AWRC: A Web-Based Reinforced Concrete Design Adaptive Testing System

Yu-Hur Chou[1], Shang-Hsien Hsieh[2]
Tung-Nan Institute of Technology[1]/ National Taiwan University[2], Taiwan

Abstract

Reinforced Concrete (RC) design is a very important course on civil engineering in higher education. Students are taught in this course how to design a structure by using reinforced concrete. Students are also trained to follow the building code (Code by ACI, American Concrete Institute) into design [11]. Because the building code is a very complicated collection of rules, students are easily misunderstanding in the learning procedures and instructors or tutors must pay more time to explain the RC theory and describe the procedures of design [8,9]. Otherwise, teachers need spend a lot of time to correct students' examinations[5]. It is also difficult to evaluate students' misconceptions by traditional test. Therefore, it is necessary to develop computer-aided tools to help teachers' teaching for the RC design course.

This paper describes AWRC, a web-based reinforced concrete design adaptive testing system developed by author's research team. AWRC includes a test bank saves over 200 testing problems, a testing generator develops adaptive testing problems to each student's achievement level, a diagnosis system may find students' misconception through the testing progression. A level manager is used to calculate the difficulty level of a problem and the achievement level of a student. An algorithm by "skills" to assign the initial difficulty level of a problem is proposed in this paper. The functions of AWRC also include test delivery, scoring, and record keeping. AWRC can help teachers to generate, score and assess an examination. Students may login AWRC to practice examples and to check the design procedures of RC structures in 24 hours.

The testing system is made on the NT platform, written by ASP (IIS), Visual Basic and uses SQL Server as a database system. The major advantage of this testing system is fully automatic and adaptive.

Keywords: on line test, adaptive testing system, reinforced concrete design, fault diagnosis, difficulty level

1. Introduction

With the fast improvement of computer technology and the Internet, educators may alternate pedagogical tactic from traditional teaching strategy to the e-learning which highly uses computer capability for creating, storing, sharing and distributing teaching curriculums and testing materials [2,3,6]. The use of computers for testing purposes, called Computer-Based Testing (CBT), has a history more than sixty years. The IBM developed the first computer machine for testing scoring in 1935 [4]. The CBT can be divided into three levels, the computer scoring, the computer testing and the adaptive computer test. The adaptive computer test is the most difficult to develop, because it depends on the Item Response Theory (IRT), Knowledge Space Theory (KST) or other approaches are more complex than the traditional testing theory. The adaptive computer test has advantages of easy finding testers' level, reducing to inputting information and a minimum of redundancy. Besides, the Internet was improved very fast in the last twenty years and will become more and more important in the future. Therefore, to develop an on-line testing system is required and necessary. The course of reinforced concrete (RC) is opened in the third years on civil engineering. The design procedures are theoretic and sequential constant. Students are trained to follow rules to make calculations and design the structure. Therefore, the RC design is a strict hierarchical structure that is easy to program and to make the adaptive testing system. This is the main reason that the RC course is selected and developed to be the first Web-course in our department.

This paper describes AWRC, a web-based reinforced concrete design adaptive testing system developed for improving the performance of teaching. AWRC have abilities of generating adaptive test depended on students' achievement, finding students' misconception by using the diagnosis system and describing the whole calculating processes of the test by using the answering system. Testing problems can be assigned by teachers or generated by system automatically.

This paper is organized as follows: We first describe the AWRC system and explain the relationships among components of the AWRC system. Secondary, we explain the design of diagnosis system of AWRC. Third, the logic of the adaptive function and the level manager are discussed. Fourth, an algorithm by "skills" to assign the initial difficulty level of a problem is discussed. Finally, we make some conclusions and illustrate the future studies we will concentrate.

2. AWRC System

The AWRC system consists two databases: Testing Database (TDB in short), Users Database (UDB in short) and four database managers: Testing Generator (TG in short), Diagnosis/Answering Evaluator (DAE in short), Level Manager (LM in short) and Adaptive Manager (AM in short). Besides those two databases and four database managers, the AWRC also includes two user-interface programs: Embedding Tags GUI program (ETG in short) and Testing Assignment GUI program (TAG in short). We use SQL Server as database system to set up TDB and UDB databases. ETG and TAG two GUI program are written by Visual Basic. The others database managers are written by ASP (IIS). The system of AWRC is shown in Fig. 1 as follows:

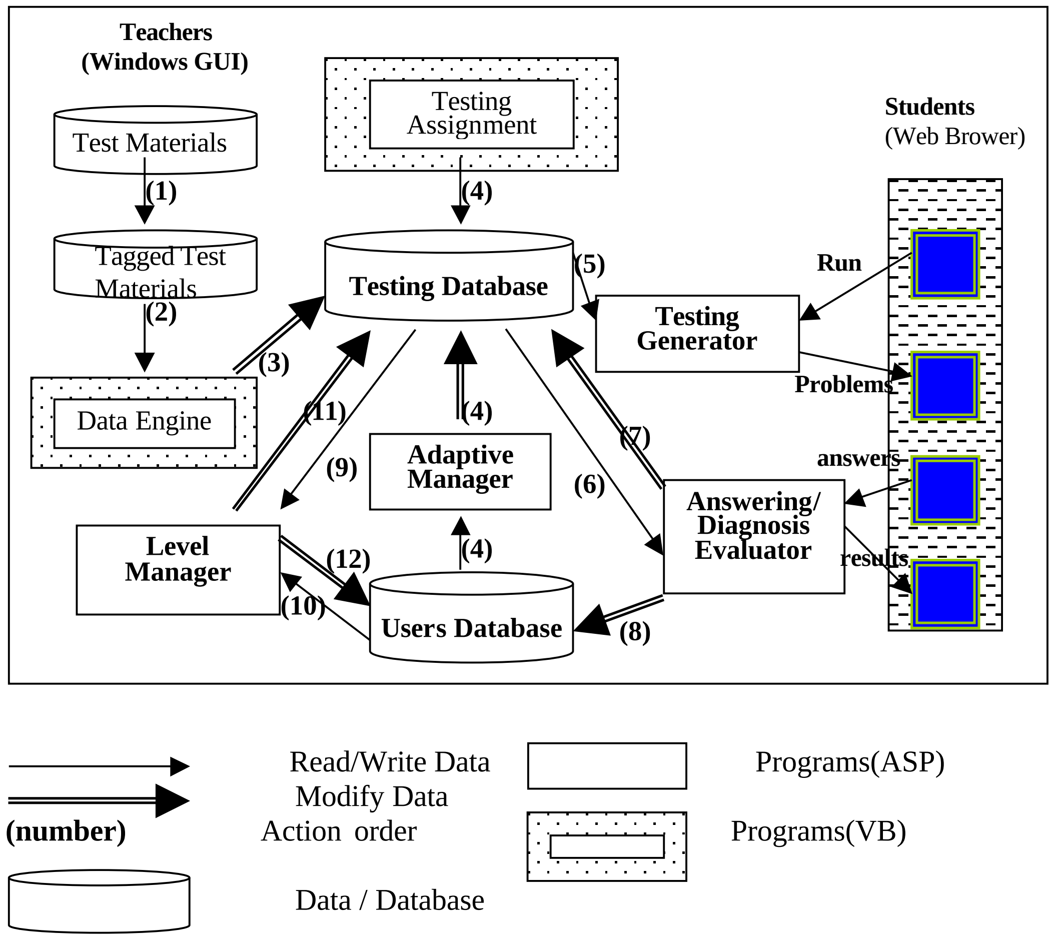


Fig. 1 AWRC system

Embedding Tags GUI program (ETG): The ETG is used to encapsulate structured information into make-up information for improving the reusability and message understanding. With the GUI programming, the ETG is easy to use and can transfer the teachers' testing materials into tagged documents in XML format. After the tagged transformation, these tagged documents will be stored into testing database through the data engine program. The tagged format of a testing problem is shown as the following table:

Table 1. The format of a tagged testing document

<?xml version = “1.0” encoding = Big5” ?>
<AA>
<Problem> Please find the capacity of nominal moment M_n of the beam shown as the following cross-section ? </Problem>
<Input Value = “X”>$M_{n.}$</Input>
<Var2 GT=”30” , LT=”50”, Unit = “cm”>b = </Var2>
<Var2 GT=”50”, LT = “70”, Unit = “cm”> d = </Var2>
<Var2 GT=”5”, LT = “12”, Unit = “cm”> d’ = </Var2>
<Var2 GT =”210”, LT =”350”,Unit = “kg/cm^2”> f'_c = </Var2>
<Var2 GT=”2800”, LT=”4200”,Unit = “kg/cm^2”> f_y = </Var2>
<Var1 GT=”20”, LT=”40”,Unit=”cm^2”>As = </Var2>
<Description>b: Width of beam</Description>
<Description>d: Effective depth of beam</Description>
<Description>f'_c: The strength of Concrete</Description>
<Description>f_y: The yielding stress of steel bar</Description>
<Description>A_s: The total tensile reinforcement</Description>
<Graphic>../graphics/test2.bmp</Graphic>
<Answer>../answers/a13.asp</Answer>
<Diagnosis> ../diagnosis/d13.asp</Diagnosis>
</AA>

In which
<AA></AA>: Problem’s begin and end tags.
<Problem></Problem>: The description of problem.
<Input value = “X”></Input>: Input frame, the attributed name is X.
<Var2 GT =”30”, LT =”50”,Unit = “cm”></Var2>: Variable of integer (30 to 50), the unit is cm.
<Description> </Description>: Description of variables.
<Graphic></Graphic>: The location of the graphic file of the problem.
<Answer></Answer>: The location of the answering program.
<Diagnosis></ Diagnosis >: The location of the diagnosis program.

Testing Assignment GUI program (TAG)**:** Teachers can use TAG to choose the problems from the testing database for setting up a test and assign the testing constraint factors, such as the time limitation, the order of testing problems and the format of problems. The test created by the TAG assignment has fixed number of problems and is used for examinations.

Testing Generator (TG)**:** The TG transfers the tagged testing document, saved in the testing database, to HTML format for browser’s showing. The testing variables are generated randomly. Therefore, the same testing problem will show in different format for different users. The answer program will calculate the correct answer for each execution. The diagnosis program will trace the processes of user’s operations and find users’ misconceptions. The processes of the TG are shown as the following graphic.

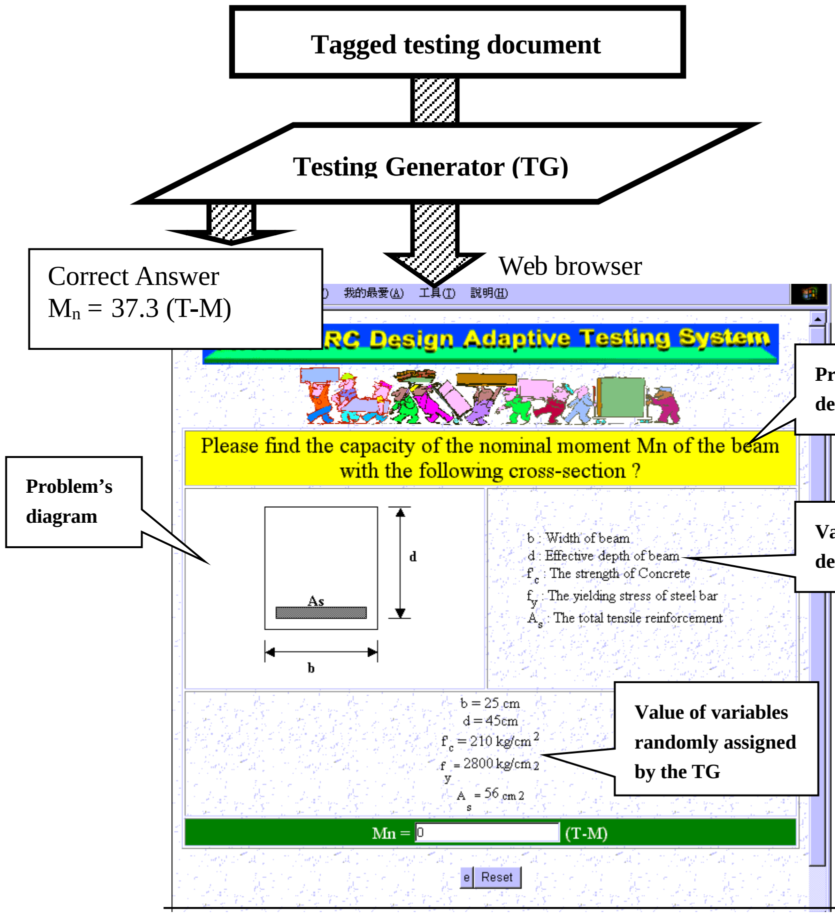


Figure 2. The output of testing generator

Adaptive Manager (AM): The AM is used to find the adaptive testing problems to students. The AM searches the proper tests that are depended on students' level and assigns to students automatically. The AM is suitable for students' practicing and evaluating students' level, but is not for examination.

Diagnosis /Answering Evaluator (DAE)**:** The DAE can find the problem's correct answer if the answering program is assigned and explore both users' mistakes and correct answer if the diagnosis program is assigned. The detail description of the DAE will be discussed in the next session.

Level Manager (LM): The LM updates both students' achievement level according to the difficulty levels of all problems they correctly answered and the difficulty levels of problems.

Testing Database (TDB)**:** The TDB saves testing problems in tagged format. It also includes the data of difficulty levels of problems and the index information that problems belong to.

Users Database (UDB)**:** The UDB saves students' basic data, testing diagnosis results and achievement levels.

The numbers in Figure 1 are the order of AWRC executions. All AWRC executions can be divided into the following four stages.

(1) Teachers create testing problems. AWRC tags problems in XML format and saves problems into database system. Number (1)~(3)
(2) Teachers assign the mode of examinations. Two distinct modes (standard and adaptive) can be selected. Number (4)
(3) Examination execution: problems generation, test delivery, diagnosis/answering and record keeping. Number (4)~(8)
(4) Level management: update data of students' achievement levels and problems' difficulty levels. Number (9)~(12)

3. The Diagnosis System of AWRC

The diagnosis function of AWRC is supported by the Diagnosis /Answering Evaluator (DAE). The procedures of developing the DAE are shown as the following three stages.

(1) Analyze problem's knowledge space and represent it as an algorithm hierarchy. For the example of the problem (RC beam analysis) is shown in Figure 2. Its knowledge space can be shown as in the Figure 3. The values of b, d, f_y, f_c' and A_s are randomly generated by the Testing Generator. The knowledge space includes the total possible executing calculations, adjustments and branches of a problem.

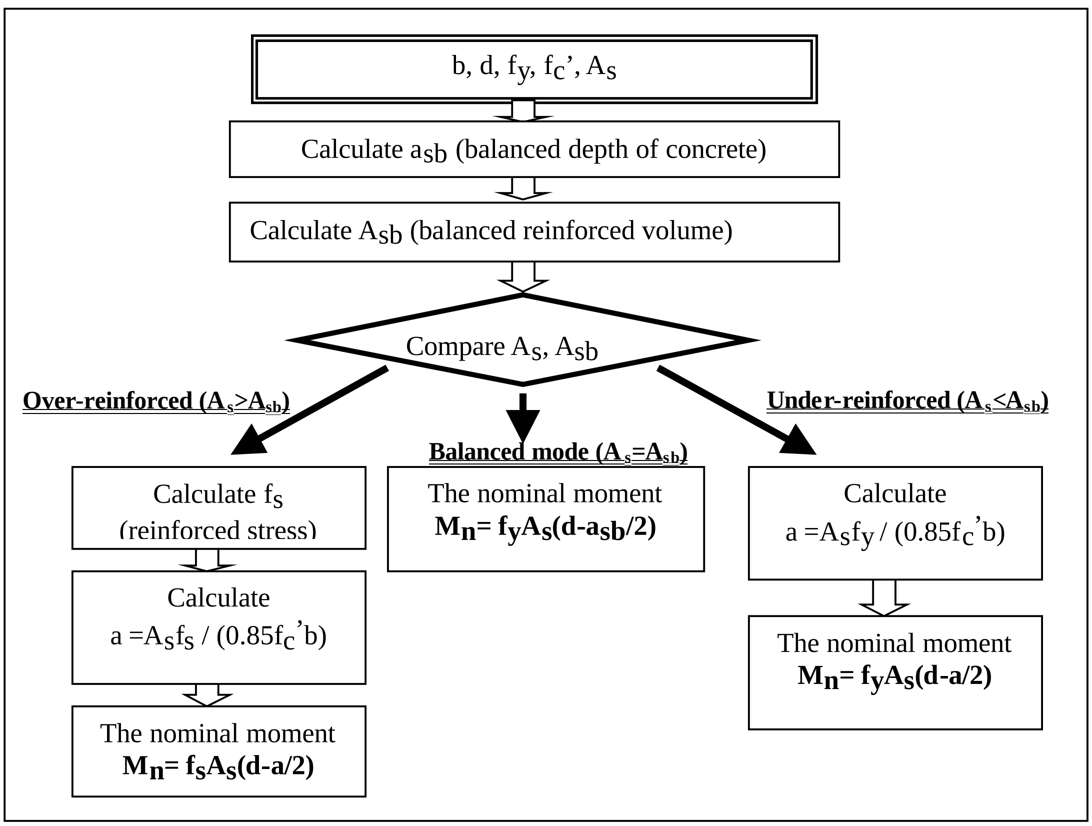


Figure 3. Algorithm hierarchy of a knowledge space

(2) Follows the above algorithm to develop the answering program. Answering program can find the problem's correct answer for different input variables and show the calculation procedures of the answer.
(3) Follows the above algorithm to develop the diagnosis program. If the diagnosis is checked the problem will ask student to answer a serial questions but fill-the-gap question. For the example is shown in figure 2. After the problem description, student will be asked to answer the progressive questions shown in Figure 4 as follows:

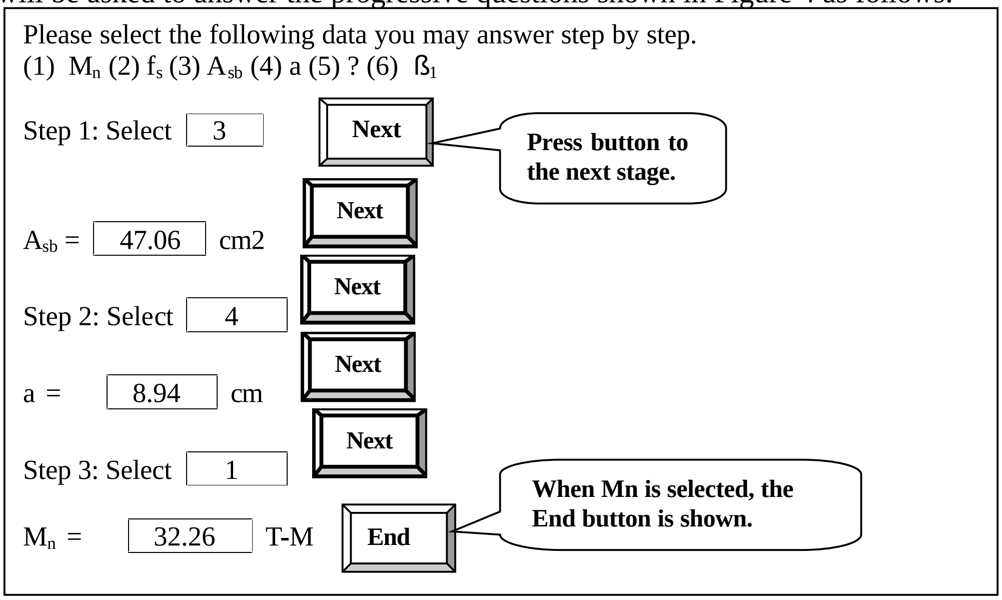


Figure 4. The progressive problems in the diagnosis mode

The diagnosis program traces student's inputs to find out mistakes. We set up the tolerance to verify the answer is 5%~10% difference by step number. The diagnosis program can recognize which steps student makes wrong calculation. Teachers may modify their teaching strategies by those statistics from diagnosis data.

4. The Adaptive Function and Level Manager

Both the achievement level of a student and the difficulty level of a question are divided into ten grades from 1 (the lowest) to 10 (the highest) in AWRC. The logic of the adaptive function for a domain is shown in Figure 5.

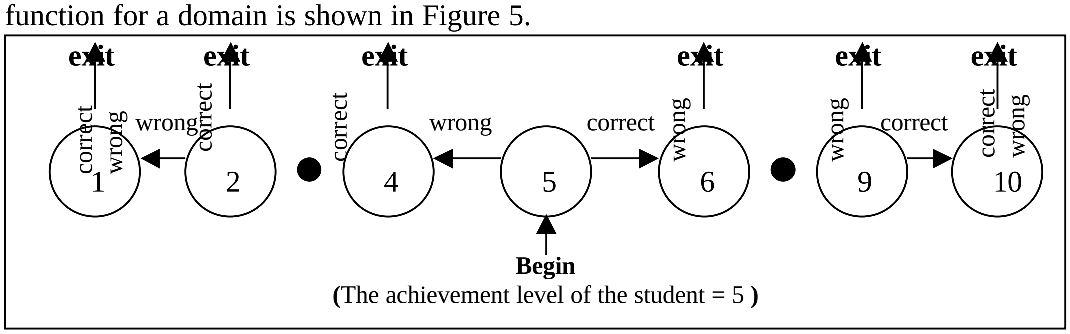


Figure 5. The logic of the adaptive function for a domain

The adaptive manager first sends the problem with difficulty level is equal to the achievement level of the student. The progressive problem is sent following the logic of the adaptive function. AWRC saves over ten problems with different difficulty levels in each domain.

Now, we describe the functions of calculating both the achievement level of a student and the difficulty level of a problem used in the level manager. The definition of calculating functions is refer to the Mine's studies [1]. The achievement level of a student is recursively calculated with the difficulty level of a problem at stated periods and vice versa. They are defined as follows:

$$s_{i.j} = \begin{cases} 1 & \text{if } m_{si,t} = 0 \\ \dfrac{1}{m_{si,t}} \sum_{j=1}^{m_{si,t}} q_{j,t} \bullet \sigma_{i,j} & \text{Otherwise} \end{cases} \qquad s_{i.j} = \begin{cases} 1 & \text{if } S_i \text{ answered } Q_j \text{ correctly} \\ 0 & \text{Otherwise} \end{cases}$$

$$q_{j,t} = \begin{cases} q_{j,t-1} + \dfrac{\sum_{i=1}^{m_{qj,T}} \left| s_{i,\tau} - q_{j,t-1} \right| \bullet \xi_{i,j}}{\sum_{i=1}^{m_{qj,T}} \left| \xi_{i,j} \right|} & \text{If } \sum_{i=1}^{m_{qj,T}} \left| \xi_{i,j} \right| \neq 0 \\ q_{i,t-1} & \text{Otherwise} \end{cases}$$

$$?_{i,j} = \begin{cases} -1 & s_{i,t} \text{ is less than } q_{j,t-1} \text{ and } S_i \text{ answered } Q_j \text{ correctly} \\ 1 & s_{i,t} \text{ is greater than } q_{j,t-1} \text{ and } S_i \text{ answered } Q_j \text{ wrongly} \\ 0 & \text{Otherwise} \end{cases}$$

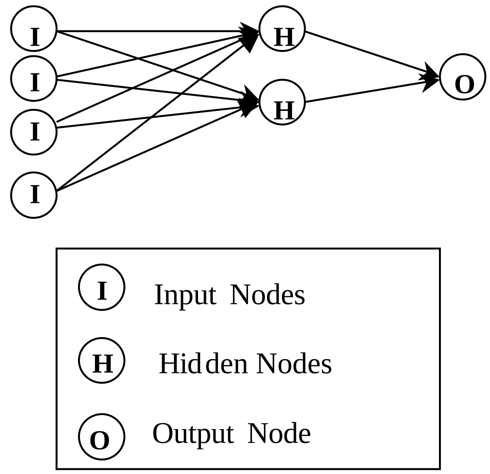
In which

$s_{i,t}$: the achievement level of student *Si* at time t, where is between 1 and 10.
$q_{j,t}$: the difficulty level of problem Q_j at time t, where is between 1 and 10.
$m_{si,t}$: the number of questions that S_i tried at t.
t : the latest time such that S_i tried to answer Q_j.
T : the set of t.
$m_{qj,T}$ the total number of students who tried Q_j in T.

The Level Manager automatically adjusts the values of the $s_{i,t}$ and $q_{j,t}$ by using the calculating functions, but the initial difficulty level of the problem Q_j ($q_{j,0}$) and the initial achievement level of the student S_i ($s_{i,0}$) must be assigned by teachers. We refer to the students' performance of the structure analysis course as the initial achievement level of the student in AWRC. We purpose an algorithm by "skills" to assign the initial difficulty level of problem and this algorithm is discussed in the next session.

5. The Algorithm by "Skills"

The "skills" algorithm could evaluate the initial difficulty level of the RC problems. The neural network is used as the modeling methodology of the "skills" algorithm. The "skills" algorithm can be presented by the simple neural network and shown in the following graphic.



In which:

Figure 6. The neural network system of the "skills" algorithm

There are one output node and four input nodes for this "skills" algorithm. The output node obtains the value, from 1 to 10, is the difficulty level of a problem. The input nodes are the descriptions of "skills" to present how to solve a problem. The definitions of input nodes "skills" are listed as follows:

(1) The first input node represents the total number of possible executing stages of a problem. For the example of the problem shown in Fig. 3, the value of the first input node is 8. There are total 8 executing stages for the problem of RC beam analysis.
(2) The second input node stands for the total branches of a problem. There are 3 branches, 3 possible results, for the problem shown in Fig. 3.
(3) The third input node is the number of executing stages to solve a problem. It is 3 to the balanced mode, 4 to the under-reinforced mode and 5 to the over-reinforced mode for the problem shown in Fig. 3.
(4) The last input node is the number of determinant to solve a problem. For the problem shown in Fig. 3, the number of determinant is 1. We need to compare A_{sb} with A_s to check the possible failure mode.

Before the process of this neural network, we have to prepare data for training and testing [10]. In Taiwan, a national examination for entrance technical universities will hold for each year. There are more than two thousands of students take this examination in the civil engineering field and the course of RC design is one of the testing courses. The bulk problems in AWRC are collected from this national entrance examination. We had collected more than 200 problems within 7 years examinations. Therefore, we use the examinational results for the training and testing data. The predictions of the neural network after training are good fit to testing results.

6. Conclusions

A web-based reinforced concrete design adaptive testing system (AWRC) had been developed. AWRC has a test bank with over 200 testing problems. Each problem owns the data of the initial difficulty level. AWRC also includes a testing generator can develop adaptive testing problems to each student's achievement level, a diagnosis system may find students' misconception and a level manager can calculate the achievement level of a student and the difficulty level of a problem. We believe the AWRC is helpful to improve the teaching efficacy.

We now are concentrating the web-based course development. The reinforced concrete design is chosen as the topic course. This web-based course will contain tagged contents for data communication and understanding and provide different feedbacks as well as study materials based on the user's responses.

Acknowledgements

The authors gratefully acknowledge the funding received from the National Science Foundation, ROC (Taiwan) Government (NSF-90-2516-S-032-001) to support this work.

Bibliography

1. Tsunenori Mine, "The Design and Implementation of Automatic Exercise Generator with Tagged Documents based on the Intelligence of Students: AEGIS" Proc. International Conference on Computer in Education, pp. 651~658, Taipei, 2000.

3. Yueh-Chun Shih, "Design and Evaluation of Constructivist Web-based Instructional Systems", Proc. International Conference on Computer in Education, pp. 500~504, Taipei, 2000.

4. Ana Vidaurre, "Interactive Tutorial for Training Physics Students in Their First Year A University" Proc. International Conference on Engineering Education, Taipei, 2000.

5. James N. Craddock, "Web based laboratory manuals in Civil Engineering" Proc. International Conference on Engineering Education, Taipei, 2000.

6. Stanislaw Majewski, "Integrated project system and supervised industrial placement -Essential cores of Civil Engineering Education" Proc. International Conference on Engineering Education, Taipei, 2000.

7. Barry M. Lunt, "Systems and automation education through Web-based Labs" Proc. International Conference on Engineering Education, Taipei, 2000.

8. P.Y. Hsieh, "Four easy pieces: development systems for knowledge-based generative instruction", International Journal of Artificial Intelligence in Education, vol. 10, no. 1, pp.1-45, 1999.

9. Wang and Salmon, Reinforced Concrete Design, Willy, 1994.

10. Christian Meyer, Design of Concrete Structures, Prentice Hall, 1996.

11. Giarratano and Riley, Expert Systems, PWS Publishing Company, 1996.

12. Building Code and Commentary (Metric) ACI 318M-95, 1995.

Biographies

YU-HUR CHOU
Yu-Hur Chou is a Lecturer of Civil Engineering at Tung -Nan Institute of Technology. He received a M.S. degree in Structural Engineering and the second M.S. degree in Computer Science at the University of Connecticut. He is currently a PhD student in Civil Engineering at National Taiwan University.

SHANG-HSIEN HSIEH
Shang-Hsien Hsieh is an Associate Professor of Civil Engineering at National Taiwan University . Dr. Hsieh received an M.S. and a Ph.D. in Civil Engineering at Cornell University. His researching interests are Parallel Computation, Finite Element Analysis an d Computer-Aided in Engineering.