initial Project Tree Structure

Since Ext 2.0 API works with the JavaScript framework, a tailor-made JavaScript file is implemented to construct the initial tree structure of the project ontology model, Project Tree. The detail of the file can refer to Appendix A5.

In the initial tree structure, it contains five main tree nodes such as Project, Overview (Project Overview), Project Design, Learning Reference and Deliverables, which represent the pre-defined classes in the project ontology model.

Retrieve Data of Project Trees

Also, the API provides some functions that the Project Tree can be implemented to invoke a servlet, “getTreeNode”, to retrieve the data of the sub-tree nodes (sub-classes and instances) by clicking the tree nodes. The node ID of the clicked tree node is passed when invoking the servlet. In case, the tree nodes are not the “leaf”, indicating that no child node belong to it.

It makes possible to generate the data in real-time and to display different project structure. The data is in JSON format and each tree node can only be

identified with the node ID, thus, the servlet needs to construct the data with the only parameter, node ID, for different classes in the model.

There are two format of tree nodes' node ID in this environment. One of which is the class in the project ontology models is set as the name of the class; another one which is the learning object is set as the LOM ID. The servlet checks the node ID and retrieve the corresponding data according to the type of the classes (node ID). And it formats the data for both its sub-classes and instances. Figure 5.10 shows some part of the codes in the servlet which formats the data for one of the types of the classes. ("id" > node ID; "text" > title for display; "leaf" > indicate whether has sub-class and instance; "cls" > CSS style)

```
String url = response.encodeURL("../ManageSubclass");
String link = "\"MainOpen(\'" + url;
String url2 = response.encodeURL("../ManageLO");
String link2 = "\"MainOpen(\'" + url2;

// construct sub-classes data
JSONObject json = new JSONObject();
json.put("id", name);
json.put("text", "<a href=# onClick=" + link + "?directClass=" +
name + "\')\">" + title + " </a>");
json.put("leaf", false);
json.put("cls", "folder");
jsonItems.put(json);

// construct instance data
JSONObject json = new JSONObject();
json.put("id", lom);
json.put("text", "<a href=# onClick=" + link + "?inclass=" + node
+ "\')\">" + name + "</a>");
json.put("leaf", true);
json.put("cls", "file");
```

Figure 5.10 Example of formatting JSON data for the tree nodes of a Project Tree

```
[{"text":"<a href=#onClick="+ link + "?directClass="+
name+"\')\">"+ title
+"</a>","id":"name","leaf":false,"cls":"folder"},
 {"text":"<a href=#onClick="+ link+ "?inClass="+ node+"\')\">"+
name +"</a>","id":"lom","leaf":true,"cls":"file"}]
```

Figure 5.11 Illustration of JSON data generated

In addition, the environment needs to direct to different JSP to display the particular information if the users clicks the tree nodes of the Project Tree. Hence, different hyperlinks, which invoke the JavaScript function to locate the web-page, are formatted with the JSON data and set in each tree node together.

Interaction with Project Display Handler in Display Module

Besides, a specific JSP is implemented to display the Project Tree which is the Project Display Handler. If the Project Tree is embedded in the web-pages and displayed in the browser's main window, the tree is being refreshed together with the web-pages. Moreover, it is ineffective to embed the Project Tree in each JSP. And it would confuse the users if the main window has too much information showing together.

Therefore, the JSP of the Project Tree is displayed with a pop-up window. The tree does not refresh together even if the web-pages in the main window change. Since the Session ID of the pop-up window in some browser may not be maintained as same as the main window, the Session ID is embedded with the hyperlink when the pop-up window is opened. Then it enables two windows to share the same Session ID such that the data and JavaBeans can be shared too.

As mentioned above, the main window of the browser is directed to different web-pages when the tree nodes of the Project Tree are being clicked. To facilitate the interaction between the main window and the pop-up window, a simple JavaScript function is designed for locating the web-pages in the main window.

```
function MainOpen(direction) {
    window.opener.location.replace(direction);
}
```

Figure 5.12 JavaScript function for changing the location of the main window of the browser

Display Project Ontology Model with getLOList

Other than displaying the project ontology model as Tree, the servlet "getLOList" enables to display the model by specific class. In the environment, specific class can be chosen for display as Figure 5.12.

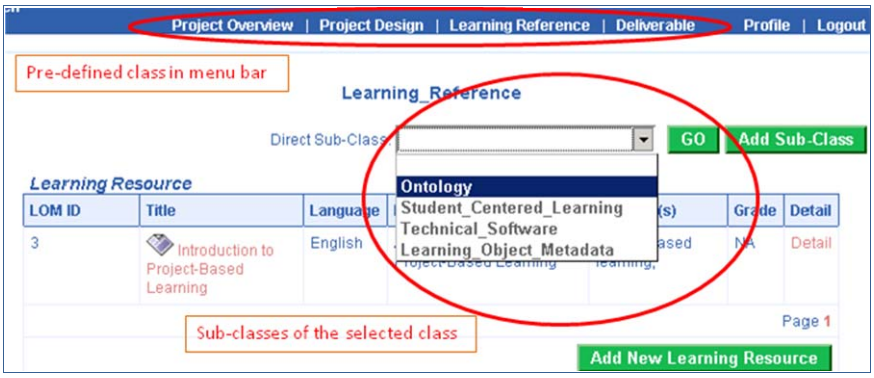


Figure 5.13 Web interface for choosing specific class to display

The servlet is invoked if the pre-defined class in menu bar was clicked or any one of the sub-classes was selected. The servlet retrieves the name of the chosen class from the request. With the Project ID which stores in the Session,

and the name of the chosen class, it calls the OWLControl functions to get the project ontology model and the ontology class of the chosen class. The servlet retrieves the list of sub-classes of the ontology class with Jena API. On the other hand, the list of LOM ID of its instances is retrieved by calling the function of DisplayControl in the Database Module. Then, it constructs the data model for the Display Module to display. Some part of the codes in this servlet is shown:

```
/** Retrieve the list of Subclass from a project model
  * @param: int Project ID, String class name (the direct class)
  * @return: String of subclass if directclass exist; otherwise null
  */
private String getSubClass(int prjID, String directclass) {
    String result = "";
    // connect to Jean DB
    IDBConnection conn =
    DBManipulator.getJenaDbConnection(Constant.JENA_DB_TABLE,
    Constant.DB_TYPE);

    OntModel model = OWLControl.retrieveProjectModel(conn, prjID);
    OntClass direct = OWLControl.retrieveClass(model, directclass);

    if (direct != null) {// the specified class exist in the model
        model.enterCriticalSection(Lock.READ); // get the read lock
        try {
            model.begin();
             // retrieve intance belong to input directclass
             ExtendedIterator iter = direct.listSubClasses(true);
             while ( iter.hasNext() ) {
                OntClass tmp = (OntClass) iter.next();
                String sub = tmp.getLocalName();
                result = result + sub + ";";
             }
             iter.close();
             model.commit();
          } catch (Exception e) {
             result = "";
             model.abort();
             e.printStackTrace();
          } finally {
             model.leaveCriticalSection();
          }
       } else
          result = null;
       model.close();
       DBManipulator.closeJenaDbConnection(conn);
       return result;
   }
```

Figure 5.14 Example of retrieving sub-classes of an ontology class in a project ontology model

5.4 Implementation of System Functionalities

In this section, the implementation of the system functions mentioned in Section 4.5 is introduced.

5.4.1 Project Information Management

Create Project

The Project Handler gathers the project title, background information, objective, deadline, description and keyword. It constructs the Project data model. Also, the Project LOM data model for "General" Category is constructed. The Data Module interacts with ProjectControl and LOMControl in Database Module to store the project information and LOM records.

After these records were created, the Project Handler receives the unique Project ID and LOM ID from Database Module. With this data, the Project Handler can call the function of OWLControl in Ontology Module to construct a Project Ontology Model. Then it directs to a JSP for adding members into the project group. After all project group member(s) added, the project can be started by the project group member(s).

Manage Instruction

For inserting a new instruction, its title and content are passed into the Project Handler. The handler constructs a LOM data model as mentioned in Table 5.1 (Section 5.2). Then the LOM of an instruction is inserted into database with the LOMControl in Database Module. Using the unique LOM ID returned, an

Instruction (learning object) instance is added into the project ontology model with the Ontology Module.

To update or delete any instruction, the LOM ID or sometimes the updated title and content are passed into the handler. The handler interacts with the Database Module and Ontology Module to manipulate the instruction data.

5.4.2 Learning Process Management

Manage ILO

The procedure of inserting, updating and deleting ILO in a project are similar to those of "Manage Instruction", but this is managed with the Project Design Handler.

Manage Activity and Achievement

Beside inserting, updating and deleting the Activity and Achievement records, they can be added the relationships with different ILO and learning objects. The Project Design Handler gets a LOM ID of the activity or achievement, and gets a LOM ID of the ILO or learning object. It checks whether these LOM ID are valid to add the relationship. Then it calls the OWLControl functions in Ontology Module to add the relationship of the instances in the project ontology model.

To view their relationships, the handler calls the functions of OWLControl with a LOM ID of the activity or achievement. The list of LOM ID of all related learning objects is returned to the handler. Then the handler constructs a list of LOM data models for display the learning objects (including ILO).

View Project Log

After a new log content of a project group member has been passed into the Member Handler, it constructs the Member Log data model with the content and member's User ID. It interacts with the Data Module and Database Module to add a new project log record in "member_log" table.

To view the project logs, the Member Handler retrieves the list of project logs of a particular project group member with the User ID. Then it instantiates the Log data models with Data Module. It responses and direct to a JSP for display the logs.

5.4.3 Learning Resources Management

Manage Project Structure

Once the Project Structure Handler gets the name and description of the new class, it checks whether the name of the class is valid for the project ontology model. The class name must be shorter than or equal to 40 alphabets. Also, the class does not exist in the project ontology model.

If it is valid, the handler manipulate the class name by replace the " " into "_", which is the valid format of ontology class name. The handler calls the function of ProjectControl in Database Module to store this class information in database. The name of its direct super-class of this class is retrieved. Then the handler can call the function of OWLControl to add the new class.

Manage Learning Resource

The Learning Object Handler receives the LOM information from the request. It checks the direct class of the learning resource belongs to. It also checks whether the learning resource is a URI link or file. Then it constructs the LOM data model and stores the LOM records with Data Module and the LOMControl in Database Module according to the learning resource type (URI link / file). A unique LOM ID is returned from the Database Module.

Moreover, the handler checks the pre-defined super-class of the learning resource with OWLControl function. Then the handler calls the function of OWLControl to add a new instance of its direct class in the project ontology model according to its super-class.

If the learning resource is a file, the handler directs to JSP for uploading files. The File Upload Handler will be invoked if the file is being uploaded. The handler checks the file content type and size to verify whether the file can be uploaded into the system. If there is no problem, the file is uploaded to specific destination. Also, it constructs LOM data model to update the information for the learning resource.

Manage Annotation

The Annotation of a learning object is defined when it is added into the project. The additional Annotation is handled by Annotation Handler. When the learning objects are retrieved for display, the Annotation Handler interacts with the Learning Object Handler. By the learning object's LOM ID, the Annotation Handler

calls the function of DisplayControl in Database Module to retrieve the list of annotations of this learning object. Then it constructs the results as a list of Annotation data models and returns to the Learning Object Handler.

In addition, after the Annotation Handler has got the new annotation of the learning object, it constructs and formats the Annotation data model according to Table 4.1 and Section 5.2. It interacts with the Data Module and the Database Module to stores the annotation records in database.

Search Learning Resources

The Searching Handler receives the searching criteria with the keyword, class of learning resource belongs and learning resource type. According to the requirement, it calls a particular function of the DisplayControl in Database Module to retrieve the list of searching results. The handler instantiates the Search Result data model and response for display the results in a JSP.

5.4.4 Learning Outcomes Management

Manage Assignment Schedule

After the Assignment Handler has received the assignment title, description, type and deadline, it validates the deadline of the schedule. The deadline of the schedule cannot be the date after the project's deadline. Then it constructs the Assignment Schedule data model with the Data Module. It then stores into the database with Database Module. After the unique Schedule ID is returned, a sub-class of Assignment is created in the project ontology model with the Ontology

Module. The sub-class's name is the prefix "Assign" and the Schedule ID (e.g. Assign1).

Project Assessment

For the Project Assessment, the Evaluation Handler calls the functions of EvaluationControl in Database Module to collect the evaluation result. In which, the statistic of the total numbers of learning objects in a project and the total numbers of learning objects gathered by all project group members are found. Also, the information of the assignments of all project group members is collected. Then the handler instantiates different Evaluation Result data models to display the assessment information.

Project Group Member Assessment

For the Project Group Member Assessment, the Evaluation Handler also collects the assessment information, but the statistic and assignments are related to a specific project group member. With the project group member's User ID, the Evaluation Handler calls the functions of EvaluationControl to retrieve the assessment statistics for the member.

Besides, the Evaluation Handler can manage the evaluation result for members. After the handler receive the User ID, grade and comment of the members, it interacts with EvluationControl in the Database Module to add or update the evaluation comment and grade of corresponding database records of the project group member.

Chapter 6. Scenarios

6.1 Background

This scenario shows the learning process of doing Final Year Project of Computer Science students, which is a typical Project-Based Learning. In addition, the scenario assumes that all user accounts of the supervisors and students have been registered by system administrators.

6.1.1 Project Information Management

After the supervisor and students discussed the topics of their FYP project, supervisor can create a new project in this Project-Based Learning Environment.

Create Project

1. The supervisor provides some background information, objective, deadline, general description and keywords to describe the project.

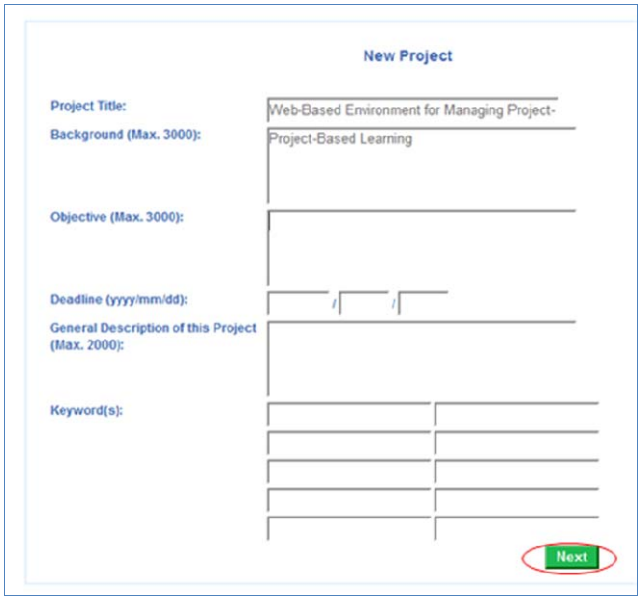


Figure 6.1 Supervisor creating a new project

And the supervisor forms the project group. The project student is set to "Member", while the students who work on other closely related projects is set to "Guest" in the project.

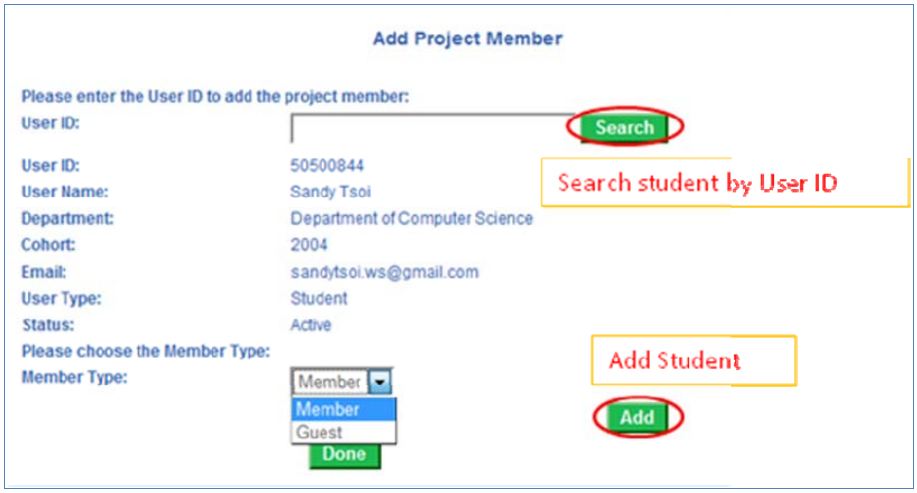


Figure 6.2 Adding Project Group Member into the project

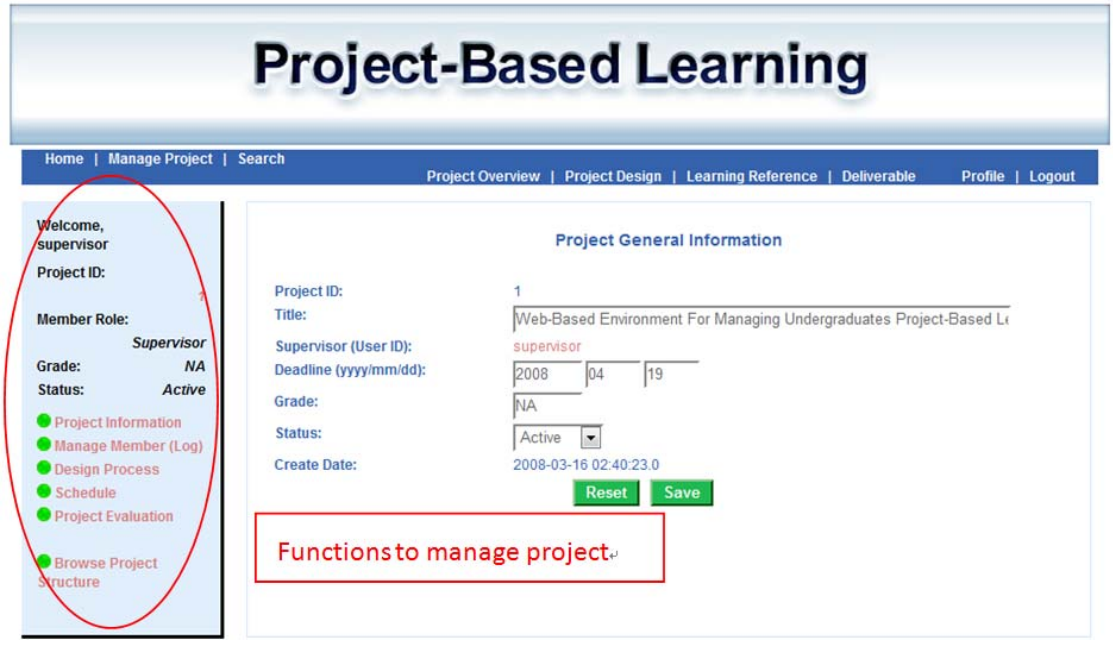


Figure 6.3 Interface to manage the new created project

2. After the supervisor has created a project, the project student, “Member”, can start to manage the FYP, while other students, “Guest”, can share their ideas of the learning resources. They read the information given by the supervisor to understand the project’s background. The project student searches information to learn more about the project’s topic. Then the project background, objectives, description and keywords can be updated by the project student as he/she better understand the project.

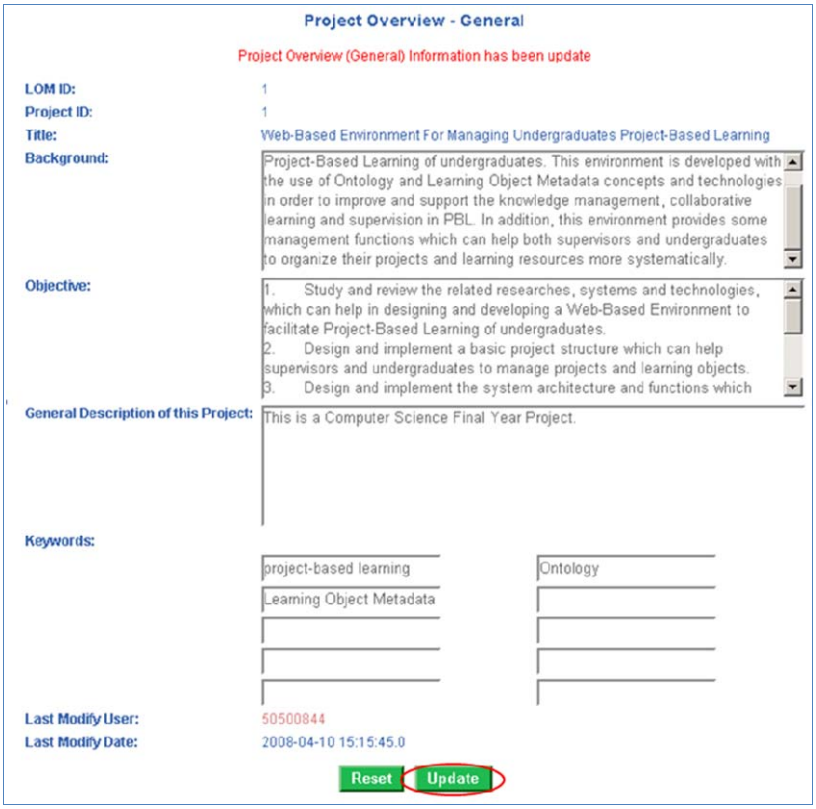


Figure 6.4 Project student updating project information

Manage Instruction

1. The supervisor provides some instructions and guiding questions that can direct the project student to work on the project.

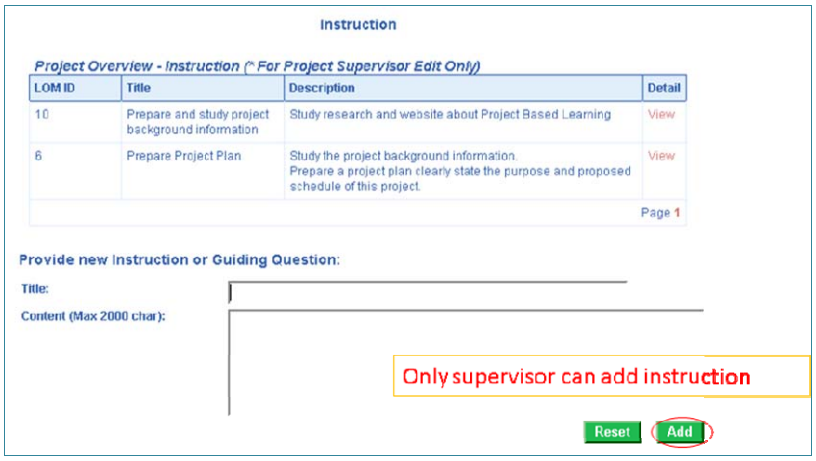


Figure 6.5 Supervisor managing project instruction(s)

2. The project student can know what should be done or even the guideline of doing the project.

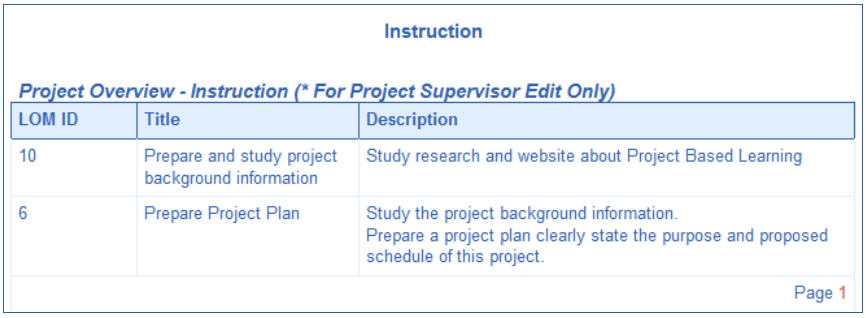


Figure 6.6 Reading instructions given by supervisor

6.1.2 Learning Process Management

Doing final year project can motivate students to actively learn and achieve some learning outcomes. To facilitate the project student learning and achieving the outcomes, the supervisor can define the learning path of the project.

Manage Intended Learning Outcome (ILO)

1. The supervisor set the ILOs, which clearly explain the expected outcomes of the student in the project. They can be the skills or actual deliverables.

2. Apart from the supervisor defined ILO, the project student can also states the additional goals as his/her own learning path.

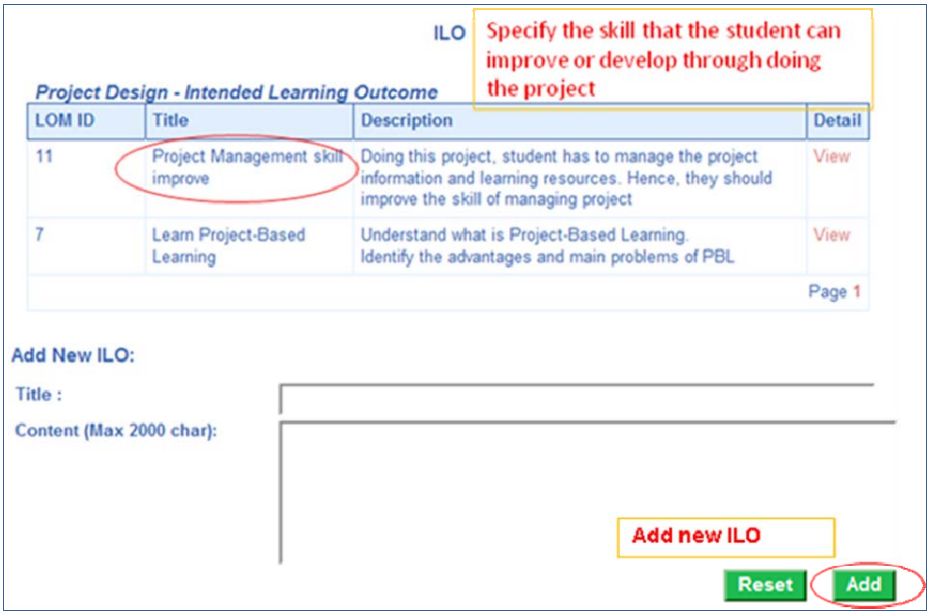


Figure 6.7 Setting the Intended Learning Outcome (ILO)

Manage Activity

1. The supervisor would like to give some exercises or learning references for the project student to do during the learning process. They are the learning activities which achieve particular ILO. The supervisor can manage this information by specifying the Activity description and mapping with the learning resources.

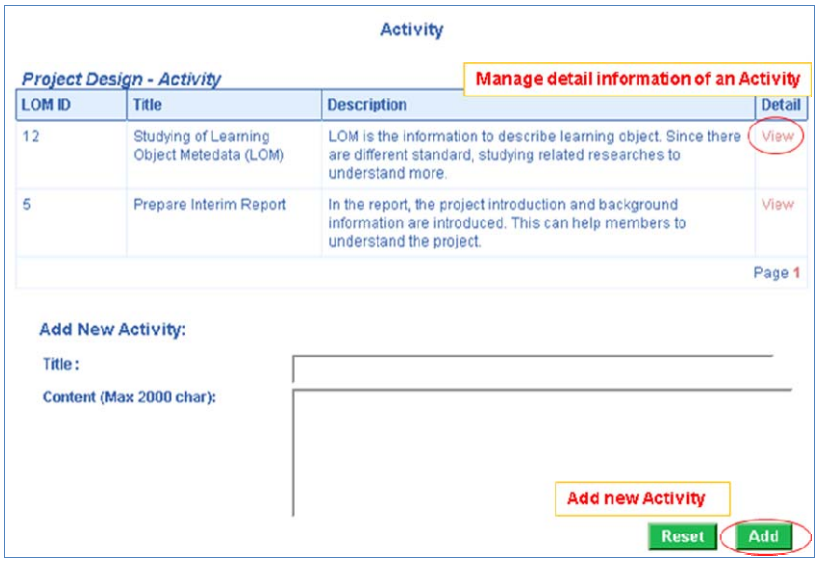


Figure 6.8 Managing project activities

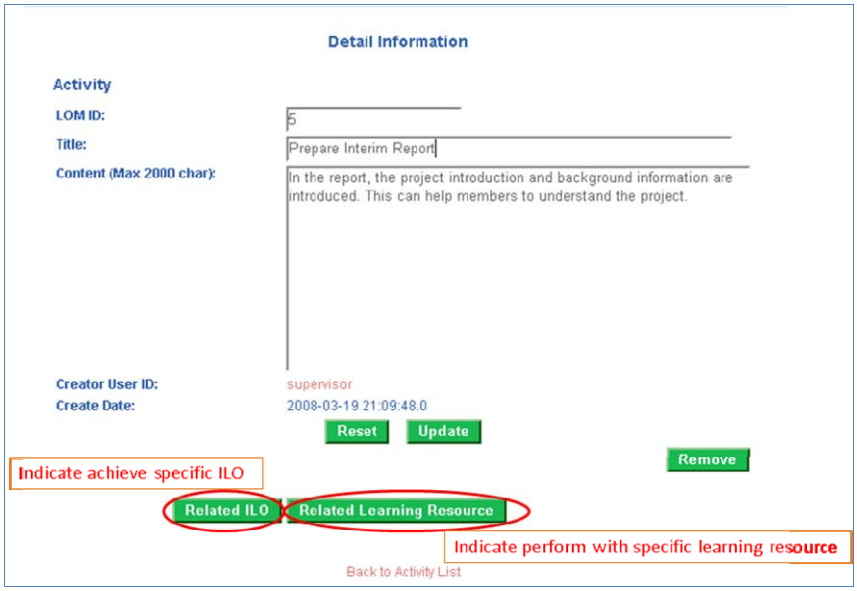


Figure 6.9 Showing an activity detail information

2. The project student checks the Activity information. He/she needs to perform the tasks relating with each Activity. And he/she knows what skills or outcomes can be helped to develop or achieve by doing the learning activities.

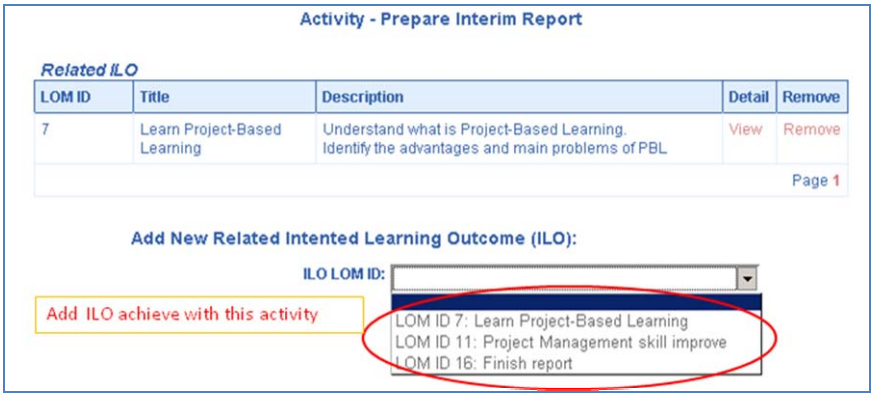


Figure 6.10 Adding an ILO which is achieved with specific activity

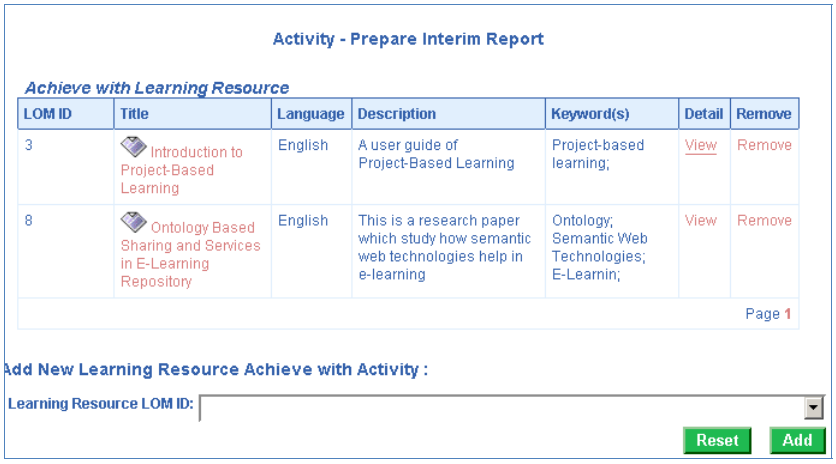


Figure 6.11 Learning resources needed to study in an activity

Manage Achievement

1. When the project student has finished some tasks and achieved some of his/her targets, he/she records these results with the learning path, “Achievement”. Hence, the supervisor can know his/her progress and learning outcomes.

2. The supervisor and project group members know the learning process and achievement of the project student by viewing “Achievement”. Moreover,

they can check the related ILO or learning resources to understand the actual outcomes.

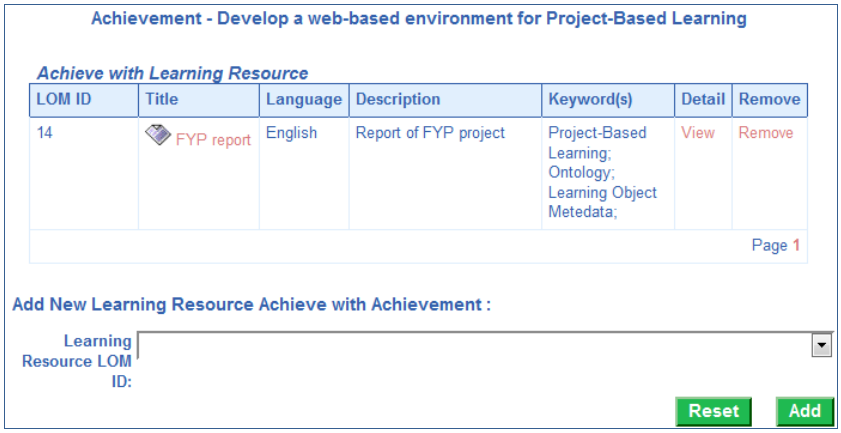


Figure 6.12 Showing learning resources related to an Achievement

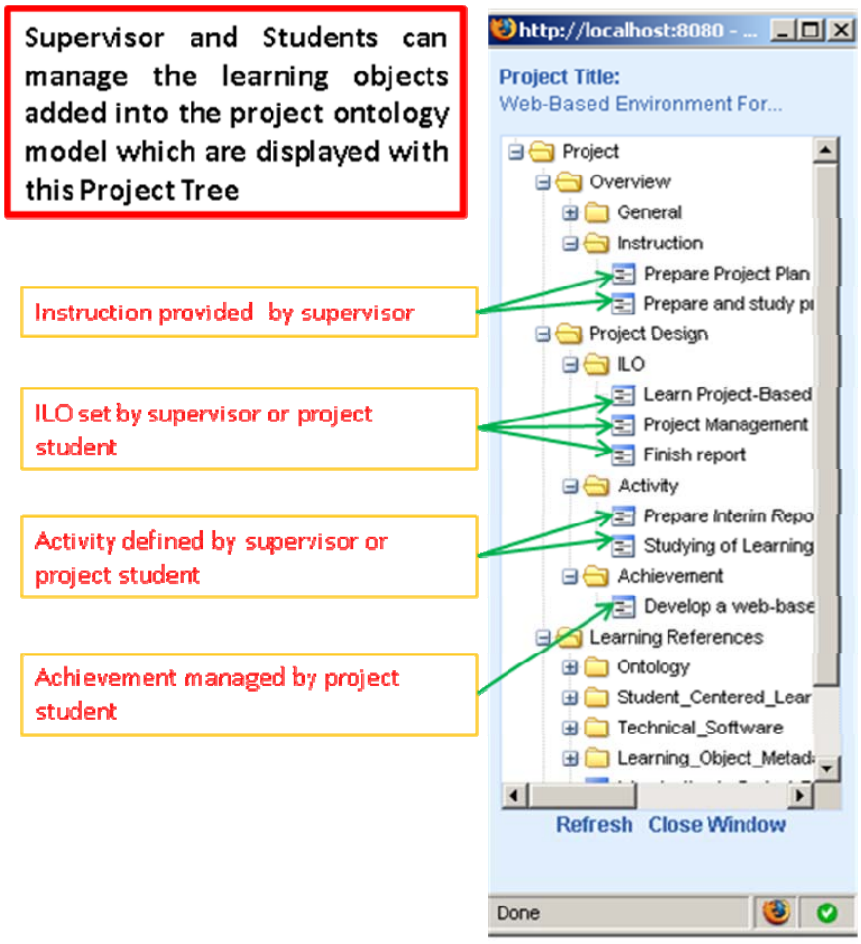


Figure 6.13 Illustration of the learning objects (Instruction, ILO, Activity and Achievement) managed with a Project Tree (project ontology model)

View Project Log

1. Although the learning path can manage the information for the learning process, the project student needs to report the progress by writing project log. This can let the supervisor know his/her current working progress from time to time.

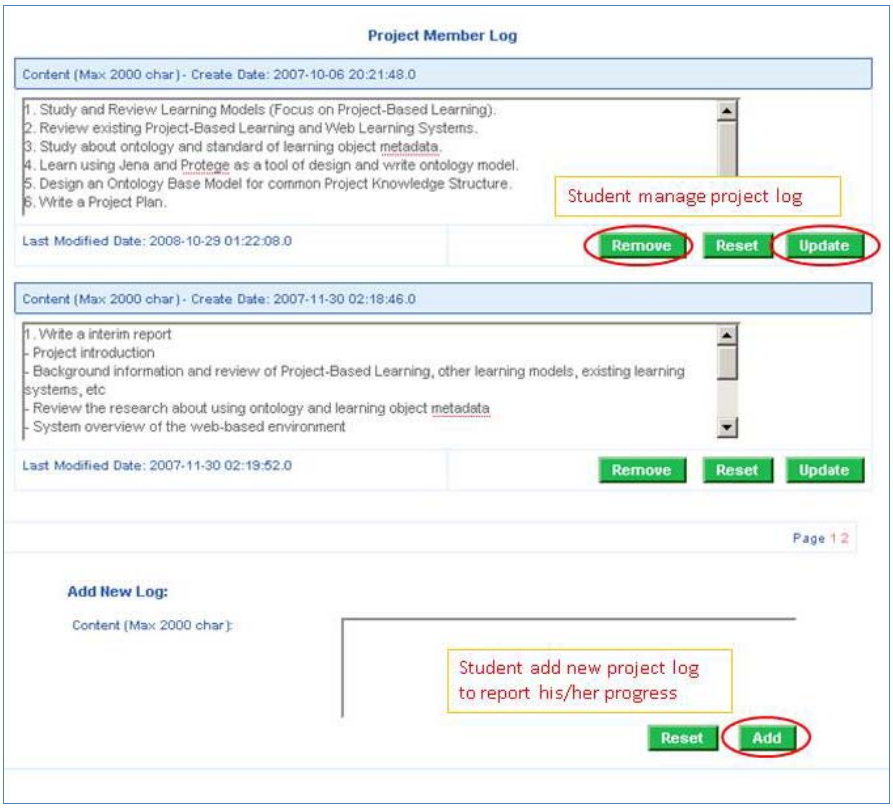


Figure 6.14 Student managing the project logs to report his/her current learning progress

2. During the learning process, the supervisor would like to know the learning progress of the project student, they could have meetings to discuss their progress. Besides, by viewing the Project Log, the supervisor can know his/her learning which may be unable to illustrate with the learning path.

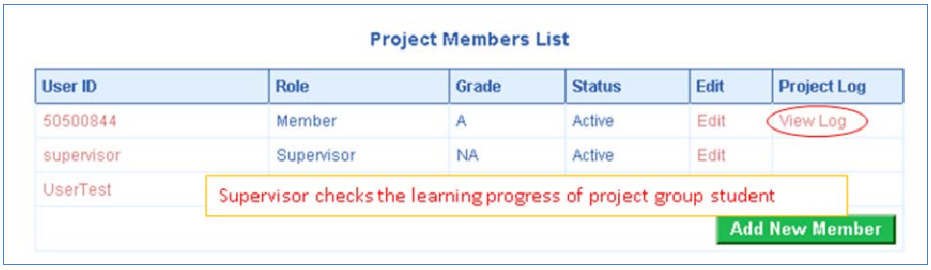


Figure 6.15 Supervisor choosing a project group member to monitor his/her learning progress

Figure 6.16 Viewing Project Logs of a project group member

6.1.3 Learning Resources Management

As large amount of books, research papers and websites are studied, the project student and the supervisor have to organize the learning resources which enhancing learning process.

Manage Project Structure

1. According to the project student knowledge, the classes are defined and described with some text to represent some conceptual knowledge or topics.

Since different classes defined, he/she builds up the project structure which can represent the project's conceptual knowledge model.

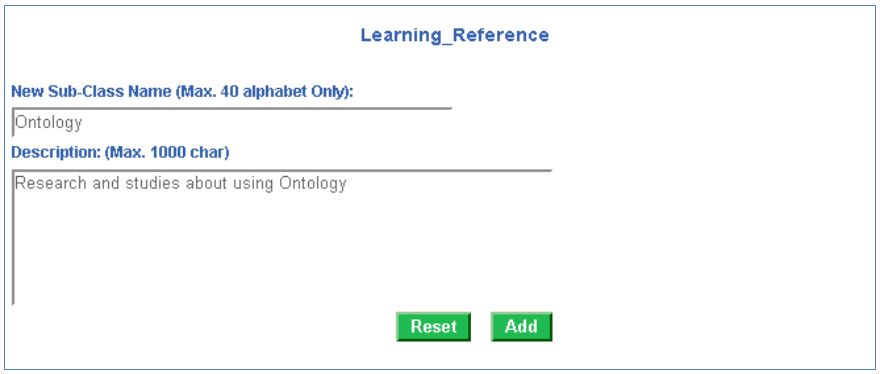


Figure 6.17 New sub-class "Ontology" adding into "Learning_Reference"

2. The supervisor can also define new class to classify learning resource. As well, the supervisor can monitor whether the project student understand the accurate conceptual knowledge.

Manage Learning Resources

1. The project student gathers and studies many kinds of learning resources.. He/she defines the LOM of each learning resource and classifies the learning resources into different classes.

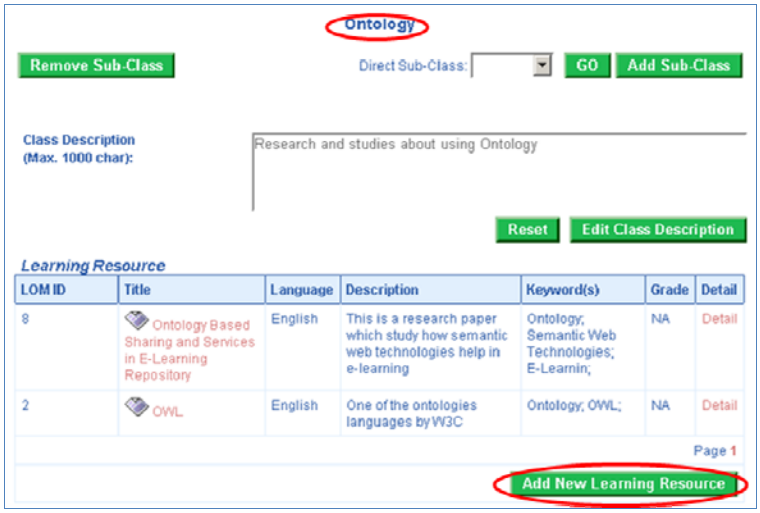


Figure 6.18 Classifying learning resource into a user-defined class

2. The supervisor reviews the learning resources and checks whether the project student is studying some relevant learning resources. From the descriptive information defined by the project student, the supervisor knows what types of materials that the project student has studied.

Figure 6.19 LOM information of a learning resource

3. In addition, the supervisor needs to verify the classifications of the learning resources. The project structure indicates how students build up the conceptual knowledge of the project. If the learning resource is imperfectly defined or classified into an unsuitable class, the supervisor and the project student can edit it and re-classify into another class.

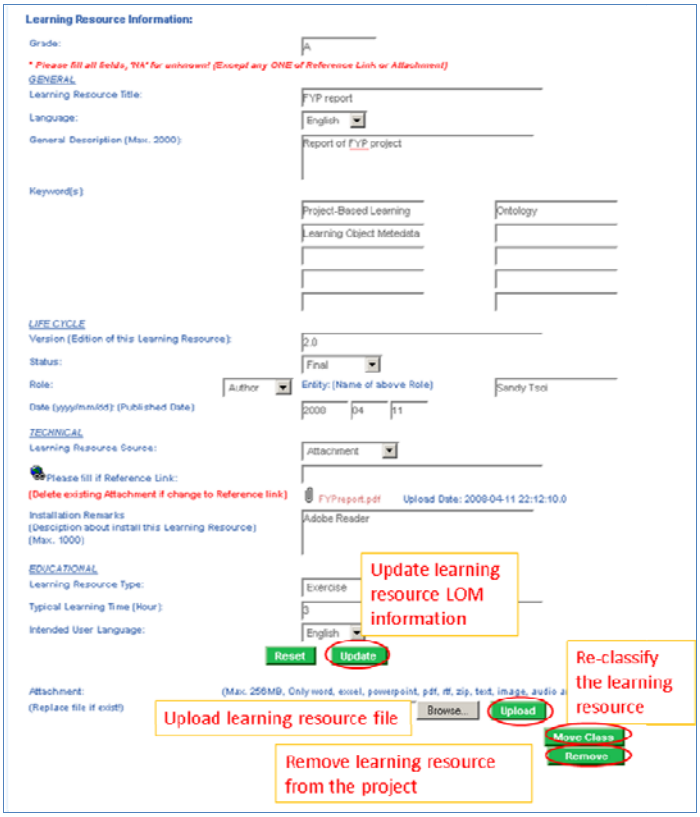


Figure 6.20 Editing learning resource's LOM information

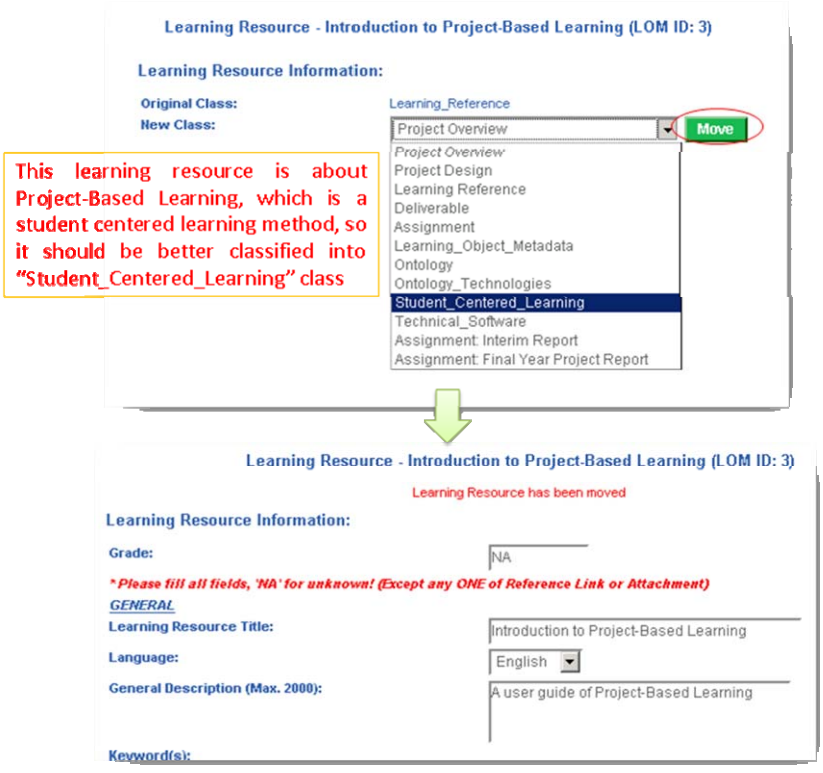


Figure 6.21 Re-classifying a learning resource from one class to another one

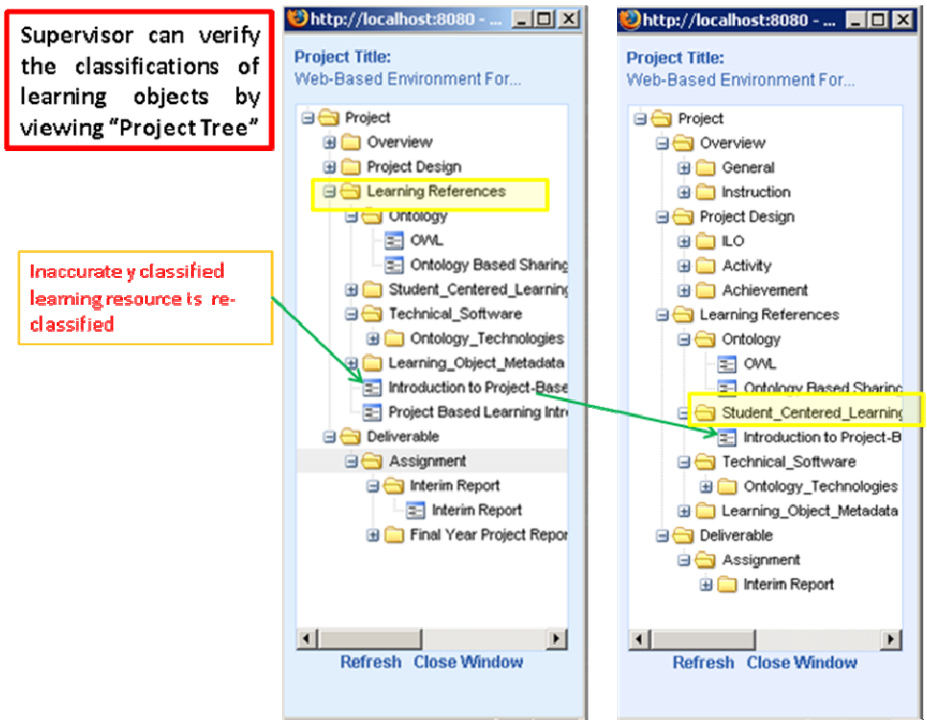


Figure 6.22 Illustration of the re-classification of a learning resource displaying in a Project Tree

Manage Annotation

1. The project student defines the annotation of the learning resource as said by his/her understandings; meanwhile, the supervisor can check whether he/she can explicit the concepts or knowledge. Then the supervisor guides or teaches the students to learn further concepts or ideas about the learning resources.

2. The supervisor and other project group members would also study the learning resources. They would give feedbacks and comments on the learning resources according to their viewpoints and understandings. With this information, the project student can gain further knowledge. Also, he/she can express the ideas and discuss with the supervisor and other members.

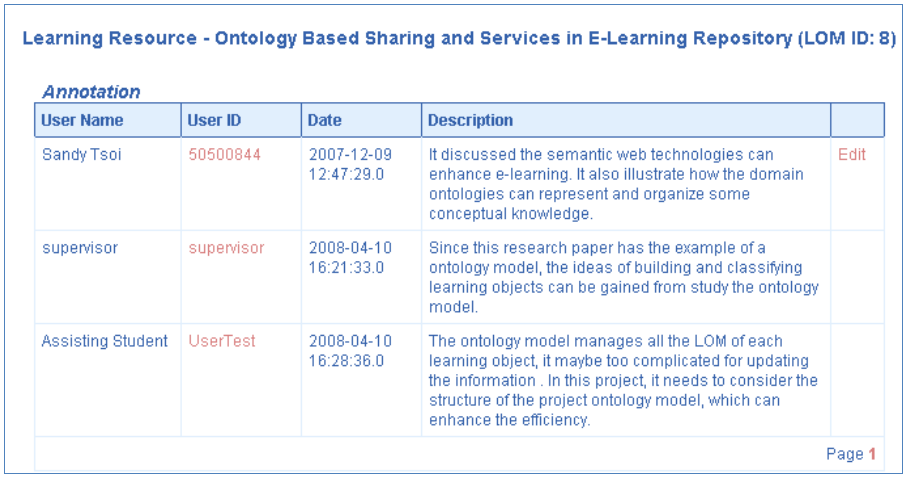


Figure 6.23 Discussion among project members and supervisor

Search Learning Resources

1. The supervisor needs to gather different information and learning resources to manage the project. Also, he/she has to provide learning references to the project student. As different learning resources which have the related topics or subjects may be studied by the supervisor, he/she can retrieve the list of learning resources by some keywords.

2. The project student may not work on the projects which have related topics or subjects. He/she can locate the learning resources in the current project.

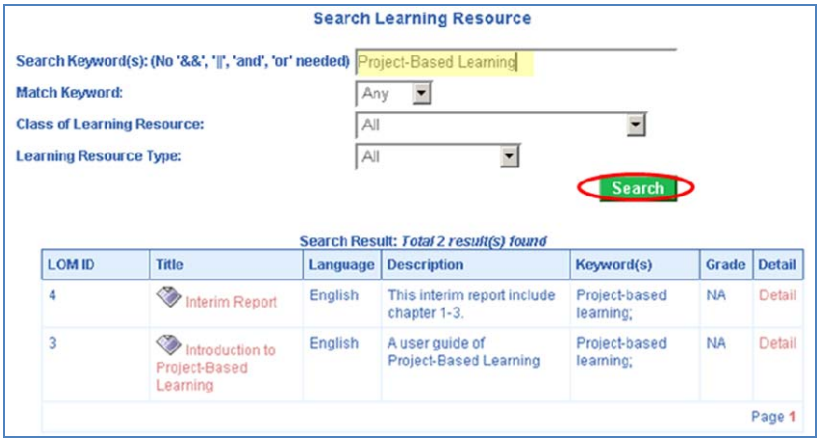


Figure 6.24 Searching learning resources in this Project-Based Learning Environment

6.1.4 Learning Outcomes Management

Manage Assignment Schedule

1. The supervisor sets the assignment schedule for the project student that he/she needs to finish in the project. The supervisor can also upload the assignment files which contain the questions or guideline of the assignments. Hence, the students can download to follow and do their assignments.

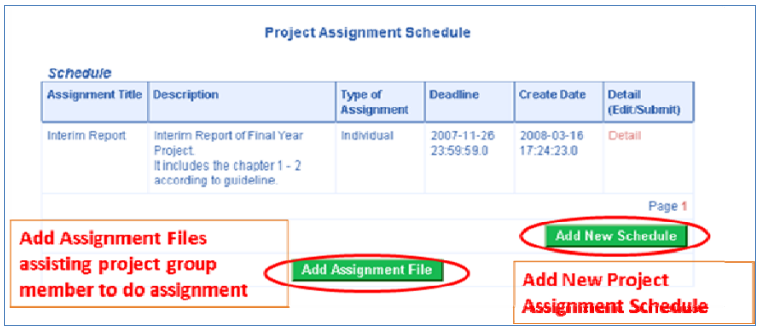


Figure 6.25 Showing the interface of Project Assignment Schedule

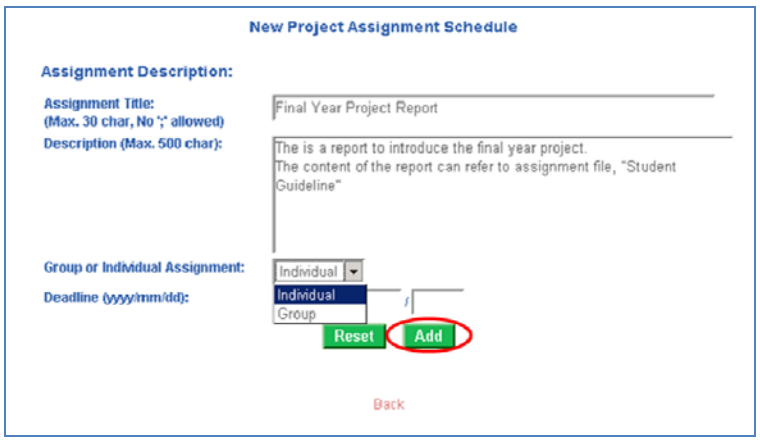


Figure 6.26 Supervisor setting a new Project Assignment Schedule

2. After the project student has studied and gained different knowledge and concepts, he/she can works on the assignments and tasks of the FYP. The products and assignments are submitted for the supervisor evaluate.

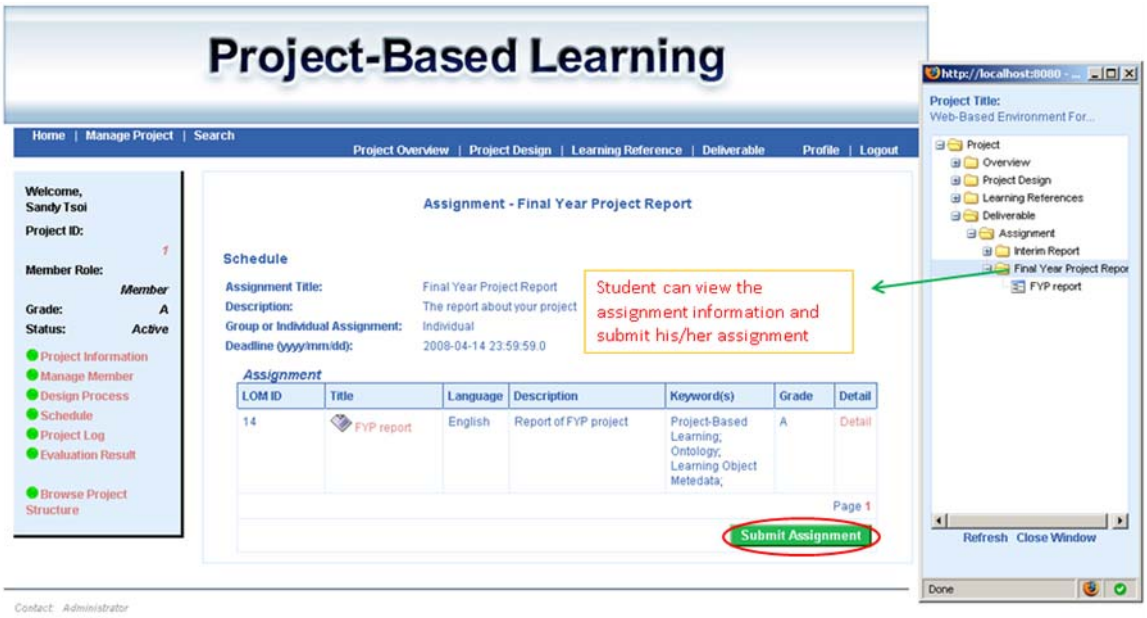


Figure 6.27 Clicking the assignment folder in a Project Tree to show the assignment detail information

Project Assessment

1. After the project has been finished, the supervisor has to assess the result of the project. From the learning path, the supervisor can view the information that shows the management of tasks, learning resources and learning outcomes during the learning process. The supervisor can easily evaluate whether the project student can achieve the intended learning outcome.

2. The supervisor has to evaluate the performance and ability of student after the FYP is completed. The supervisors can assess their assignments to check whether the project student has submitted the assignments on time.

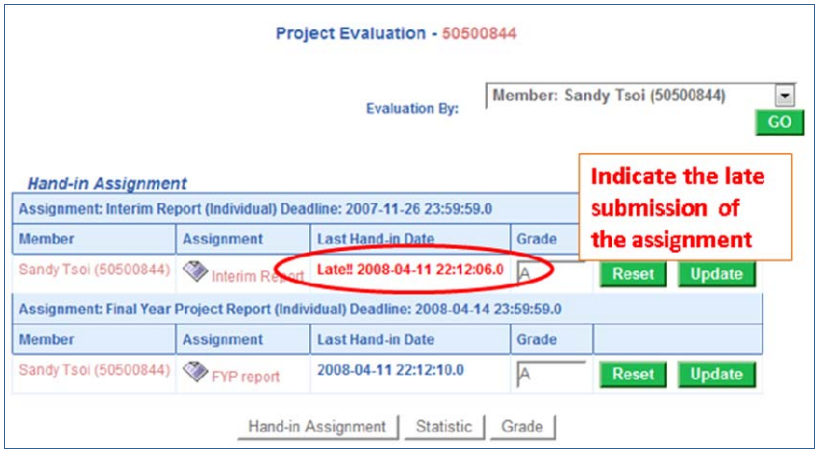


Figure 6.28 Supervisor evaluating the assignments of project student

3. The supervisor also would like to know the involvements of the project student and his/her attitudes in doing the project. From the numbers of learning resources and comments the project student studied and gave. The supervisor knows his/her involvements and checks whether he/she actively discuss the knowledge with others. Hence, the supervisor can judge the quality of each student's learning from other viewpoints. Also, the supervisor can know the effectiveness of the learning from this project.

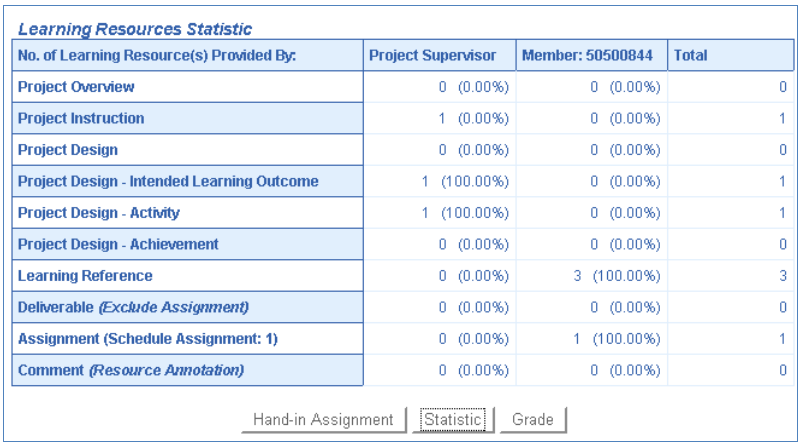


Figure 6.29 Learning Resources Statistic of a project group member

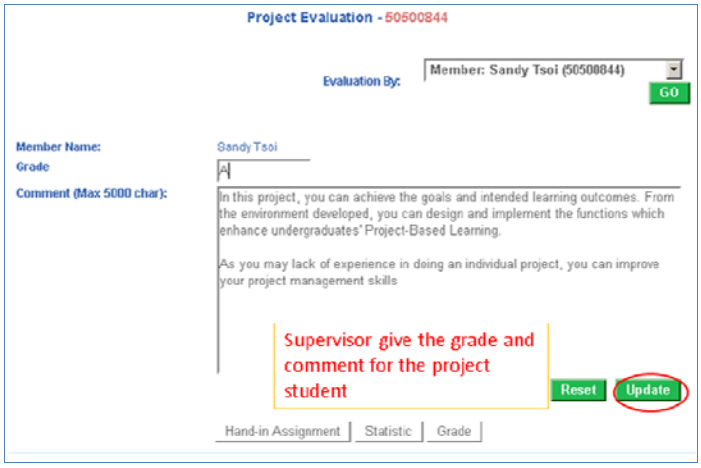


Figure 6.30 Supervisor evaluating on the project student's performance

4. After the FYP has been finished, the project student can check the result of the whole project and the grade of each assignment. From the supervisor evaluation comments, he/she knows his/her strength and weakness in doing the FYP.

Figure 6.31 Evaluation result for a project student

6.2 Evaluation of using this Project-Based Learning Environment

6.2.1 Benefit of using this PBL Environment

Knowledge Management

This environment provides interfaces for the supervisor and students to manage the project information. It is convenient for them to share and describe the background and purposes of the project.

Also, the supervisor can define the learning path in the project design process. The expected learning outcomes, project tasks, assignment schedules are clearly defined and well managed. With this organized information, students can effectively plan the project tasks and schedule. In addition, students can define the learning path themselves so that they can share the ideas of project design with others. Adding the relationships of ILO, Activity, Achievement and other learning resources, students can easily manage the learning resources which associate with their learning activities and outcomes.

The supervisor and students can also define new classes into the project structure to represent their conceptual knowledge. Different kinds of learning resources can be uploaded and classified with these classes. Since all the learning resources are defined with the tailor-made LOM structure, they would be easier to handle and share large numbers of learning resources in the project.

During the FYP learning process, many research papers or websites have been studied and visited. It is difficult to remember what have been learned from

different resources. As they can view how the learning resources including the comments were defined, they can effectively review the knowledge of different learning resources. Furthermore, since the project structure is visualized as a Tree Hierarchy, they can view the organization of the project and learning resources easily. This enhances searching and reviewing the learning resources.

Besides, this environment provides function for them to search learning resources by entering some keywords. They can find the suitable learning resources which can be reused in different projects. It reduces their time and efforts in searching information.

Collaborative Learning

LOM provides a standard format for sharing and storing different comments of the learning resources. The students can review others' learning resources descriptions. Therefore, they are able to understand the learning resources easier as they could share and discuss ideas with the supervisor and project group members. By reviewing the comments of learning resources, supervisor can also give advice for them to learn the knowledge in-depth. This could increase the learning resources' readability and enhance further knowledge exchange.

Support Supervise Students

The supervisor can provide instructions and set the learning path by defining ILO and Activity. This helps the students to follow the path to work on the project. Also, the supervisor can assign the assignments for them that the supervisor can know their levels of learning from these assignments.

Moreover, the supervisor can monitor the students' working progress by the project logs. The supervisor can also check how they classify the learning resources. Hence, the supervisor cannot only know the numbers of learning resources they have studied, but also know the level of their understandings of learning resources. It could enhance the flexibility for supervisors to review and monitor the progress.

This environment provides functions for the supervisor to evaluate the project's results and students' performance. The supervisor can know the involvement of the students from the statistics. Hence, the supervisor can manage or monitor their learning and evaluate their performances.

6.2.2 Limitation of using this PBL Environment

This environment can enhance the Project-Based Learning. Both supervisor and students can organize the project and the learning resources effectively. However, some issues of PBL are still difficult to be facilitated and solved by this environment.

Incorrect Classification of Learning Resource

The supervisor and students can add different sub-classes in the project structure for classifying their learning resources. They can also define the description of each sub-class to represent some conceptual knowledge. However, it cannot ensure that the classifications of learning resources are absolutely correct. The students may incorrectly get the meanings of the conceptual knowledge. Consequently, the learning resources cannot be well organized and shared among the project group.

Hence, the supervisor needs to control and monitor this process that they can provide professional advices to the students.

Unable to Manage all Learning Process

In this environment, the supervisor and students can manage some of their learning process with the learning path. This can only facilitate them to manage the information about the project goals, activities and outcomes. The concrete processes of how they think and learn the conceptual knowledge from the project or learning resources may not be described and managed with this environment. Moreover, the students may learn the knowledge from the discussion with the supervisor face-to face. The environment cannot provide functions to record these kinds of information.

Chapter 7. Conclusion

7.1 Achievement

In this project, it is proposed to develop a web-based environment which can facilitate the Project-Based Learning of undergraduates. To enhance the learning of the undergraduates, the environment is planned to provide interfaces for supervisors and students to manage the project and learning resources systematically and effectively.

In order to develop the proposed environment, many research and information about the student-centered learning and Project-Based Learning are studied. They showed that Project-Based Learning can enhance students' learning, but students would find the difficulties in knowledge management and collaborative learning. In addition, supervisors would be difficult to supervise the learning of the students.

To solve these main problems, the researches about the technologies used in e-learning environments and the existing learning platforms are also studied. Moreover, the concepts of using Learning Object Metadata and Ontology used in learning platform and knowledge-based application are reviewed. By studying these two technologies, a standard format and project structure are designed to manage the projects and learning resources.

In the Project-Based Learning Environment developed, all the learning objects, including project information, project design process information and learning resources are described with LOM. In addition, with using different Ontology technologies, a Project Ontology Model is designed and implemented to manage the

knowledge of projects. Supervisors and students can make uses of “Project Management”, “Learning Process Management”, “Learning Resource Management” and “Learning Outcome Management” functions of the environment to organize their projects and learning resources effectively.

From the scenario, it showed that the LOM of the learning resources can provide additional information which help students to understand the concepts or knowledge. Also, the learning objects organized with the project ontology model and LOM can enhance the collaborative leaning among students and supervisor. Since the environment can help sharing of knowledge and streamline the process of doing projects, it can reduce the problems of knowledge management, collaborative learning and supervision of students’ learning. This can also facilitate undergraduates to manage projects.

7.2 Limitation on Technologies

Learning Object Metadata

As the LOM of the learning objects are stored in database, the information can only be accessed through this environment. This may not enhance the knowledge sharing of learning objects with other learning platforms. In addition, the LOM are stored in database, so the sizes and formats of some data elements are restricted in this environment.

Accessing Time of Project Ontology Model

Each project ontology model is stored as a persistence model in the database with Jena API. It has the limitation on this technology that the time of retrieving and manipulating data in the ontology model is slower than accessing normal database records. This may affect the performance of the environment if the project ontology models of the projects manage large amount of data.

Searching Learning Objects Detail Information

Considering the accessing time of ontology models, each project ontology model only records the title and LOM ID of the learning objects. Hence, this model can only show the hierarchy of the learning objects in a project structure. As it is necessary to access database with the LOM ID for further information, it cannot search the learning objects detail information directly with the project ontology model.

7.3 Challenges and Difficulties

Design of Project Basic Structure

The Project Basic Structure is needed to use for different kinds of projects. Hence, it is very difficult to design the structure which can suitable for different Project-Based Learning projects. In addition, the project basic structure is implemented with ontology. Different kinds of Ontology applications and API are needed to study in order to choose a suitable one for the system development.

Visualization of Project Ontology Model

Although each project structure can be manipulated with Ontology, it must be represented in a format that users can easily understand its hierarchy. Even the system chose using Tree for display the project structure, it is difficult to find a suitable API to manipulate the Tree.

In addition, the project structure is not fixed for different projects. Ext 2.0 API can manipulate the Tree structure in real-time. However, it just provides the interface for importing specific format data to construct the tree. A generic handler is needed to implement that can handle the data for different tree structure.

Design of Generic JSP and Handlers

The project structure of each project is different that users can define different sub-classes. It is impossible for implementing different separated web-pages for all user-defined classes in real-time. Therefore, the JSP and Handlers are needed to design to handle the data and display for different project structures.

7.4 Further Development

Visualize Project Ontology Model in Different Format

In this environment, an OWL file of the project ontology model are generated and stored in the system when the project is closed. These files are used to clone the project structure and stored as a backup. Also, the content of OWL files is machine interpretable on the Web. For the further development, they can be used to display as a concept map which can give a clear picture of the classifications and relationships of learning objects in different projects.

Export of Learning Object Metadata

Since the LOM of the learning objects are stored as database records, it is difficult to exchange the learning information with other learning systems. To enhance knowledge sharing, the system should provide function for user to export the LOM records to the XML format, which is interchangeable in the web. Then they may share the information of their learning objects with other learning platforms.

Global Knowledge Sharing

In the developed environment, the learning objects are classified with different project ontology models, in which the conceptual knowledge of projects is represented and managed with their own ontology models separately. However, the knowledge of some domains should be common in the world. In the further development, the research on merging different project ontology models to represent a global conceptual knowledge can be studied. Hence, it would facilitate the further knowledge sharing within the environment.

Design of Personal Ontology Model

The implementation of the project ontology models in this system shows that ontology enhances knowledge management. Different learning objects classified in ontology can facilitate searching and managing learning resources. Therefore, the system can develop a personal ontology model for each user. Users can organize their learning resources with the personal ontology models which can represent their own knowledge rather than the projects one. This may further improve the knowledge management and reduce the time on searching learning resources in the system.

Design of Supervisor Project Ontology Model

From time to time, each supervisor would have huge amount of projects which have been assigned for different students. It is difficult to organize and reuse the projects. Hence, the system can develop a Supervisor Project Ontology Model for the supervisors to handle their projects. The project can be classified by its topic, year or subject, etc. Then the supervisors can efficiently find out the project and reuse them for other new project. Also, they can simply find the relevant learning resources used in the projects for their students to study.

Chapter 8. Reference

Alberta Education (2004). [Online]. Focus on inquiry: a teacher's guide to implementing inquiry-based learning. Available: http://www.education.gov.ab.ca/k_12/curriculum/bySubject/focusoninquiry.pdf [2007 November 20].

Batatia, H. Markkanen, H. and Ponta, D. (2002). Netpro: Tools for web-based project learning. International Conference on Computers in Education (2002 December 3-6). Auckland, New Zealand.

Blackboard [Online]. Blackboard Learning System. Available; http://www.blackboard.com/ClientCollateral/100506/Bb_Learning_System_%20Brochure.pdf [2007 October 23].

Bransford, J. D. and Stein, B. S. (1993). The IDEAL problem solver (2nd ed.). New York: Freeman.

Dodge, B. (1997 May 5). [Online]. Some Thoughts About WebQuests. Available: http://webquest.sdsu.edu/about_webquests.html [2007 November 15].

Ext JS [Online]. Ext JS - JavaScript Library. Available: http://extjs.com/ [2007 November 28].

Gasevic, D., Jovanovic, J. and Devedzic, V. (2004). Enhancing Learning Object Content on the Semantic Web. Proceedings of Advanced Learning Technologies, 2004. IEEE International Conference (2004 August 30 – September 1), p. 714 – 716.

Heiyanthuduwage, S.R. and Karunaratne, D. D. (2006). A Learner Oriented Ontology to Make Effective Learning Management Systems. Digital Information Management, 2006 1st International Conference (2006 December 6), p. 476 – 481.

IEEE Learning Technology Standards Committee (2002). [Online]. Draft Standard for Learning Object Metadata. New York, USA: Institute of Electrical and Electronics Engineers, Inc. Available: http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf [2007 September 11].

IEEE Learning Technology Standards Committee [Online]. IEEE LTSC | WG12. Available: http://ltsc.ieee.org/wg12/ [2007 September 11].

International Society for Technology in Education (ISTE) (1997). [Online]. Foundations for the Road Ahead: Project-Based Learning and Information Technologies. Available: http://www.iste.org/Content/NavigationMenu/Research/Reports/The_Road_Ahead_Background_Papers_1997_/Project-Based_Learning.htm [2007 November 21].

Jena Semantic Web Framework. [Online]. Jena – A Semantic Web Framework for Java**.** Available: http://jena.sourceforge.net/index.html [2007 July 15].

Kurubacak, G. (2005). Sharing power and culture through Project-Based Online Learning (PBOL): Designing online knowledge based on multicultural education. In C. Crawford et al. (Eds.). Proceedings of Society for Information Technology and Teacher Education International Conference 2005. Chesapeake, VA: AACE, p. 466-470.

Land, S. M., and Greene, B. A. (2000). Project-Based Learning with the World Wide Web: A qualitative study of resource integration. Education Technology, Research and Development, Vol.48 (1), p45-68.

MacGregor, S.K. and Lou, Y. (2005). Web-Based Learning: How Task Scaffolding and Web Site Design Support Knowledge Acquisition. Journal of Research on Technology in Education, Vol.37 (2), p. 161-17.

Maedche, A.D. (2002). Ontology Learning for the Semantic Web. Massachusetts, USA: Kluwer Academic Publishers Group.

Markham, T., Larmer, J. and Ravitz, J. (2003). [Online]. Introduction to Project Based Learning. Project Based Learning Handbook 2nd ed. California: Buck Institute for Education, p. 3-9.

Mergendoller, J.R., Markham, T., Ravitz, J. and Larmer, J. (2005). [Online]. Scaffolding Project Based Learning: Tools, Tactics and Technology to Facilitate Instruction and Management. Available: http://www.feg.unesp.br/~rioparaiba/palestras_pbl/John%20R%20Mergendoler%20artigo.pdf [2007 November 20].

Mergendoller, J.R. and Thomas, J.W. (2000). [Online]. Managing Project Based Learning: Principles from the Field. Available: http://www.bie.org/files/researchmanagePBL.pdf [2007 November 22].

McGuinness, D.L. and Harmelen F.V., ed. (2004). [Online]. OWL Web Ontology Language Overview. World Wide Web Consortium (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). Available: http://www.w3.org/TR/owl-features/ [2007 September 13].

NetPro [Online]. NetPro. Available: http://netpro.evtek.fi/frames/tools.htm [2007 Novemeber 15].

Noy, N.F. and McGuinness, D.L. (2001). [Online]. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford, CA: Stanford University. Available: http://www.ksl.stanford.edu/KSL_Abstracts/KSL-01-05.html [2007 September 12].

Ponta, D., Donzellini, G., Markkanen, H. (2003). Project based learning in internet. Frontiers in Education, 2003. FIE 2003. 33rd Annual, Vol.3, S2F-7-12.

Project Foundry [Online]. Project Foundry™: Engaging Students and Ensuring Results with Project-Based Learning Systems. Available: http://www.projectfoundry.org/demonstrate.html [2007 September 3].

Protégé [Online]. The Protégé Ontology Editor and Knowledge Acquisition System. Available: http://protege.stanford.edu/ [2007 August 12].

Smith, M.K., Welty, C. and McGuinness, D.L., ed. (2004). [Online]. OWL Web Ontology Language Guide. World Wide Web Consortium (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). Available: http://www.w3.org/TR/owl-guide/ [2007 September 13].

Solomon, G. (2003 January 15). [Online]. Project-Based Learning: a Primer. TechLEARNING. Available: http://www.techlearning.com/db_area/archives/TL/2003/01/project.php [2007 October 27]

Starr, L. (2005). [Online]. Project-Based and Problem-Based Learning. Education World. Available: http://www.education-world.com/a_curr/virtualwkshp/virtualwkshp002.shtml [2007 November 20].

Teaching and Research in Engineering in Europe (TREE) (2007). [Online]. Problem Based Learning. Available: http://www.unifi.it/tree/dl/oc/b5.pdf [2007 November 19].

Thirteen Ed Online [Online]. WebQuests:Explanation. Available: http://www.thirteen.org/edonline/concept2class/webquests/index.html [2007 Novemeber 12].

Thomas, J.W. (2000). [Online]. A review of research on project-based learning. Available: http://www.bie.org/files/researchreviewPBL.pdf [2007 August 15].

WebQuest. [Online]. WebQuest.Org. Available: http://www.webquest.org/index.php [2007 November 11].

Wikipedia. [Online]. Ontology (computer science). Available: http://en.wikipedia.org/wiki/Project-based_learning [2007 September 12].

Wikipedia. [Online]. Project-based Learning. Available: http://en.wikipedia.org/wiki/Project-based_learning [2007 August 12].

Wong, A., Quek, C.L., Divaharan,S., Liu, W.C., Peer, J. and Williams, M. (2006). Singapore Students' and Teachers' Perceptions of Computer-Supported Project Work Classroom Learning Environments. Journal of Research on Technology in Education, 38 (4), p449-479

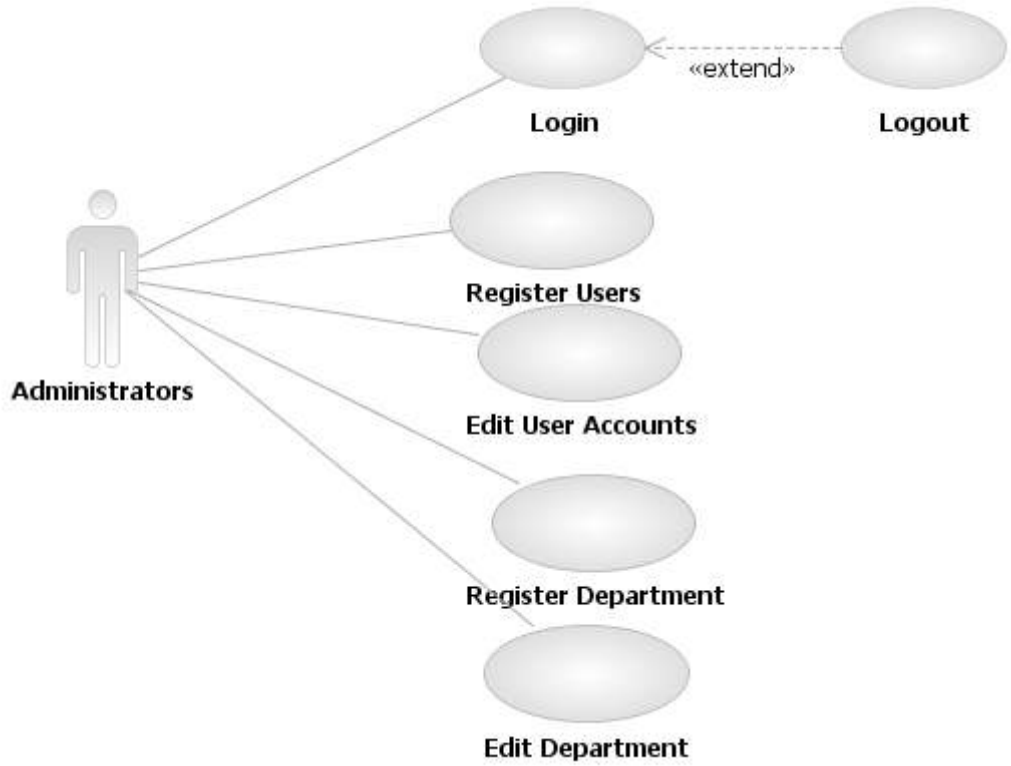
Xie, H. and Shen, W. (2006). Ontology as a Mechanism for Application Integration and Knowledge Sharing in Collaborative Design: A Review. Computer Supported Cooperative Work in Design, 2006. CSCWD '06. 10th International Conference (2006 May), p.1-7.

Zhu, X., Li, X. and Guo, W. (2007). Ontology Based Sharing and Services in E-Learning Repository. 2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007), p. 957-962.

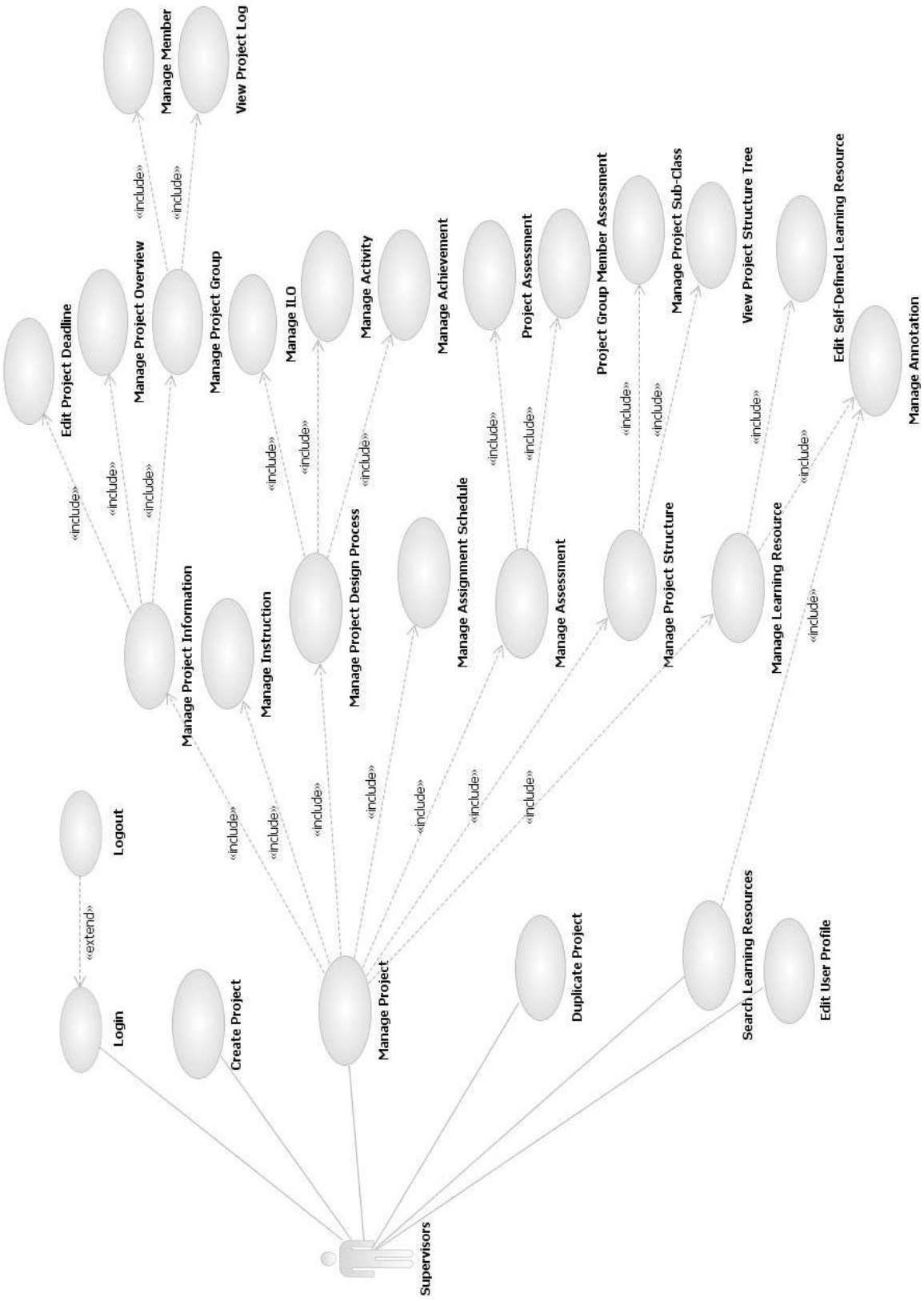
Appendix

A1: Use Case Diagrams of Project-Based Learning Environment

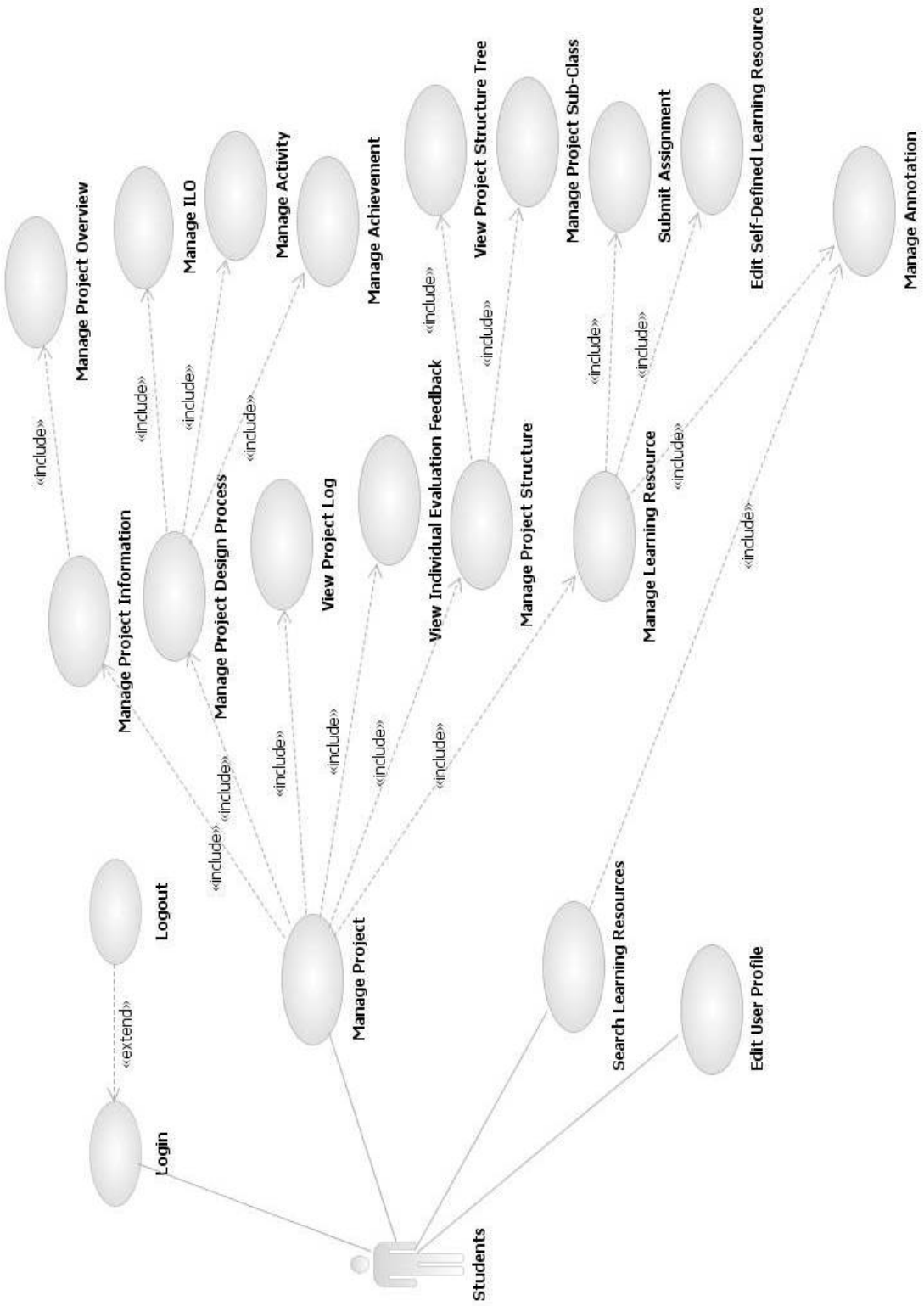
A1.1 Administrator



A1.2 Supervisor

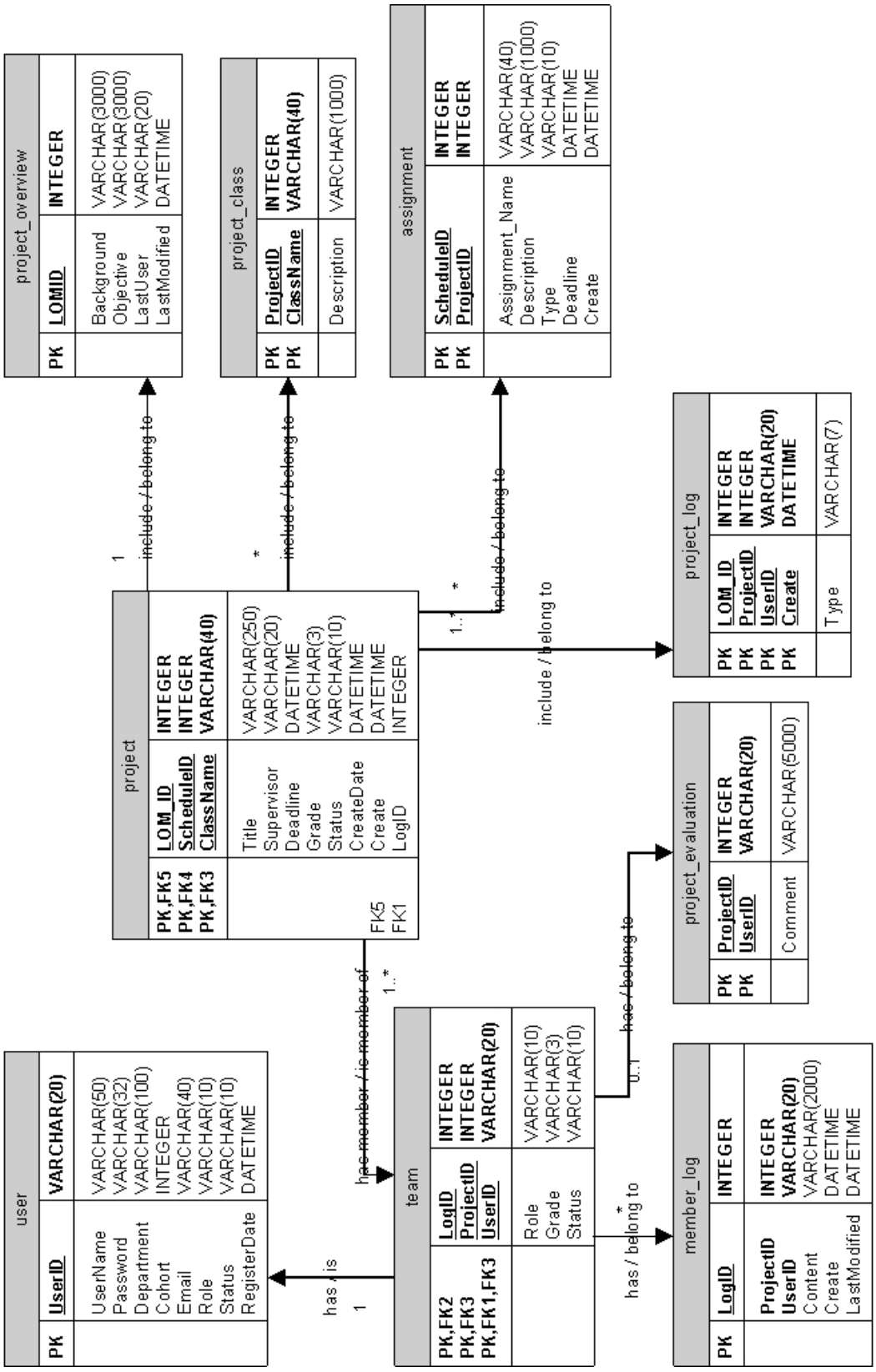


A1.3 Undergraduate

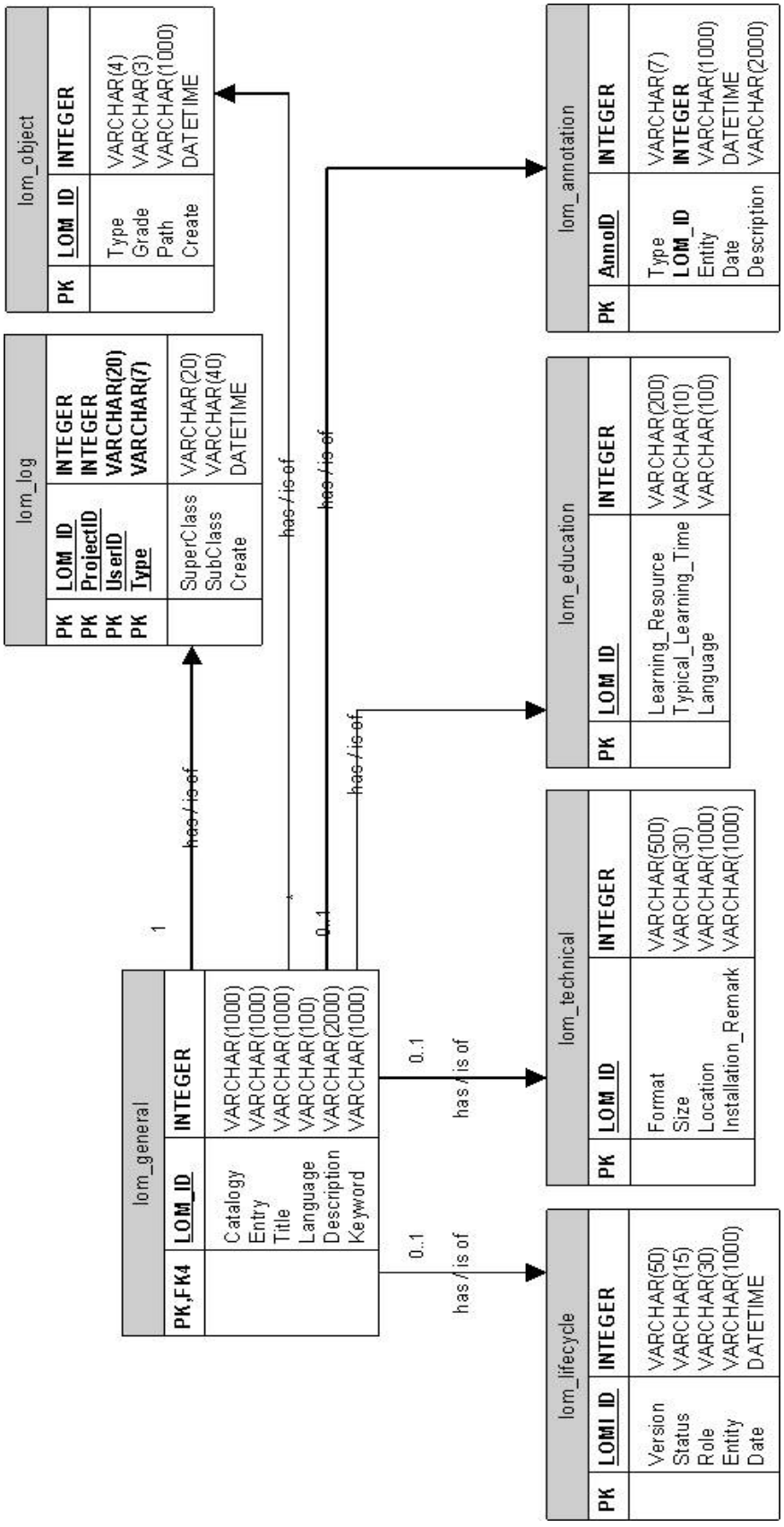


A2: Database Diagrams

A2.1 Project Database Diagram



A2.2 Learning Object Metadata Database Diagram



A3: Detail Information of System Module

Detail Information of System Module	
User Module	This module handles the manipulations of the user information.
Registration Handler	Construct the information from client-sides to register new user
Authentication Handler	Control users to login and log out.
Account Handler	Manage the information of user accounts including the projects of the users.
Project Module	This module manages projects and projects' information.
Project Handler	Manage the creations and general information of projects such as background, objective, deadline, instruction and etc.
Member Handler	Manage project group members and members' information of the projects.
Account Handler	Manage the information of user accounts including the projects of the users.
Project Design Handler	Manage the project design process the projects ◆ ***Manage ILO*** It manages the intended learning outcomes of the projects. ◆ ***Manage Activity*** It manages the activities of the projects, and their relationships with ILO and learning objects. ◆ ***Manage Achievement*** It manages the achievement of the projects, and their relationships with ILO and learning objects.
Project Structure Handler	Manage the information and description of the user defined sub-classes of the projects.
Assignment Handler	Manage the information and schedules of the project assignments.
Evaluation Handler	Manage the evaluation information of the projects and project group members.
Ontology Module	This module manages the project ontology models of each project.
OWLControl	Provide functions to manipulate the project ontology models. It works with other modules to handle the project structures, which manages the instances, relationships of instances, sub-classes of different ontology models.
Ontology Display Handler	Manage two different methods to retrieve the data of project ontology models for display. ◆ ***getLOList*** It retrieves the sub-classes and the instances of a specific class requested and directs to corresponding JSP for display. Also, it works with Database Module to display the general information of the instances together. ◆ ***getTreeNode*** It also retrieves the sub-classes and the instances of a specific class requested, but it works with Project Structure Handlers. It generates the retrieved data to display as the Tree Hierarchy.

Detail Information of System Module	
Learning Object Module	This module manages the information of learning objects.
Learning Object Handler	Work with other modules to manage the learning objects' information, including LOM, in different projects.
Annotation Handler	Manage the annotation records of the learning objects' LOM.
File Upload Handler	Handle the files of learning objects uploading to the system. Also, it checks whether the file type and size of the learning objects can be uploaded into the system.
File Download Handler	Handle the files of learning objects downloading from the system. It authenticates the users before the file is being downloaded.
Searching Handler	Manage the searching of learning resources. Different learning resources are being searched according to the users.
Database Module	This module provides services for other modules to access the system database records.
DBManipulator	Manage the connections with the databases, Jena Database and System Database. Jena Database is used for Jena API manipulating ontology models; System Database is used for all other tables. Also, this class provides functions to manipulate the results of other handlers in Database Module.
AdminControl	Manage the database records for supporting administration functions.
UserControl	Manage database records of user accounts.
ProjectControl	Manage database records for supporting the functions of managing projects.
LOMControl	Manage all LOM tables' records for learning objects.
EvaluationControl	Collect and organize the database records in different tables for project evaluation.
DisplayControl	Manage the database records for display the different lists of records in the system.
DisplayUtilityControl	Provides functions for assisting display information in JSP with database records.
Utility Module	This module contains different Java classes providing the functions, which are used within the system.
Constant	Manage all the constant variables used in the system, which allows system administrator to configure the environment settings more efficiently.
Checker	Provide the functions used to validate the data passing from the client-sides.
MD5	An open-source java class provides one-way encryption.
Data Module	This module is used to handle the information for display

Detail Information of System Module	
Filter Module	This module contain different filter classes to filter the request and response among client-sides and server-sides
Display Module	This module contains all JSP for displaying interfaces for users in the web browsers.
Project Display Handler	Handle the interfaces of the project structure which interacts with the Ontology Module to display the hierarchy of the Project Ontology Models.

A4: Initial Project Ontology OWL File (Simply Version)

```
<?xml version="1.0" encoding="UTF-8" ?>
- <rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
<owl:Ontology rdf:about="" />
+ <owl:Class rdf:ID="Learning_Reference">
<owl:Class rdf:ID="Project_Design" />
+ <owl:Class rdf:ID="ILO">
+ <owl:Class rdf:ID="Activity">
<owl:Class rdf:ID="Project" />
+ <owl:Class rdf:ID="Deliverable">
+ <owl:Class rdf:ID="Achievement">
+ <owl:Class rdf:ID="Assignment">
+ <owl:Class rdf:ID="General">
+ <owl:Class rdf:ID="Instruction">
+ <owl:ObjectProperty rdf:ID="hasProjectDesign">
+ <owl:ObjectProperty rdf:ID="hasOverview">
+ <owl:ObjectProperty rdf:ID="hasDeliverable">
+ <owl:ObjectProperty rdf:ID="hasReference">
+ <owl:ObjectProperty rdf:ID="achieveTo">
+ <owl:ObjectProperty rdf:about="#achieveWith">
+ <owl:ObjectProperty rdf:ID="relateBy">
+ <owl:ObjectProperty rdf:about="#relateTo">
- <owl:FunctionalProperty rdf:ID="LOMID">
<rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"
/>
- <rdfs:domain>
- <owl:Class>
- <owl:unionOf rdf:parseType="Collection">
<owl:Class rdf:about="#Project" />
<owl:Class rdf:about="#Project_Overview" />
<owl:Class rdf:about="#Project_Design" />
<owl:Class rdf:about="#Learning_Reference" />
<owl:Class rdf:about="#Deliverable" />
</owl:unionOf>
</owl:Class>
</rdfs:domain>
<rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
</owl:FunctionalProperty>
+ <owl:FunctionalProperty rdf:ID="belongTo">
</rdf:RDF>
<!-- Created with Protege (with OWL Plugin 3.3.1, Build 430)
http://protege.stanford.edu -->
```

A5: Project Tree Structure JavaScript File

```
// Open specific page in parent window of popup
function MainOpen(direction) {
   window.opener.location.replace(direction);
}
Ext.onReady(function() {
   // shorthand
   var Tree = Ext.tree;
   var tree = new Tree.TreePanel({
      el:'project_tree',
      autoScroll:true,
      animate:true,
      enableDD:false,
      containerScroll: true,
      loader: new Tree.TreeLoader({
         dataUrl:'../GetTreeNode'
      })
   });

   // set the root node
   var root = new Tree.TreeNode ({
      text: '<a href="#"
onclick="MainOpen(\'../project/viewProject.jsp\')"
>Project</a>',
      id:'Project',
      draggable:false
   });

   // set the Class Project Overview node
   var overview = new Ext.tree.AsyncTreeNode({
      text: '<a href="#"
onclick="MainOpen(\'../ManageSubclass?inclass=Project_Overview\
')" >Overview</a>',
      id:'Project_Overview',
      draggable:false
   });

   // set the Class Project Design node
   var prjManage = new Ext.tree.AsyncTreeNode({
      text: '<a href="#"
onclick="MainOpen(\'../ManageSubclass?inclass=Project_Design\')
" >Project Design</a>',
      id:'Project_Design',
      draggable:false
   });

    // set the Class Learning References node
   var references = new Ext.tree.AsyncTreeNode({
      text: '<a href="#"
onclick="MainOpen(\'../ManageSubclass?inclass=Learning_Referenc
e\')" >Learning References</a>',
      id:'Learning_Reference',
      draggable:false
   });
```

```
    // set the Class Deliverables node
    var deliverables = new Ext.tree.AsyncTreeNode({
        text: '<a href="#"
onclick="MainOpen(\'../ManageSubclass?inclass=Deliverable\')"
>Deliverable</a>',
        id:'Deliverable',
        draggable:false
    });

    // reload node
    var refresh = function(node) {
        if (!node.isLeaf())
            node.reload();
    };

    overview.on('dblClick', refresh);
    prjManage.on('dblClick', refresh);
    references.on('dblClick', refresh);
    deliverables.on('dblClick', refresh);


root.appendChild(overview,prjManage,references,deliverables);

    tree.setRootNode(root);

    tree.render();
    root.expand();

});
```

A6. Progress Log

Project Month	Progress Log
Oct	1. Study and Review Learning Models (Focus on Project-Based Learning). 2. Review existing Project-Based Learning and Web Learning Systems. 3. Study about ontology and standard of learning object metadata. 4. Learn using Jena and Protégé as a tool of design and write ontology model. 5. Design an Ontology Base Model for common Project Knowledge Structure. 6. Write a Project Plan. 7. Design web-application system (Possible Function).
Nov	1. Write a interim report - Project introduction - Background information and review of Project-Based Learning, other learning models, existing learning systems, etc - Review the research about using ontology and learning object metadata - System overview of the web-based environment - Proposed ontology model and learning object metadata structure 2. Use Protégé to prepare the base Project Structure model 3. Learn Jean API to handle ontology models with java programming 4. Learning Extjs (open source) which may be used in the environment presentation interfaces 5. Reviewing the design of the environment 6. Implementing the basic interfaces of the environment
Dec	1. Implement ontology model for Project Structure 2. Use Extjs (open source) to present the Project Structure Tree 3. Implement the basic interfaces of the environment 4. Implement the functions of the web-based environment -User Manage -Project Manage 5. Implement the system database
Jan	1. Implement the functions of the web-based environment

Project Month	Progress Log
Feb	Refine and Implement the functions of the environment: Admin: 1. Register and Manage User Account and Information 2. Register and Manage Department of User Supervisor: 1. Create New Project (Manage Project Structures) 2. Manage Instructions in a Project 3. Project Design - Manage Intended Learning Outcomes (ILO), Activities, Achievements - (Ontology) Manage the relations among ILO, Activities, Achievements and Learning Resources with the Project Ontology Model 4. Manage Learning Resources edit, upload and download in the environment with Project Ontology Model and Learning Object Metedata (Including Annotation) 5. Manage Project Assignment Schedules 6. View Project Members Progress Log 7. Evaluate Learning Resources, Assignments, Project Members 8. Copy Exisitng Project as a new Project Members: 1. Manage Supervisors assigned Projects (Manage Project Structures) 2. Manage Project Design as Above 3. Manage Learning Resource as Above 4. Manage Hand-in Assignment according to the Project Assignment Schedule 5. Manage Individual Project Progress Log 6. View Evaluation Result All Users: 1. Search Learning Resources in the environment
Mar	1. Continue and refine all the remaining functions Perform the testing of all functions of the environment 2. Preparing the Final Year Report: - Refine Chapter 2 - Additional Background Information 3. Writing the first version of the report