A Moodle Plugin for Socratic Questioning

Dissertation

submitted in partial fulfillment of the requirements
for the degree of

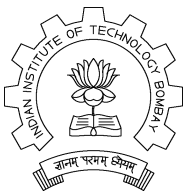
Master of Technology

by

Manish Chouhan
Roll No: 113050082

under the guidance of

Prof. Sridhar Iyer



Indian Institute of Technology, Bombay
Mumbai - 400076.

2013

Acknowledgments

I express my sincere gratitude towards my guides **Prof. Sridhar Iyer** for his constant help, encouragement and inspiration throughout the project work. Without his invaluable guidance, this work would never have been a successful one. Last, but not the least, I would like to thank the whole KReSIT family which made my stay at IIT Bombay a memorable one.

Manish Chouhan
IIT Bombay
June 28, 2013

Abstract

Web based distance education is used in higher education as a means of providing knowledge as a teacher to different community of individuals. Teaching a subject to a number of students is always a challenge for the teacher. Teachers uses many teaching strategies to teach a subject student to students. But when it comes to a computer based system it becomes very difficult. So providing an effective Intelligent Tutoring System (ITS) Framework is a challenge.

While many approaches exist for solving this problem, I select Socratic Questioning teaching strategy. I have developed a moodle plugin to implement this teaching strategy, which will help the teachers to teach their course using Socratic Questioning method in distance learning environment. This plugin provides functionality to create quiz using Socratic Questioning strategies. Then I have discussed functionality of moodle plugin developed by me which will be used as an activity in moodle, followed by implementation of this plugin, and challenges in integrating my plugin into moodle.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Students usually have to attend classes physically to learn subject. Only a single instructor handles the entire class. Now a days as population is increasing so as number of students in single class. Instructor may not be able to take care of each individual student. Then teaching subject to students is a tough challenge for teacher. These days technology is improving rapidly so integration of technology into teaching field also increasing. As a result on-line learning and distance education systems are gaining attention which can be available at any place and at any time. So the education system can provide support to such place and time independent learning. Also this system treats each learner with equal care.

1.2 Goal

Goal of my research project is to provide a framework which can help the teachers to create ITS for their course and teach their course in distance learning environment. This ITS should be created as a plugin for moodle which should provide responsive and interactive teaching facilities for teacher, sequence the curriculum and help the learners to improve their knowledge without the help of human teacher. This system will be easy to use for both teacher and students.

To achieve this goal I have developed a plug-in for moodle which allows teachers to teach their course using Socratic method. This plugin works as an activity in moodle. This activity provides a interface for teacher to sequence the curriculum of the course and to teach their course using Socratic questioning. In this system teacher should be able to enter questions in the quiz and also should be able to connect them to each other. Teacher should also be able to edit question and their connection with other questions. system should be easy enough for teacher to add a quiz and edit quiz. I have implemented all the functionality which is reported in chapter 5.

1.3 Thesis Outline

chapter 2 have discussion about literature survey on present ITSs. In chapter 3, I have discussed about some teaching strategies. after that I explain Socratic Questioning teaching strategy, Types of Socratic questioning and principles used in this teaching method. In chapter 4, I describe how moodle is organized, how moodle works and how to develop a new activity in moodle. Chapter 5 explains System specification like operating environment, users of the system and functional requirements. Chapter 6 describe Implementation details of this moodle plugin for socratic questioning. This chapter describe functional logic and database schema. This chapter also has screenshots of system. Chapter 7 explains the details of challenges I have faced throughout developing this Socratic module. This chapter also explain testing of system. chapter 8 contains conclusion of this project and future work needed for this plugin.

Chapter 2

Teaching Strategies

There are many teaching strategies which can be implemented in ITS to provide one-to-one teaching to a large number of students. Some of them are:

- **Socratic Questioning teaching strategy [2] :** Socratic questioning is a pedagogical technique in which teacher don't give information directly but ask a series of questions. By giving the answer to the teacher's questions, student discovers the knowledge or gain deeper insight of the knowledge.
- **Guided Discovery teaching strategy [8] :**In guided discovery the teacher is responsible for guiding the students throughout the learning process. Teacher plans to teach a concept and asks a focus question about the concept. In Ideal guided discovery each question will have single answer.
- **Scaffolding teaching strategy [6]:**Scaffolding strategy is based on vygotskin theory which is (Zone of proximal development (ZPD)) [7]. According to this theory ZPD is distance between student's level of development through independent problem solving and level of development through problem solving under adult guidance or with the help of more capable students. In scaffolding first teacher determine student's ZPD then provide hints to the student based on his/her ZPD.

I select Socratic Questioning method to implement as a plugin of moodle because this is very well known teaching technique. This method also helps students to develop critical thinking attitude and inquiring attitude which is essential for a learner. Socratic Questioning helps the students to debug their own theories. It also prevents them from making mistake again in future. This method uses questioning about student's existing beliefs. And lead their beliefs/ideas to a contradiction.

2.1 Socratic Questioning

Socratic questioning is basically a dialogue conversation between two teacher and student. First, instructor starts the question and student response. In return instructor reformulates a new question according to the response given by student. Questioning and answering is structured systematically to reach at ultimate goal. This method also helps students to develop critical thinking attitude and inquiring attitude which is essential for a learner. Socratic Questioning helps the students to debug their own theories. It also prevents them from making mistake again in future. This method uses questioning about student's existing beliefs. And lead their beliefs/ideas to a contradiction. After contradiction student are inspired to rethink the primary question.

I used following example to illustrate Socratic questioning.

```
#include<iostream>
using namespace std;
int main()
{
int m,n;
float x,y;
m=x=n=y=7.89;
cout <<m<<n<<x<<y;
}
```

Teacher shows above program and ask the question as:

Teacher: what is the value of m,n,x,y?

Student: m=7, n=7, x=7.89, y=7.89

Here the teacher finds two things which are:

1) Student knows difference between int and float.

2) Student has not realized that x is set to value of n.

So teacher leads the dialog as:

Teacher: What is y=7.89?

Student: An expression.

Teacher: What is the value of this expression?

Student- 7.89

Teacher: y = 7.89 gives the value 7.89. It means m=x=n=y=7.89 is same as

m=x=n=7.89?

Student: yes

Teacher: Then what is the value of n=7.89?

Student: 7

Teacher: Now it means m=x=7?

Student: yes

Teacher: what is the value of x now?

Student: 7

This last answer is contradicting student's first answer which is x= 7.89.

Teacher: x can't be 7.89 then what is value of m,n.x,y?

Student: m=7, n=7, x=7, y=7.89

Above example shows that teacher didn't give direct answer to the student but he asked series of questions according to student's belief and lead the student to contradiction to make him realize his mistake. This kind of questioning helps the student to know that they don't know about concept properly and inspired them to rethink on primary question.

2.2 Types of Socratic Method

- Classic Socratic Method
- Modern Socratic Method

2.2.1 Classic Socratic Method [1]

Classic Socratic Method uses questioning to lead on-going conversation towards contradiction of other person's belief. In classic Socratic Method one person challenge the other person's belief, and forces them to rethink the primary question under discussion again. In this method neither questioner nor respondent (other person) knows the correct answer. The only aim of questioning is to rethink other persons belief.

The result of classic Socratic Method is failure to find a satisfactory answer to the primary question. This failure helps the respondent to realize his/her ignorance about the topic of on-going discussion. This failure may inspire the respondent to think about question from the beginning. If satisfactory answer is found, this method becomes the modern Socratic Method. This method is difficult to implement in real conversation because of handling the wide range of possibility of answers is impossible. That's why modern Socratic Method gains attention of people because of possibility of answers is limited in this method.

2.2.2 Modern Socratic Method [1]

In modern Socratic Method, questioning is used to lead a student's mind to knowledge through small steps (series of questions). This knowledge can be specific data, training approach to problem solving. In this method teacher ask direct question that have predefines range of answers and allow student to answer correctly before moving to the next question. If student is unable to answer question correctly then teacher should lead the student by asking series of question to a point where he/she can answer the question correctly. Teacher should ask helpful question to achieve desired lesson. As modern Socratic Method is suitable for training approach I am using this method to design questions for cs101 course.

2.3 Principles used in Socratic questioning [3]

- **Generate different examples to students:** At the beginning tutor can generate example to start the dialogue. To verify whether student really understand the concept tutor can generate example.
- **Recall knowledge the student already has:** If students response is incorrect than review the concepts they have already studied.
- **Determine students belief:** If student make prediction then ask the question to know why he make prediction and try to verify it with the concept student just studying.
- **Ask for prediction:** Present new scenario to the student and ask the student to make prediction and produce new concept.
- **Present counterexamples:** If students response or prediction is incomplete or inconsistent then give him a counterexample.
- **Testing of hypothesis formed by student:** If student makes hypothesis then present a scenario on which he can test his hypothesis.
- **Entrap the student and lead him/her to contradiction. When he/she has not identified all the relevant factor:** If students response is incomplete then pose misleading question to contradict the students responses.
- **Question further to elaborate the concept:** If student missed any issue or fact about concept then generate sub-question to bring his attention to this issue or fact.
- **Help the student in establishing new rules:** If students hypothesis is verified by testing than help him to establish universal rule.
- **Ask the student to apply new rules:** Present the scenario to the student and ask him to apply universal rule which is made by him.

2.4 Qusetions used for Testing

Concept description: Working of assignment operator, difference between int and float value
Question 1 : What is the output of following program?

```
#include<iostream>
using namespace std;
int main()
{
int m,n;
```

float x,y;
m=x=n=y=7.89;
cout <<m<<n<<x<<y;
}

option 1: m=7 n=7 x=7.89 y=7.89 (Q2)
option 2: m=7 n=7 x=7 y=7.89 (Q1 of next concept)
option 3: m=7.89 n=7.89 x=7.89 y=7.89 (Q2)
option 4: m=7 n=7 x=7 y=7 (Q10)

Question 2 : In statement m=x=n=y=7.89; what is y=7.89?
option 1: An expression whose value is 7 (Q8)
option 2: An expression whose value is 7.89 (Q3)
option 3: Only a statement not an expression (Q2 Try again)

Question 3 : It means m=x=n=7.89. Now what is n=7.89?
option 1: An expression whose value is 7 (Q4)
option 2: An expression whose value is 7.89 (Q5)
option 3: Only a statement not an expression (Q3)

Question 4 : It means m=x=7. Then what is value of x?
option 1: 7 (Q1)
option 2: 7.89 (Q4)

Question 5 : In our program type of n is integer. What kind of value stored by int type of b variable?
option 1: Only integer (Q6)
option 2: Fractional (Q5)
option 3: Character (Q5)

Question 6 : In expression n=7.89 value 7.89 is of type ____?
option 1: Integer (Q6)
option 2: Fractional (Q7)
option 3: Character (Q6)

Question 7 : What is integer value of 7.89 in expression n=7.89?

option 1: 7 (Q3)
option 2: 0.89 (Q7)

Question 8 : Type of variable y is float, what kind of value a float variable can store?
option 1: Integer (Q8)
option 2: Fractional (Q9)
option 3: Character (Q8)

Question 9 : 7.89 is fractional so expression y= 7.89 will give _____?
option 1: 7 (Q8)
option 2: 7.89 (Q2)

Question 10 : In statement m=x=n=y=7.89; what is y=7.89?
option 1: An expression whose value is 7 (Q8)
option 2: An expression whose value is 7.89 (Q1)
option 3: Only a statement not an expression (Q10)

Chapter 3

Related Work

Socratic questioning has been widely used as a method to implement an ITS. In this report I described architecture of some web based ITS, which have been developed for different domains.

3.1 Web based Socratic tutor for tree recognition [9]

This web based ITS was developed to provide tutoring for the classification and identification of different European forestry species.

3.1.1 Architecture

Diagram in fig 3.1 is reproduced from: Web based Socratic tutor [9] There are two types of modules present in this architecture. First one data modules that contains data and knowledge and another one is functional modules which are responsible for proper functioning of ITS.

Domain knowledge Conceptual model

This module contains content of domain knowledge that is acquired by the students. It also contains sub-domain model that contains the knowledge that is to be known to the students.

Student model

It stores all the information of students interaction session with ITS (number of session, session duration, session trace, pages visited)

Tutorial strategies model

It contains set of rules and session configuration parameter given by ITS designer. This parameter is uses by the teacher to define tutorial strategies.

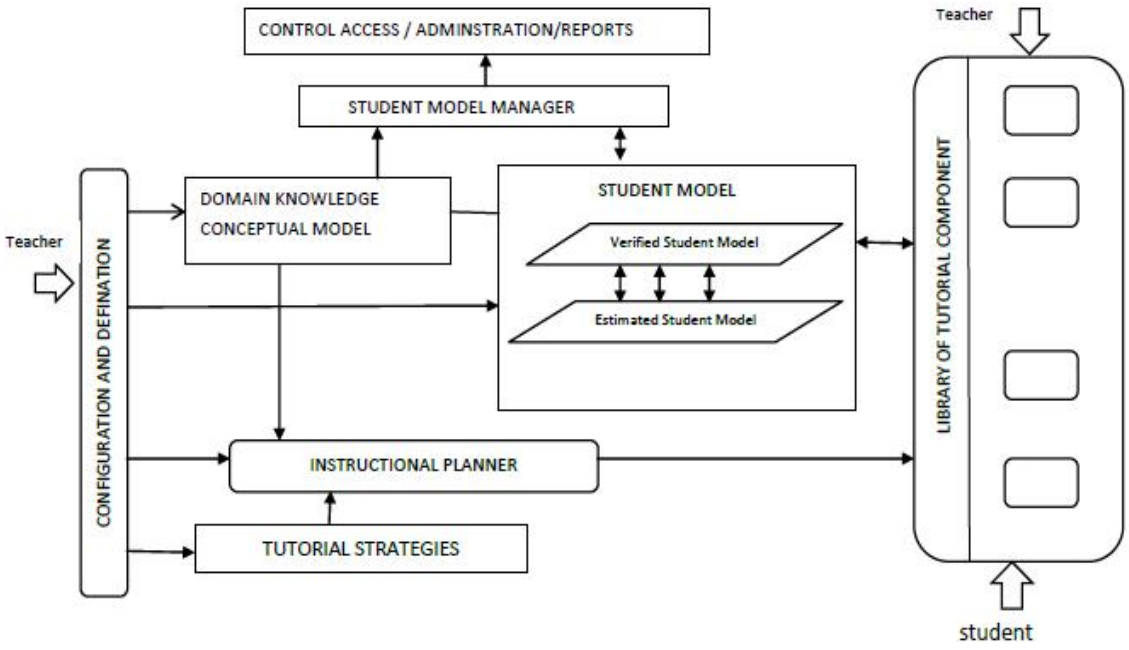


Figure 3.1: ITS architecture for tree recognition

Configuration and definition model

This module allows experts and teachers to define content for the conceptual domain model and tutorial strategies model.

Student model manager

This module control changes in student model according to students performance in a particular domain.

Access control model

It controls the students system access (login, password) and makes reports about the students.

Instructional planner model

This module takes the decision during the learning process. It takes the decision about next tutoring concept and best tutoring strategy for student based on his/her student model.

Tutorial component library

In this library every component implements a set of tasks that has to be performed by student in his/her learning process. These tasks are included by the teacher in the component. In this architecture Socratic tutor is

a tutorial component that teaches the student by asking questions to the student.

3.2 ITS for Q&A strategy [2, 6, 8]

This software system was developed to provide a generic framework which can support more than one teaching strategies and independent from subject domain. Then building an ITS from this framework that may provide responsive and interactive teaching facilities for students, track their progress , assess their performance, sequence the curriculum and help the learners to improve their understanding without human teacher intervention. This system achieved this goal by using a pattern which is independent from subject domain called multiple choice questions. This system was built for 4 different strategies which are supported by underlying architecture of system. These strategies are

- Socratic Questioning strategy
- Scaffolding Strategy
- Guided Discovery Strategy

3.2.1 Architecture

In this ITS shown in figure 3.2 there are only four basic component student model, teaching model, domain model and user interface model. Figure 3.2 is reproduced from [2]

Domain Model

This model also contains the content of domain knowledge which will be acquired by students. It has two parts first part contains course structure (CS) and topic structure (TS), and second part contains question module which will be formed by the experts using Socratic questioning strategy.

Student Model

It represents all the knowledge and skills which students already have. It also contains students performance record. In this ITS performance criteria is problem solving ability.

Teaching Model

Teacher decides all content of domain that is to be taught to the student, with the help of this model and teacher can also decide teaching strategy. This decision of selecting content and teaching strategy is depends upon student model.

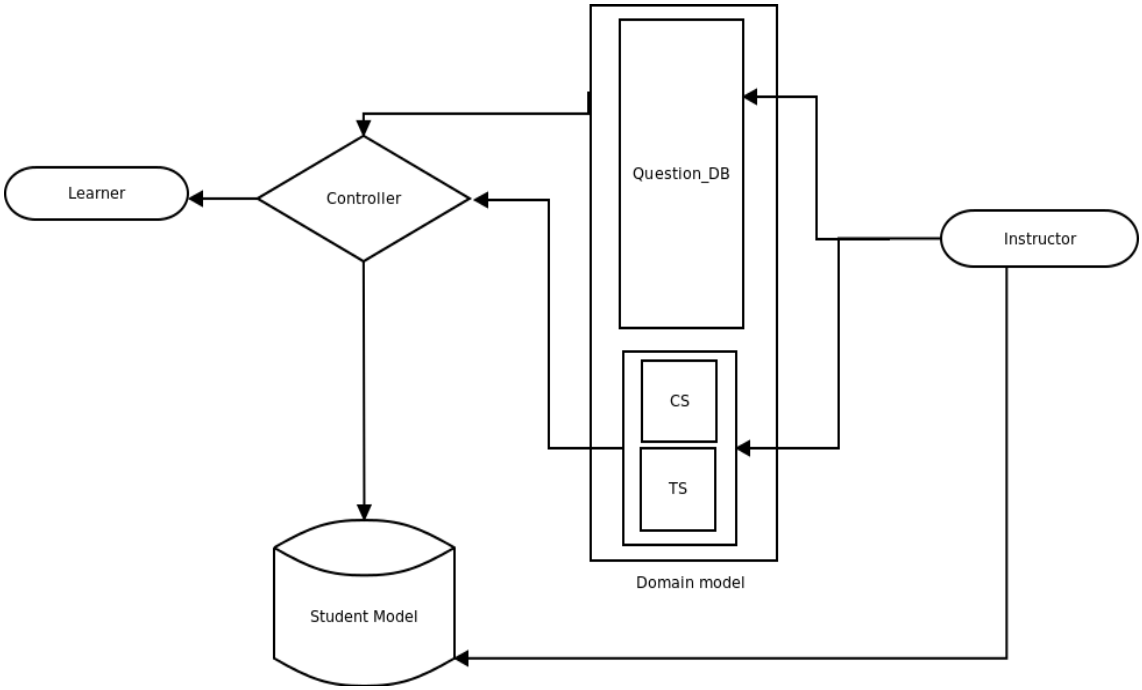


Figure 3.2: Architecture of ITS Framework

User Interface Model

This component controls the interaction between user and ITS. It translates systems internal representation to interface language and vice versa.

Chapter 4

Moodle Internals

4.1 Introduction of Moodle

Moodle is a Learning Management System, or Course Management System. Its goal is to give teachers and students the tools they need to teach and learn respectively. The idea of Moodle came from Social Constructionist pedagogy[5], however, it can be used to support any style of teaching and learning.

Moodle is organized as an application core, which contains a large number of plugins to provide different functionality. Moodle is designed such a way that some one can easily extend and customize moodle's functionality without modifying the core libraries. When somebody wants to customize or extend their own Moodle ,he should always do so according to the plugin architecture.

The standard Moodle is distributed among Moodle core and a number of plugins, because of that we can use new Moodle installation immediately to start teaching and learning. After installation we can change Moodle site for our purpose by changing the default configuration option in configuration file, and by installing and removing plugins.

Plugins in Moodle are of specific types. A plugin communicates with Moodle core using different APIs. Functionality common to all plugins like installation, upgrade, permissions, configuration etc. are handled consistently across all plugin types.

Physically, a Moodle plugin is just a folder of PHP scripts (and CSS, JavaScript, etc.). To communicates with the plugin, moodle core looks for particular entry points, which should be defined in the file lib.php within the plugin.

4.1.1 Overview of Moodle core

Moodle core provides all the structure that is necessary to build an LMS. Key concepts of moodle is that all the different plugins should need to work with some necessary module. These includes the following

- Courses and activities

- Users and Groups
- Enrollments and access control
- Navigation, settings and configuration
- JavaScript library
- Installation and upgrade
- Logs and statistics

Some important plugin types are Activities and resources, Blocks, Course formats, Themes etc.

Activities and resources are most basic and important component of moodle. That actually make up a course. And these are the main tools for teaching and learning. pages,links and IMS content packages are some example of resources. Forums, wikis, quizzes, and assignments are some example of activities. Both activities and resources are installed in the $HOME/mod folder. Here $HOME represents the current directory where moodle is installed.

4.1.2 Organization of moodle code

There are two layers in moodle implementation which is used to separate presentation from the business logic. The outer layer is the theme which controls the all visual aspects of the Moodle interface. Then there are renderer classes which generates the HTML to be output from the data supplied by the domain model.

4.2 Configuration

Inside the HOME directory we can find the Moodle configuration file, config.php. In this configuration file all the configuration settings regarding the database and different configuration variables are stored. Among these variables, three variables are used frequently while developing new module. These are,

$CFG $\rightarrow$ *wwwroot* : Root web address for Moodle.

$CFG $\rightarrow$ *dirroot* : Root directory address for Moodle.

$CFG $\rightarrow$ *dataroot* : Root data directory address for Moodle.

4.3 Library Function

- All the function to update, insert, delete database entries in Moodle are defined in the **$HOME/lib/dml/moodle_database.php** file. For example
  function delete_records($table, array $conditions=null){}
  Above function is used to delete the records from a table where all the given conditions met. If conditions not specified, table is truncated.

- General purpose Moodle functions are defined in the $HOME/lib/moodlelib.php. This file contains the function related to authentication of user. For example
  function require_login()
  checks that the current user is logged in, and optionally whether they are allowed to be in a particular course and view a particular course module.

- Library of functions for web output are defined in the $HOME/lib/weblib.php file. For example
  function get_referer($stripquery)
  It returns the URL of the HTTP_REFERER, less the querystring portion if required. If $stripquery,variable is true it also removes the query part of the url.

4.4 Database of Moodle

The Moodle database structure is defined in install.xml files. And database structure for every plugin is defined inside the db folder in each plugin directory. For example $HOME/lib/db/install.xml defines the tables used by Moodle core.

4.4.1 XMLDB

XMLDB is Moodle's database abstraction layer. it is the library of code which is used by Moodle to interact with and access the database. XMLDB enables moodle to work with some more RDBMS (MSSQL and Oracle). Uses of XMLDB in moodle are

- It provides one layer for DB creation/upgrade.
- It provides one layer for DB handling.
- It is Simple, usable and effective.

4.4.2 XMLDB editor

The XMLDB editor is a tool for making the .xml files that specify how moodle should set up its' database tables. We can edit database structure

of All module defined in install.xml files present under each db directory in Moodle using this editor.

procedure to edit database table using XMLDB editor

- Log into Moodle site as Administrator
- click on the site administration than Development than XMLDB editor in setting blog as shown in fig 4.1 (a)
- Load the database of a module and click on Edit, as shown in fig 4.1 (b)
- It will show list of tables click on Edit whichever table we want to edit,as shown in fig 4.1(c)
- It will show Fields of table. now click on Edit whichever field we want to edit,as shown in fig 4.1(d)
- It will show page for editing field where we can edit the field of table.4.1(e)
- similarly we can also delete and add a table or a field in the table.

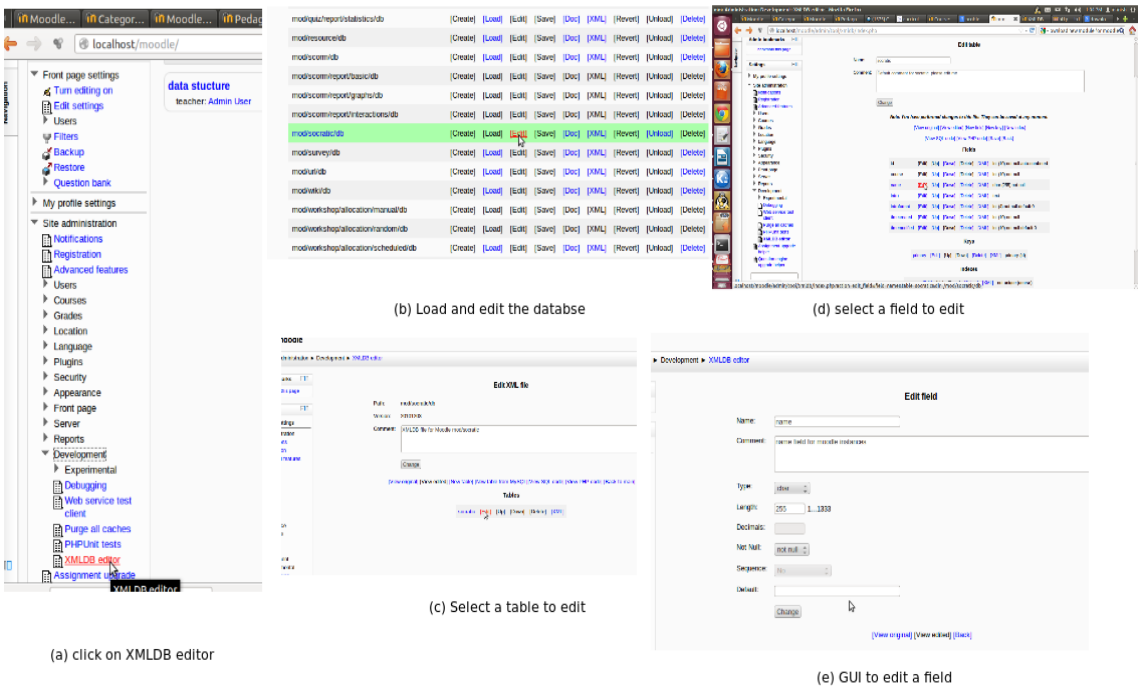


Figure 4.1: Use of XMLDB editor

4.5 Development of a new Activity in moodle

- Download new module template code from www.moodle.org.
- Unzip the archive.
- Rename the newplug/ folder to the name of our module (eg "socratic"). The module folder MUST be lower case.
- I edit all the files in that directory and its subdirectories and change all the instances of the string "newplug" to my module name ("Socratic"). To do this I use the following command
  $ find . -type f -exec sed -i 's/newplug/socratic/g' { } \
- Rename the file lang/en/newplug.php to lang/en/socratic.php
- Place the Socratic folder into the /mod folder of the moodle directory.

Chapter 5

System Specification

This system helps the instructor to teach their course through Socratic Questioning strategy in distance learning environment. It has been integrated with moodle as a plug-in. So it has been developed by following the rules methods and technology which are necessary to build a plug-in for moodle. Moodle is mainly used to manage different types of courses. Here a teacher can add different activities as part of their course, like: quiz, wiki, assignments, survey as shown in fig 5.2. Students need to carry-out these activities. Socratic module is one type of activity, which comes under the Add an activity or Resource link. as shown in fig 5.1. In this activity instructor creates an activity for each topic. Socratic quiz can be created concept vice. For each concept there will be a cycle of questions.

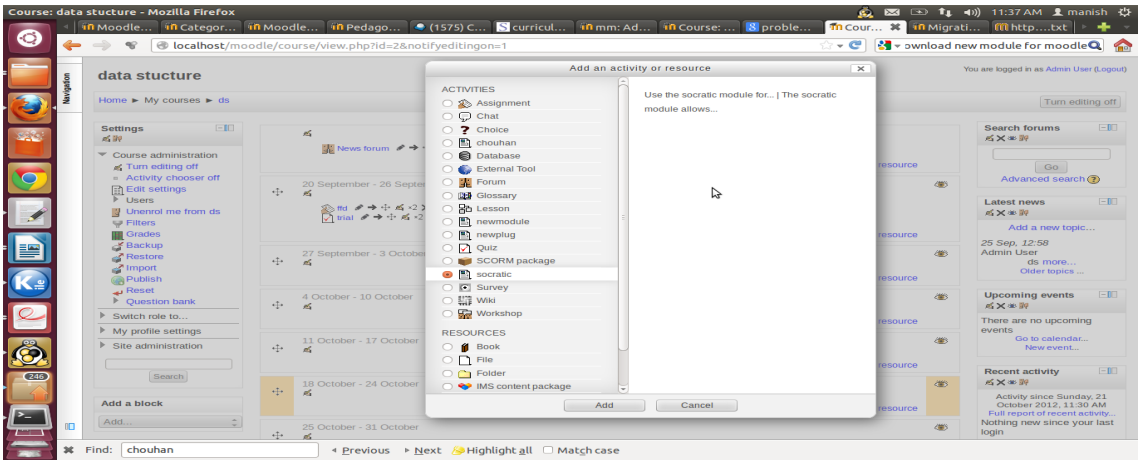


Figure 5.1: Socratic ITS in moodle as an activity

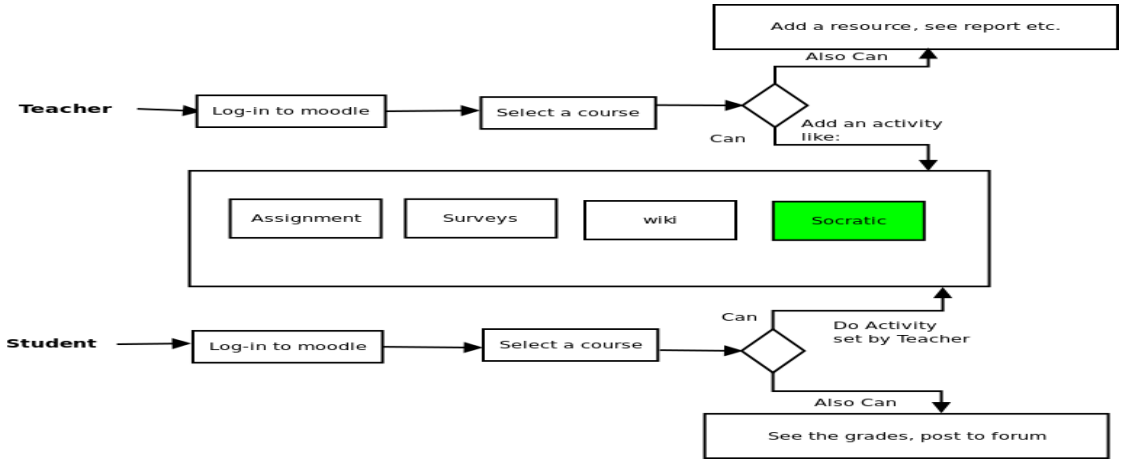


Figure 5.2: Socratic ITS module in moodle

5.1 System Specification

5.1.1 User characteristics

User must be familiar with basic handling of computer and Internet browsing to use this new Socratic module.

5.1.2 Operating Environment

This module will be developed using PHP, HTML, MySQL, JavaScript and CSS. So this module will run on different operating environments, which support these technologies and where Moodle runs.

5.1.3 Users of the system

There are 2 types of users, Teacher and Student. Roles of users defined below.

Teacher Teacher is the person who is the instructor for the course, where Socratic activity can be added. Teacher can add one Socratic quiz activity to teach students. Teacher can edit Socratic quiz. In editing teacher can either edit the question or change the sequence of questions.

Student Student is one, who is attending the course. Student can attempt quiz created by teacher.

5.2 Functional Requirements

5.2.1 User as a Teacher

- User will select Socratic activity from add and activity menu.
- System will prompt teacher to select one option either add question or edit question.
- If user will select add question then system will check whether cycle of previous concept is completed or not. If not then system will prompt user to complete previous concept otherwise system will prompt user to add new concept in the quiz.
- After that system will prompt user to enter question details and options. It will also ask for right option.
- After submitting this information system will prompt user to enter next question corresponding to each option or to link that option to previous question.
- This process will continue till user wants to add concepts in the quiz .

5.2.2 User as a Student

- Student can see the link of Socratic activity created by teacher.
- Student will click the link and system will show him quiz exercise created for this subtopic.
- If quiz has incomplete cycle then student is not allow to attempt quiz. it will show a message to student that 'quiz is not completed'

Chapter 6

Implementation

6.1 Database Structure

There are five database tables in the Socratic Module which I have created. Database structure of this module is shown in fig 6.1

- **mdl_socratic:** It stores the information regarding the Socratic module, like name, introduction etc.
- **mdl_socratic_concept:** It stores the information regarding concepts of each course module. like description, course id course module id
- **mdl_socratic_parsing:** It stores the information regarding of status of question. How many options of a question is linked with another question.
- **mdl_socratic_questions:** It stores the questions entered by teacher.
- **mdl_socratic_sequencing** It stores the information of sequence of questions.

6.2 Development Documentation

6.2.1 Authorization

To view any of the file in the Socratic module user need to be logged in. To check if the user is logged in or not, we use the require login() function defined in the lib/moodlelib.php file. **require login($course_id, $setwantsurltome,$cm)**

- **$course_id:** It is the course module or id of the current course
- **$cm:** Course module object

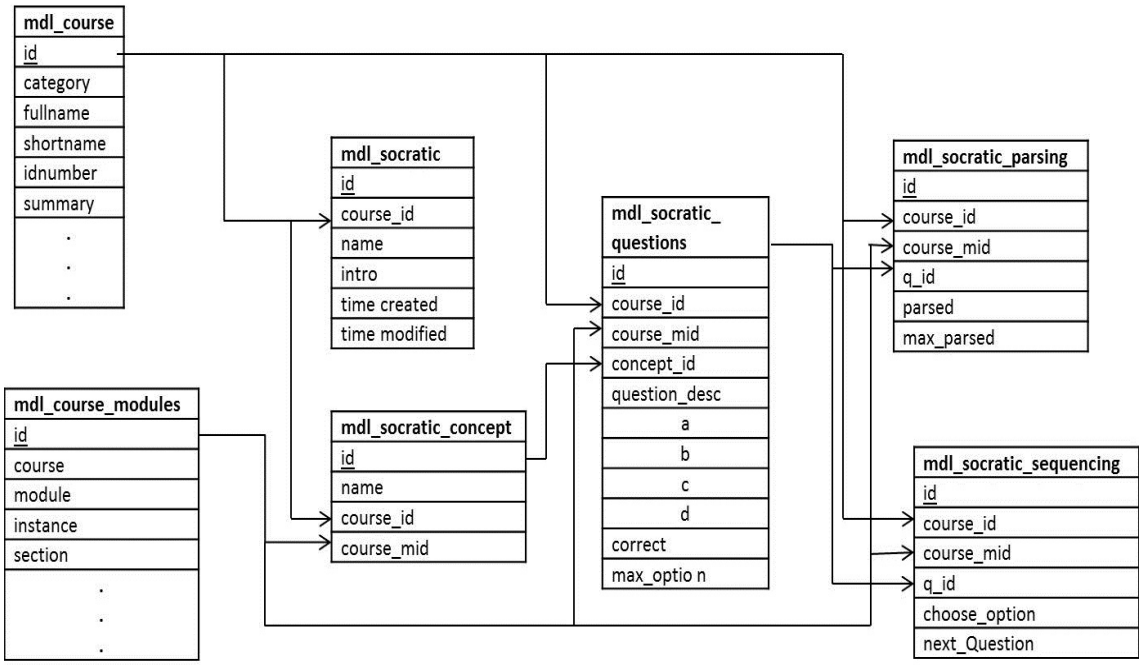


Figure 6.1: Databse structure

- **$setwantsurltome:** True, if we want to set the $SESSION-wantsurl variable

This require login() function checks whether the current user is logged in and is allowed to be in the particular course to view this particular course module. If user is not logged in then he will be redirected to the Moodle log-in page. If $course_id is given and the user is not enrolled in that particular course then he will be redirected to the course enrolment page. If $cm is given and the module is hidden, then it will not be shown in the course homepage if the current user role is not teacher.

Now the local access control capabilities are defined in the local /db/access.php file. For this Socratic module we have defined two capabilities. These are,

- **edit**: It is to authorize teacher to edit the socratic quiz
- **attempt**: It is to authorize studnt to attempt the socratic quiz

To check for the capability of user I have used the has_capability(capability name, $context) function.

6.2.2 Creation of Socratic module

Table 6.2.2 shows purpose of different files in Socratic activity module in Moodle. Fig 6.2 shows organization of the main files in socratic module

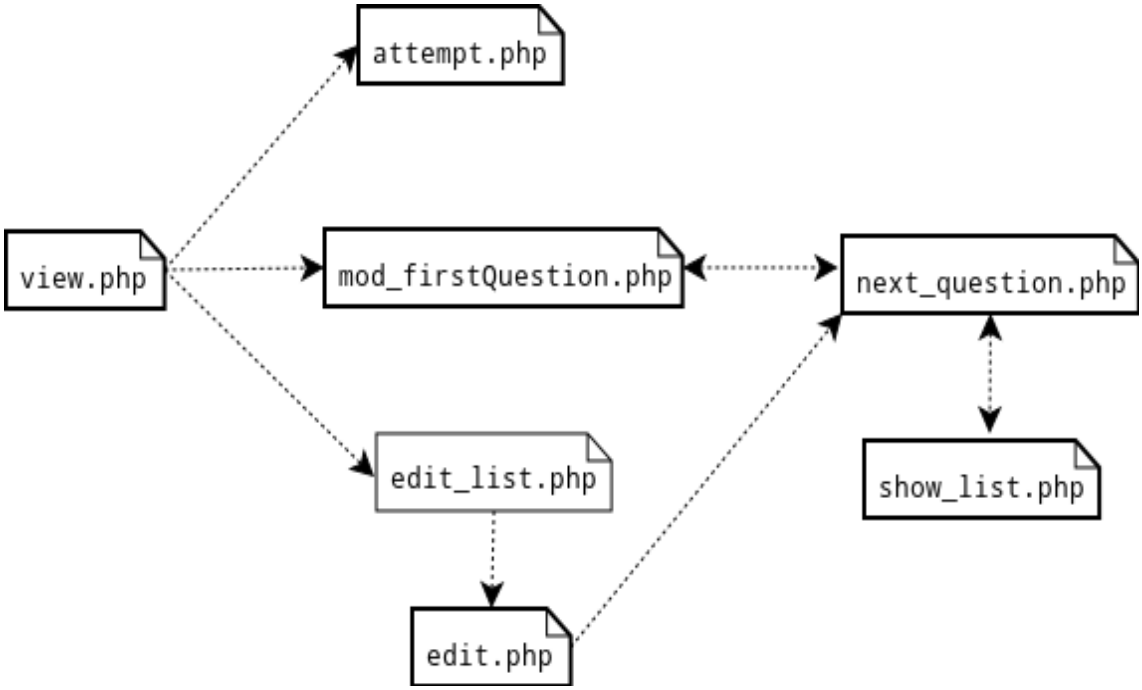


Figure 6.2: Organization of files

6.2.3 Functional Logic

Socratic quiz will be created concept wise. For each concept there will be a series of questions which forms a cycle. For creating cycle we provide the functionality of linking options of a question to the previously added question. Every option of a question will be connected to some question. Correct option of first question will automatically connect to first question of next concept. as shown in fig 6.3.

When user will click on Socratic activity it will redirect to view.php. here it will find out role of user and display the form according to role of user. There are two role teacher and student.

- **Teacher:** If user is a teacher than it will show a form containing two button labeled as "Add question" and "edit question".
- **Student:** If user is a student than it will show a form containing one button labeled as "attempt quiz".

Fig 6.4 shows activity diagram of Socratic module. When teacher will click on 'Add question' button, system will check if there is incomplete cycle of previous concept, If so then teacher will be prompted to complete the cycle first else teacher will be prompted to add new concept in the quiz.

If teacher will click on 'edit question' button, system will show list of question added for this course module(activity). teacher will select question from the list. After selection teacher can edit selected question. when teacher will click on save changes button, system will delete all entry of

File Name	Purpose
view.php	When a user select an already created activity, he is directed to this page. This page displays form created for either student or teacher based on their role capabilities
mod_firstQuestion.php	This page displays the form to enter details of first question of a concept. The same page store the question details submitted by teacher. Then redirect to next_question.php
next_question.php	This page displays the form to enter details of subsequent questions of a concept. form displays the question description and option for which next question is to be entered. form also contains the hyper link, which redirect user to show_list.php. The same page store the question details submitted by teacher. Then redirect to next_question.php
show_list.php	This page displays the form which contains list of previously entered questions for current concept. teacher will choose one question to connect the given option of a question to previous question.
edit_list.php	This page displays the form which prompt teacher to select one question among previously entered question for editing.
attempt.php	This page displays the form which prompt the user to attempt the quiz.
version.php	This file contains the current version number of the module. Whenever we make changes to database we need to update the version number of the module in this file
lib.php	This file contains all the functions which are needed to integrate the module with Moodle and all the other functions which are required to implement the module logic.
db/access.php	This file contain all the capability definition. This capabilities will be used to restrict the access of the users.
db/upgrade.php	Whenever we need to update the database we need to paste proper php code generated by Moodle XMLDB editor in this file.
db/install.xml	In this file all the database tables are defined in xml format

Table 6.1: Purpose of different files in socratic activity module in Moodle

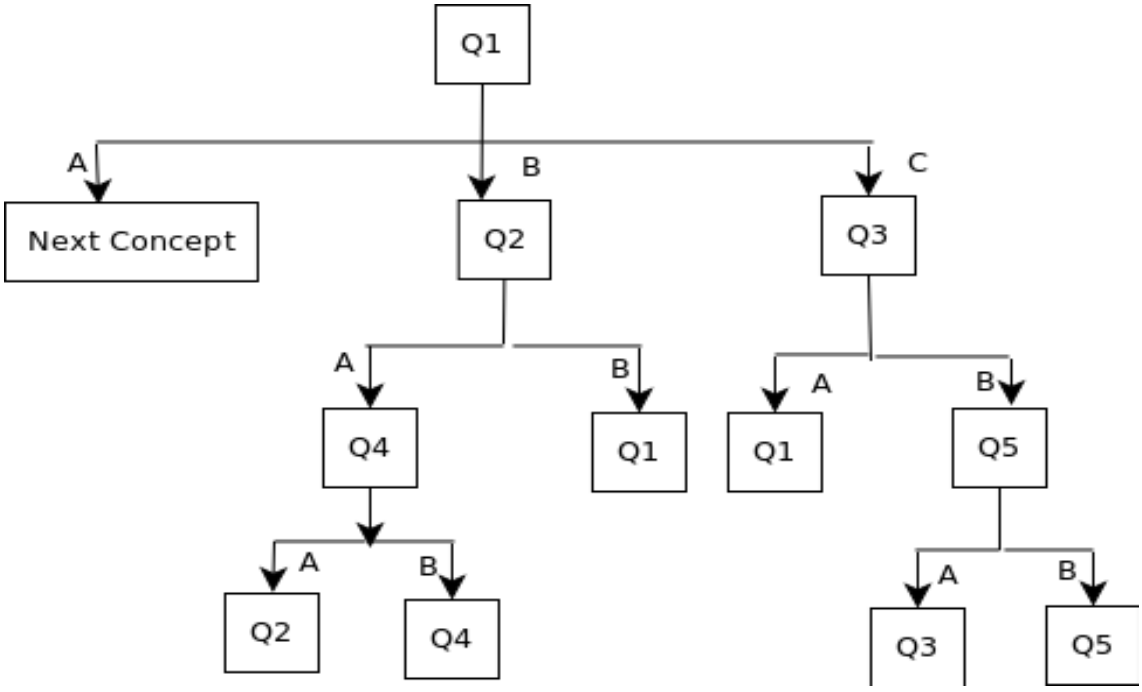


Figure 6.3: Example Cycle for a concept

this question from mdl_socratic_sequencing table. teacher will be prompted either to link options of this edited question to already added question or to connect that option to new question.

When student will click on attempt quiz system will check if there is an incomplete cycle of a concept for this activity. If so then student can't attempt the quiz. A message will be shown to student "Quiz is not completed".

To find out incomplete cycle I have used mdl_socratic_parsing table. This table contains the information about status of question. how many option are not connected to next question. If all option are connected to some question. then system will delete entry for that question from the parsing table. for incompleteness system will search parsing table. If there is an entry for current activity in parsing table that means there is an incomplete cycle in current Socratic activity.

6.2.4 Database Layer

For the Socratic module all the database access are done using the Data manipulation API defined by Moodle. The functions which we have used are

- **get_record_select($table, $select, $strictness=IGNORE_MISSING):** This function is used to get a single database record as an object which match a particular WHERE clause. $table contains table name without prefixes. $select contains conditions which goes in

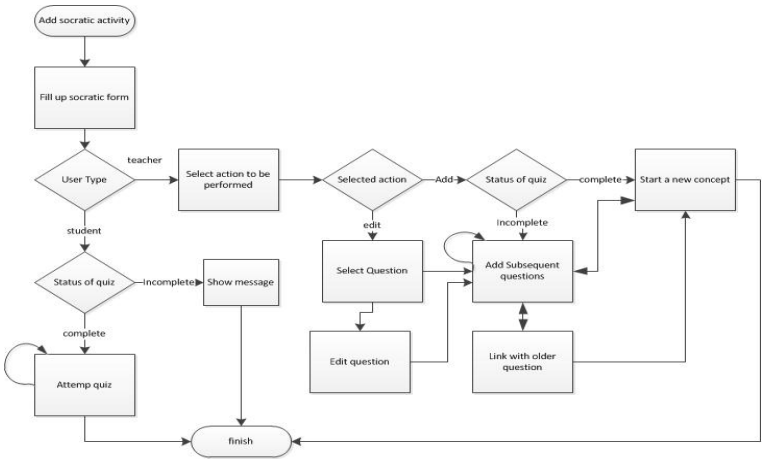


Figure 6.4: Activity diagram of socratic module

WHERE clause. $strictness IGNORE_MISSING means compatible mode, false returned if record not found, debug message if more found. IGNORE_MULTIPLE means return first, ignore multiple records found(not recommended).MUST_EXIST means throw exception if no record or multiple records found

- **get_record_sql($sql, $strictness=IGNORE_MISSING):** This function is used to get a single database record as an object using a SQL statement. where $sql contains a sql statement.
- **get_field_sql($sql, $strictness=IGNORE_MISSING):** This function is used to get a single field value (first field) using a SQL statement.
- **insert_record($table, $dataobject):** This function is used to insert a record into a table. $dataobject is an object containing needed data.
- **delete_records_select($table, $select):** This function is used to delete one or more records from a table which match a particular WHERE clause.
- **update_record($table, $dataobject):**This function is used to update a record in a table. $dataobject is An object with contents equal to fieldname=> fieldvalue. It must have an entry for 'id' to map to the table specified.

All the function mention above are public methods of the $DB global object, so we need to "import" it within our functions using **"global $DB;"**. The $DB global object is an instance of the moodle_database class, which is defined in moodle_database.php.

6.3 Views of System Usage

6.3.1 Teacher View

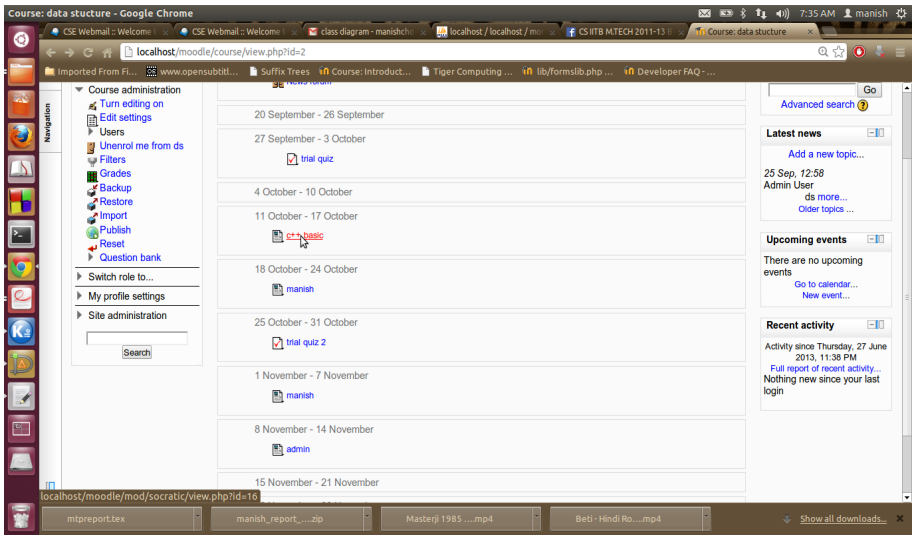


Figure 6.5: Teacher clicks on activity

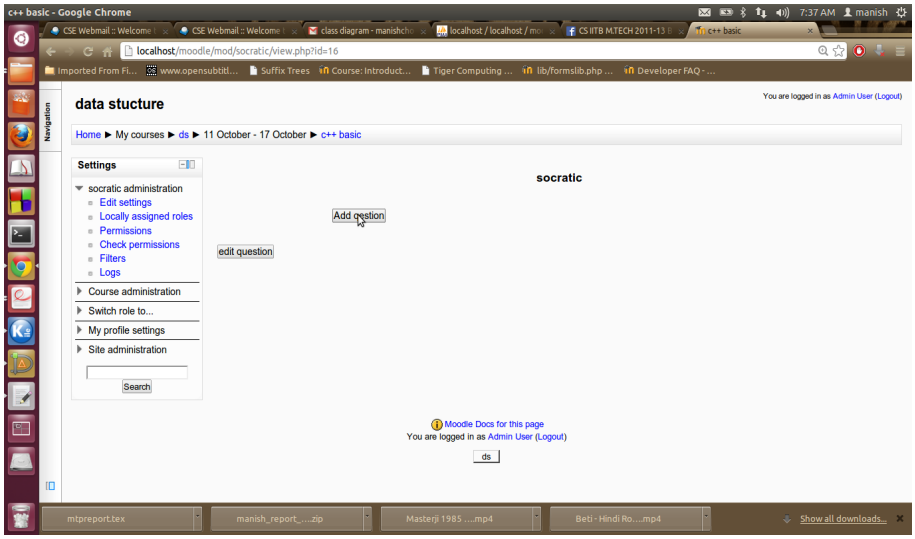


Figure 6.6: Teacher select Add Question

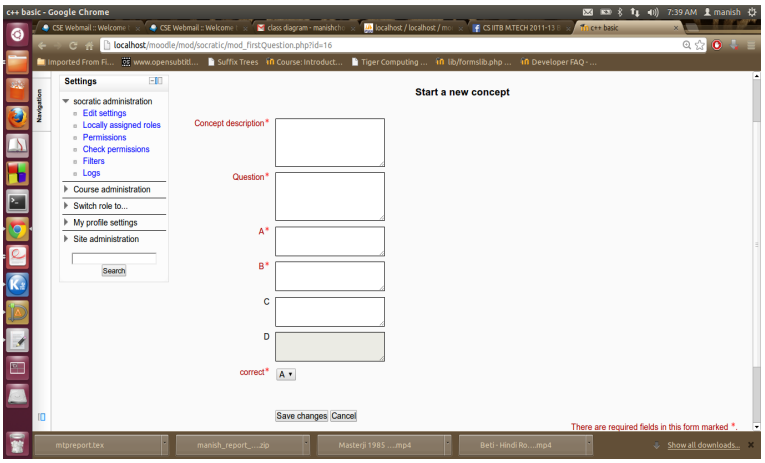


Figure 6.7: Teacher enter question details

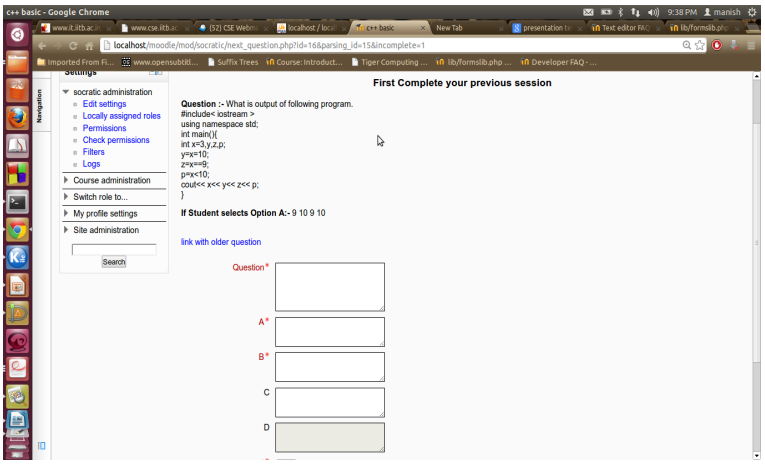


Figure 6.8: Teacher enter connecting question details

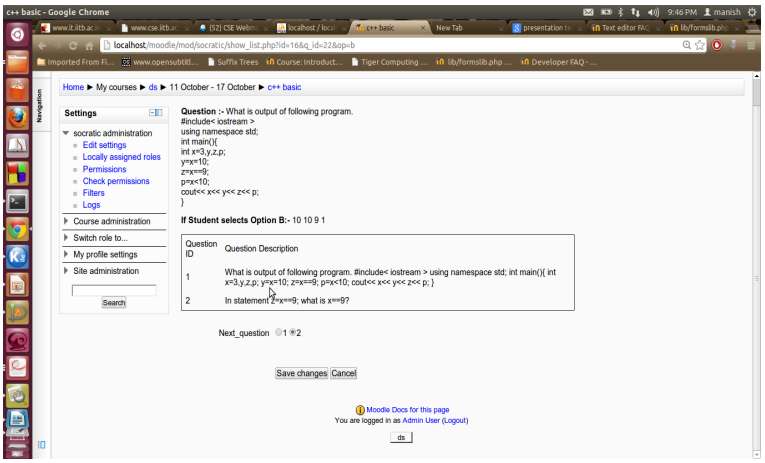


Figure 6.9: Teacher select older question to connect it with current question

6.3.2 Student view

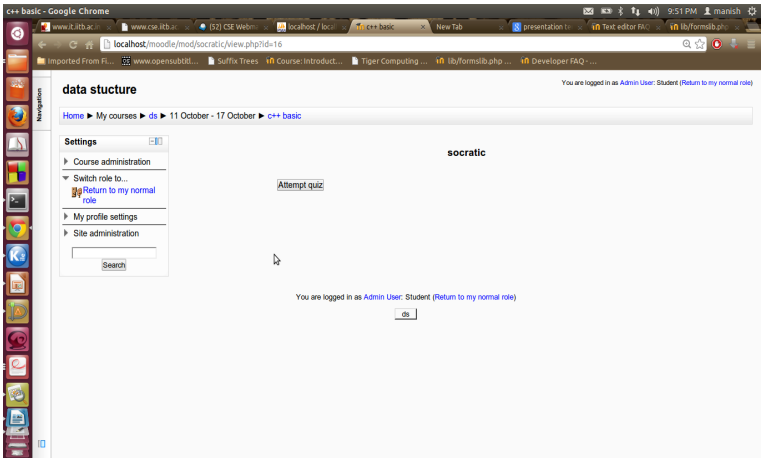


Figure 6.10: Student clicks on Attempt quiz

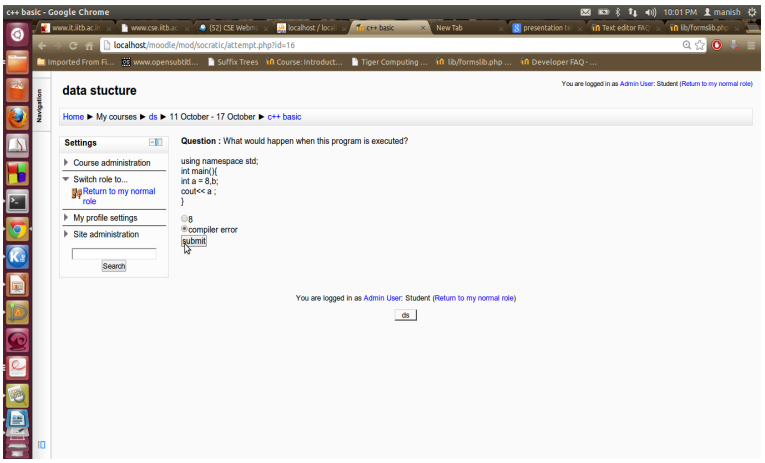


Figure 6.11: Student attempts quiz

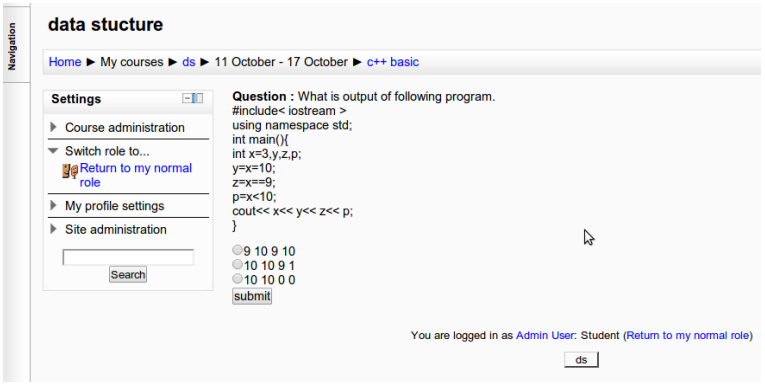


Figure 6.12: If student choose correct option then attempt new concept

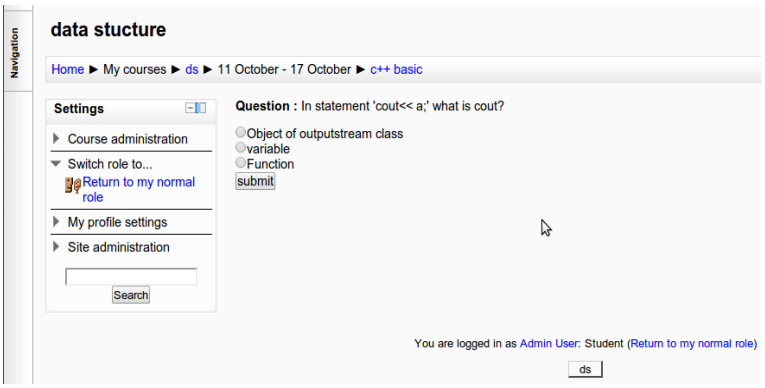


Figure 6.13: If student choose wrong option then attempt related question

Chapter 7

Challenges

7.1 Understanding Moodle

Moodle is a very large software. The installation of Moodle creates almost 10000 files in the local directory. The library files which are used frequently, like moodlelib.php, weblib.php, accesslib.php each having more than 10000 lines of code. Therefore, initially it took some time to figure out how a particular feature is working internally. Other than this, before developing a module I need to know basic databases structure of Moodle, how access is controlled in Moodle, how the XMLDB editor works, about the directory structure of an activity module, how the Moodle-form extension works etc.

7.2 Moodle-form

Forms in moodle are created using forms API. Moodle uses forms API for easy development, validation and processing of forms with advanced accessibility and security. The main challenges in dealing with moodleform is to take input in tabular format, Passing parameter to the form etc. Use of htmleditor to take input from user is difficult instead of this I use normal text area to take the input from user. Another challenge in creating form is to use of renderer which is used for display all the output for component of moodle. This concept is introduced in moodle 2.0 with $OUTPUT class and some associated php files.

7.3 Testing

7.3.1 Testing for teacher

Test Case 1 : Click on Socratic activity user should get different form based on their role (i.e. teacher or student).
Expected Output : Redirect to view.php. Teacher should get form with 'add Question' and 'edit Question' button.
Actual Output : same as expected output.

Comment : This system will show the form to the user based on their role use of has_capability() function.

Test Case 2 : Directly type a Url of a php page of socratic module.
Expected Output : Redirect user to login page.
Actual Output : same as expected output.
Comment : Use of require_login() function.

Test Case 3 : Teacher clicks on 'Add Question' button
Expected Output : If there is an incomplete cycle of questions redirect to next_question.php else redirect teacher to mod_firstQuestion.php.
Actual Output : same as expected output.
Comment : use of logic for incomplete cycle detection.

Test Case 4 : Teacher clicks on 'Edit Question'.
Expected Output : Redirect to edit_list.php which shows only list of questions entered for current activity.
Actual Output : same as expected output .

Test Case 5 : Teacher clicks on the link 'link with older question'.
Expected Output : Redirect to show_list.php which shows only list of questions entered for current concept.
Actual Output : same as expected output.

Test Case 6 : Link all option of last question to previous questions.
Expected Output : Redirect to mod_firstQuestion.php and ask to add new concept in quiz.
Actual Output : same as expected output.

Test Case 7 : After completing first concept add new concept in quiz.

Expected Output : In sequencing table 'next_question' field of correct option of first question of previous concept should contain question id of first question of new concept.

Actual Output : same as expected output.

Comment : use of logic for incomplete cycle detection.

Test Case 8 : add a c++ program as a first question of new concept

```
#include<iostream>
using namespace std;
void main()
int var = 5;
cout<<var;
}
```

option A : 5
option B : 6

Expected Output : Page containing text
Question:- #include<iostream>
using namespace std;
void main(){
int var = 5;
cout<<var;

}

If student selects Option B:- 6

Actual Output : Page cotaining text
Question :- #include
using namespace std;
void main(){
int var = 5;
cout< }

If Student selects Option B:- 6

Reason : After Retriving data from database system will add data to .php file where "<" is treated as opening php tag that's why in above program it does not print anything after "<" symbol until it gets matching closing symbol ">".

Prevention : Teacher can prevent this problem by Using space character after "<" symbol for example #include< iostream > and cout<< var;

Solution : I have use text area for taking input from user. Moodle also provides htmleditor which is used for taking rich contents as a input from user. One can solve this problem by replacing text area with htmleditor.

7.3.2 Testing for student

Test Case 8 : when user click on socratic activity user should get form based on their role (i.e. teacher or student).

Expected Output : Redirect to view.php student should get form with 'attempt Question' button.

Actual Output : same as expected output.

Comment : This system will show the form to the user based on their role.

Test Case 9 : student clicks on 'Attempt Question' button.

Expected Output : If there is an incomplete cycle of questions it shows a message saying "Quiz is not completed! " else shows first question of quiz for attempting.

Actual Output : same as expected output.

Test Case 10 : student attempts last question correctly.

Expected Output : Shows a message "you have completed quiz"

Actual Output : same as expected output.

Chapter 8

Conclusion

I have developed an activity module as my project. Socratic module is a complete tool to teach a subject using Socratic method. It has all the features described in previous chapter. So it can be used in distance learning environment. This module is limited to teach any subject using multiple choice questions (MCQs) only.

To build this tool as a plug-in for learning management system, moodle has some advantages, like Data manipulation api to interact with api. moodle also has a very good features to handle databases, using XMLDB editor. Initially it took too plenty of time, in understanding how moodle works and if we want to add functionality how to do it.

8.1 Future Work

- This system is limited to MCQs(objective) only, it can be extend to subjective type quiz.
- One can add an assessment method to judge student's progress
- GUI may be improved of this plugin using renderer classes of moodle

Bibliography

[1] http://www.Socraticmethod.net.

[2] vikash kumar "Development of Intelligent Tutoring System Framework: Using Socratic strategy" Mtech. Thesis IIT Bombay june 2012

[3] K. E. Chang, M. L. Lin, and S. W. Chen, Application of socratic dialogue on corrective learning of subtraction, Comput. Educ., vol. 31, no.1, pp. 5568, Aug. 1998.

[4] http://docs.moodle.org.

[5] http://docs.moodle.org/23/en/Pedagogy.

[6] Chandra pal "Development of Intelligent Tutoring System Framework: Using Scaffolding Teaching Strategy" Mtech. Thesis IIT Bombay june 2012. http://docs.moodle.org/dev/Data_manipulation_API

[7] L.S.Vygotsky:MindinSociety:DevelopmentofHigherPsychologicalProcesses, p.86

[8] Raja Shekhar "Development of Intelligent Tutoring System Framework: Using Guided Discovery learning" Mtech. Thesis IIT Bombay june 2012

[9] Monica Trella, ricardo Conejo and Eduardo Guzman, A web based Socratic tutor for trees recognition