# Development of Intelligent Tutoring System Framework: Using Socratic Strategy

By Vikash kumar

10305059

Mtech.2(CSE)

# Outline

- Introduction
- Literature Survey
- Problem And Our Solution
- Demo
- Our ITS  Architecture
- Time Sequence Diagram
- Socratic  Teaching Strategy
- Integration Of  All Strategies
- Future Work
- Limitation

# Introduction

- **What is ITS?**
  - A system that provides response and interactive teaching facilities to learners,
  - tracks their progress and past performance,
  - sequences the curriculum and helps the learners to improve without human instructor intervention.

- **Why ITS?**
  - Research Need[1]: Study of cognitive behavior
  - Practical Need[1]: One to one teaching environment.
    - Why One-to-one teaching environment?

# Literature Survey

# Some ITSs

- Wayang Outpost
- SQL-Tutor
- Auto-Tutor
- Thermo-Tutor
- Smart Tutor

# Wayang Outpost[3]

- It is a web-based intelligent tutoring system designed for helping student in SAT exam.

**GOAL:**

- Improve student performance on SAT-Math geometry problems
- Decrease the gender gap
- Determine the most effective way, using either visual or analytic help, to tutor students of various abilities

# Wayang Outpost(cont.)

- If the student answers incorrectly, or requests help, then system provides step-by-step instruction and guidance in the form of Flash animations with audio.
- It had focused on both mental rotation abilities and memory retrieval speed.

# Architecture

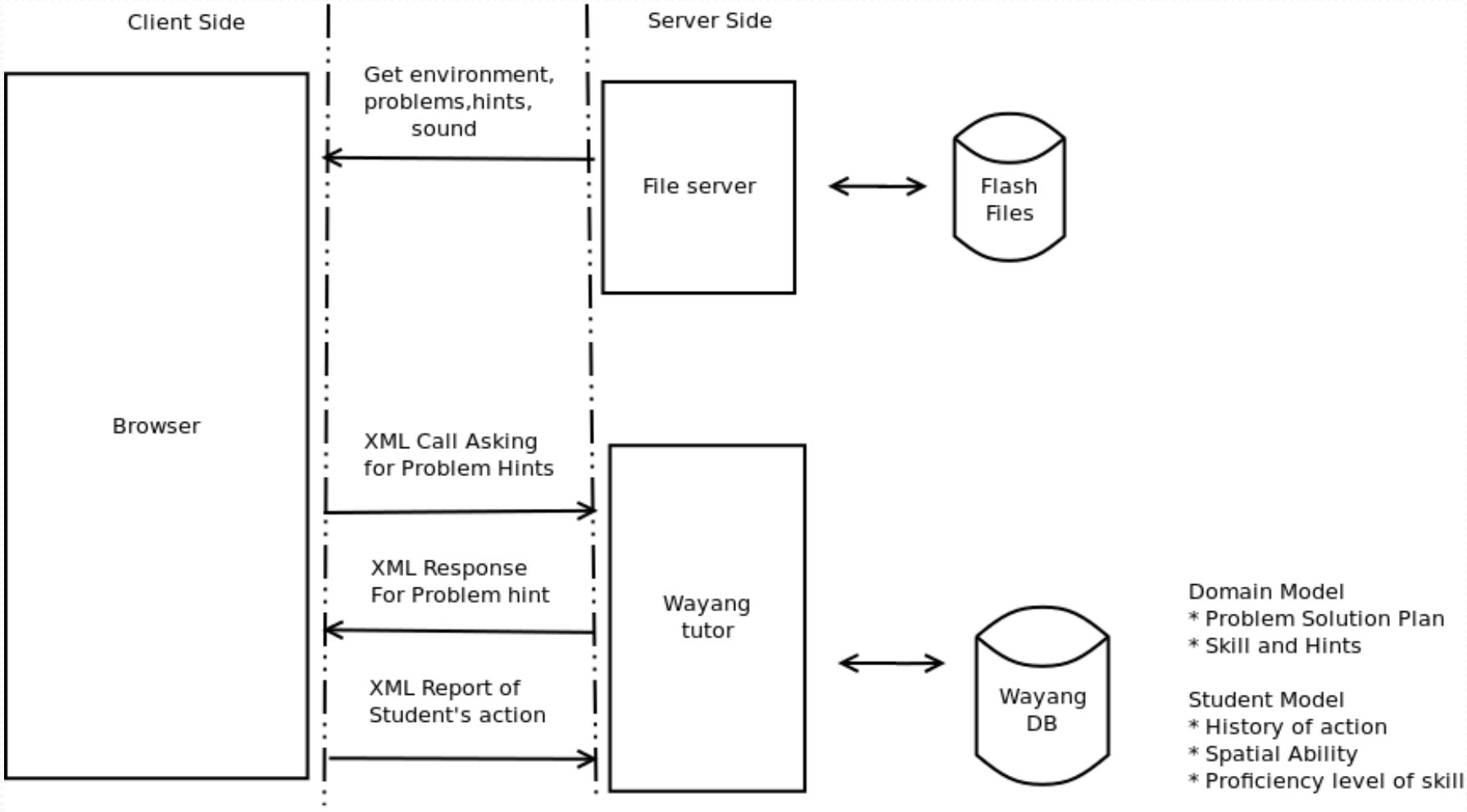



Fig1: Wayang Outpost Architecture[3]

1. In Wayang Outpost, there is a centralized database.

2. Every Data like ***problem, its solution, hint, student work, his performance*** are stored in a centralized database.

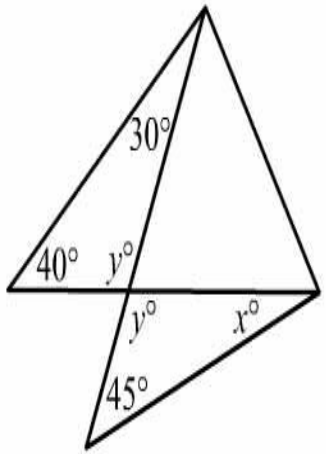
3. From this data, the system makes inferences the student's performance and select problems at the appropriate level of challenge, and

4. Also chooses hints that will be helpful for the student.

# Help logic

- One hint provides a **computational** and numeric approach
- The second provides **spatial transformations** and visual estimations

# Two techniques
## for selected problems




In the figure above, what is the value of $x$?

(A) 65
(B) 45
(C) 40
(D) 30
(E) 25

$x + 45 + y = 180$

$40 + 30 + y = 180$

$70 + y = 180$

$y = 180 - 70$

$y = 110$

$x + 45 + y = 180$

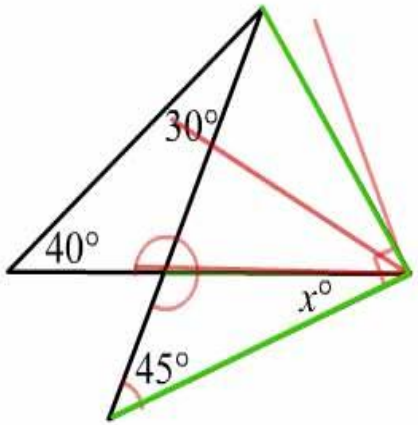
$x + 45 + 110 = 180$

$x + 155 = 180$

$x = 180 - 155$

$x = 25$

Choose (E)




In the figure above, what is the value of $x$?

(A) 65
(B) 45
(C) 40
(D) 30
(E) 25

How are the rest of the angles related to $x$°?

x is about a third of the green angle

The green angle is a bit less than 90 degrees

x is a bit less than 90/3

x is a bit less than 30

Choose (E) for an answer

Fig2: example1[3]

# Some points

- Wayang Outpost doesn't trace each step of the students solution.
- It uses the concept of data-centric approach with Bayesian Networks to categorized 3 types of student
  - who already knows a skill ,
  - is learning a skill and
  - is not learning skill?
- The tutor observes the hints requested by the student to reach the solution

- **Problem with most of ITS**
  - They are designed only for one strategy.
  - All these ITSs are made to teach only one specific subject.
- **Our Solution**
  - Make ITS framework generic (it means independent from subject domain and can support more than one strategy).
  - We have built a system which can support at least 3 different strategies together.

# Solution(Cont.)

- **Why we have tried more than one strategy ?**
  - Student Need
  - Strategy Specific subject

- **Example**
  - For Introduction part of each subtopic ***scaffolding strategy*** is good.
  - For Practical point of view ***Guided Discovery*** is better
  - For Conceptual study ***Socratic Strategy*** is Best.

# DEMO

# Our ITS Architecture

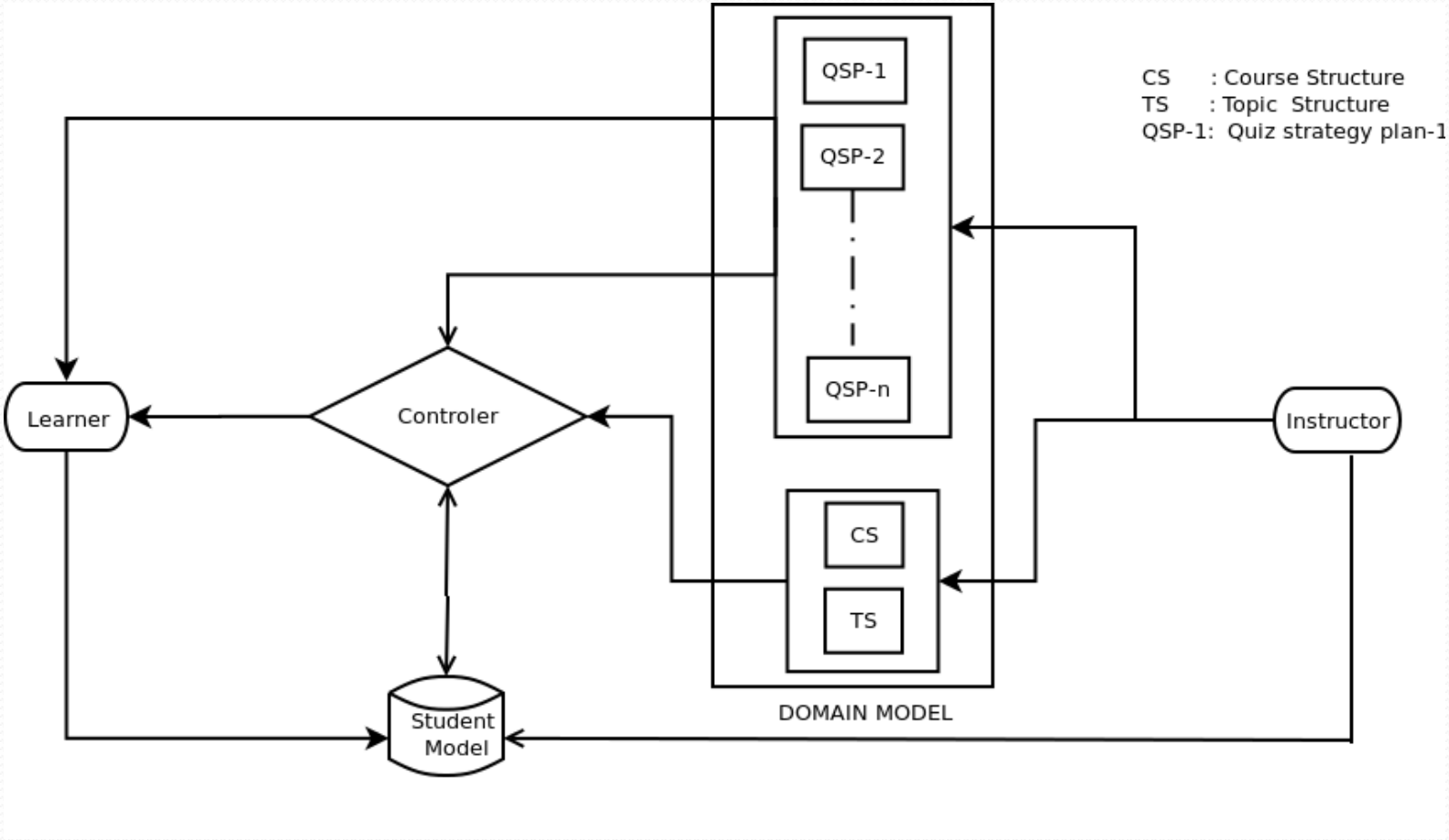



**Fig 3 : Top Level Architecture**

# Components Of ITS

- Domain Model
- Student Model
- Teaching Model
- User Interface Model

- **User Interface Model**
  - communicating component of the ITS which controls interaction between the user and the system.
  - It works bidirectional.

# Domain Model

- It represents the content knowledge that the student is acquiring. i.e it mainly deals with the what-to-teach part of an ITS.

- It has two parts
  - first part contains course structure (CS) topic structure (TS) and Subtopic Structure.
  - other part contains Question_DB according to Each strategy.

# Student model

- It refers to the dynamic representation of the emerging knowledge and skill of the student.
- It contains information of
  - Student's profile;
  - Student's Response table: It contains the information about student performance question wise.
  - Student's performance status: It contains the final student's performance topic wise.
- Using only one criteria **program solving ability.**

# Teaching Model

- This model decides all knowledge base of ITS.
- It is the nucleus of the ITS which communicates with the other modules and
- plans the teaching strategies to be taken for individual students.

- Here instructor has to perform:

  - Design Course Structure
  - Design Topic Structure
  - Design Rule structure.

# Time Sequence Diagram For Instructor

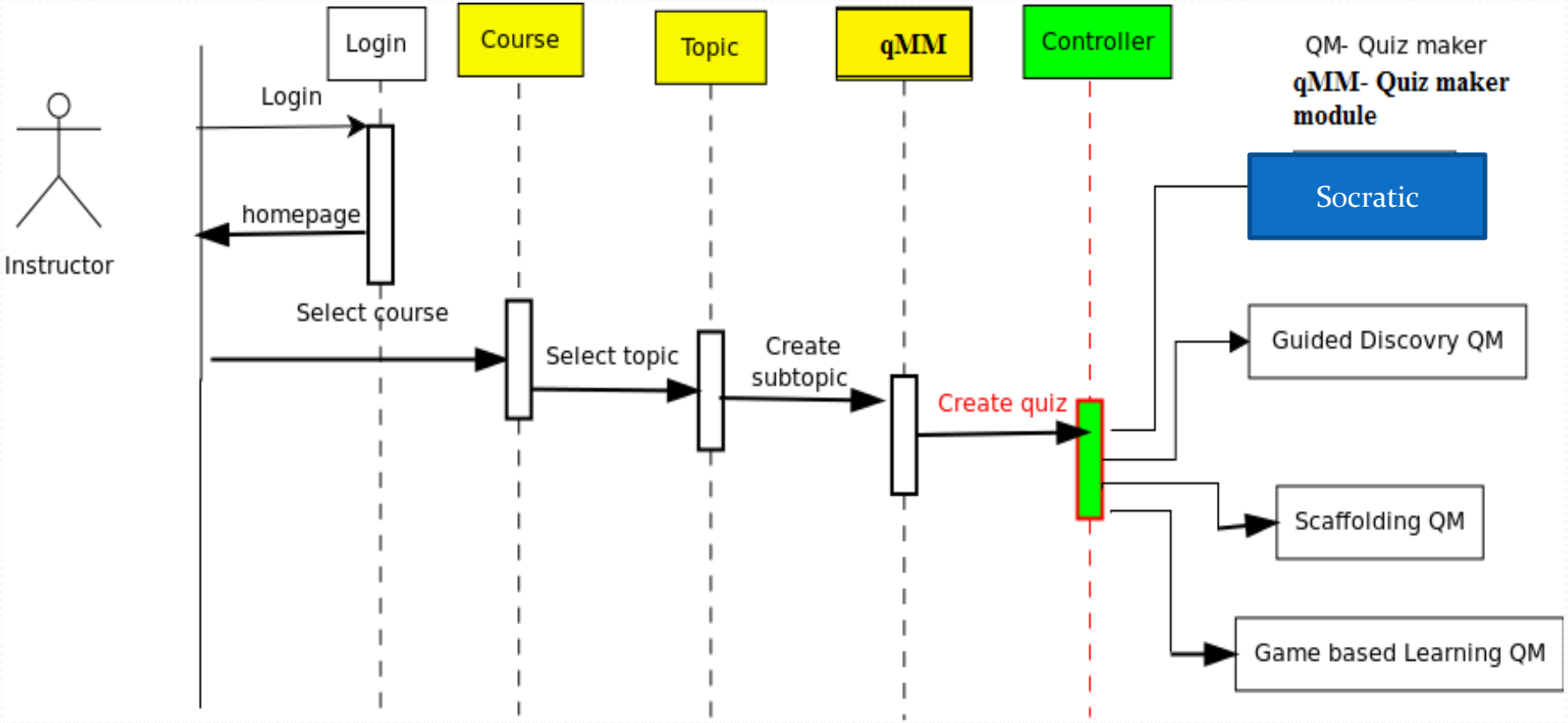



**Fig 4**

# Time Sequence Diagram For learner

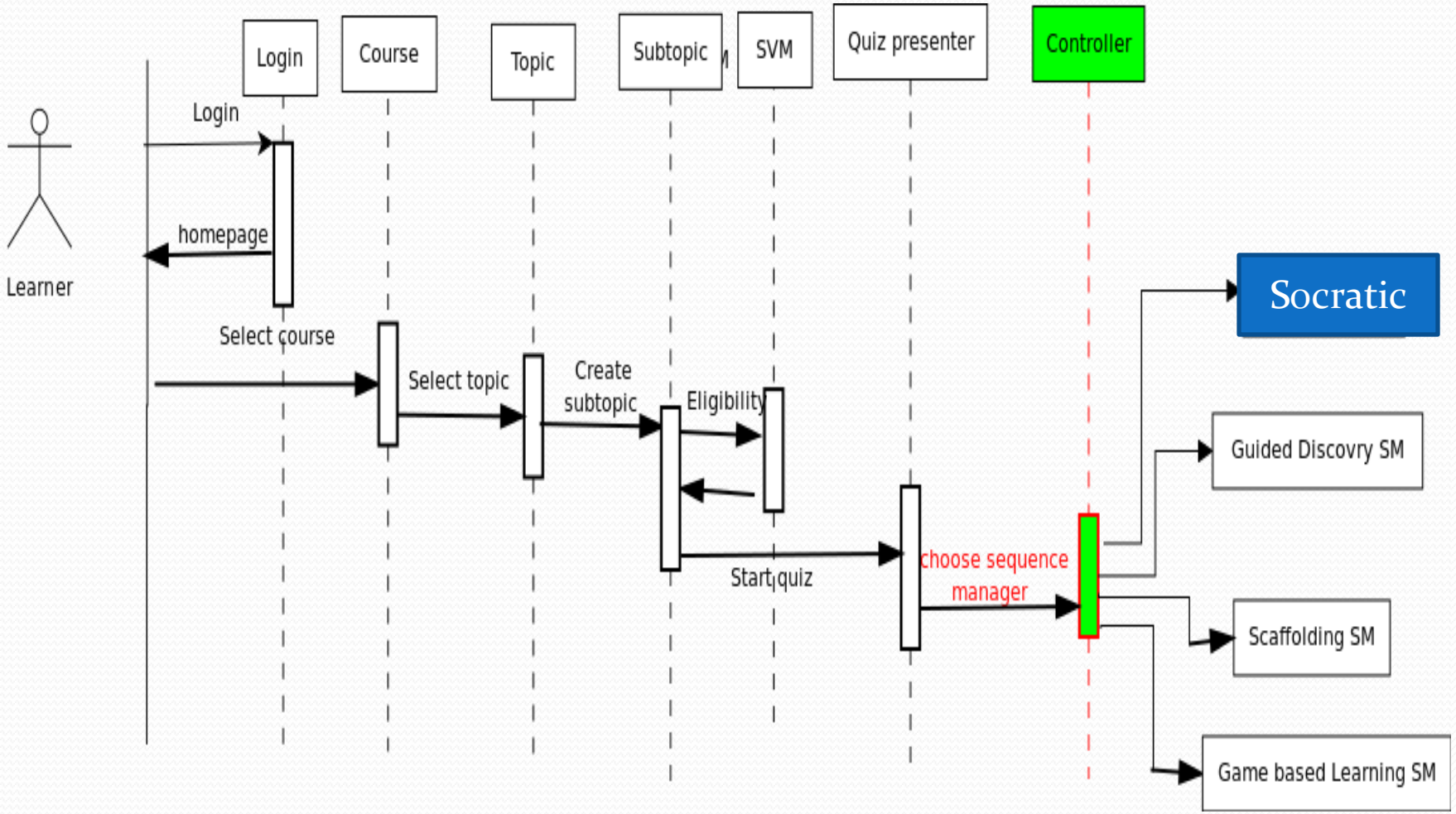



**Fig 5**

# Work of Controller

- Center of the system
- Follow all rules applied by the instructor
  - Strategy Switch for quiz creation
  - Present question According to selected Strategy.
  - ***Select higher priority strategy if more than one strategy is available for particular subtopic.***
  - ***Change Strategy when student doesn't perform well***
  - Update student performance or response table

# Socratic Teaching Strategy

# Introduction

- Socratic Questioning: It is an approach in which teaching-learning is performed in the form of question and answer.

- Follow  Systematic engagement, Bottom Up Approach

- **Steps in Socratic Questioning**
  1. Ask question
  2. Wait for response
  3. Take response
  4. Ask next question based on response

# How I have  implemented?

- With the help of MCQs.
- Each question has 4 multiple answers next question will depend on the option chosen by student.
- Instructor will create relationship  between option and  next question
- Our System(ITS) will follow.

# Example

- Que: int k=5/2; the value of a will be
  - (a) 2.5
  - (b) 0
  - (c) 2
  - (d) error
- Que: If data type of a variable is integer than it return
  - (a) Always zero
  - (b) Depend upon assigned value
  - (c) Always integer
  - (d) Can't Say
- Que:If float k=2.5; the value of k will be
  - (a) 2.5
  - (b) 2
  - (c) 0
  - (d) we cannot store oat value in integer type variable.

| CourseId | TopicId | SubopicId | QueId | Option | NextQueId |
|---|---|---|---|---|---|
| CS101 | T1 | St1 | 23 | A | 23 |
| CS101 | T1 | St1 | 23 | B | 24 |
| CS101 | T1 | St1 | 23 | C | 25 |
| CS101 | T1 | St1 | 23 | D | 27 |
| CS101 | T1 | St1 | 24 | A | 24 |
| CS101 | T1 | St1 | 24 | B | 25 |
| CS101 | T1 | St1 | 24 | C | 24 |
| CS101 | T1 | St1 | 24 | D | 24 |
| CS101 | T1 | St1 | 25 | A | 25 |
| CS101 | T1 | St1 | 25 | B | 26 |
| CS101 | T1 | St1 | 25 | C | 27 |

**Question Sequencing Table**

# Time Sequence Diagram for Instructor In Socratic Strategy

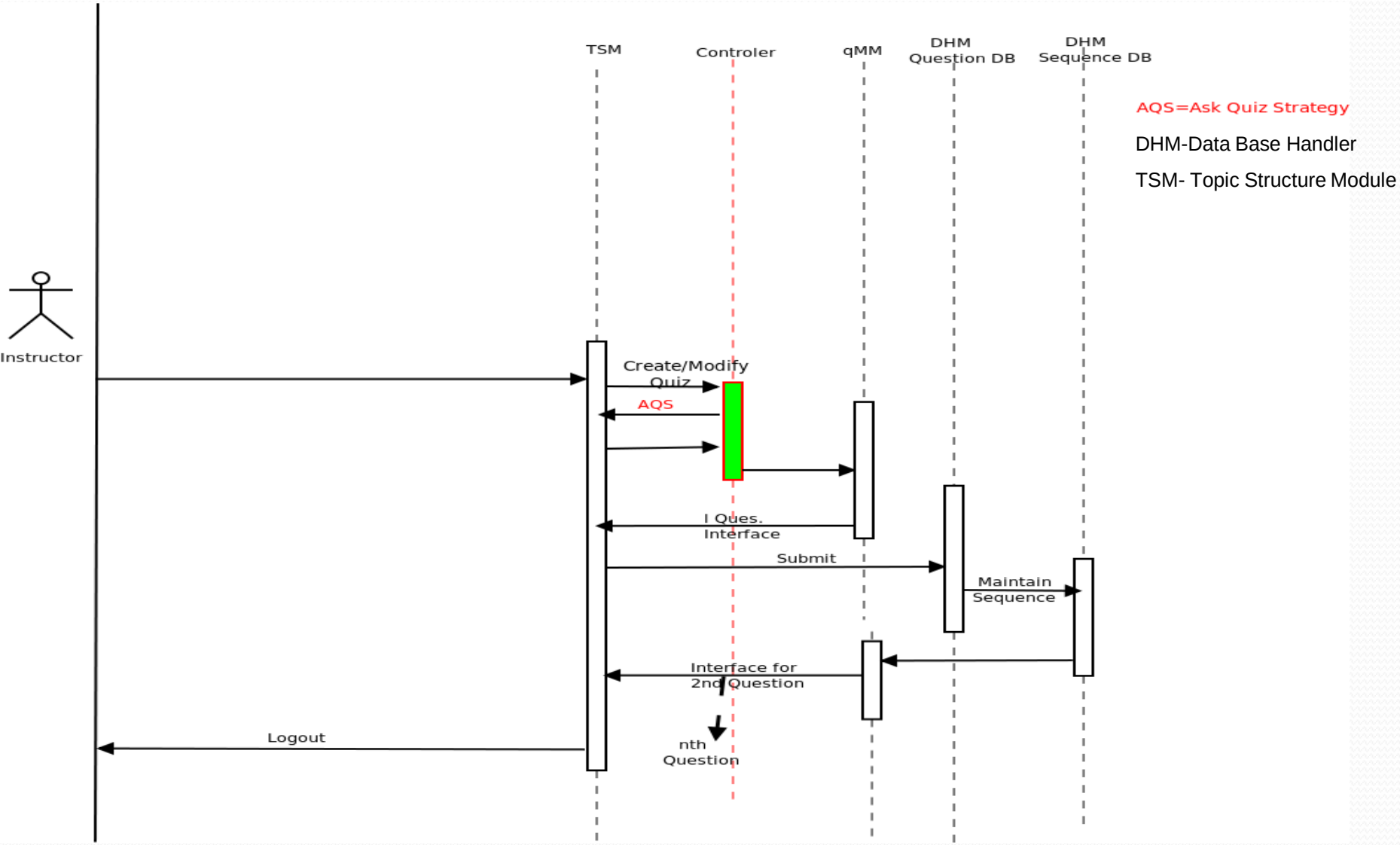



**Fig 6:Extended Version of Socratic Strategy**

# Integration Of All Strategies

- **Approach**
  - Find Common Modules
  - Find Non Common Modules
  - Find Non Common Tables
  - **Developed some modules which helps in integration**
    - Strategy Sequencing Module
    - Strategy Selector Module
    - Strategy  Changer Module

# Common modules to all 3 strategies

- GUI module(GUIM)
- Input Validation module(IVM)
- Student module
- Authentication module(AM)
- Course Validation module (optional for my strategy)
- Quiz Maintainer Module
- Evaluation module
- Database handling module
- Log module
- Feedback module
- Result generator module
- Logic module
- Course Module
- Topic Structure Module
- Quiz maker Module

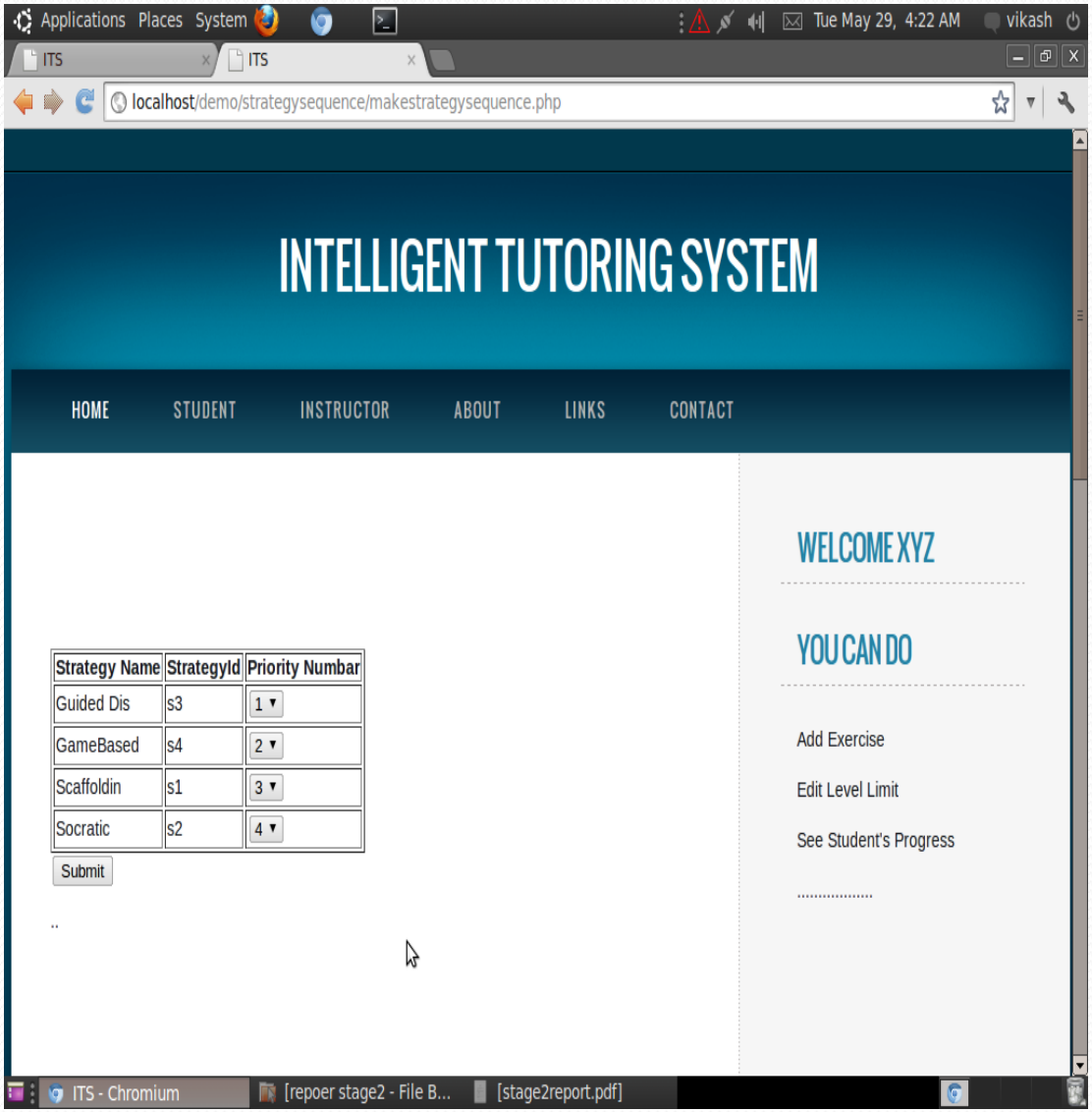
# Non Common Module

- **Socratic Strategy**
  - Sequencing Module (use of parsed_table)
    - BFS implementation
  - Topic and Subtopic Validation Module
    - DFS implementation
  - Student Previous Progress
- Scaffolding Strategy
  - Hint Module
  - Logic Module
- Guided Discovery Strategy
  - Quiz maker module
  - Feedback module

# Module used for integration

**[A] Strategy Sequencing Module**

- I f there is single strategy No need.
- If more than one strategy Then maintain Priority



# Integration(cont.)

**[B] Strategy Selector Module**

First check subtopic selected by student is associated with single strategy or not.

1. if single strategy than redirect to that strategy
2. If more than one strategy than goes to *strategy sequencing table* and checks the priority order of strategy and choose highest priority order strategy
3. give this strategy to Strategy Changer Module.

# Integration(cont.)

**[C] Strategy Changer Module(SCM)**

1. take the strategy from SSM and check the cutoff
2. If student is participating with this strategy **1st time** then redirect to that strategy
3. if strategy is above cutoff then redirect other wise pick another strategy
4. after all strategy fails then check student best performance and redirect to that strategy.

# Future work

- Collecting material for teaching
- Subjective Questions
- Time  Response  Theory
- More teaching strategies
- Collaborative learning

# Limitation

- At this time Our ITS is limited to teach any subject with the help of multiple choice questions (MCQs) only.
- It is limited one-to-one learning. We haven't work for collaborative learning i.e. discussion forum and chat forum etc.
- It is very hard to implement Socratic questioning for subjective problem. (content creation is very tough)

# References


1. Bloom, B.S. The 2 Sigma Problem: The search for Methods of Group Instruction as Effective as One-to-One Tutoring, Educational Researcher, vol. 13, no. 6, pp. 416, 1984.
2. D. Sleeman, and J. S. Brown, Introduction: Intelligent Tutoring Systems" Ed. Academic Press, New York, 1982, pp. 1-11.
3. Ivon Arroyo, Rena Wales, Carole R. Beal, Beverly P. Woolf, Tutoring for SAT-Math with Wayang Outpost University of Massachusetts, Amherst
4. Hyancith S.Nwana. Intelligent Tutoring Systems Department of computer Science University of Liver pool, Liverpool L69,3BX,UK
5. Cotton, K. (2001). Classroom questioning. School Improvement Research Series(SIRS). http://www.nwrel. org/scpd/sirs/3/cu5.html (25.10.2005).
6. PETE BOGHOSSIAN How Socratic Pedagogy Works Portland State University College of Education
7. Albert T. Corbett, Kenneth R. Koedinger, and John R. Anderson. Handbook of Human-Computer Interaction,chapter 37. USA, 1997.
8. Wikipedia.http://en.wikipedia.org/wiki/Intelligent_tutoring_system .

Thank you!!!