International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 2 | ISSN : 2456-3307

Intelligent Adaptive Learning System

Avishkar Sawant, Rahul Waghmare, Yogita Kotwal, Niket Gawde, Prof. Prachi Kshirsagar

Department of Information Technology, PVPPCOE, Mumbai, Maharashtra, India

ABSTRACT

The main purpose of the Intelligent Adaptive Learning System is to conduct Adaptive Learning Examination using Multiple Choice Questions. The Adaptive Learning System will help students in learning and grasping in an efficient way. The Results will demonstrate how the Students can adapt to changing difficulty level of the MCQs.

Keywords : Arduino, Wi-Fi (ESP 8266), Load cell, Database System

I. INTRODUCTION

The Self Learning System will help students in learning and grasping in an efficient way. Also, this system will help students in clearing their doubts and understanding the topics in a very clear manner, the way in which teacher wants to explain. The objective of the Self Learning System is to provide better information for the users of this system for better results for their maintenance in online test schedule details.

II. METHODS AND MATERIAL

The existing system is a system in which the questions do not change according to the Caliber or Problem Solving Capability of the Student. The difficulty level of the questions remains the same so the students do not find it challenging. In existing system two students giving the exam may generally get the same questions.

The existing system is a manual one in which the student is listening to the teacher. This leads to lack of interest and grasping capacity of the student. The minimum number of doubts and confusions are cleared.

The following subsection contains a brief description and features for existing online examination systems:

SIETTE

Guzman and Conejo (2005) proposed an online examination system called System of Intelligent Evaluation using Tests for Tele-education. SIETTE is a web-based environment to generate and construct multiple tests. SIETTE supports secure login and portability features. On the other hand, the other features including different question sets and adaptive questions as per students calibre are missing.

Sheshadri et. al. (2011) proposed a web-based Online Non-choice-based Examination System. WONES is an effective solution for massive education evaluation; it employs special authentication protocols to ensure transactions between the examination server and the students.Similar to the other systems it doesn't have different question sets and it's not adaptive as per students calibre.

IEMS

Vasupongayya et. al. (2010) proposed an interactive Examination Management System. iEMS is a web-based application test management system, with ease of uses, rich features, flexibility. The iEMS supports secure login, portability, random questions distribution, and random choices distribution features. On the other hand no different question sets are available. The questions don't change difficulty according to the students' calibre.

III. RESULTS AND DISCUSSION

Self Learning System has a self adaptive feature which will be helpful in self learning. Various tests will be available for the students without any limit of the subject. Each question will be having the timer set; if the student is able to give the answer within 15 seconds of time, the level of the next question will upgrade to a difficult one. If the student takes time longer than 15 seconds, the next question will be comparatively an easy one. So the questions will be depending on the caliber of the student.

Administrator can enroll any new subject and can create a topic-wise or chapter-wise mcq type question. So it will be easy for the student to know the topic well and it can also replace the assignment system, as the main aim of assignments is to let student understand about the topic in depth.

Students enrolled in the Self Learning System may access through the electronic details they provide and perform various functions with the system in order to participate in the on-line innovative learning test sessions. Students can receive an on-line test, having multimedia content, for the course, and they can select appropriate answers in the test. And after Completion of their test, they are provided with the right answers entered by the teacher. The students can sit at individual terminals and login to give the answers for test questions in the given duration.

The Self Learning System will perform correction, display the result immediately and also store it in database. This system provides the administrator with a facility to add new exams. The Self Learning System provides the teacher to add questions to the exam, modify questions in a particular test. This system takes care of authentication of the administrator, teacher as well as the student.

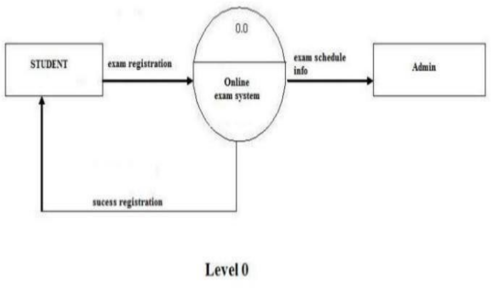
On-line test contents provide to focus on creating effective assessment questions and focusing on test's feedback delivery to students. In the paper, we present techniques that are pertinent to the elements of assessment process: answers submission, computerized grading, and feedback after submission. As the modern organizations are automated and computers are working as per the instructions, it becomes essential for the coordination of human beings, commodity and computers in a modern organization.

The administrators, teachers, students who are attending online test can communicate with the system through this project, thus facilitating effective implementation and monitoring of various activities of online learning like conducting test as per scheduled basis and delivering result to that particular student. Also, the details of students who attempted online test are maintained with the administrator.

Our web portal provides features like computerized adaptive testing as per the student's caliber, generation of different question sets based on difficulty level. Also, no repetitive questions will appear for different examinee. The most important profit of this intelligent adaptive learning system will be that no student can copy or cheat easily. The system will provide a secure authentication mechanism for the administrator, teachers and students. In the paper, we present techniques that are pertinent to the elements of assessment process: Answers submission, Computerized grading, and feedback after submission. Each question will be having a critical time limit and maximum time. If the student is able to give the answer within a critical time limit then, the difficulty level of the next question will upgrade to a difficult one. If the student takes longer than critical time limit, his next question will be of same difficulty level. If the student selects a wrong answer or is unable to answer before maximum time, then the next question will be of easier difficulty level. To avoid repetitions of questions, the framework will provide a number of different SET options. In each SET, there will be a subset. Each will be having a different difficulty level number. The Administrator can create new courses or edit existing course list. The Teachers can create topic-wise or chapter-wise MCQs or Edit Existing MCQs and decide the Difficulty level of the Questions. The Adaptive Learning System will perform correction, display the result immediately and store the results in a database.

3.1 LEVEL 0 DFD:

The level-0 DFD shows the basic functionality of the system in IALS i.e. exam registration and schedule information. The input and output of the system is also shown



Figire 3.1 DFD LEVEL 0

3. 2 LEVEL 1 DFD:

The level-1 DFD in the figure below showing the role of student and administration in detail

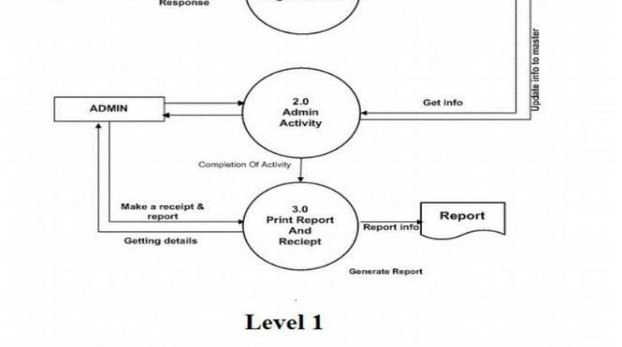


Figure 3.2 DFD LEVEL 1

3.3 ALGORITHMS FOR IALS:

We have implemented the following algorithms:
Best First Search Algorithm
A * Algorithm

3.3.1 BEST FIRST SEARCH ALGORITHM:

Best-first search is a search algorithm which explores a graph by expanding the most promising node chosen according to a specified rule.

An algorithm implementing best-first search follows.

```
OPEN = [initial state]

while OPEN is not empty or until a goal is found
do

 1. Remove the best node from OPEN, call it n.
 2. If n is the goal state, backtrace path to n (through
    recorded parents) and return path.
 3. Create n's successors.
 4. Evaluate each successor, add it to OPEN, and record
    its parent.
done
```

This version of the algorithm is not complete, i.e. it does not always find a possible path between two nodes, even if there is one. For example, it gets stuck in a loop if it arrives at a dead end,that is a node with the only successor being its parent. It would then go back to its parent, add the dead-end successor to the OPEN list again, and so on.

The following version extends the algorithm to use an additional CLOSED list, containing all nodes that have been evaluated and will not be looked at again. As this will avoid any node being evaluated twice, it is not subject to infinite loops.

```
OPEN = [initial state]
CLOSED = []

while OPEN is not empty
do

1.  Remove the best node from OPEN, call it n, add it
    to CLOSED.
```

2. If n is the goal state, backtrace path to n (through recorded parents) and return path.

3. Create n's successors.
4. For each successor do:

a. If it is not in CLOSED and it is not in OPEN: evaluate it, add it to OPEN, and record its parent.

b. Otherwise, if this new path is better than previous one, change its recorded parent.

i. If it is not in OPEN add it to OPEN.

ii. Otherwise, adjust its priority in OPEN using this new evaluation.

done

3.4. A * ALGORITHM:

In computer science, A* is a computer algorithm that is widely used in path finding and graph traversal, the process of plotting an efficiently traversable path between multiple points, called nodes.

A* is an informed search algorithm, or a best-first search, meaning that it solves problems by searching among all possible paths to the solution (goal) for the one that incurs the smallest cost (least distance travelled, shortest time, etc.), and among these paths it first considers the ones that appear to lead most quickly to the solution. It is formulated in terms of weighted graphs: starting from a specific node of a graph, it constructs a tree of paths starting from that node, expanding paths one step at a time, until one of its paths ends at the predetermined goal node.

At each iteration of its main loop, A* needs to determine which of its partial paths to expand into one or more longer paths. It does so based on an estimate of the cost (total weight) still to go to the goal node. Specifically, A* selects the path that minimizes

$f(n) = g(n) + h(n)$

where n is the last node on the path, g(n) is the cost of the path from the start node to n, and h(n) is a heuristic that estimates the cost of the cheapest path from n to the goal. The heuristic is problem-specific. For the algorithm to find the actual shortest path, the heuristic function must be admissible, meaning that it never overestimates the actual cost to get to the nearest goal node.

The following pseudocode describes the algorithm:

```
functionA*(start,goal)

// The set of nodes already evaluated.
closedSet:={}

// The set of currently discovered nodes still to be evaluated.

// Initially, only the start node is known.
openSet:={start}

// For each node, which node it can most efficiently be reached from.

// If a node can be reached from many nodes, came From will eventually contain the

// most efficient previous step.
cameFrom:=theemptymap

// For each node, the cost of getting from the start node to that node.

gScore:=mapwithdefaultvalueofInfinity

// The cost of going from start to start is zero.
gScore[start]:=0

// For each node, the total cost of getting from the start node to the goal

// by passing by that node. That value is partly known, partly heuristic. fScore:=mapwithdefaultvalueofInfinity

// For the first node, that value is completely heuristic.
fScore[start]:=heuristic_cost_estimate(start,goal)

whileopenSetisnotempty
current:=thenodeinopenSethavingthelowestfScore[]value

ifcurrent=goal
returnreconstruct_path(cameFrom,current)
```

```
openSet.Remove(current)

closedSet.Add(current)
foreachneighborofcurrent
ifneighborinclosedSet

continue // Ignore the neighbor which is already
evaluated.

// The distance from start to a neighbor
tentative_gScore:=gScore[current]
+dist_between(current,neighbor)
ifneighbornotinopenSet // Discover a new node
openSet.Add(neighbor)
elseiftentative_gScore>=gScore[neighbor]
continue // This is not a better path.

// This path is the best until now. Record it!
cameFrom[neighbor]:=current
gScore[neighbor]:=tentative_gScore
fScore[neighbor]:=gScore[neighbor]
+heuristic_cost_estimate(neighbor,goal)

returnfailure

functionreconstruct_path(cameFrom,current)
total_path:=[current]     whilecurrentincameFrom.Keys:
current:=cameFrom[current]
```

IV.CONCLUSION

Adaptive Learning systems can be implemented on the Internet for use in distance learning and group collaboration applications.The field of distance learning now incorporating aspects of adaptive learning. Initial systems without adaptive learning could provide automated feedback to students who are presented questions from a preselected question bank. Current trends in distant learning call for the use of adaptive learning to implement an environment with intelligent dynamic behavior in the learning environment.

V. REFERENCES

[1]. Adaptive and Intelligent Web Based Educational System(2003), Brusilovsky , Peter.

[2]. A Study of Adaptive Learning for Education (11 December 2014), Lavieri , Edward.

[3]. An architecture for multidimensional adaptive test (2012) , Piton-Gonsalves J.