Indian Journal of Science and Technology, Vol 9(26), DOI: 10.17485/ijst/2016/v9i26/93178, July 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Test Paper Generation System using Multi Agents

Saritha Madhavan*, M. A. Bharathy and R. Dhivya

Department of Computer Science and Engineering, SSN College of Engineering, Chennai – 603110, Tamil Nadu, India; sarithamadhesh@ssn.edu.in, bharathy.athma@gmail.com, kutty.dhivya.dhivya@gmail.com

Abstract

Objective: To build a test-paper generation system for the welfare of students to improve their knowledge thereby developing their skills. **Methods:** In this paper, Multi Agent System is employed to generate the test paper for students. Different intelligent agents such as Question Selector Agent, Performance Analysis Agent and Co-ordinating Agent are designed, which interact with each other, to find the performance of the student in the test. The proposed algorithms assign utility values to the questions stored in the database and based upon the subject preferences given by the students, test paper is generated. **Findings:** A new intelligent system is proposed which helps the students to understand their knowledge level in the preferred subjects. The developed test paper generation system overcomes the drawback of normal class room test by reducing the problem in crowd management and also by making the test paper evaluation process easier and faster. The student can take the test from any place and the system is designed in such a way that each time when the user takes the test, he would get different set of questions. The difficulty level of the test paper is automatically computed for every test the student takes, based upon his previous test performance. **Applications:** The test paper generation system is a helping hand for students' community to improve their study skills in their subject areas. The proposed system using Multi Agents improve the efficiency of the learning process of students.

Keywords: Coordinating Agent, Multi Agent System, Performance Analysis Agent, Question Selector Agent, Test Paper Generation System

1. Introduction

An agent is an entity which observes the environment through sensor and act upon the environment with its actuator to achieve its goal. An agent is autonomous[1] as it can work without the intervention of humans and can control its actions on its own. It can be either social or reactive. An agent environment can be either deterministic or non-deterministic, static or dynamic. If changes in the environment are predictable, then the environment is deterministic else it is non-deterministic. If there is no change in the environment when the agent is "thinking", it implies that the environment is static. If there is any change in the environment when the agent is thinking, then it is called as dynamic environment. Multi agent is more advantageous than autonomous agent because in a multi agent, even though something goes wrong with any one agent, the system can survive. But in the case of a single agent, if anything goes wrong with the agent then the system may get crashed. While performing complex operations using single or individual agent, the response time would be more than using Multi Agent System (MAS). Decision making[2] is an essential work of MAS in order to compose the decisions made by each agent in the entire system. MAS is built up of intelligent agents[3] which play a vital role in different domains that include medical and industrial applications, e-commerce, entertainment and education system.

In today's competitive world, to develop lifelong learning skills, the students need to learn and understand the socio-cognitive and meta-cognitive processes[4] that are involved. The multi-agents give strategic advice and guide the students to expertise in their specific area of interest. Adaptive learning and teaching strategies are in great demand in order to improve the efficiency and effectiveness of the education system; but a very few intelligent education systems exist, to satisfy individual students' requirements. An intelligent agent-assisted environment[5] was proposed to enhance a highly interactive learning approach. The environment used the student's learning related profile such as learning style and background knowledge in selecting, organizing and

* Author for correspondence

presenting learning material and it also presented an environment that delivered the content based on smart instructional components that can be integrated into a wide range of courses.

Utility based agent[6] for test paper generation achieved the difficulty level of the test paper by selecting appropriate level of the questions. It used only one agent called TPAgent to perform this task. First, TPAgent calculated highest utility value for the given difficulty level. It retrieved few questions with highest utility value and the utility value of the remaining questions is one less than highest utility value. Particle Swarm Optimization (PSO)[7,8] is a process of obtaining the best value by making a swarm of particles fly through the problem space. In PSO approach, test paper generation was made dynamically by considering many criteria such as difficulty level of the questions, the weight of each topic and the exposure frequency of the question. Exposure frequency was used to know the number of times the question appeared in the test paper. If the exposure frequency is higher, then it implies that this question has appeared maximum number of times, so student might remember the answer of this question. Hence the system should control the exposure frequency of this test item. Test paper can also be generated using criteria such as teaching requirements and difficulty level[9].

We have made use of the MAS to build a Test Paper Generation System (TPGS). In our work, MAS is used to generate the questionnaire for the students in order to improve their knowledge in the subjects in which they are not well-versed. The agents in MAS use different algorithms to get the current knowledge of the student and make him more knowledgeable in the less-familiar subjects.

2. Proposed System for Test Paper Generation

In early days, test paper was generated and evaluated by the teachers. Teacher spends more time to analyze the questions and to set the test paper. With the introduction of the intelligent agents, test paper can be generated automatically and the result will be given instantly to the student through the fast evaluation. Thus it reduces the burden of teacher and saves their time to concentrate on other academic work. TPGS generates the test paper automatically using multi-agents. In order to generate the test paper, the user has to provide the difficulty level of the test paper and preference of the different subjects. There are many services that generate the test paper automatically. But this work not only generates the test paper automatically, it also shows the performance of the student in each subject, so that he can identify the knowledge level in each subject and can concentrate in the subject in which he is not well versed. When he takes the test next time, the previous performance is taken into account for generating the next test paper. It also reduces the search time of questions, because the search is done in parallel. TPGS generates the test paper automatically using three agents; Performance Analysis Agent (PAA), Coordinating Agent (CA), Question Selector Agent (QSA). The interaction among all three agents is shown in Figure 1.

- Coordinating Agent: It calculates the number of questions in each subject, evaluates the test paper and acts as coordinator between user interface, PAA and QSA. This is also used to create and kill the QSA.
- Performance Analysis Agent: It computes the percentage of number of questions in each subject to be displayed in the test paper.
- Question Selector Agent: It decides the utility value of the questions and the number of questions in each utility value. It is used to update the utility value of the question depending upon the user performance. The QSA also takes care of non repetition of questions.

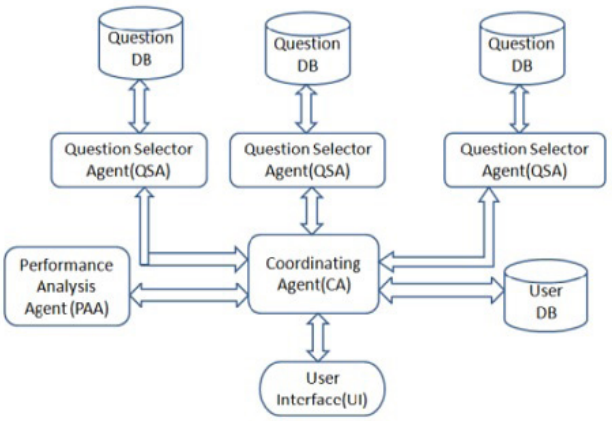

Figure 1. Architecture of test paper generation system.

The system works as follows: The teacher has to create an account by giving her name and subject. CA generates the user id and password for the teacher and this information is updated in the teacher database. She can update the question database by logging in. The user should create an account, while creating an account the user is asked to provide his name, email id, subject preferences and the overall difficulty level of the test paper. CA gets the information from the user and stores it in the user database. It also generates the user id and the password. The subject preferences are passed to the PAA, which in turn returns the percentage of the questions in the test paper for a particular subject.

When the user logs in, QSA is created by CA and depending on the percentage of the questions in the test paper for a particular subject, number of questions for each subject is determined and sent to the QSA along with difficulty level. QSA decides from which level (difficult, medium, easy) the questions need to be selected and how many questions have to be selected from that particular level. To achieve the overall difficulty of the entire test paper specified by the user, it selects the questions that have not yet been selected. If the required number of questions is more than the selected questions, then the remaining questions are selected from the databases which are least recently used. Now the set of questions are sent back to the CA. CA collects the questions from all QSAs and displays the test paper. CA evaluates the test paper and the result is sent to the PAA and the user. It also sends the selected questions along with their result status such as correct or wrong to QSA. Once QSA receives the information sent by CA, it sets the lag value for the questions that are answered correctly. If a particular question has appeared more than ten times, then the utility value of the question is changed depending on the performance of the students for that particular question. Depending upon the results, PAA determines the difficulty level and percentage of the questions in the test paper for a particular subject for the next test to be taken by the same user. The determined difficulty level is sent to the CA by the PAA and CA uses it for the next login of the same user. The coordination among agents for generating and evaluating test paper is shown in Figure 2 and Figure 3 respectively.

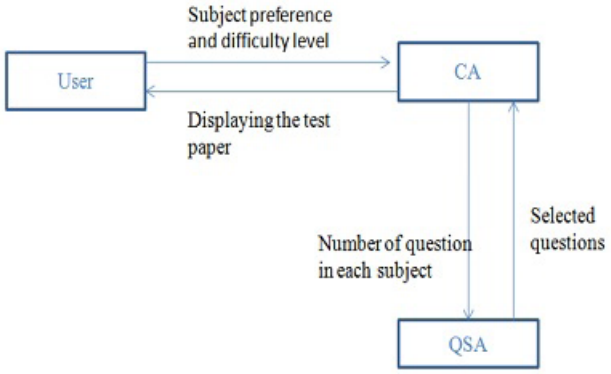

Figure 2. Generation of test paper.

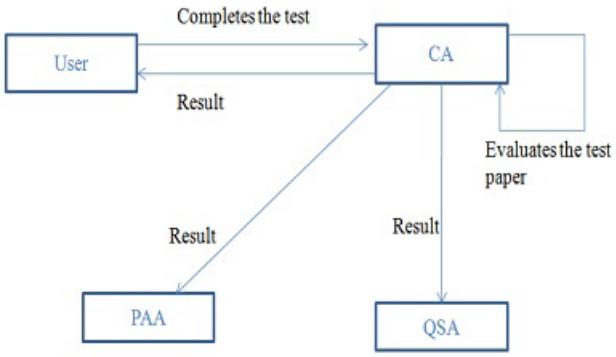

Figure 3. Evaluation of test paper.

2.1 Algorithm used in Question Selector Agent (QSA)

The QSA computes the number of questions in each utility level and recomputes the utility value of each question.

2.1.1 Calculating Number of Questions in Each Utility

Step 1: Calculate the highest utility value (*huval*) by using the formula,

$$huval = d/20$$

Where d (given in percentage) is difficulty level of the test paper.

Step 2: Calculate hn_i, the number of questions with highest utility value by using the formula,

$hn = n_i * d / 100$

Where n_i is total number of questions in each subject.

Step 3: Calculate sn_i, the number of questions with one less than highest utility value by using the formula,

sni=(((100-d)/100)*(10-(d/10))*n_i)/10

Step 4: Retrieve hn_i number of questions with highest utility value from the database.

Step 5: Retrieve sn_i number of questions with one less than highest utility value from the database.

Step 6: The remaining number of questions (n_i-(hn_i-sn_i)) with 0.5 less than highest utility value is also retrieved from the database.

The utility value of each question is changed based upon the number of times the question is answered correctly.

2.2 Algorithm used in Performance Analysis Agent (PAA)

The percentage of number of questions in each subject and the new difficulty level for the next test paper for the same user is determined by using the PAA.

2.2.1 Computing Percentage of Questions in Each Subject

Step 1: Check whether login count is zero. If it is zero then initialize the percentage value for each subject otherwise go to step 2.

Step 2: Calculate the ratio of pass percentage for all subjects by using the following formula:

$r_1=(t_1/(p_2 * p_3)) * 100$

$r_2=(t_2/(p_1 * p_3)) * 100$

$r_3=(t_3/(p_1 * p_2)) * 100$

Where r_i is ratio of pass percentage of each subject, t_i is percentage secured by the user in the i^{th} subject, p_i is percentage of questions in each subject in the previous test.

Step 3: Find the two subjects which have minimum and next minimum ratio.

Step 4: Increase or decrease the percentage of the questions for the subject which is having minimum and next minimum ratio based upon how the user performs the test.

Step 5: Add the percentage of the number of questions in two subjects which is calculated on the above step and subtract it from 100 which will be taken as the percentage of number of questions for the third subject.

2.2.2 Finding Difficulty Level

Step 1: If resultant percentage scored by the user is in the range of *0-25*, then *d = d-20,* if it is in the range of *25-50*, then *d = d-10,* if it is in the range of *50-75*, then *d = d+10,* if it is in the range of *75-100*, then *d =d +20,*

Where *d* is the difficulty level of the test paper.

Step 2: If calculated difficulty value is less than minimum threshold then set the *d* value as minimum threshold.

Step 3: If calculated difficulty value is greater than maximum threshold then set the d value as maximum threshold.

2.3 Implementation

The work is implemented in Java. The popular tool used to implement Multi Agents is JADE (Java Agent Development Environment). It uses the standard FIPA ACL communication language. JADE platform is a popular FIPA compliant platform for the development of multi agent system. JADE is a software framework fully implemented in Java language. It facilitates development of interoperable intelligent multi agent systems which is distributed under an open source license. JADE[10] is a very mature product, being used by heterogeneous community of users both in research activities and in industrial applications. It simplifies the implementation of MAS (Multi Agent System). JADE includes a library of classes to create agent, an agent platform that provides the agent service such as life cycle, message encode and message transport. JADE supports the mobility of agent by allowing the agents to migrate between the containers. An agent can have different behavior and different agents can be in the same container.

3. Results and Discussion

In order to take the test, student has to register first. After registration, the information about the user is updated in the user database by CA. CA calculates the number of questions in each subject according to the preferences given by the user. The calculated number of questions in each subject and the overall difficulty level

for the test paper is sent to the corresponding QSA. In order to obtain the overall difficulty level of the test paper, QSA selects the questions from different utility values. The highest utility value and the number of questions in each level (difficult, medium, easy) are computed. If the difficulty level given by the user for the test is high, then the number of questions from easy level will be less than the other two levels. For example, in our work, we have taken three subjects and the Figure 4 shows the number of questions selected in a subject in each level.

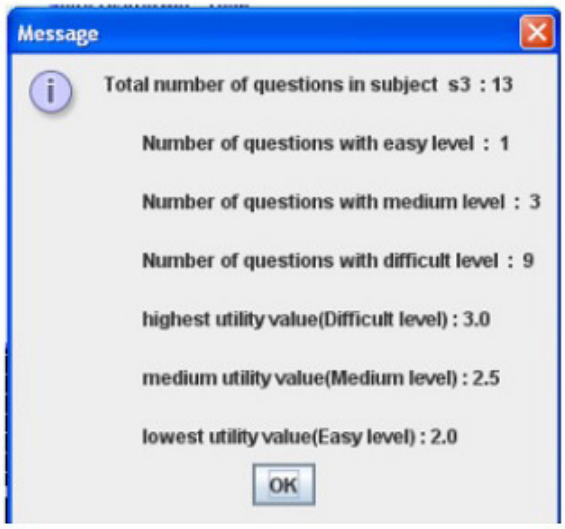

Figure 4. Number of question in a subject.

Thus the test paper is generated based upon the number of questions in each subject which matches the utility values. The marks secured by the student is shown in Figure 5.

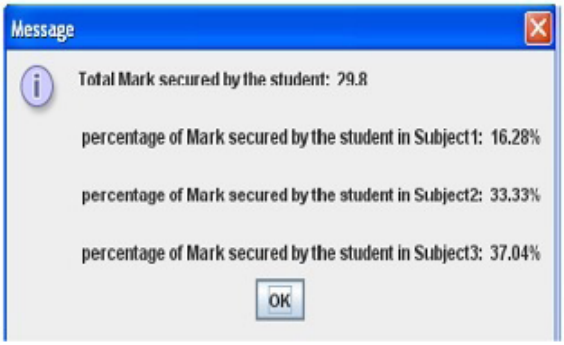

Figure 5. Marks secured by the student.

CA sends the marks secured by the student to the PAA. PAA calculates the overall difficulty level and the number of questions in each subject for the same student automatically for the next test. If the student performs poor in the test, it shows that the user is not capable for this difficulty level and so the system decreases the difficulty level for the nest test; otherwise the system increases the difficulty level. QSA updates the utility value for the questions based upon the user's performance. When the same student takes the next test, new question paper is generated with the updated difficulty level, thereby TPGS showing his performance variations at different levels of taking the test.

4. Conclusion

The proposed test paper generation system is used to generate the test paper automatically depending upon the user's performance. The system uses multi-agents to generate and evaluate the test paper automatically without human intervention and it also analyzes the student's performance. The system can be used by the student who prefers self study. It helps the students to categorize their knowledge level in each subject so that the user can identify the subject in which he has to concentrate more. The TPGS helps the students to improve their skills by analyzing the test performances on their own. The current work is developed as a desktop application. It can be made more useful for the students by providing as a web service.

5. References

1. Bellifemine FL, Caire G, Greenwood D. Developing Multi Agent Systems with JADE. John Wiley and Sons; 2007.
2. Amudhavel J, Sruthy G, Padmashree D, Pazhani Raja N, Saleem Basha MS, Bhuvaneswari B. A comprehensive analysis on Multi Agent decision making systems. Indian Journal of Science and Technology. 2016 Mar; 9(11):1–5.
3. Das TK. Intelligent techniques in decision making: A survey. Indian Journal of Science and Technology. 2016 Mar; 9(12):1–6.
4. Sun S, Joy M, Griffiths N. The use of learning objects and learning styles in a Multi Agent education system. Journal of Interactive Learning Research. 2007 Jul; 18(3):381.
5. Shang Y, Shi H, Chen SS. An intelligent distributed environment for active learning. JERIC. 2001 Aug; 1(2):4.
6. Naz M, Aslam M, Dar E. Utility based agent for test paper generation. International Journal of Multidisciplinary Sciences and Engineering. 2010; 1(1):1–5.
7. Peng C, Anbo M, Chunhua Z. Particle swarm optimization in Multi Agent System for the intelligent generation of test

papers. Evolutionary Computation (IEEE World Congress on Computational Intelligence); 2008 Jun. p. 2158–62.
8. Weng MM, Hung JC, Lin F, Shih TK. A personalized quiz game based on Multi Agent System. New Horizons in Web Based Learning: Heidelberg, Berlin: Springer; 2012 Sep; 7697:100–9.
9. Cheng SC, Lin YT, Huang YM. Dynamic question generation system for web-based testing using Particle Swarm Optimization. Expert Systems with Applications. 2009 Jan; 36(1):616–24.
10. Bellifemine F, Caire G, Poggi A, Rimassa G. JADE: A software framework for developing Multi Agent applications. Lessons learned. Information and Software Technology. 2008 Jan; 50(1):10–21.