Int. J. Engng Ed. Vol. 20, No. 1, pp. 13–23, 2004
Printed in Great Britain.

0949-149X/91 $3.00+0.00

A Web-based Intelligent Learning Environment for Digital Systems*

ASHRAF A. KASSIM, SABBIR AHMED KAZI and SURENDRA RANGANATH
Department of Electrical and Computer Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260. E-mail: ashraf@nus.edu.sg

Many recent web-based teaching and learning environments that involve the use of fixed sets of problems are unable to accurately measure a student's understanding about the subject matter. Such systems do not contribute significantly to the learning process. A good pedagogical framework is needed to develop effective web-based teaching and learning systems. This paper presents just such an effective system, which is designed for a specific engineering subject, Digital Logic Systems Design. Our system is called Web-based Intelligent Learning Environment for Digital System or WILEDS. WILEDS has been built on a robust framework of intelligent tutoring systems realised on a three-tier Java client-server architecture. It comprises five components: a communications module, a pedagogical module, a student model, an expert model and the domain knowledge. WILEDS incorporates the entire process needed to assess students, starting from automatically generating problems to checking the students' solutions and keeping track of their progress.

INTRODUCTION

HIGHLY INTERACTIVE learning environments that are independent of time and space constraints can now be realised through computer and Internet technologies. Distance learning is becoming important for several reasons. More people are interested in higher education and educational institutions have been increasingly turning to the Internet to deliver their services to this group of people [1].

Many of these institutions also supplement regular classroom teaching with additional web-based material. Key features of computer and Internet-based learning environments [2] include: interactivity, global accessibility, availability of online resources, learner-controlled pace, convenience, non-discriminatory, cost-effective, collaborative learning, online evaluations, etc. Individual (one-on-one) tutoring is the most effective mode of teaching [3] but it poses financial and logistical problems. Also, in engineering education, a major focus is on developing problem solving skills. Although doing exercises given in textbooks enable students to gain insight into how to solve problems, this lacks interaction and feedback. Web-based intelligent learning environments have the potential of bringing the individual tutoring experience to a broader audience.

Our comprehensive Digital Logic Systems Design curriculum covers topics from traditional logic design to modeling and design of digital systems using hardware description languages. The curriculum involves a series of laboratory sessions where students are exposed to design and realization of digital circuits beginning with the use of small-scale integration (SSI) logic devices to complex programmable logic devices (CPLDs) and application-specific ICs (ASICs) using hardware description languages. To complement the learning of key concepts in the curriculum, we developed a web-based intelligent learning environment for digital systems or WILEDS [4]. WILEDS utilizes information, multimedia and Internet technologies to provide individualized attention and enhance student learning and problem-solving skills. WILEDS provides students with a unique set of problems, each time it is used and can be accessed easily anywhere through an Internet browser that supports Java. In this paper, we present the internal model and system architecture of WILEDS.

TECHNOLOGY IN TEACHING AND LEARNING

The approach used for designing effective web-based learning environments should be based on an understanding how learning occurs. Learning is a very complex psychological process that refers to the way in which an individual processes information. Cognitive or learning styles can be classified into two major categories: behaviourism and constructivism. **Behaviourism** [5] views learning as the acquisition of new behavior and discounts mental activities. **Constructivism**, the dominant learning theory, claims that knowledge is actively constructed by learners based on their experiences [6] and not passively transmitted, say through listening to lectures or reading textbooks. The constructivism model assumes that conceptual

* Accepted 30 September 2003.

growth comes from the negotiation of meaning, the sharing of multiple perspectives and the change of our internal representations through collaborative learning [7].

In an instructional situation there are usually at least three entities: the human teacher, the student and the domain knowledge that the student is expected to obtain. Computer Aided Learning (CAL) introduces a fourth entity, namely the computer, to the teaching and learning process. Traditional CAL resources primarily consist of tutorials that are essentially computer-based forms of 'programmed instructions' drawing heavily on behaviourist views. With the advent of Artificial Intelligence (AI), a newer breed of CAL systems called Intelligent Tutoring Systems (ITS) [8] emerged. ITS instruct students in an intelligent way and typically consists of an internal model of the expert knowledge, the learner's current knowledge and the pedagogical principles. As learning progresses, the model of the learner's knowledge and the model of the expert's knowledge are compared, and using AI, the sequence of instructions is dynamically generated to suit the needs of the learner [9].

The ITS architecture [10], as shown in Fig. 1, consists of at least four basic components:

- the expert knowledge module;
- the student model module;
- the tutoring module;
- the user interface module.

ITS's **expert module**, like a human expert, has knowledge about the particular subject matter to be conveyed to the student. This module uses different techniques (e.g., sets of rules, semantic nets or frameworks) for sorting this knowledge. The expert module also includes reasoning capabilities that are used to determine correct solutions to problems. The **student model**, ITS's core module, represents the student's current state of knowledge at any point of time. The student model distinguishes ITS from CAL by being able to respond to learning styles of different students by delivering customized instructions [11]. Ideally, this model gathers information about the students' knowledge and behavior including prior relevant learning, progress with the curriculum, preferred learning styles, and other types of learner-related information that could have possible implications on their performance and learning styles. The student model is also used as the basis for corrective feedback.

Based on the relationship with the expert's knowledge, student models are classified into three types [12–13]: overlay, differential and perturbation models. The overlay student model is the simplest of modeling techniques where student knowledge is assumed as a subset of the expert knowledge and the goal of tutoring is to enlarge this subset. The limitation of this model is that the student will not learn anything that the expert does not know and it does not cater for misconceptions that a student may have or acquire during tutoring. The differential student model, an extension of the overlay model, partitions the domain knowledge into already presented knowledge and that which has not been presented to the student. The perturbation student model, the most sophisticated technique, caters for knowledge possessed by the student that is not present in the expert domain knowledge. In a perturbation or bug model, the standard overlay model is combined with a representation of misconceptions or 'bugs'.

ITS's **tutoring module**, also called the instructional module or pedagogic module, designs and regulates instructional interactions with students. It gathers information about student's current performance level correlates this information

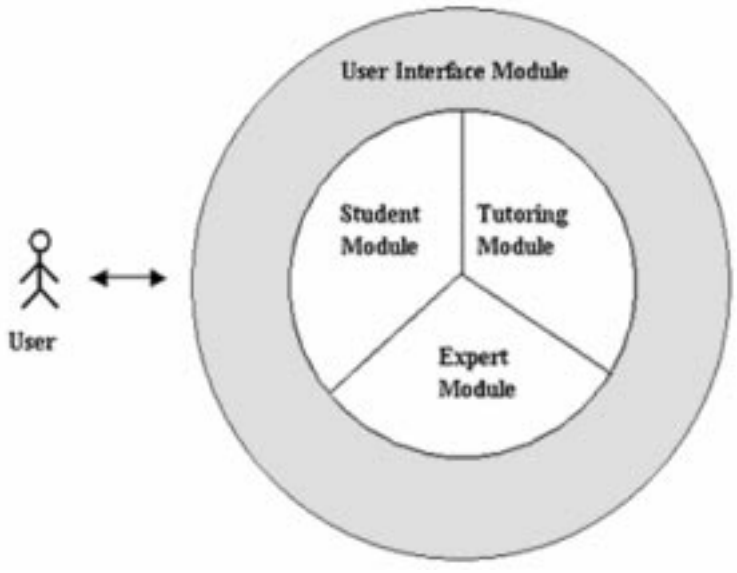

Fig. 1. Idealised structure of an intelligent tutoring system.

with its own pedagogic goal and finally decides which pedagogic activities will be presented [11]. In other words, this module is responsible for the curriculum sequencing. Student interacts with ITS through the **user interface module.** This module is responsible for presenting the information to the student and provides a high level of meaningful interaction between ITS and the students. The acceptance of any ITS highly depends on the ease of use and attractiveness of this module.

Web-based learning environments

There was a great paradigm shift in CAL systems with the advent of the World-Wide-Web. The possibility of using the Web for education has generated a great deal of interest among educators worldwide. A Web-based educational system is an Internet-based environment in which students and educators can perform learning-related tasks. Oliver *et al.* [14] identified information access, interactive learning, networked communication and material development as learning strategies in web-based learning environments. Web-based teaching and learning is flexible and easily fits around the lifestyles of students. Web-based classes offer advantages to active and involved learners who are motivated to learn. The relatively self-paced learning style of most web-based classes can be adapted to students who each learn differently and at different rates.

An important advantage of web-based courses is that instructional work areas can be available for use at any time. Collaboration among distributed learners is possible and often more convenient than face-to-face teaching in classrooms [15]. However, the web-based educational system has a number of limitations including high costs of putting the technology in place, and getting teachers and students to get used to new web-based environments. Additionally, it requires a great deal of effort from instructor's part to convert the conventional teaching materials to a web-ready form. However, once this conversion is done, future updates are much easier. A web-based educational system could include on-line tests that are automatically graded and enables instructors to easily track their student's performance and progress. Instructors can facilitate collaborative learning through chat rooms and forums.

A number of web-based learning systems have been developed. Early versions were hypertext-based information retrieval systems [16] and while later versions were adaptive and intelligent thereby making the web a more important educational medium. Examples of Web-based intelligent learning environments can be found in [17]. The adaptive nature of more recent systems is achieved by incorporating a student model that maintains up-to-date information about the learner's background, current stage of knowledge, goal and etc.

In recent years, the Internet and related technologies have been used in various ways to supplement the teaching of digital systems design concepts. In the virtual classroom developed by Burks [18], students are given a truth table of input variables and asked to construct digital circuits by 'dragging and dropping' the various logic gate components. Java applets have been used to demonstrate the functionality of digital logic circuits [19], and logic minimization with Karnaugh-maps [20].

WEB-BASED INTELLIGENT LEARNING ENVIRONMENT FOR DIGITAL SYSTEM—WILEDS

At the National University of Singapore, the digital systems design course is taught to second-year electrical and computer engineering students. In this course, students are taught digital circuit design techniques including combination and sequential logic circuit design, algorithmic state machine design and VHDL modeling of digital circuits. WILEDS aims to provide an environment where students solve problems related to the various topics in digital systems. WILEDS incorporates the entire process needed to assess students, starting from automatically generating problems to checking the students' solutions. Details about student progress are kept in a database for easy reference and monitoring by the course instructors. Students who need additional help can thus be easily identified.

WILEDS covers several major topics in the digital systems design course including number systems, combinatorial logic design, sequential logic design, programmable array logic (PAL) and programmable logic device (PLD) based design. In number systems, students are asked to do conversions between different types of randomly generated numbers systems, such as binary to octal, octal to hexadecimal to decimal, and so on. In sequential logic design, students are given a randomly generated state table of 'Present States and Next States', and asked to solve the 'Remapped Next State' for different types of flip-flops. In PAL and PLD based design, students are given a randomly chosen logic circuit and asked to find the output at the different nodes of the circuit. For each topic, WILEDS presents the problems in one of several levels of difficulties. Students have to show a certain level of competency to move to next level of difficulty and subsequently to the next topic.

In this section, the internal and external architecture of WILEDS are presented in the context of combinational logic design problems. These problems involve dynamically generated logic expressions that have to be simplified and minimized by students. The student's solution is assessed by the system and appropriate feedback is given based on the correctness of the solution.

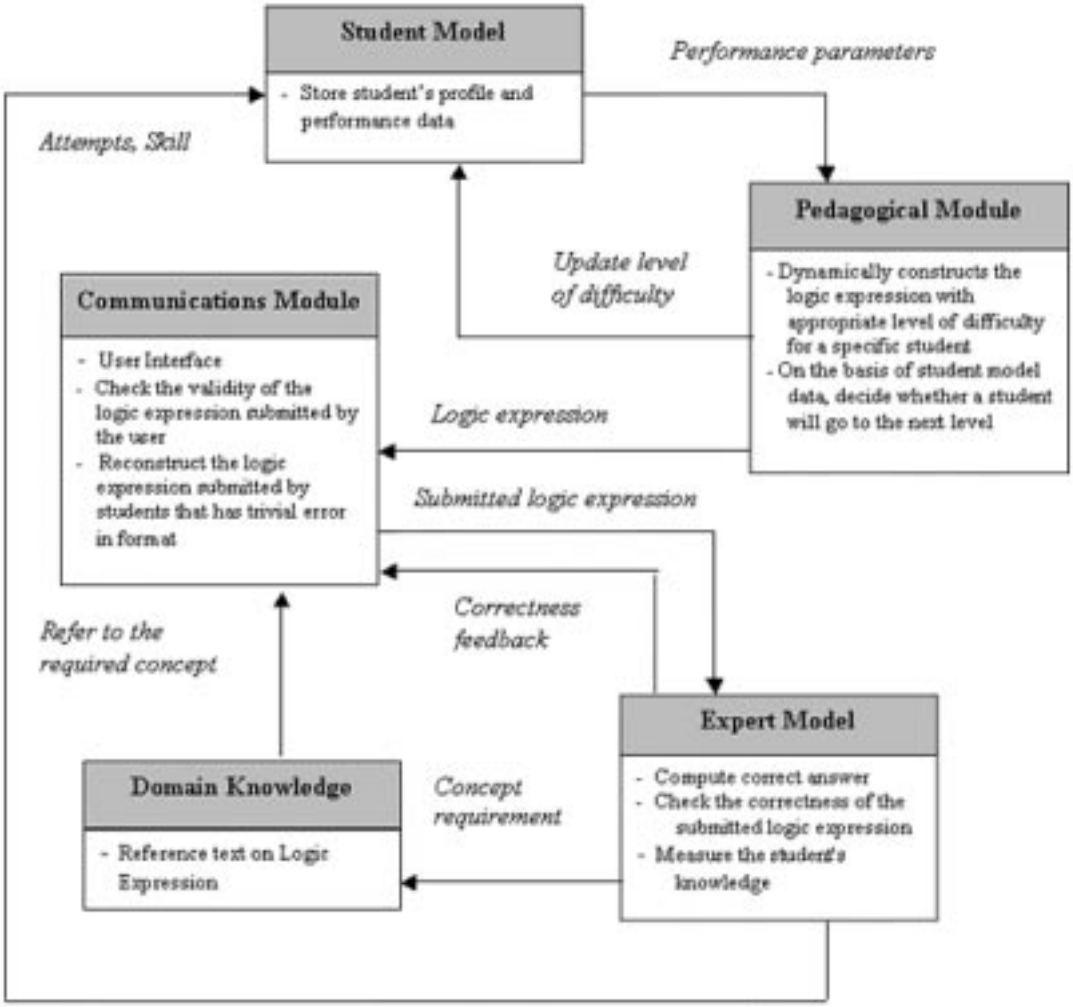

Fig. 2. The internal architecture of WILEDS.

Internal architecture of WILEDS

The internal architecture of WILEDS (Fig. 2) is based on work presented in [21], and comprises the following five components: communications module, pedagogical module, student model, expert model and domain knowledge. The pedagogical module generates problems dynamically and adapts the level of difficulty to a particular student based on the student model. Students access the system through the communications module and solve problems given to them. The expert module generates the expert answer, which is used to assess the student's solution. The student's perceived level of understanding is used to update the student model, which in turn helps the pedagogical module to generate the next problem for the particular student. The goal of WILEDS is to provide an environment where students are able to try out problems related to the various topics in digital systems.

The **communications module** is the user interface module through which all interactions (mainly the presentation of the problems and the acquisition of student's responses) with the system take place. Logic expressions are logical combinations of independent binary input variables. A simple convention is used to write logic expressions in WILEDS. For example, the logic expression $(\overline{A \cup B}) \cap (\overline{A} \cup C) \cap (\overline{B} \cup \overline{C})$ is written as $[\{!(A \vee B)\}\hat{}\{(!A \vee C)\}\hat{}\{(!B \vee !C)\}]$, where the symbols '!', '$\vee$' and '$\hat{}$' represent logical *NOT*, *OR* and *AND*, respectively. An error checking mechanism in the communication module prevents students from inputting empty strings, incomplete truth-table values, expressions that do not conform to the pre-set format, independent variables that are different from those in the given logic expression, and logic expressions that are identical to the given logic expression. The system is also able to detect logic expressions that are identical to the generated expression except that the sub-expressions are arranged in a different order as illustrated in Fig. 3. WILEDS has a pedagogical agent, which is an animated character implemented as a Java animation. Animated pedagogical agents [22] are 'life-like' autonomous agents that facilitate human learning by interacting with learners and make computer-based learning more engaging. In addition to this, animated pedagogical agents provide another

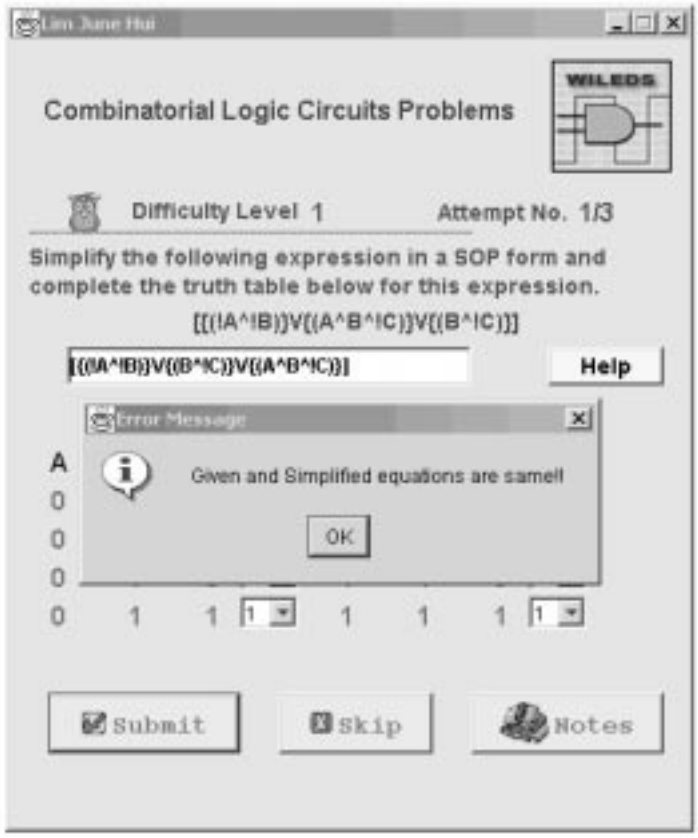

Fig. 3. Checking of user's inputs.

important benefit: motivation. WILEDS' pedagogical agent appears with a waving hand ('congratulatory' behavior) when a student successfully solves a problem. It also shows his score and number of attempts made by the student to solve a particular problem.

The **student model** represents the dynamic representation of the emerging knowledge and skills of the student. WILEDS' student model, which is based on the *overlay model* [12], records the student's proficiency for each topic in a database developed in the Microsoft Access RDBMS environment. WILEDS' communications, pedagogical and expert modules, which are implemented as Java Applets, access the student database in the server through Java servlets.

When a student logs into WILEDS, he is brought to the main environment (Fig. 4). The main environment has four buttons for the different topics, but only the button for the topic that is due for the particular student, is active. By clicking on the active button, the student will be brought to a problem of his or her current difficulty level based on that particular topic on the data kept in the student model. The topic of combinatorial logic has four problems that are arranged in a sequence based on the level of difficulty based on the number of independent variables as depicted in the Table 1. Two different factors are taken into consideration to determine the level of difficulty:

- nature of the problem: simplification or minimization;
- number of independent variables in the logic expression: 3 or 4.

Student's performance data are kept in two different databases: the first for simplification problems (i.e., difficulty levels 1 and 2) and the other for minimization problems (i.e., difficulty levels 3 and 4). For simplification problems, students need to simplify and find the truth table of a given logic expression generated by the pedagogical module. For minimization problems, students need to find the minimized expression of a given logic expression using Karnaugh-maps (K-maps). The databases store the generated logic expressions, the system-generated solution, up to three solutions attempted by the student, the student's correct answer, and the time spent by the student on the particular problem. Instructors can monitor the student's performance and progress

Table 1. Level of difficulties for combinatorial problems

Level of difficulty	Problem type	Number of independent variables	Independent variables
1	Simplification	3	A B C
2	Simplification	4	A B C D
3	Minimization	3	A B C
4	Minimization	4	A B C D

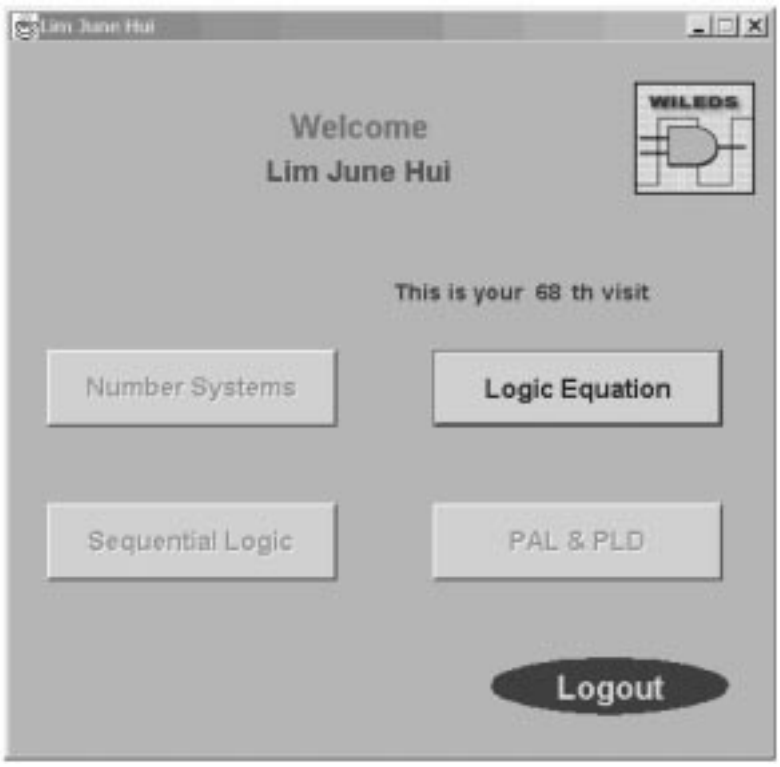

Fig. 4. Main tutoring screen.

using WILEDS' student monitoring system. This system shows general information about all students including current status of each topic, last access date, and other specific information (Fig. 5) about each topic covered by the student including difficulty level, question and score obtained by the student.

The **pedagogical module** dynamically constructs problems on a particular topic at the right level of difficulty for a particular student based on the information provided by the student model. WILEDS' expert module evaluates the solutions for the problems generated by the pedagogical module and determines whether the student's

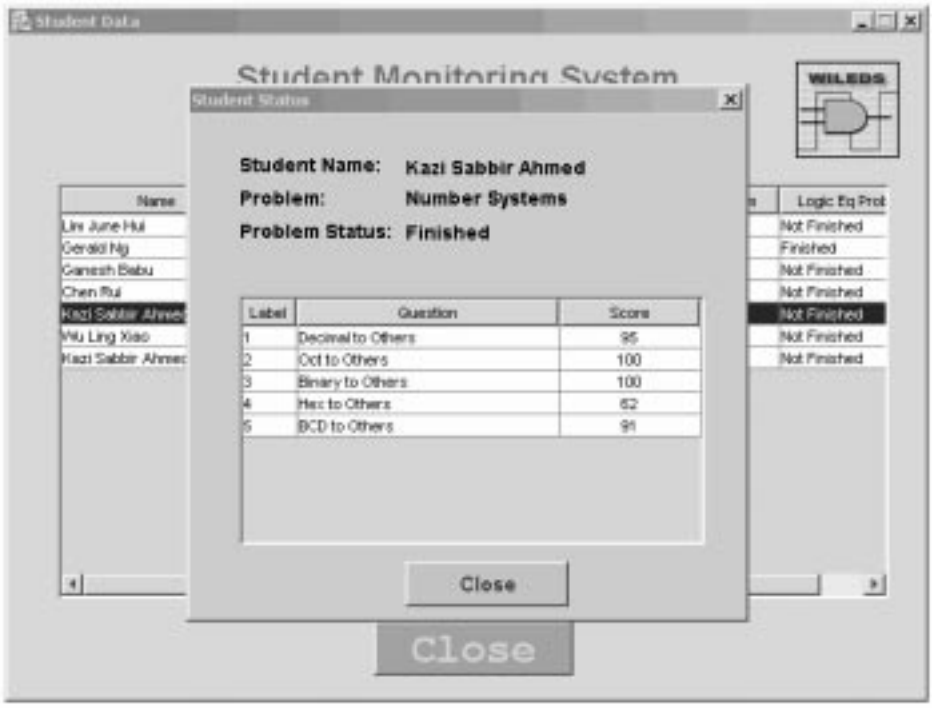

Fig. 5. Student monitoring system of WILEDS.

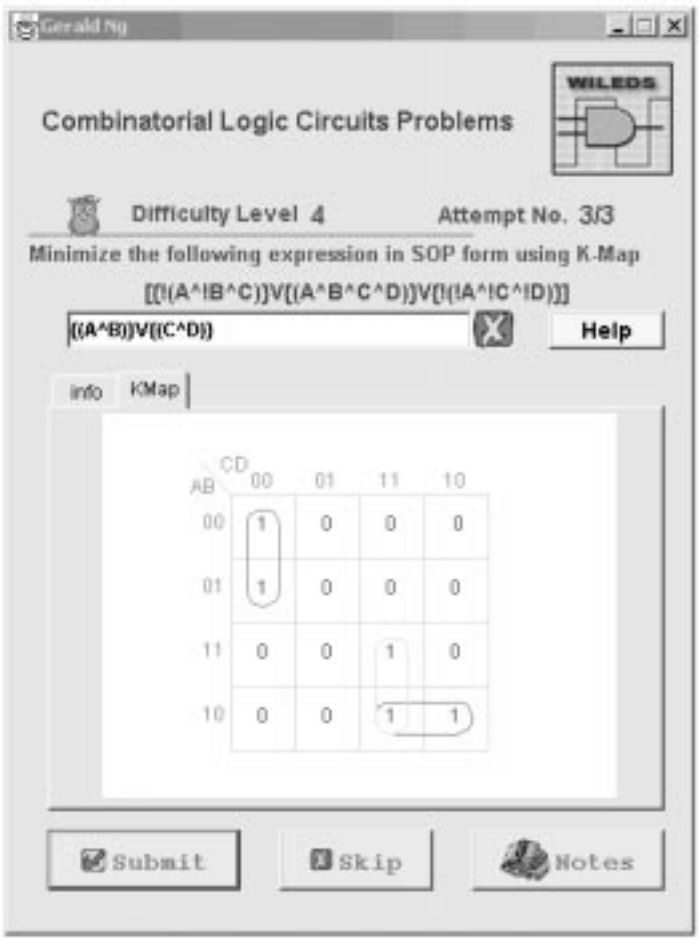

Fig. 6. K-map drawn by expert module.

answer is correct. The pedagogical module generates two types of logic expressions: sum-of-products (SOP) and product-of-sums (POS) expressions. A SOP expression consists of product terms that are logically OR-ed together, while a POS expression consists of summed terms that are logically AND-ed together. The total number of independent variables and the number of sub-expressions (terms) in the generated logic expression depend on the level of difficulty. Whether the generated logic expression is a SOP or POS, is chosen on a random basis. A student can skip any particular problem by clicking the 'Skip' button (Fig. 3). The pedagogical module records a score of -1 for the skipped problem in the student model data.

The **expert module** in an intelligent tutoring system computes the 'expert solution' for a given problem and compares the student's solution with the 'expert solution'. For the tutorial problems that are static in nature, 'expert solutions' are usually computed by the 'human expert' and stored in a database. The expert module accesses this database to get the 'expert solution'. In WILEDS, the pedagogical module generates all problems dynamically and the solutions are computed by expert module at the runtime, which is implemented as a set of Java sub-routines (i.e., Java classes). For simplification problems, which are solved by applying Boolean algebra, the expert model compares truth tables for the generated logic expression with student's solution to ascertain the correctness of the answer. Karnaugh-map (K-map) analysis is used to assess minimization problems, which involves finding the smallest number of gates to realize a logic network. In minimization problems, the truth table of the randomly generated logic expression is used to create the corresponding K-map function table and the minimized expression is formed by appropriate grouping of the logic terms.

Students have the option of viewing the K-map created by the expert module (Fig. 6) to help them to arrive at the solution to the minimization problem, if they fail to get the correct answer in the first two attempts. The system keeps record if a student solves the problem after referring to the K-map. The pedagogical module updates the student model based on the correctness of the student answer and the communication module provides the feedback to the student.

WILEDS' Domain Knowledge is basically the course material that can be accessed by students by clicking the 'Notes' button. The notes in HTML, PDF and other formats will open in a separate window. Students can access the course material

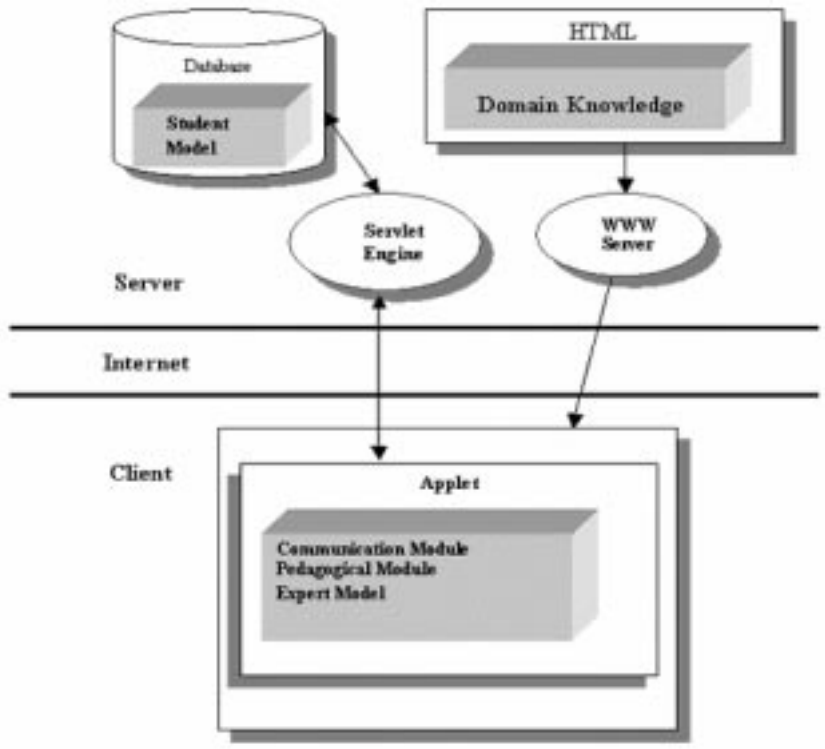

Fig. 7. External architecture of WILEDS.

for digital systems course over the web at anytime and while doing the problems on WILEDS. Also, students are able to participate in on-line chat rooms and discussion forums.

External architecture of WILEDS

Possible ways of deploying system architectures of web-based intelligent learning environments include:

- *Java-only solution:* The complete tutoring system that interacts with the student resides in a Java Applet, which is downloaded by visiting a specific URL and executes on the client (student's) machine.
- *HTML-CGI architecture:* The user interacts with HTML entry forms in a web browser; information entered by the user is sent to a web server, which forwards it to the CGI (common gateway interface) program that then replies with new HTML pages. The tutorial functionality, which is implemented by CGI programs, resides on the server and the user interacts with it using a standard web browser. This architectural model has several constraints, the most serious of which is the lack of immediate interactivity for certain user actions.
- *Java 3-tier client/server architecture:* A downloadable Java applet contains the user interface of the tutoring environment and communicates directly with the student model through middleware servlets. Thus, some of the system resides in the client, while others, especially the student model, reside in the server.

WILEDS is developed using the third approach. A major benefit of this architecture is that students can log in from anywhere and the system always 'knows' each student's current level of performance. Recent systems of this nature are also based on the Java 3-tier client/server technology.

The external architecture of WILEDS is illustrated in Fig. 7. The students use a web browser and run the Java applets that implement the communications, pedagogical and the expert modules of WILEDS on their computer (i.e., client). On the server computer, resides the World-Wide-Web server that implements communications over the Internet, and the student model and domain knowledge of WILEDS. The client-side Java applets communicate with the student model through the Java servlet.

WILEDS—the complete system

Besides combinational logic design, WILEDS covers several other major topics in the digital systems course including number systems, sequential logic design, programmable logic device (PLD) based design, state machine design and hardware description language (HDL) based modeling. For each topic, WILEDS presents the problems in one of several levels of difficulties. Students have to show a certain level of competency to move to the next level of difficulty, and subsequently to the next topic. Figures 8 and 9 show the number systems and PLD based design environments respectively. The problems are generated dynamically and the parameters for each problem are

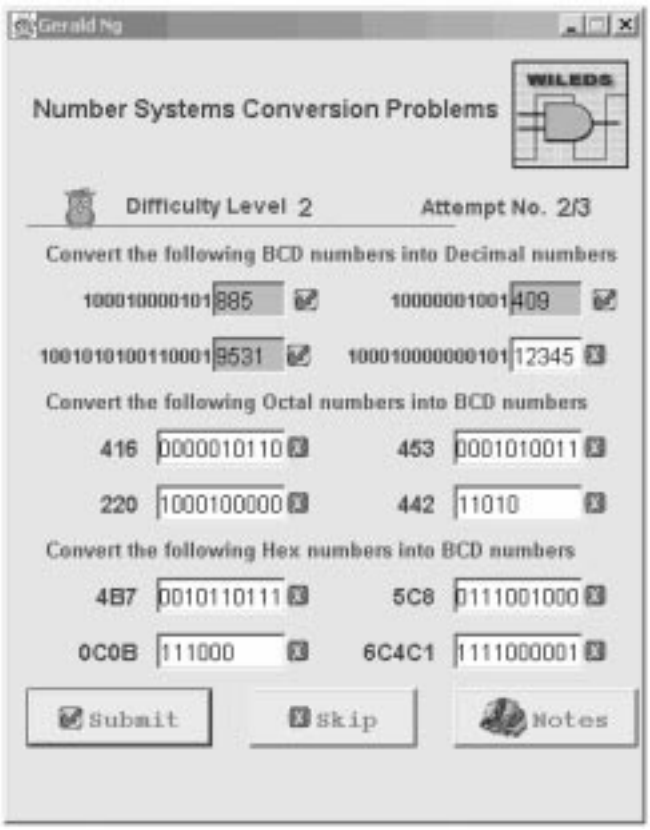

Fig. 8. WILEDS' number systems environment.

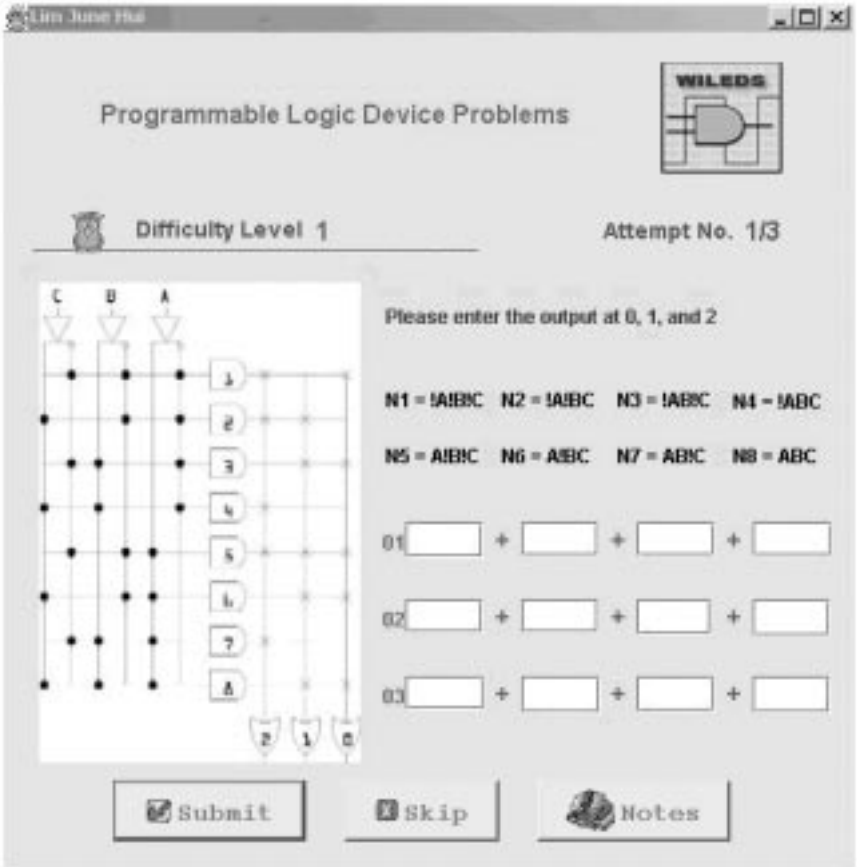

Fig. 9. WILEDS' PLD-based design environment.

chosen randomly. In WILEDS' PLD environment (Fig. 9), for example, the programmable connections to the AND array and OR array are randomly generated and the student has to find the minimized logic expression in SOP form at every OR gate.

CONCLUSION

The internal and system architecture of the Web-based Intelligent Learning Environment for Digital System (WILEDS) is presented in this paper. WILEDS allows students to solve problems related to the digital systems course, at any time and from any Internet-linked computer. WILEDS helps to enhance student learning and problem solving skills. Feedback on performance is given in real-time and the performance data is also captured. Course instructors are also kept informed of the progress of students.

Acknowledgment—The authors gratefully acknowledge the contributions of CH Lao, CT Tan and JC Wee.

REFERENCES

1. K. O. Jones, Issues in moving toward web-based teaching and learning, *IEE Int. Symp. Engineering Education: Innovation in Teaching, Learning and Assessment*, Vol. 2 (2001) pp. 36/1–36/5.
2. B. H. Khan, Web-based instruction (WBI): what is it and why is it? *Web-based Instruction*, Educational Technology Publications Inc., Englewood Cliffs, NJ (1997) pp. 5–18,
3. B. S. Bloom, The 2-Sigma problem: the search for methods of group instruction as effective as one-to-one tutoring, *Educational Researcher*, **13**, 1984, pp 3–16.
4. A. A. Kassim, S. A. Kazi and S. Ranganath, A web-based intelligent approach to tutoring, *Proc. Int. Conf. Engineering Education*, Oslo, Norway, 2001, pp. 8B4-25–8B4-30.
5. T. L. Good and J. E. Brophy, *Educational Psychology: A Realistic Approach* (4th Ed.), White Plains, NY: Longman (1990).
6. D. H. Jonasson, Objectivism versus constructivism: do we need a philosophical paradigm? *Educational Technology Research and Development*, **39**(3), 1991, pp. 5–14.
7. M. D. Merrill, Constructivism and instructional design, *Educational Technology*, May 1991, pp. 45–53.
8. Kinshuk, Computer Aided Learning for Entry Level Accountancy Students, Ph.D. thesis, De Montfort University, Leicester, England (1996)
9. M. A. Orey and W. A. Nelson, Development principles for intelligent tutoring systems: integrating cognitive theory into the development of computer-based instruction, *Educational Technology Research and Development*, **41**(1) 1993.
10. P. L. Brusilovsky, The construction and application of student models in intelligent tutoring systems, *J. Computer and Systems Sciences International*, **32**(10) 1994, pp. 70–89.
11. J. A. Self, Student models: what use are they? *Artificial Intelligence Tools in Education*, pp. 73–86 (1988).
12. D. H. Sleeman and J. S. Brown, *Intelligent Tutoring Systems*, Academic Press, New York (1982).
13. R. Kass, Student modelling in ITSs—implications for user modelling, *User Models in Dialogue Systems*, Springer-Verlag, Berlin, (1989) pp. 386–410.
14. R. Oliver, A. Omari, J. Herrington and A. Herrington, database-driven activities to support Web-based learning, *ASCILITE 2000*, Coffs Harbour, Australia (2000).
15. M. W. Goldberg, Student participation and progress tracking for web-based courses using WebCT, *Second International NA Web Conference*, October 5–8, NB, Canada (1996).
16. Kinshuk and Patel, A conceptual framework for Internet based intelligent tutoring systems, in A. Behrooz (Ed) *Knowledge Transfer*, Vol. II, Pace, London UK (1997).
17. P. Brusilovsky, Adaptive and intelligent technologies for web-based education, in C. Rollinger and C. Peylo (eds.), special issue on intelligent systems and teleteaching, *Künstliche Intelligenz*, **4**, 1999, pp. 19–25.
18. O. Burks, A Virtual Classroom Approach to Teaching Circuit Analysis, *IEEE Trans. Education*, **39**(3), August 1996, pp. 287–296.
19. Virtual Laboratory: Circuit Builder, Johns Hopkins University. Website: http://www.jhu.edu/virtlab/logic/logic.htm
20. Karnaugh Map Demonstration, Universität Hamburg. Website: http://tech-www.informatik.uni-hamburg.de/applets/kvd/index.html.
21. Joseph, M. K. Stern and H. Erik, Applications of AI in education, *ACM Crossroads*, Sept. 1996.
22. W. L. Johnson, Pedagogical agents, *Proc. ICCE'98*, vol. 1, China Higher Education Press and Spring-Verlag, Beijing (1998) pp 13–22.

Ashraf A. Kassim received his B.Eng. (First Class Honors) and M.Eng. degrees in Electrical Engineering from the National University of Singapore (NUS) in 1985 and 1987, respectively. From 1986 to 1988, he worked on the design and development of machine vision systems at Texas Instruments. He went on to obtain his Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University, Pittsburgh, in 1993. Since 1993, he has been with the Electrical & Computer Engineering Department at NUS, where he is currently an Associate Professor and Deputy Head of Department. Dr Kassim's research interests include image analysis, machine vision, video/image processing and compression.

Sabbir Ahmed Kazi is now working as a Systems Analyst in Singapore Polytechnic where he is involved in developing eLearning applications. He received his B.Eng. degree in Computer Engineering from Beijing University of Posts and Telecommunications, China in 1992. He obtained his M. Eng. (by research) degree from the Department of Electrical and Computer Engineering, National University of Singapore (NUS) in 2002. His current areas of interest include distributed learning systems, intelligent agent technology and game programming.

Surendra Ranganath received the B. Tech. degree in Electrical Engineering from the Indian Institute of Technology, Kanpur, in 1975, the ME degree in Electrical Communication Engineering from the Indian Institute of Science, Bangalore, in 1977, and the Ph.D degree in Electrical Engineering from the University of California at Davis in 1983. From 1982 to 1985, he was with the Applied Research Group at Tektronic, Inc., where he was working in the area of digital video processing for enhanced and high definition TV. From 1986 to 1991, he was with the medical imaging group at Philips Laboratories, Briarcliff Manor, NY. In 1991, he joined the Department of Electrical and Computer Engineering at the National University of Singapore, where he is currently an Associate Professor. His research interests are in digital signal and image processing, computer vision, human computer interfaces and neural networks.