Design of Adaptive Question Bank Development and Management System

Vijay Krishan Purohit*, Abhijeet Kumar*, Asma Jabeen*, Saurabh Srivastava*, R H Goudar* ,Shivanagowda+, Sreenivasa Rao**
vijaykrishan1990@gmail.com, abhijeetchar@gmail.com, asma9geu@gmail.com, saurabhsrivastavacs@ieee.org, rhgoudar@gmail.com, shivana.gowda@gmail.com, srmeda@gmail.com
*Department of Computer Science and Information Technology, Graphic Era University (GEU), Dehradun, India.
+Dept. of CSE, SDMCET, Dharwad, Karnataka
**SIT, JNTUH and CIHL, MSIT Division, IIIT Hyderabad

***Abstract*—In the changing present competitive scenario, Intelligent development of question model is indispensible for intellectual growth of students and also to fulfill learning objectives of the course (Blooms Taxonomy). Although there are several computer based question paper generators but they typically use random selection from question bank. Also these databases (question bank) are not rich enough to avoid recalling (repetition of questions). Moreover, History is also not maintained for the previous question module generated. Such systems lack intelligency as they do not use concepts maps and are unable to provide quality questions with variations. In this paper, we are designing an adaptive question bank management system that is intelligently picking questions from rich database (question bank) and representing the question model according to the inputs or parameters provided by the question paper designer (QPD). This question bank uses concept map developed using graphical method that uses hierarchical knowledge of particular domain. The concept map integrated with question bank (question database) will ensure the question modeling process based on degree of certain criteria like Bloom's Taxonomy, difficulty level, marking scheme etc. The evaluation of generated question model will provide a feedback to check overall student's level of understanding and this will be an advantage for an organization for enhancing the growth of their students. In sum, our proposed system would be great aid for the organization in effective question modeling and its assessment.**

Keywords-Question modelling, Bloom's Taxonomy, Question paper designer engine, Question storage engine, Question bank (Database), Concept map.

I. INTRODUCTION

Learning is a complicated activity. Student's aim is to master knowledge and skills for their career. Most of the times, students get only recall (bottom of the Bloom's Taxonomy) type questions during assessment phase but professionally they are assessed on all aspects as per blooms taxonomy. This creates a knowledge gap which is not revealed in the former part. The current education system does not ruminate on this concern and intellectual growth of students founders. Teachers' aim is to pass on knowledge and enhance students' learning ability. Every Courses have LO's (Learning Objectives), Assessment for such course is verification by the course's teacher to check whether the students have met or achieved the objectives or not. But designing such questions to test learning (categorized by Bloom) of the students with respect to particular courses needs enormous experience and is time consuming activity. This would be difficult for budding teachers and even experienced teacher may not be able to create such questions frequently as there are several tests, quizzes and exams in a year in any academic house. The complex curriculum puts pressure on the question paper designer (QPD) to assess students along with covering all the objectives of the course. So, there is a need to develop a intelligent technology based system for generating standard question paper that examines the overall growth of examinee. A standard question paper contains eclectic questions from rich question bank (database) according to organization's course curriculum. Therefore, we are implementing an adaptive question bank management system that is intelligently picking questions from rich graph database (question bank) and representing the question model to the inputs or parameters provided by the question paper designer (QPD).

II. BACKGROUND

Considerable effort has been devoted to research in the area of technology-enhanced education system over past several years. Several online programs have been developed for questioning system and online evaluation of a student. Commercial products like "WebAssign" [1] [6] and "WebCT" [1] [7] are existing which include immediate feedback given on every question attempted. Similar work has been done for online physics homework system (Andes) [5]. These research efforts are online and are emphasized on students' interaction and their evaluation as per different category (Blooms Taxonomy [12] [13]). Above works do not focus on the question paper generation. In the current education system there is a need to develop an adaptive question bank management system that engenders a standard question paper which can be used by an educational organization or institution for assessing their students all together (offline) as per their course curriculum. Although there are several computer based question paper generators but they typically use random selection from pool of questions. Basically, these systems use RDBMS and linear searching for picking questions which is inefficient.

One of the important constituents of all these systems is **Concept Map** [2][3][4][8]. Much of efforts, study and development on concept map is already being carried out with inclusion of the cross linking feature [2] [3]. Construction of a preliminary concept map can be done by writing all of the concepts on Post-its™, [2] or preferably by using the IHMC CmapTools, a computer software program. In this paper we are not focusing on concept map rather on the methodology and algorithm for efficiently and intelligently picking up questions for QP from question bank integrated with concept map (discussed in Proposed Mechanism).

III. PROPOSED SYSTEM

A. System Overview

The objective of this software system for an academic institution is to exploit the potential of expertise around the world in collecting questions and effectively managing the collected questions. Then inculcate the system with intelligence to generate the question papers as per the requirement (Marks, Skills, Difficulty level, Syllabus, Types test etc) of the Course teachers of the Institute for particular courses.

Basically, our proposed system has three users:

- Question Designer (QD)
- Question Paper Designer (QPD)
- Administrator (for authentication)

Every educational organization has a fixed course curriculum. Our system will be used in complaisant with the organization and uses organizational course curriculum for designing concept map. The above mentioned users have different roles in the system and they will provide inputs accordingly to Adaptive Question Bank Development and Management System (AQBDMS). The work flow and key features of our system can be summed up as:

- After verification the institute or organization recognizes the registered people as question designers (QD) and question paper designer (QPD) and provides them Login ID and Password for the portal.
- QD donates questions of their interested area/subject/topics as per specification of institute along with scheme of evaluation.
- QPD feeds his specification and input parameters to the intelligent system in order to fetch the editable question paper in a stipulated amount of time.
- Every submitted question will be verified for repetition equivalence, correctness and realistic by special committee of institute (Administrator).
- The system maintains the history of the questions appearing in question paper (QP) to take off repeating issues.
- A question which is added to the question bank has various dimensions. They are divided according to Blooms Taxonomy, marks, difficulty level, Type, Topic.

AQBDMS is comprised of three basic components: the user interface, the questions bank (Database) and the intelligent development and management system as shown in figure 1. An appropriate user drives the system via user interface. Intelligent development and management system for questions gives the user (QD) graphical concept map for augmenting questions. Model of our proposed system is shown below:

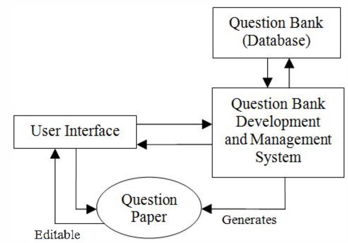

Fig. 1: Adaptive Question Bank Development and Management System Model.

Depending upon the input parameters of user (QPD), it presents standard question paper (QP) by intelligently picking questions from the backend. We are inculcating intelligence to our system by traversing the concept map with the efficacy of our algorithm which in turn results in generation of good combination of questions.

B. System Architecture

To implement "Adaptive Question Bank Development and Management System" several databases are required. The Question Database or Question Bank is implemented using graphical hierarchy with the integration of the concept map of that curriculum which stores questions pertaining to a specific curriculum. User Database is a relational database which stores information of the users, authorized to interact with the system i.e. QPD & QD. Question Paper Database is a relational database which stores the Question Papers generated by QPDE and is accepted by the QPD for the future review. AQBDMS comprises of two engines i.e. Question paper designer engine (QPDE) and Question storage engine (QSE). QD drives the QSE via web server. Function of QSE is to augment questions in the Question DB. QPD drives QPDE and its functionality is to fetch questions from Question DB in order to generate a balanced question paper and also to make a record of the same in Question paper DB. Administrator will have access to all the databases for monitoring of resources and authentication of questions.

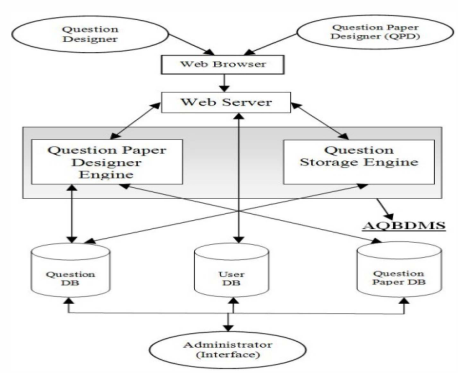

Fig. 2: System Architecture.

C. Case Study (Example)

We are developing this system for the domain *"C-Programming"*. Note that AQBDMS is under development and has not yet been fully built and tested. This paper describes the theory, concepts and algorithm to be implemented for developing adaptive question bank management system.

C.1. Entity Relationship (ER) Model

To implement this system, we are using database that contains a huge collection of questions which are attached below the leaf nodes of concept map graph. The relationship among various entities of our system is shown in fig.3. Here we can observe the entities Question Paper, Question Paper Designer, Question Designer, Course and Question are represented through relationships.

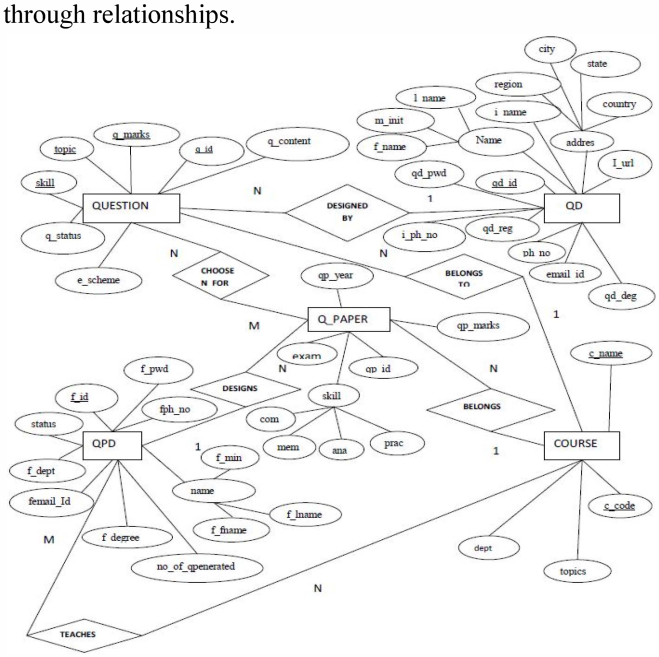

Fig. 3: ER Diagram.

Various attributes of each entity are also shown. Observe that entity: question has several attributes like scheme, skill, status, topic, marks, content, etc and entities like QD, QPD have attributes: id, password which are used for their authentication while logging-in for the portal. Besides this, they also have other information for maintaining their profile. The QPD is related with the particular Course in M:N relationship and QPD is only allowed to generate the Question Paper for the Department (f_dept : foreign key) they teach. The Question paper belongs to a particular course and is designed by QPD. The responsibility of Administrator of this complete system (Portal) is to verify the questions added by the QD. Moreover, the administrator is entitled to maintain the course curriculum of an institution or an organization by modifying concept map (discussed in next section).

C.2. Concept Map

A concept map is tool to organize and represent powerful knowledge frameworks for a specific *focus question[2]*. In our proposed system, the focus question is: "*What is the course curriculum of C-Programming?*". The concepts are represented in a hierarchical fashion with the most inclusive, most general concepts at the bottom of the map and the more specific, less general concepts arranged hierarchically above (see fig.4). In curriculum planning, concept maps are enormously useful. The hierarchical organization of concept maps suggests more optimal sequencing of instructional material. Since the fundamental characteristic of meaningful learning is integration of new with the learners' previous concept and propositional frameworks, proceeding from the more general, more inclusive concepts to the more specific information which usually serves to encourage and enhance meaningful learning. Entire curriculum can be redesigned by special committee working collaboratively and then updated by administrator. In our case, we are considering only a few sub-domains of *"C-Programming"* as listed below.

- Memory Allocations
- Arrays
- Strings
- Functions
- Structures
- Unions
- Pre-Processor Directives
- Output Finding

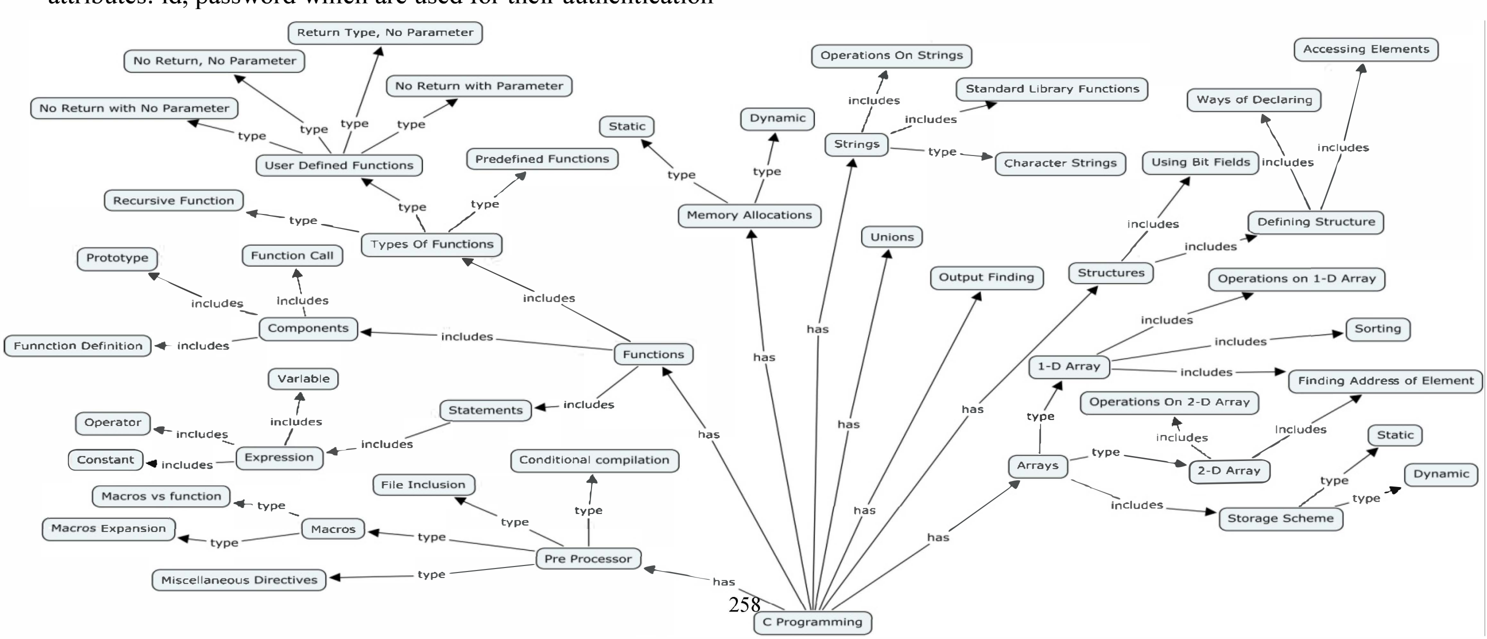

Fig. 4: Concept Map of "*C-Programming*".

For each sub-domain, we have further selected set of concepts (parking lot [2]) defining it. The questions are arranged in a data structure below the most precise concept of sub-domain. Even if a particular question is using the concepts of different sub-domains then the question is linked to all those concepts of respective sub-domain. We have constructed the concept map of our case with ***Cmap tool*** [10] [11] that is described in fig. 4.

C.3. Algorithms

This algorithm is based on intelligent distribution of questions among the chapters selected. Though our algorithm is more general, we are considering a problem statement for proposing our algorithm. ***Problem Statement:*** A paper is generated for "C-Programming" for first four chapters (given in concept map section). QPD had given his inputs as below:

- 30% chapter1, 20% chapter2, 30% chapter3, 20% chapter4.
- Objective questions-30, short-10 and long-6.
- 30% easy, 30% medium and 40% hard.
- 20% remembrance, 30% understanding, 20% application and 30% analyzing (Blooms Taxonomy).

Now we will analyze the distribution and picking of objective type questions from graph (concept map integrated with questions). There are 30 questions of objectives which is divided in 9, 6, 9, 6 among respective chapters and 9, 9 and 12 for respective level (easy, medium, hard). Similarly 6, 9, 6, 9 in remembrance, understanding, application and analyzing respectively according to their percentage weight. First of all, our aim is to generate a good combination of questions which are equally distributed among levels, taxonomy and chapters. For this, we are using Matrix Method algorithms for establishing a relationship among Bloom's Taxonomy, Levels and Chapters. In fact, we have designed three algorithms: first for generating combination of questions for Bloom's Taxonomy and Chapters (algorithm1), second for levels and chapters (algorithm2) and third for Bloom's taxonomy, Levels and Chapters (algorithm3). Generated output matrix from first and second algorithm is input of third algorithm which in turns establishes the final relationship among the attributes of questions. According to the parameters provided by QPD,

We have, bloom[p] ←{6, 9, 6, 9} ; p←6;

level [q] ←{9, 9, 12} ; q←3;

chap [r] ←{9, 6, 9, 6}; r←4;

Algorithm 1:

Algorithm for creating 2-D matrix to establish a relationship between Bloom's Taxonomy and Chapters by Matrix Method is shown below:

```
CreateMatrix_BloomsChaps (bloom [ ], chap [ ])
Initialize matx1 [p][r] ← 0 ; ch[ ] ←0; b←0; c←0;
for b ← 0 to p-1 do
    while bloom[b]!=0
        do
            if c>r-1 then
                set c←0;
            if ch[c]<chap[c] then
                matx1[b][c] ←matx1[b][c]+1;
                ch[c] ← ch[c]+1;
                bloom[b] ← bloom[b]-1;
                c←c+1;
            else
                c←c+1;
```

After passing through these steps we get a 2-D matrix as:

$$
\begin{array}{cc}
 & \text{Chapters} \\
\begin{array}{c}\text{Blooms}\\ \text{Taxonomy}\end{array} & \left(\begin{array}{cccc|c}
2 & 2 & 1 & 1 & 6 \\
2 & 2 & 3 & 2 & 9 \\
2 & 1 & 1 & 2 & 6 \\
3 & 1 & 4 & 1 & 9 \\
\hline
9 & 6 & 9 & 6 &
\end{array}\right)
\end{array}
$$

Matx1 [p] [r]

We can observe here that number of questions is distributed among chapters and blooms level. At the same time rows and columns are validated according to the provided input limit intelligently. Similarly we got the second matrix from algorithm 2.

Algorithm 2:

Algorithm for creating 2-D matrix to establish a relationship between Level and Chapters by Matrix Method:

```
CreateMatrix_LevelChaps (level [ ], chap [ ])
Initialize matx2 [q][r] ← 0 ; ch[ ] ←0; l←0; c←0;
for l ← 0 to q-1 do
    while level[l]!=0
        do
            if c>r-1 then
                set c←0;
            if ch[c]<chap[c] then
                matx2[l][c] ←matx2[l][c]+1;
                ch[c] ← ch[c]+1;
                level[l] ← level[l]-1;
                c←c+1;
            else
                c←c+1;
```

Again, after passing through these steps we get a 2-D matrix as:

$$
\begin{array}{cc}
 & \text{Chapters} \\
\text{Levels} & \left(\begin{array}{cccc|c}
3 & 2 & 2 & 2 & 9 \\
2 & 3 & 2 & 2 & 9 \\
4 & 1 & 5 & 2 & 12 \\
\hline
9 & 6 & 9 & 6 &
\end{array}\right)
\end{array}
$$

Matx2 [q][r]

Number of questions selected for particular level or chapter is validated in rows and columns. Now when we have got two matrices, our aim is to create the 3-D matrix with the help of these two matrices. These two are given as input to the third algorithm (shown below in algorithm 3).

Algorithm 3:

Algorithm for creating 3-D matrix to establish a relationship among Bloom's Taxonomy, Levels and Chapters by Matrix Method:

```
CreateMatrix_BloomsLevelChap (matx1 [ ][ ], matx2 [ ][ ])
Initialize matx3 [p][q][r] ← 0 ; l←0; c←0; b←0;
for c← 0 to r-1 do
        for b ← 0 to p-1 do
              while matx1[b][c]!=0
                    do
                    if l>q-1 then
                        set l←0;
                    matx2[l][c] ←matx2[l][c]-1;
                    matx1[b][c] ←matx1[b][c]-1;
                    matx3[b][l][c] ←matx3[b][l][c]+1;
                    l←l+1;
```

After executing this function, we get a 3-D matrix whose relationship is used for querying the graph and picking the questions from database attached to leaf node. It is shown below:

$$
\begin{array}{cc}
 & \text{Levels} \\
\text{Blooms taxonomy} & \left(\begin{array}{ccc|c} 1 & 1 & 0 & 2 \\ 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 2 \\ 1 & 0 & 1+1 & 3 \\ \hline 3 & 2 & 4 & \end{array}\right)
\end{array}
\quad \left(\begin{array}{c}\text{Chap.} \\ 1\end{array}\right)
$$

Matx3 [p][q][r]

No. of questions from 2-D array of chapter1= chap [0] =9 Questions from each blooms level=1st column of matx1[p][r] Questions from each difficulty level = 1st column of matx2[q] [r]. This proves that a valid combination of questions have been generated by th7e algorithm for chapter1, likewise questions are generated for other chapters selected. Similarly, questions are generated for the short and long type questions. The next step that comes in process is to traverse the graph and pick questions with those attributes which are generated in the 3-D matrix. For this we are proposing the algorithm below (algorithm 4).

Algorithm 4:

Algorithm for traversing the concept map (Graph-with questions attached below leaf node), distributing questions (The good combination generated in Algorithm-3) to different nodes (According to question paper designer (QPD) input) and then picking them (from leaf node) for generation of standard question paper.

```
   Traversal (chapter [ ], root)
1  for c ← 0 to r do
2        for each vertex u∈ adj (root) do
3                if name[u] ==chapter[c] then
4                        Distribute (name[u]);
   Distribute (name [ ])
1  for each vertex v ∈ adj (chapter_node)
2        count←count+1;
3  if count == 0 then
4        Query & maintain history;
5  else
6        d ←a% count;
7        e ←a/ count;
8  j←0;
9  for each vertex v ∈ adj (chapter_node) do
10        question[v] ←e;
11        childnode_name[j] ←name[v];
12        j←j+1;
13        if d! =0 then
14                question[v] ←question[v]+1;
15                d← d-1;
16  for j=0 to count do
17        DISTRIBUTE (childnode_name[j]);
```

Here, traverse () takes the chapter [ ] as input ie: provided by QPD for paper generation and root is the initial node ("C-programming"). This function searches for each adjacent node of root and calls distribute () when the chapter node is found (step 2-4). The Chapter which is found is passed to Distribute () and then Distribute () counts the nodes adjacent to it (step1-2) and divides the questions (generated in algo.3) among it's child nodes (step 9-15). Note that it is traversing its adjacent child nodes for distribution of questions from a **random set of child nodes.** It uses recursive call to traverse the graph upto leaf node (step 16-17). When it reaches the leaf node, the count becomes 0 as there is no child node further (step3). Then it executes the query (data retrieval query) for picking the questions of prescribed attributes from data structure defined below leaf node (step4). The query contains the various dimensions of the questions to be fetched as shown in Table I.

TABLE I. ATTRIBUTES ASSOCIATED WITH QUESTION.

Attributes of Question	Explanation
Content Area	A key that points to the content of the question.
Scheme	Objective, Short, Long.
Blooms Taxonomy (Category)	Blooms level of Learning Objectives.
Difficulty Level	Easy, Medium and Hard.
Topic	Concepts used in question.
Marks	According to scheme.

Bloom's Taxonomy is a classification of learning objectives of education [12] [13]. We have classified questions on this taxonomy as it is considered to be foundational and essential elements of educational objectives (categorization shown below).

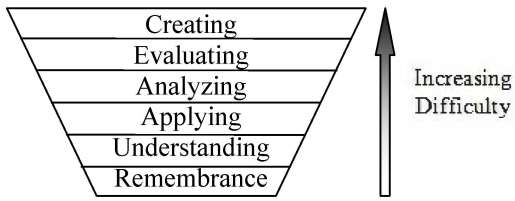

Fig. 5: Bloom's Taxonomy.

In this way, the editable generated question paper is represented to the QPD in the user interface. The QD (Expert) add questions to the database with the help of graphical concept map represented to it. QD has to provide all the attributes (dimensions) of questions as shown in Table I. above. The *'topic'* attribute of a question is set by selecting the leaf concept of the graph. The question will be inserted in the data structure attached to the leaf node automatically by intelligent development system. As a question may use more than one concept so the topic attribute can also be multi-valued.

IV. ADVANTAGES

Setting a balanced question paper is a complex and demanding process. Adaptive question bank development and management system is intelligent system that takes care of fetching a balanced and standard question paper by distributing the questions equally among the levels, taxonomy and chapters as provided by question designer. This intelligence will reduce the pressure from the question paper designer and will free them from responsibility of setting learning objectives properly. The proposed system is using graph for constructing concept map [9] and a database for storing questions. The database for storing the questions can be a graph database or a relational database as linear search is employed in both cases. Thus, traversing the graph for fetching the questions will reduce the search to greater extent as it has to pick questions from database attached to a particular node. Linear searching is only required for the maintained data structure attached to leaf node. Its complexity decreases dramatically as pool of questions increases in comparison to relational or linear databases. The time complexity of linear search is **O(n)** and the complexity while traversing the graph is **O(log n).** The graph is plotted for ***O(n) vs. O(log n)*** which is shown below.

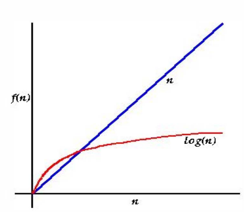

Fig. 6: Time Complexity O (n) vs. O (log n)

Generally, redundancy of questions in the question bank is a major concern. Suppose there are **n** questions and each using **m** concepts then the space complexity for storing questions is **m*n** unit space as it is linked to every concept node separately. We are eliminating this problem by storing the questions in question database once and linking it to all those leaf node concepts which are used in the question. So, by this we are reducing the space complexity by a multiplicative factor **m** and in this case it is **n** unit space. We are maintaining history (q_status attribute) of each question so that recalling is prohibited and different questions are generated every time.

CONCLUSION

An approach to implement *"Adaptive Question Bank Development and Management System (AQBDMS)"* is a flexible and intelligent web based question paper generator. While this application is still a work in progress, it is believed that the success of this project will provide a great aid for the organization in effective question modeling and its assessment. The future goal is to create a flexible system which manages question paper generation for various domains of a particular educational organization or institution. This System would interact with Evaluation system (under planning) which generates the student's courses comprehension-maps which will be used by the course teacher offline for fixing a proper grade letter.

REFERENCES

[1] Stephen A. Zahorian, Vishnu K. Lakdawala Oscar, R. González, Scott Starsman, and James F. Leathrum, Jr., "Question Model for Intelligent Questiong Systems in Engineering Education", 31st ASEE/IEEE Frontiers in Education Conferencee, October 10 - 13, 2001 Reno, NV, © 2001 IEEE.

[2] Joseph D. Novak, Alberto J. Cañas, "The Theory Underlying Concept Maps and How to Construct and Use Them", Technical Report IHMC CmapTools 2006-01 Rev 01-2008.

[3] Gul Tokdemir, Nergiz Ercil Cagiltay, "A Concept Map Approach for Introduction to Computer Engineering Course Curriculum", IEEE EDUCON Educational Engineering 2010- The future of learning and global education, April 14-16, 2010, Madrid, Spain, © 2010 IEEE.

[4] Jie Wu, "The introduction of contemporary teaching approaches to improve student learning in Fundamentals of C Programming", The China Papers, November 2006.

[5] Schulze, K., Shelby, R., Treacy, D., & Wintersgill, M., Andes: A coached learning environment for classical Newtonian physics, Proceedings of the 11th International Conference on College Teaching and Learning, Jacksonville, Florida, 2000.

[6] URL: http://www.webassign.com; WebAssign.

[7] URL: http://www.webct.com; WebCT.

[8] A conceptual map model for developing intelligent tutoring systems. # 2002 Elsevier Science Ltd. All rights reserved.

[9] John H. Milam, Jr., Susan A. Santo, Lisa A. Heaton, Concept Maps for Web-Based Applications, ERIC Technical Report.

[10] Alberto J. Cañas, Rodrigo Carvajal, Roger Carff & Greg Hill CmapTools, Web Pages & Websites, Technical Report IHMC CmapTools 2004-01.

[11] Concept mapping: How can we utilize it in our class?, Center for Instructional Technology, Carrier Rm 22 540.568.7061.

[12] Elizabeth Dalton The "New Bloom's Taxonomy,"Objectives, and Assessments, December 3,2003.

[13] Mary Forehand, "Bloom's Taxonomy-Emerging Perspectives on Learning, Teaching and Technology", The University of Georgia.