Int. J. Engng Ed. Vol. 25, No. 4, pp. 723–728, 2009
Printed in Great Britain.

0949-149X/91 $3.00+0.00

Teaching Problem Solving in Engineering Education: Expert Systems Construction*

WEI-FAN CHEN
College of Information Sciences and Technology, The Pennsylvania State University, Wilkes-Barre Campus, USA. E-mail: weifan@psu.edu

This study presents a pedagogical model of constructing an expert system knowledge base for an undergraduate computer networking class. This model included: identifying a suitable problem, defining the problem domain, specifying goals or solutions, specifying problem attributes and values, generating rules and examples, and selecting the right tool. In addition, the effectiveness of this problem-based learning approach was verified by an experimental study. Results indicated that students in a student-created expert system group achieved significantly higher scores than a system-provided expert system group ($F = 5.042$, $p < .05$) when they were solving story problems. Creating such an authentic learning environment by asking students to develop their own knowledge base is the main theme of the study. The same instructional technology can be applied to other disciplines that focus on teaching engineering problem solving.

Keywords: expert systems; problem solving; experimental design; computer networking; learning assessment

INTRODUCTION

EDUCATORS charge that there exists an incongruence between educational goals and instructional methods. Students are taught dry facts, but are expected to be able to apply higher order thinking skills to solve complex problems [1, 2]. Assuming that the central goal of education is to teach people to think critically and to become good problem solvers, employing effective instructional strategies and technologies for facilitating students' critical thinking and problem solving skills becomes vital. In engineering education, students are trained to be good problem solvers. College engineering programs are designed to help students to identify, formulate, and solve engineering problems [3]. While existing pedagogical models for teaching problem solving are available to adopt [4, 5], in reality, classroom teachers still heavily rely on textbook questions that generally do not elicit students' relevant problem solving skills [6]. This study presents a pedagogical model of constructing an expert system knowledge base for an undergraduate computer networking class. The problem-based learning approach produces a significantly better learning outcome than a traditional approach, which is verified by an experimental study.

REVIEW OF LITERATURE

An expert system, an intelligent computer program, is often used to assist people in making decisions in a problem solving environment. Its applications include job aids, corporate training, and classroom tutoring [7–9]. Expert systems also can be used to model human mental structures and cognitive processes [10, 11]. Edward. A. Feigenbaum, widely regarded as the pioneer of expert systems, defined an expert system as " . . .a computer program that has built into it the knowledge and capability that will allow it to operate at the expert's level" [12, p. 23]. In other words, an expert system program is designed to mimic the way human experts solve problems. For example, one goes to an expert (say a medical doctor) about a problem, the expert diagnoses symptoms, recalls facts and rules from memory and experiences, and eventually arrives at a decision. This process, when computerized appropriately, is an expert system.

A rule-based expert system is typically composed of three major components: (1) the human–computer interface capable of handling input and providing output to the system, (2) a knowledge base, and (3) an inference engine. The human–computer interface is the interface used to create the knowledge structure on which the computer program makes decisions. The knowledge base contains diverse knowledge which can be brought to bear on a given task. It includes facts and rules that can be represented by using IF-THEN formats. The IF states a condition and the THEN an action. The inference engine is built into the computer programming language so that when the expert system has been programmed with the facts and rules from the knowledge base the inference engine automatically examines the relationship within these structures and produces a decision.

* Accepted 24 February 2009.

Research in the last two decades indicated that requiring students to create their own rule-based expert systems (either paper-based or computer-based) helped them to understand domain-specific knowledge. Starfield *et al.* [13] found that paper-based expert systems construction as a pedagogical approach facilitated college students in learning engineering concepts. Jonassen and Wang [10] reported a case study where students, using inexpensive and easy-to-use expert system shells, were able to develop simulations of cognitive processes. Marra and Jonassen [11] investigated the effect of students building semantic networks using expert systems. They found that students who constructed semantic network created expert systems with significantly more rules than a control group. Hatzilygeroudis and Prentzas [14] used a hybrid rule-based approach to develop an expert system to enhance students' knowledge acquisition and their capability to make decisions. Kassim, Kazi and Ranganath [15] developed a Web-based learning system using Java client–server architecture to improve the engineering students' learning process. The system can automatically generate problems to check the students' solutions to specific digital logic questions and keep track of their progress. Buts, Duarte and Miller [16] designed an expert system to assist electrical engineering undergraduate students in learning circuit design. This system creates real-life engineering scenarios for the students to solve problems and make decisions.

Based on previous research findings, creating rule-based expert systems as an instructional technology engages students' higher-order thinking skills such as analytical reasoning, synthesis of knowledge, meta-cognitive awareness and self-regulation, and thus further improves their problem solving skills.

EXPERT SYSTEM DEVELOPMENT

To adopt rule-based expert systems as an instructional technology in engineering classrooms, an undergraduate computer networking class is used to implement the development process of the expert system by using a six-stage model, which is explained as follows.

IST 220: Networking and Telecommunications is offered in the College of Information Sciences and Technology at The Pennsylvania State University (Penn State), USA. It is an introductory course focusing on a foundational knowledge of the telecommunications and networking industry, as well as the basic concepts inherent to the application of data communications and computer networks. Given the large and rapidly changing landscape of operating systems and home networking vendors, the expert system cannot be expected to give detailed instructions for every possible configuration. Users are queried for basic information, and based on these inputs, the expert system returns a network topology recommendation, along with specific checklists, basic instructions and links to the Web that can be used to implement the topology. The expert system uses a JavaScript-enabled Web page to obtain user inputs and dynamically generate HTML codes to render the recommended solution.

Grabinger [17] developed a six-stage model for creating such an expert system. These stages include:

1. Identifying a suitable problem
2. Defining the problem domain
3. Specifying goals or solutions
4. Specifying problem attributes and values
5. Generating rules and examples
6. Selecting the right tool.

This framework serves as a guideline for developing the networking expert system. The process of creating such a system is explained as follows.

1. Identifying a suitable problem

In this stage, the proper problem for developing an expert system is a problem that requires a specific solution from among several selections. In a home network expert system, the problem that requires an appropriate network topology recommendation from among a collection of situations (e.g. router type, Internet connection type) is qualified to be a good problem choice for building an expert system.

2. Defining the problem domain

The objective of defining problem domain is to acquire the domain knowledge which an expert uses to make decisions. Before obtaining the domain knowledge, it is necessary to identify the types of problems first. One question needs to be answered: "Is this a well-structured or an ill-structured problem?" In this case, the problem that requires students to choose an appropriate network topology is obviously a well-structured problem, which is appropriate for developing expert systems. Therefore, the domain knowledge can be acquired from subject matter experts.

3. Specifying goals or solutions

From the system inputs and network topologies, eight potential roles for the network PCs emerge (Table 1).

Any PC assigned the role of Gateway must also be assigned a Connection Manager role based on the connection type selected by the user. The reverse is not true: in a *1 PC, No Router* network, the PC simply manages the connection because there are no clients that need gateway services.

4. Specifying problem attributes and values

The home networking expert system uses five attributes to determine a recommended solution. Table 2 provides the name and a brief description of each attribute.

Table 1. PC role and descriptions

PC role and abbreviation	Description
Windows Dial-Up Connection Manager (WINDU)	Establishes Dial-Up Connection with ISP
Windows DSL/Cable Connection Manager (WINDSL)	Establishes DSL/Cable Connection with ISP
Linux Dial-Up Connection Manager (LINUXDU)	Establishes Dial-Up Connection with ISP
Linux DSL/Cable Connection Manager (LINUXDSL)	Establishes DSL/Cable Connection with ISP
Windows Gateway (WINGATE)	Acts as a Network Gateway for Client PCs
Linux Gateway (LINUXGATE)	Acts as a Network Gateway for Client PCs
Windows Client (WINCLIENT)	Relies on Gateway or Router for Internet connection
Linux Client (LINUXCLIENT)	Relies on Gateway or Router for Internet connection

Table 2. Attributes and values

Attributes	Values
connectionType	The type of service connecting the home network to the Internet. Allowed inputs include *DIAL-UP* or *DSL/CABLE*.
pcNumber	The number of PCs in the home network. Allowed inputs are *1*, *2* or *3*+.
Router	This parameter indicates the use of a dedicated router/firewall to manage the Internet connection. This parameter is not used if *connectionType = DIAL-UP*. Allowed inputs are *YES* or *NO*.
Gateway	The operating system of the PC controlling the Internet connection. This parameter is not used if *router = YES*. Allowed values are *WINDOWS* or *LINUX*.
Clients	The operating system(s) of the remaining PCs in the home network. Allowed values are *WINDOWS, LINUX* or *BOTH*. This parameter is not used if *pcNumber = 1*. The *BOTH* value is not allowed if *pcNumber = 2* and *router = NO*.

Table 3. Topology and descriptions

Topology abbreviation	Description
1 PC, No Router (1NR)	One PC manages its Internet connection to the ISP.
2 PCs, No Router (2NR)	One PC manages the Internet connection as a gateway for a client PC.
3+ PCs, No Router (3NR)	One PC manages the Internet connection and acts as a gateway for multiple client PCs. A simple network hub provides inter-connectivity between the PCs
Router (RTR)	A dedicated firewall/router manages the Internet connection and acts as a gateway for one or more client PCs.

Based upon system inputs, the expert system recommends one of four possible network topologies (Table 3).

5. Generating rules and examples

Rules are a series of "IF . . .THEN" statements that describe the means of reaching a specific decision in a narrative form. Based on the possible solutions and the problem attributes and values listed above, the rules can be designed as the following form.

IF (pcNumber = 2 OR pcNumber = 3+) AND (router == YES) THEN

{
  USE **RTR**;
  IF (client = WINDOWS) USE **WINCLIENT**;
  IF (client = LINUX) USE **LINUXCLIENT**;
  IF (client = BOTH) USE **WINCLIENT** AND
    USE **LINUXCLIENT**;
}

6. Selecting the right tool

Once the network topology and PC roles are defined, the expert system can provide specific HTML output based on the input parameters.

EXPERT SYSTEM EVALUATION

To evaluate the quality of the expert system construction as an instructional technology, an experimental study was designed to explore two different uses of expert systems in higher education. The first instructional treatment asked students to create an expert system knowledge base by using IF-THEN rules. The second treatment asked students to use a provided expert system program as a consulting tool to solve problems. Based upon the purpose of this study, one major research null hypothesis may be drawn as follows: there were no statistically significant differences in students' learning achievements when they received two different instructional treatments: (1) student-created expert system rules; and (2) a system-provided expert system program.

To avoid potential threats to internal validity of the proposed instructional technology in the computer networking class [18], the experiment selected one independent unit of instruction as experimental material that was unrelated to the content of the computer networking course. The selected class for conducting the experiment was an introductory statistics class at Penn State.

Subjects

Twenty-two undergraduate students in an introductory statistics class (STAT 200) at Penn State participated in the study. They were 19–21 year old sophomores from a wide range of majors who took this course to fulfill General Education requirement—Quantification. They volunteered to participate in the study and were given extra points from the course instructor.

Instructional materials

One statistical topic in STAT 200 was chosen to conduct the experimental study. It was to choose appropriate statistical tests (t-test, Z-test, etc.) based on certain problem situations that included variable type, the number of samples, sample size and variance. For example, if a given problem is to deal with the mean score within one sample, and its variance is known, then a suggested statistical test is to use z-test about one mean.

Independent variables

One independent variable was examined in this study: type of expert system usage. Two groups (the student-created expert system group versus the system-provided expert system group) were randomly assigned to the variable of instructional treatment.

Dependent variables

The measurement instrument in this experiment was a post-test for assessing students' learning achievement after they receive different instructional treatments. This post-test included two separate problem solving types of questions. Each had ten multiple-choice items. The following details the two sub-tests in the post-test.

Part 1: Procedural problem test (10 items): This procedural problem test measures students' procedural knowledge by asking them to identify an appropriate test statistic in a given scenario. For example, we want to know whether the averages in two populations are the same or not, so samples of size 5 are taken from each population. What test procedure should be used to answer the question?

Part 2: Story problem test (10 items): This story problem test not only asked students to demonstrate their understanding of statistical procedural knowledge but also expected them to solve a real-world statistical problem. It involved more complex critical thinking and problem solving skills. For example, in 1975, 9.0% of all physicians in the U.S. were women. A journalist thought that the percentage of physicians who are women in 1996 had increased since 1975, so she obtained a random sample of 400 physicians (in 1996) and found that 64 (or 16%) were women. At the 0.05 level, can she confidently assert that the percentage of physicians who are women has increased? Please test.

Instructional treatments

Students in the student-created expert system group followed the aforementioned six-stage model to develop a statistics expert system using IF-THEN rules based on the selections of statistical methods in different situations. On the other hand, students in the system-provided expert system group solved statistical problems by using an expert system program already developed by the course instructor and the author. After receiving the two different instructional treatments for both groups, the students took an achievement post-test.

Experiment design

This study investigated the effect of a variety of types of instructional treatments (student-created expert system versus system-provided expert system) on learning achievement for undergraduate students. The experimental design is one factor with two levels. Since the dependent variables are typically related statistically and conceptually, and the statistical correlation should be in the range of a low to moderate level, a Multivariate Analysis of Variance (MANOVA) was performed to analyze the results. Multivariate analysis of variance in the Statistical Package for the Social Sciences (SPSS) was used to examine the main effects of categorical variables on multiple interval/ratio dependent variables. An alpha level of 0.05 was set for analyzing the significant difference of the testing hypothesis. All statistical analyses were conducted using SPSS for Windows.

Results and discussions

Table 4 reports the results of MANOVA for the effect of the studied independent variable on the dependent groups.

According Cronk [19], Wilks' Lambda determines whether an independent variable has any effect on dependent variables. Table 4 indicated that a significant effect of instructional treatments was found (Lambda = 0.718, $p < 0.05$). To further investigate the effect of the instructional treatments on dependent variables, univariate analyses (Table 5) revealed that the independent variable had a significant main effect on dependent variables in the story problem test ($F = 5.042$, $p < 0.05$), but not in the procedural problem test ($F = 0.010$, $p > 0.05$). In other words, students in the student-created expert systems group achieved significantly higher scores (Mean=26.82) than the system-provided expert system group (Mean=18.18) when they

Table 4. Results of multivariate tests

Effect	Wilks' Lambda	F	*P*
Intercept	0.112	75.693	0.000*
Treatment groups	0.718	3.726	0.043*

* Significant at 0.05 level.

Table 5. Tests of between-subjects effects

Source	Dependent variables	Type III sum of squares	Df	F	Sig.
Groups	Part 1	1.136	1	0.010	0.920
	Part 2	410.227	1	5.042	0.036*
Error	Part 1	2222.727	20		
	Part 2	1627.273	20		
Total	Part 1	15725.000	22		
	Part 2	13175.000	22		

* Significant at 0.05 level.

were solving story problems. However, no significant differences were found between the two groups when they solved procedural problems.

The procedural problems in the post-test can be viewed as well-structured problems; the story problems as ill-structured problems [20]. The results of the study confirms Jonassen's comments on creating problem solving environment that solving a well-structured problem needs instruction that facilitates students' information processing, while solving an ill-structured problem requires instructional designers to embed instruction in an authentic learning context [4]. It is also in accordance with Jonassen's finding [21] that merely presenting knowledge structure does not automatically improve acquisition of structural knowledge. The constructive processing of the learner results in effective learning. The student-created expert system group did learn more than the system-provided group when they were asked to solve complex story problems.

While solving a procedural problem can be taught by providing direct instruction, such as the system-provided expert system program in the study, as students solve a story problem that involves higher critical thinking and problem solving skills, an authentic problem solving environment should be designed for them to apply their learned knowledge to solve a real-world problem.

CONCLUSIONS

This study presented a pedagogical model of constructing an expert system knowledge base for an undergraduate computer networking class. An experiment using a different subject matter was conducted to elicit the quality of the proposed instructional technology. Creating an authentic learning environment by asking students to develop their own knowledge base is the main theme of the study. The same instructional technology can be applied to other disciplines that focus on teaching engineering problem solving.

Although the study indicated that students in the student-created expert system group obtained better learning outcomes than the student-provided expert system group, there might be other extraneous factors that caused this difference. These factors might include time spent on the student-created expert system, randomization of the experiment, sample size of each group, and other unidentified reasons. In addition, factors jeopardizing the internal validity of experimental designs should be considered, which is cautioned by Campbell and Stanley [18] in their classic work on experimental research design. They include history, maturation, testing, instrumentation, statistical regression, bias in selecting sample subjects, experimental mortality, and selection–maturation interaction. Therefore, future research should take the above factors into consideration and continue to explore students' cognitive processes and individual differences when they solve varied types of engineering problems.

While teaching engineering problem solving using instructional technologies may be manipulated to influence students' learning achievement in a positive way, particular attention should be given to guidelines derived from teaching problem solving and experimental methodology, as well as consideration of learner characteristics and styles. Only by initiating a systematic investigation where instructional technologies for teaching problem solving are judiciously manipulated to determine their relative effectiveness and efficiency for facilitating specifically designated learning objectives will the true potential inherent in teaching engineering problem solving be realized.

REFERENCES

1. R. M. Gagne, Learnable aspects of problem solving, *Educational Psychologist*, **4**(1), (1980), pp. 84–92.
2. D. R. Woods, How might I teach problem solving, In J. E. Stice (ed.), *Developing Critical Thinking and Problem solving Abilities*, Jossey-Bass, N.J., (1987).
3. ABET Criterion for Accrediting Engineering Programs, Retrieved February 9, 2009 from http://www.abet.org/Linked%20Documents-UPDATE/Criteria%20and%20PP/E001%2007-08%20EAC%20Criteria%2011-15-06.pdf

4. D. H. Jonassen, Toward a design theory of problem solving, *Educational Technology Research and Development*, **48**(4), (2000), pp. 63–85.
5. D. R. Woods, An evidence-based strategy for problem solving, *Journal of Engineering Education*, **89**(4), (2000), pp. 443–459.
6. M. Prince and B. Hoyt, Helping students make the transition from novice to expert problem-solvers, *Proceedings of the 32nd ASEE/IEEE Frontiers in Education Conference*, pp. F2A7–11, IEEE, (2002).
7. W. Cheung, L. C. Leung and C. F. Tam, An intelligent decision support system for service network planning, *Decision Support Systems*, **39**(3), (2005), pp. 415–428.
8. G. V. Post and T. G. Whisenand, An expert system helps students learn database design, *Journal of Innovative Education*, **3**(2), (2005), pp. 273–293.
9. S. Robinson, T. Alifantis, J. S. Edwards, J. Ladbrook and A. Waller, Knowledge-based improvement: Simulation and artificial intelligence for identifying and improving human decision-making in an operations system, *Journal of the Operational Research Society*, **56**(8), (2005), pp. 912–921.
10. D. H. Jonassen and S. Wang, Using expert systems to build cognitive simulations, *Journal of Educational Computing Research*, **28**(1), (2003), pp. 1–13.
11. R. M. Marra and D. H. Jonassen, Transfer effects of semantic networks on expert systems: Mindtools at work, *Journal of Educational Computing Research,* **26**(1), (2002), pp. 1–23.
12. E. A. Feigenbaum and P. McCorduck, *The Fifth Generation*, Reading, Addison-Wesley, MA, (1983).
13. A. M. Starfield, K. L. Butala, M. M. England and K.A. Smith, Mastering engineering concepts by building an expert system, *Engineering Education*, **74**, (1983), pp. 104–107.
14. I. Hatzilygeroudis and J. Prentzas, Using a hybrid rule-based approach in developing an intelligent tutoring system with knowledge acquisition and update capabilities, *Expert Systems with Applications*, **26**(4), (2004), pp. 477–492.
15. A. A. Kassim, S. A. Kazi and S. Ranganath, A Web-based intelligent learning environment for digital systems, *International Journal of Engineering Education*, **20**(1), (2004), pp. 13–23.
16. B. P. Butz, M. Duarte, and S. M. Miller, An intelligent tutoring system for circuit analysis, *IEEE Transactions on Education*, **49**(2), (2006), pp. 216– 223.
17. R. S. Grabinger, B. W. Wilson, and D. H. Jonassen, *Building Expert Systems in Training and Education*, Praeger Publishers, Westport, CT, (1990).
18. D. T. Campbell and J. C. Stanley, Experimental and quasi-experimental designs for research on teaching, In N. L. Gage (ed.) *Handbook of Research on Teaching*, pp. 171–246, Rand McNally, Chicago, (1963).
19. B. C. Cronk, *How to Use SPSS,* Pyrczak Publishing, Glendale, CA, (1999).
20. D. H. Jonassen, Instructional design models for well-structured and ill-structured problem solving learning, *Educational Technology Research and Development*, **45**(1), (1997), pp. 65–94.
21. D. H. Jonassen, Effects of semantically structured hypertext knowledge bases on users' knowledge structures, In C. McKnight, A. Dillon and J. Richardson (eds.), *Hypertext: A Psychological Perspective,* Horwood, Chichester, UK, (1993). pp. 153–168.

Wei-Fan Chen received his BS in Information and Computer Engineering in 1993, from Chung Yuan Christian University, Taiwan. He received his MEd and Ph.D. in Instructional Systems in 1999 and 2002, respectively, both from The Pennsylvania State University, USA. He is currently an Assistant Professor of Information Sciences and Technology at The Pennsylvania State University. Dr. Chen's research and teaching interests include cognitive and information sciences and technology as related to learning. He studies human–computer interaction, especially for cognitive learning for undergraduate students. Other interests include the use of innovative instructional technologies as a teaching method and the use of artificial intelligence systems in cognitive learning.