

Animations and Intelligent Tutoring Systems for Programmable Logic Controller Education*

SHENG-JEN ('TONY') HSIEH

*Department of Engineering Technology and Industrial Distribution, Texas A&M University,
College Station, TX 77843, USA. E-mail: hsieh@tamu.edu*

PATRICIA YEE HSIEH

*System Administrative and General Offices, The Texas A&M University System, College Station,
TX 77840, USA. E-mail: phsieh@tamu.edu*

Programmable logic controller (PLC) technology is central to the development of modern automated manufacturing systems. However, due to limited equipment availability, high student/faculty ratios, limited lab access, and limited resources to support students outside labs and the classroom, many educational institutions do not provide the resources necessary to help students to become proficient with PLC technology. To alleviate some of these problems, an Integrated Virtual Learning System (IVLS) is being developed to teach about PLCs. The IVLS will be used to teach information students need to know about equipment before using it, so that lab time can be spent more productively and efficiently. In addition, unlike a human instructor, it will be available anytime, anywhere there is a computer connected to the Internet. IVLS lessons have thus far been developed to teach PLC applications; information flow from an external input device to a PLC processor image table to output devices; and PLC timer instructions. Three different timer instructions—Timer-on Delay (TON), Timer-Off Delay (TOF) and Retentive Timer On (RTO)—are covered. The lessons were developed using a combination of intelligent tutoring system, animation and presentation development tools. An evaluation of the lessons on timer instructions was conducted with 90 undergraduate manufacturing engineering students. Students made statistically significant learning gains as a result of taking the lessons, and rated them positively in terms of ease of use and understanding, clear objectives, amount of interaction, ability to motivate, relevance, and pace.

INTRODUCTION

A PROGRAMMABLE logic controller (PLC) is a solid-state control system with a user-programmable memory, used to read input conditions and set output conditions to control a machine or process. Thousands of PLCs are used in manufacturing plants for such applications as monitoring security, managing energy consumption, and controlling machines and automated production lines. It has been said that the programmable logic controller is among the most ingenious devices ever invented to advance the field of manufacturing automation [1]. Research by Frost & Sullivan indicates that the world market for programmable logic controllers will continue to grow as units become smaller, more functional, and more able to work in tough environments [2]. PLC sales are now about \$1 billion per year and there are more than 30 manufacturers [3]. Such statistics indicate that there is a great need for engineers with strong skills and knowledge in this area.

PLCs are covered in many undergraduate automation and control-related courses, such as:

- Computer-Aided Manufacturing
- Computer-Integrated Manufacturing
- Industrial Control and Networking
- Sensors and Instrumentation
- Manufacturing Automation and Robotics.

However, the equipment can be expensive; an industrial scale PLC such as the Allen Bradley PLC Family 5, for example, costs about USD \$8000. Consequently, schools often can only purchase a limited number of sets, which means that students must work in groups. Other obstacles to learning include high student/faculty ratios, limited lab access, and limited resources to support students outside labs and the classroom.

High student/faculty ratios and limited equipment availability are practical realities of engineering technology education. Needed are force multipliers that enable the most to be made of limited resources. Personal computers hold enormous promise in this regard because they are relatively inexpensive, widely available, and can be used to supplement or replace existing educational methods in a variety of ways.

* Accepted 28 August 2002.

Intelligent tutoring system (ITS) technology, for example, can be used to provide individualized instruction, much like that of a human tutor [4]. Although ITSs cannot replace the experience of using actual equipment, they can be used to teach preliminary information that students need to know about equipment before using it, so that lab time can be spent more productively and efficiently. The authors have previously used an ITS authoring tool to develop an intelligent tutoring system to teach students to operate a computer numerical control (CNC) machine [5]. Results indicated that most of the students who participated in this evaluation (1) learned from the lessons, (2) changed their perspectives on learning with computers, and (3) believed that the lessons were easy to use, motivating, and relevant to their education.

Intelligent tutoring systems have been developed primarily to teach military, medical, and liberal arts and science-related subjects, such as electronics troubleshooting, medical diagnosis, physics, programming, and geometry [4, 6]. A few systems have recently been developed or are underway for manufacturing-related subjects, including:

- visual inspection [7];
- nondestructive inspection [8];
- unit operations and process systems [9];
- design for manufacturing [10];
- progressive die design [11];
- factory management [12].

However, we found no reports of use of intelligent tutoring technology to teach about programmable logic controllers.

Personal computers can also be used to create and display animations. Instructional animations can be powerful tools for helping students to visualize concepts and relationships, and to develop mental models of processes. The word *mental model* is used here to mean an organized collection of knowledge about how something (such as a device or system) works that allows one to predict what will happen under certain conditions. Good mental models are essential to the mastery of cognitive skills such as programming and troubleshooting, and aid in the recall of procedures.

There have been a few attempts to employ animation in teaching about PLCs. A particularly notable one is an educational software package called LogixPro, which utilizes animated simulations of various processes, such as traffic control and batch mixing, to show how a ladder logic diagram relates to an automated process (see <http://www.thelearningpit.com/> for more information). Students can start and stop the animations, and study the corresponding ladder logic for certain conditions or cases. LogixPro's simulations are case-based and at a relatively advanced level, however. Students who are just beginning to learn about PLCs need a more complete, tutorial-like instructional presentation

that progressively introduces new PLC concepts and utilizes animation as needed.

The authors have recently undertaken a project to integrate ITS, animations, and other computer-based instructional approaches into a single learning system, which we call an Integrated Virtual Learning System (IVLS). This system is intended to alleviate current problems arising from low availability of equipment and help students and engineers to learn independently. At the time of this writing we have completed and evaluated lessons on PLC applications; information flow from an external input device to a PLC processor image table to output devices; and PLC timer instructions. The remainder of this paper will focus on the development and evaluation of these lessons.

LESSON DEVELOPMENT

Background

Before PLCs were invented, electronic and mechanical devices (such as motors, robots, vision systems, and conveyors) had to be physically connected using relays. Revisions or corrections to a certain layout would require rewiring—a complex and cumbersome process. PLCs made it possible for input and output devices to be connected directly to the PLC and controlled via software. PLCs are programmed using *ladder logic* to develop *ladder diagrams*. Ladder logic bears some resemblance to general-purpose programming languages such as BASIC, but includes hardware and control-related commands (called *instructions* in PLC terminology). Bit-level instructions for example, are used to examine inputs or energize outputs at specific PLC interface bit addresses. Timer instructions are used to control the timing of events in a process, such as delaying motors from starting at the same time or opening a valve for a given time. Counter instructions turn outputs on or off after a certain number of input transitions.

In short, PLC programming requires knowledge not only about programming commands, but also about basic electronics concepts, hardware addressing schemes, and information flow. In addition, students need to understand a little about all these things just to get started—which is a much more complex cognitive task than writing a ‘Hello, World!’ program (a typical introductory programming assignment for students learning general-purpose programming languages).

Instructional goals

Current approaches to teaching about PLCs—i.e., lectures and labs—have their advantages. For example, lectures are good for efficiently introducing large volumes of material, and for providing students the opportunity to interact with an instructor. Labs provide hands-on experience,

which is important for becoming familiar with equipment and practicing procedures.

However, lectures and labs are less effective in other areas. For example, a common problem that students have when learning introductory PLC concepts such as image tables and timers is visualizing what happens when a PLC executes a command. Lectures are often too abstract. Labs are more concrete, but merely using a PLC does not necessarily enable students to understand how it works; the PLC is still a black box.

Another common problem that students have is in learning relatively complex cognitive skills, such as writing PLC programs. Prevailing theories of cognitive skill acquisition suggest that—due to limitations in human short-term memory—enabling knowledge and skills (such as the functions and syntax of individual commands) need to become automatic before complex skills can be mastered [13]. Furthermore, skills become automatic as a function of practice. This suggests that the problem with traditional lecture and lab-based instructional approaches is that students do not have enough opportunities for interactive practice.

To address these problems, we decided to develop animations to help students to visualize PLC concepts and an intelligent tutoring system to provide adaptive interactive practice. A commercially available tool called Macromedia Flash was used to develop the animations, due to small file size of Flash animations and the wide availability of the Flash player.

An intelligent tutoring system authoring tool called XAIDA was used to develop the intelligent tutoring system components [14, 15]. The principal benefits of using XAIDA were that 1) it facilitates rapid development of adaptive instruction based on a student model, and 2) it is free for educational use. Disadvantages include 1) to reap XAIDA's

benefits, developers must conform to its rather rigid knowledge representation requirements, and 2) because it was developed in the early-to-mid 1990s as an experimental prototype, it has a very plain appearance and does not run over the Internet. However, because the type of knowledge XAIDA is most suited to teach (characteristics/properties of entities) is a good fit for a large portion of the PLC knowledge we wish to teach (information about programming commands), we wanted to take advantage of its adaptive practice features. We reasoned that if evaluations proved these lessons to be instructionally effective, we could later develop an Internet-based ITS that implements some of XAIDA's instructional strategies. We also figured we could work around XAIDA's glitz-challenged interface by substituting PowerPoint slides for a portion of XAIDA's instructional presentation. Ironically however, for reasons explained later in the paper, we eventually discontinued using most of the PowerPoint slides we developed and successfully used XAIDA instead.

Animation development

The animations were designed to allow users to visualize PLC concepts. That is, they were intended to represent not the physical appearance of PLCs, but rather, their theory of operation. It was also important to us that the animations be attractive to users and allow them to manipulate components of the animation to see what would happen.

Thus, for example, the animation developed to teach about PLC image tables (Figs 1 and 2) contains the following elements:

- A virtual switch and push button to represent external input devices such as normally open switches and push buttons. The user can

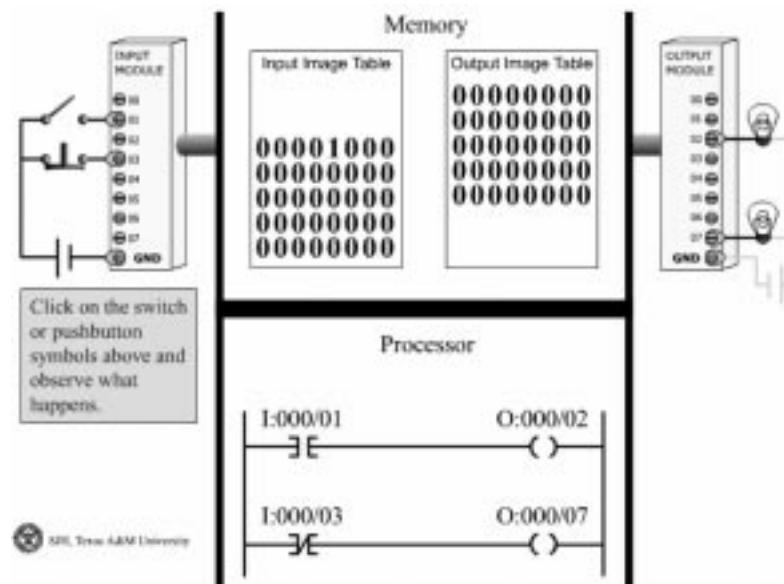


Fig. 1. PLC image table animation (pushbutton closed).

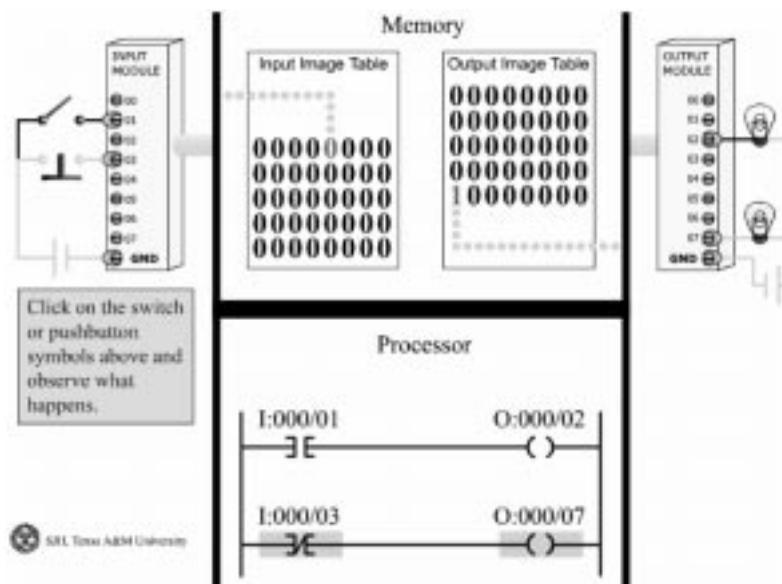


Fig. 2. PLC image table animation (pushbutton open).

manipulate the switches and observe the consequence of their actions. Each input device is connected to the input module of the programmable logic controller.

- A graphical representation of an image map in the PLC's memory. Whenever a ladder (programming) instruction is used, memory space is allocated to record the status of each bit. Each bit has its own operating principles. The animation lets students see how bits within the image map are updated as the external input devices are manipulated from Open to Closed or from Closed to Open.
- A ladder diagram to represent the logical relationship between input and output devices at the bit level. In other words, each input/output

device and ladder instruction has a memory address assigned to it. The processor turns bits On or Off based on the logic described in the ladder diagram.

- Lights to represent external output devices. These devices will be On or Off based on the instructions in the ladder diagram and the status of the input devices. The device states are represented by colors.

Animations were also developed to illustrate each of the three timer instructions. Figures 3 and 4 show screens from the animation for the TON instruction. In Fig. 3, the student is instructed to turn a switch to On to start the animation. When the switch is activated, the hand on the clock

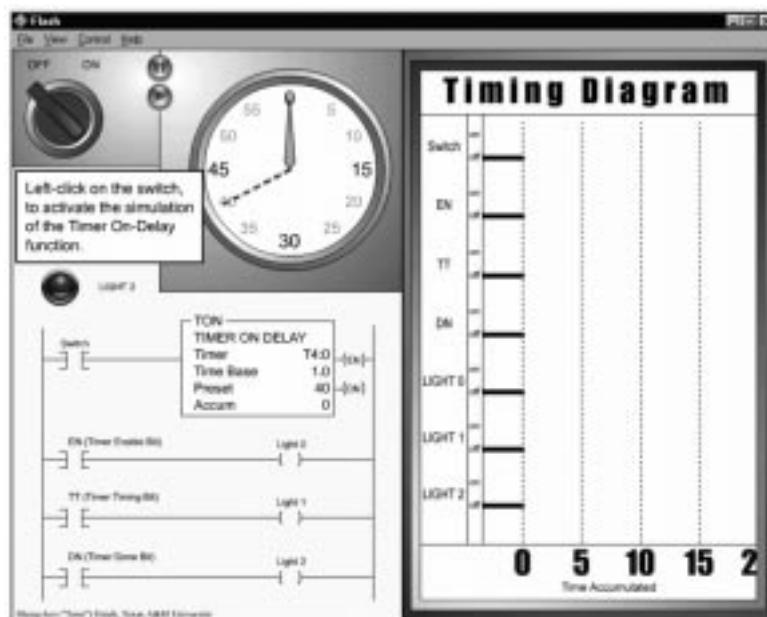


Fig. 3. Animation screen for TON instruction—Inactive.

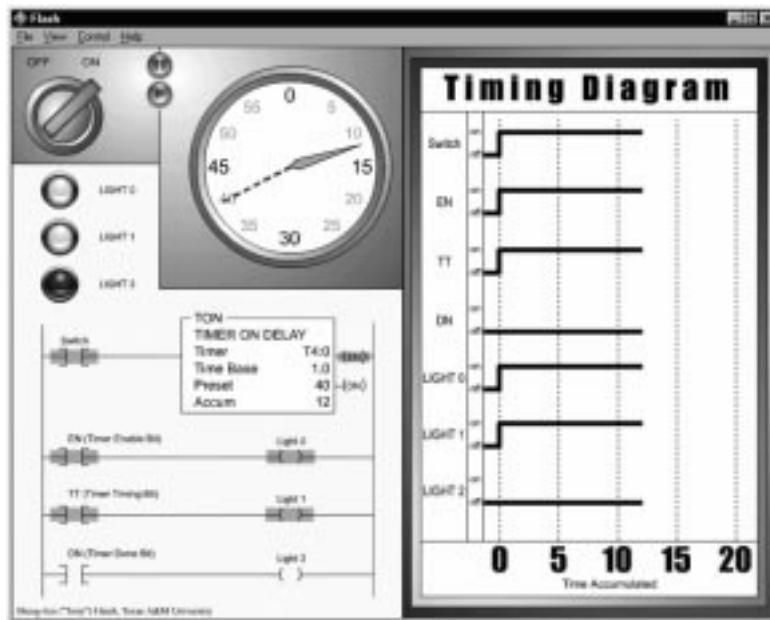


Fig. 4. Animation screen for TON instruction—Active.

begins to move from 0 to 40 s (the duration of the timer). As time progresses, rungs in the ladder diagram are activated and are highlighted in yellow. The timing diagram shows the status of various inputs, outputs, and bits as time progresses. Figure 4 shows the animation after 12 s have passed.

Intelligent tutoring system development

Authoring lessons in XAIDA involves using a lesson development interface to develop a database. The process is fairly straightforward as long as the lesson content conforms to the structure of XAIDA's database. Using the knowledge in the database and its own instructional algorithms and templates, XAIDA automatically generates an instructional presentation that includes adaptive practice with response-specific feedback. Very little instructional design is required (or indeed allowed), because the design is built into the system.

XAIDA has four instructional 'shells,' each of which is designed to teach a different type of knowledge (physical characteristics, theory of operation, procedures, and troubleshooting). The physical characteristics shell was used for this research because it is the most mature of the four shells. Knowledge in a physical characteristics database must be of the form *{part, attribute, value}* where a *part* is a component of a system, *attribute* is a property of the component, and *value* is the value of the property for that component. In a lesson about the fuel system of an automobile for example, we would represent the capacity of the fuel tank as follows *{fuel tank, capacity, 16 gallons}*. We could also represent part-sub-part relationships as follows: *{fuel system, has part, fuel tank}* and *{fuel tank, has part, gas cap}*.

Historically, XAIDA's physical characteristics

shell was designed to teach about the parts of a system. However, this shell has been successfully co-opted to teach steps in procedures, lexical information about words (e.g., meaning, synonyms, antonyms, etc.), periods in history, and a variety of other topics. For this study, we used XAIDA to teach about PLC timer instructions. The content used for these XAIDA lessons can be found in Table 1. The parts—shown in the three column headings in Table 1—were the three types of timer instructions (TON, TOF, and RTO). The attributes—shown in the row headings of Table 1—were ways of characterizing the timer instructions. The values—shown in the intersection of the columns and rows were specific characteristics for each of the three timer instructions. So for example, a fact about the application of the TON instruction was represented in XAIDA as *{TON, can be used, to delay motors from starting at the same time}*.

An XAIDA lesson consists of presentations of lesson material, followed by adaptive practice sessions. Figure 5 shows a presentation screen. Figures 6–9 illustrate XAIDA's adaptive practice question generation capabilities. Figures 6 and 7 show a multiple-choice practice question and the resulting feedback. This multiple-choice question has more than one correct response. The student selects the desired response(s), then clicks the Done button. XAIDA maintains a student model that keeps track of what each student knows and doesn't know based on his or her responses to practice questions. Facts that XAIDA thinks a student doesn't know tend to be asked about more often than facts a student seems to know. Thus no two students will be asked exactly the same set of practice questions, and brighter students will be asked fewer questions than slower ones. The thermometer at the bottom

Table 1. Timer instruction characteristics.

	TON	TOF	RTO
stands for	TIMER DELAY ON (also Timer On-Delay)	TIMER DELAY OFF (also Timer Off-Delay)	RETENTIVE TIMER DELAY ON (also Retentive Timer)
can be used	<ul style="list-style-type: none"> • to delay start of a function for a defined period of time from start of some other function • to delay motors from starting at the same time • to open a valve 27 seconds after a switch is turned on • to turn a mixing motor off 2 minutes after a switch is closed 	<ul style="list-style-type: none"> • to open a valve for 10 seconds • to hold a clamp on for 25 seconds after glue has been applied • to turn a mixing motor off 2 minutes after a switch is closed 	for maintenance functions, such as keeping track of downtime to keep track of the total time to make one batch of product
time delay period	<ul style="list-style-type: none"> • from instant the rung goes TRUE until DN bit is turned on 	<ul style="list-style-type: none"> • from time the rung goes FALSE until DN bit turns off 	from instant the rung goes TRUE until DN bit is turned on
when rung goes TRUE	<ul style="list-style-type: none"> • EN and TT bits go on 	<ul style="list-style-type: none"> • EN and DN bit turn on 	EN and TT bits go on
when AC reaches PR (when accumulated value reaches preset value)	<ul style="list-style-type: none"> • TT bit goes off, EN bit stays on, and DN bit turns on 	<ul style="list-style-type: none"> • DN bit is turned off and accumulated value is reset to 000 	TT bit goes off, EN bit stays on, and DN bit turns on
when rung goes from TRUE to FALSE and AC < PR	<ul style="list-style-type: none"> • EN and TT bits turn off and accumulated value is reset to 000 	<ul style="list-style-type: none"> • DN bit stays on; and timer begins to accumulate time for the delay period; • TT bit goes on and stays on for preset time, then resets 	EN and TT bits turn off and accumulated value is not reset
DN bit is set when	<ul style="list-style-type: none"> • accumulated value is equal to or greater than the preset value 	<ul style="list-style-type: none"> • rung conditions are true 	accumulated value is equal to or greater than preset value
DN bit remains set until	<ul style="list-style-type: none"> • rung conditions go false 	<ul style="list-style-type: none"> • rung conditions go false AND accumulated value is greater than or equal to preset value 	the appropriate RES instruction is enabled
TT bit is set when	<ul style="list-style-type: none"> • rung conditions are true AND accumulated value is less than preset value 	<ul style="list-style-type: none"> • rung conditions are false AND accumulated value is less than preset value 	rung conditions are true AND accumulated value is less than preset value
TT bit remains set until	<ul style="list-style-type: none"> • rung conditions go false OR when the done bit is set 	<ul style="list-style-type: none"> • rung conditions go true OR when the done bit is set 	rung conditions go false OR when the done bit is set
EN bit is set when	<ul style="list-style-type: none"> • rung conditions are true 	<ul style="list-style-type: none"> • rung conditions are true 	rung conditions are true
EN bit remains set until	<ul style="list-style-type: none"> • rung conditions go false 	<ul style="list-style-type: none"> • rung conditions go false 	rung conditions go false

right of Fig. 6 indicates what percent of the presented facts have been mastered so far.

XAIDA also generates detailed response-specific feedback. Figures 8 and 9 show a fill-in-the-blank practice question and the resulting feedback. In this example, the question has been answered incorrectly. Note that incorrect answers are marked with a red X and omitted correct answers are provided in red and marked with a red O. The feedback tells the student 1) that his response was incorrect; 2) correct facts about his response; and 3) the correct response (or responses in cases where there is more than one correct answer).

Note that Table 1 is presented primarily as a summary for the reader of the information in the XAIDA lessons we developed. For students, it can serve as a useful review tool, but they are not required or expected to learn by rote memorization

of the knowledge in the table. Rather, the knowledge in the table flows naturally from understanding how each of the timer instructions work.

EVALUATION

In evaluating how the lesson fared with students, we were particularly interested in the following:

- Did the instructional activities help students to learn more about timer instructions?
- Student opinions about the IVLS, such as its features, objectives, instructional sequence, difficulty, repetition, emphasis on important information, question-and-answer sessions, use of multimedia, interactivity, pace, quality, and relevance to their education.

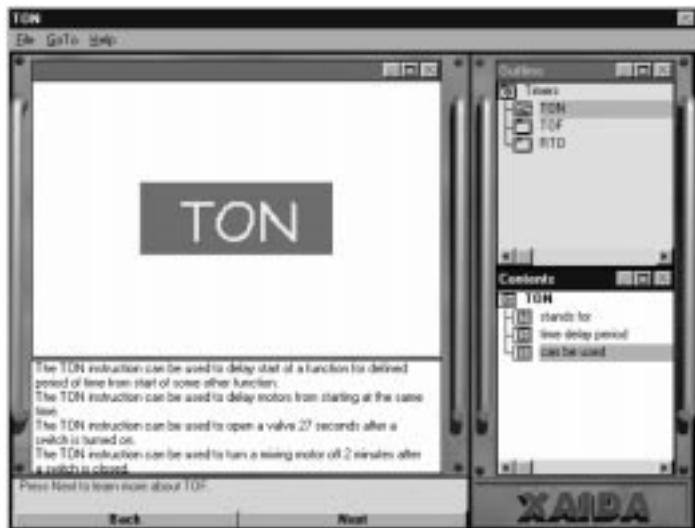


Fig. 5. XAIDA presentation screen showing applications of TON instructions.



Fig. 6. Multiple-choice practice question with responses selected by student.



Fig. 7. Correct answer feedback for practice question.



Fig. 8. Sample fill-in-the-blank practice question.



Fig. 9. Incorrect answer feedback for practice question.

SAMPLE TEST QUESTION

Select all answers that apply. A TOF instruction can be used:

- (1) for maintenance functions
- (2) in diagnostic programs
- (3) to create a short pulse at the beginning of a longer input condition
- (4) to create longer output functions derived from short input functions
- (5) to delay start of function for a defined period of time from start of some other function
- (6) to generate a short pulse at the end of a long input function

SAMPLE OPINION SURVEY QUESTION

Lesson was _____ to understand than I would have liked it to be.

More difficult 1-----2-----3-----4-----5-----6-----7 Easier

Fig. 10. Sample test and opinion survey questions.

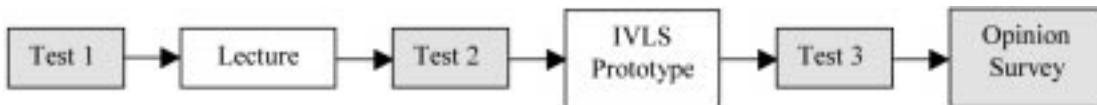


Figure 11. Evaluation event sequence.

Student comments

Participants. Participants in this evaluation were 90 undergraduate Manufacturing Engineering Technology and Industrial Distribution students who were studying a unit on programmable logic controllers. Evaluation activities took place during lab time; there were five labs of approximately 18 students each.

Materials. The evaluation focused on the lessons related to timer instructions, because this unit included examples of each type of instructional activity (i.e., presentations, interactive practice sessions, and animations). There are three types of timer instructions: Timer On Delay (TON), Timer Off Delay (TOF), and Retentive Timer On (RTO). For each instruction, students need to learn facts such as when to use it, length of delay, what happens to various bits when a rung with a timer instruction is activated. Table 1 summarizes the information students need to master.

Evaluation instruments included three parallel 10-item multiple-choice tests which were developed based on the content in Table 1 and an opinion survey. This survey asked students to rate various characteristics of the prototype on a 7-point Likert scale. Students rated prototype features (e.g. buttons, hot boxes), question-and-answer sessions, objectives, use of multimedia, instructional sequence, interaction with computer, difficulty, pace, content repetition, emphasis on important information, relevance to education, and overall quality. Figure 10 contains sample questions from the tests and opinion survey.

Procedure. Because the instructional effectiveness of the prototype was unknown and because we did not want the evaluation activities to interfere with students' learning, the unit on PLCs was taught in the usual way with lectures followed by lab. The evaluation activities were treated as additional lab activities. To avoid confounding learning progress obtained as a result of the lecture with that resulting from use of the prototype, we evaluated students' knowledge at three points in time: before the lecture, after the lecture, and after the prototype. Figure 11 shows the sequence of the evaluation activities.

Our goal was for students to be able to complete all evaluation activities within 45 minutes. With our initial design, however, we found that students required $1\frac{1}{2}$ to 2 hours to finish and were complaining that the XAIDA portion of the prototype was too long and repetitious. This was probably because 1) the interactive practice provided by the lesson was the last instructional activity in the sequence and it covered too much material; 2) some of XAIDA's default parameter settings

needed to be changed so that it did not revisit questions as often.

Thus after three sections (approximately 60 students), we felt it was necessary to change the sequence of instructional activities in the prototype. We divided what had been one long XAIDA lesson into three separate lessons and spread them out, meaning that students had shorter and more frequent practice sessions. We also adjusted some of XAIDA's default parameters so that the questioning would be less repetitious. Finally, we replaced the Powerpoint presentations with XAIDA's built-in instructional presentations instead, which allowed us to have fewer items on the menu. Figures 12 and 13 show the original (long) and revised (short) versions of the menu for the prototype. These changes allowed us to shorten the time required for evaluation activities to 45–60 minutes with no loss in content.

Data analysis and results

Because three of the five sections took a long version of the lesson and two took a short version—and because we didn't know what the impact of the change would be—we treated the subjects as two samples (long treatment and short treatment) in analyzing the data.

Test data. We analyzed the test data to see if there was a statistically significant score improvement between tests. The null hypothesis H_0 was that there would be no change. One-tailed paired t-tests were performed to compare student performance on:

- Test 1 (before lecture) with Test 2 (after lecture)
- Test 2 (after lecture, but before prototype) with Test 3 (after prototype)
- Test 1 (before lecture) with Test 3 (after prototype).

In cases where subjects were not present for all three tests, we only used complete pairs. So, for example, if a subject took Test 2 and Test 3, but was absent for Test 1, the subject's data was included only in the Test 2 vs. Test 3 analysis.

Figure 14 shows that for both the longer and shorter lesson samples, the null hypothesis was rejected for all three pairs of tests. This suggests that the lecture caused a significant improvement in learning, and the IVLS prototype subsequently caused further significant improvement in learning.

A test of means (Fig. 15) and a test of variance (Fig. 16) were also performed to see if subjects in the long lesson and short lesson samples performed differently. Students in the longer lesson sample performed slightly better than those in the shorter

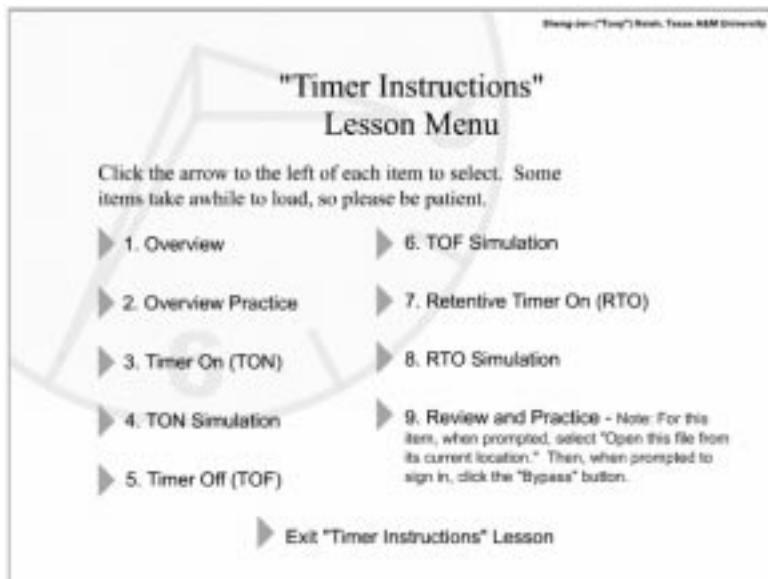


Fig. 12. Lesson menu—original.

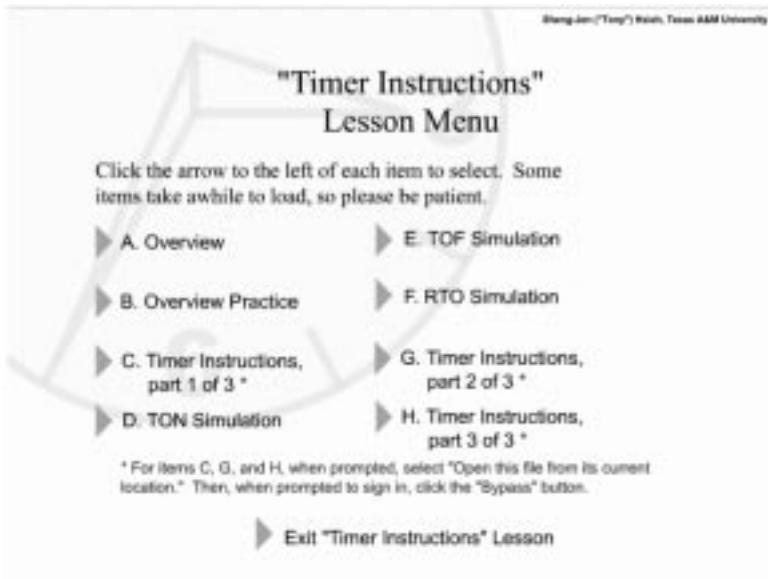


Fig. 13. Lesson menu—revised.

Shorter Lesson	Confidence Level: 0.95				
	t-Value	Probability	DF	Critical t-Value	Results
#3 vs. #2	4.36	1.00E-04	23	1.71	Reject Ho
#3 vs. #1	13.05	1.88E-07	9	1.83	Reject Ho
#2 vs. #1	13.46	1.44E-07	9	1.83	Reject Ho

Where #1 = test taken before lecture; #2 = test taken after lecture but before IVLS prototype; and #3 = test taken after completion of IVLS prototype.

$H_0: z_D = -\alpha$ where $D = X - Y$ is the difference between the first and second observations within a pair ($z_D = z_1 - z_2$)

$H_a: z_D > -\alpha$

Fig. 14. Results from one-tailed paired t-test analysis.

Testing difference between means with unknown sample variance

Null Hypothesis $H_0: U_1 = U_2$

Alternative Hypothesis $H_a: U_1 \neq U_2$

Test statistic: 1.4196

Significant level: 0.0500

Critical value: 2.0010

	Long Lesson	Short Lesson
Sample Size:	37	24
Sample Mean:	84.73	80.21
Standard Deviation:	13.64	11.08
Standard Error:	2.24	2.26
Probability:	0.16	

Conclusion: Do not reject H_0 .

Fig. 15. Test of longer lesson vs. shorter lesson sample means for test scores.

lesson sample. However, in both cases, there was no significant difference between the sample means and variance. Therefore we can conclude that there was no lesson length effect on test scores.

Opinion survey. We also computed means for the two groups on the opinion survey. Figure 17 summarizes these results. Student ratings were positive for all items, and there was no significant difference in ratings between the two groups. In general, students felt that the prototype was interactive, relevant, adequately paced, and easy to use and understand.

We again performed tests of means and variance to see if there was a significant difference in ratings between the long and short lesson samples. We used the data from the question about Ease of Use. Figures 18 and 19 show results from the tests of means and variance. In this case, students who took the shorter lesson rated the lesson's Ease of Use slightly more positively than students who took the longer lesson. However, the differences in means and variance were not statistically significant.

Student comments. Student comments about the

Test of Variance

Null Hypothesis $H_0: S_1 = S_2$

Alternative Hypothesis $H_a: S_1 \neq S_2$

Long Lesson vs. Short Lesson

Sample Size	37	24	
Mean	84.73	80.21	Difference = 4.5214
Variance	186.04	122.78	Ratio = 1.5152
Degree of Freedom	36	23	

F-Value = 1.52

Probability = 0.3 (Confidence Level = 0.95)

Critical F-Value = 1.93

Conclusion: Do not reject H_0 .

Fig. 16. Test of longer lesson vs. shorter lesson sample variance for test scores.

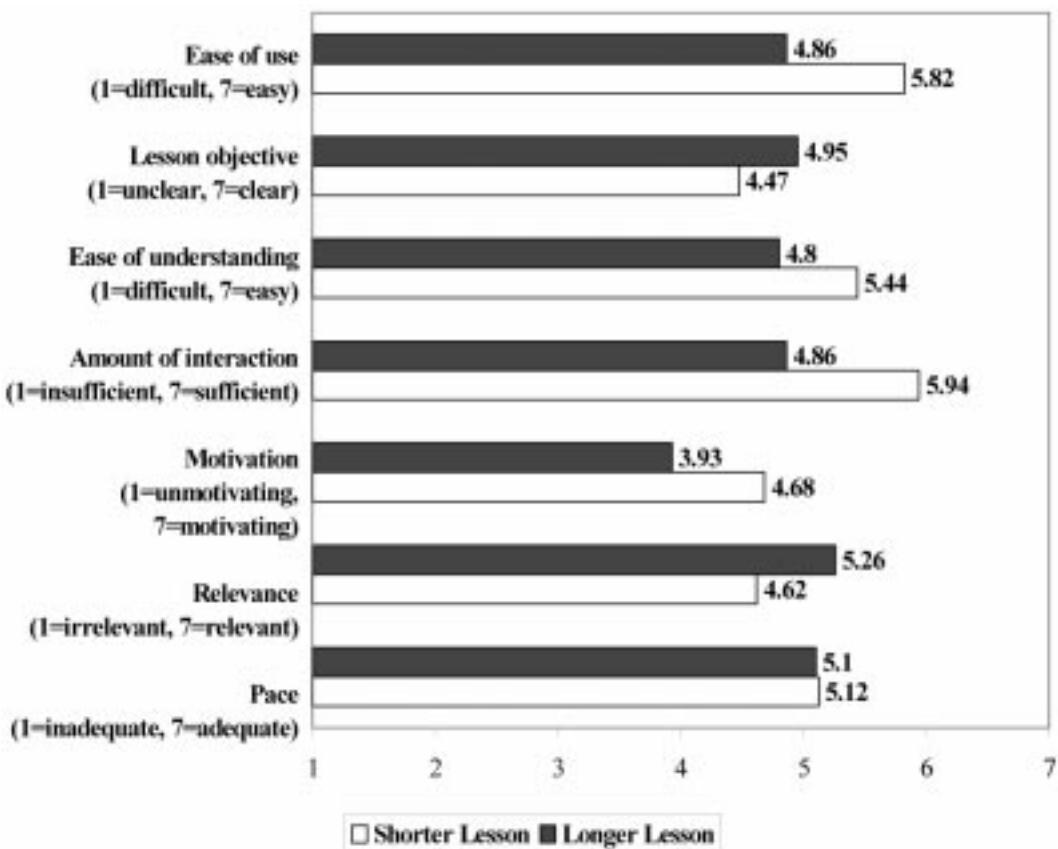


Fig. 17. Means for opinion survey items.

prototype are listed in the Appendix, and can be summarized as follows:

1. Several students in the 'long' group felt that the XAIDA-generated practice was repetitive and sometimes difficult to understand.

2. Many students liked the timer animations; they felt being able to visualize how the instruction worked was very helpful.
3. Some students commented that they would like to have examples of real-world timer applications (i.e., not just switches and lights).

Testing difference between means with unknown sample variance	
Null Hypothesis $H_0: U_1 = U_2$	
Alternative Hypothesis $H_a: U_1 \neq U_2$	
Test statistic: 1.30	
Significant level: 0.0500	
Critical value: 1.99	
Long Lesson Short Lesson	
Sample Size:	53 28
Sample Mean:	4.70 4.36
Standard Deviation:	1.10 1.13
Standard Error:	0.15 0.21
Probability:	0.1963
Conclusion: Do not reject H_0 .	

Fig. 18. Test of longer lesson vs. shorter lesson sample means for Ease of Use opinion survey item.

Test of Variance			
Null Hypothesis $H_0: S_1 = S_2$			
Alternative Hypothesis $H_a: S_1 \neq S_2$			
Long Lesson vs. Short Lesson			
Sample Size	53	28	
Mean	4.70	4.36	Difference = 0.3410
Variance	1.21	1.28	Ratio = 0.9527
F-Value = 1.05	Probability = 0.86	Confidence Level = 0.95	
Critical F-Value = 1.70	Conclusion: Do not reject H_0 .		

Fig. 19. Test of longer lesson vs. shorter lesson sample variance for Ease of Use opinion survey item.

4. Overall, the students thought the lesson was helpful and supplemented the lecture well. They wanted more.

DISCUSSION

Based on these results, we may conclude that the IVLS prototype is instructionally effective, and that students' subjective impressions of the system were positive overall. It appears that we may safely continue to develop similar types of lessons.

There was also no significant difference between the long and short lesson samples. Students in the 'long lesson' group did slightly better on the final test (Test 3) than students in the short lesson group, but not significantly so. Students in the 'short lesson' group tended to have slightly more positive opinions about the lesson, but again not significantly so. In any case, we plan to keep future lessons as short as we can while preserving instructional effectiveness.

The most popular instructional activity appears to be the animations. We are in the process of creating more realistic animations as requested.

Acknowledgements—This work was supported by a National Science Foundation Course, Curriculum, and Laboratory Improvement (CCLI) program grant (Grant No. 0088873) and a gift from Rockwell Automation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Some of the courseware described here can be accessed at http://entcweb.tamu.edu/hsieh/Hsieh_VirtualPLC.html

XAIDA is a prototype intelligent tutoring system authoring tool originally developed by MATCOM (<http://www.matcom-corp.com>) for the Air Force Research Laboratory (AFRL/HEA) under Contract No. F41624-93-F5002.

We thank Jarry Ho for his assistance in programming the timer simulations. The animations could not have been completed without his Flash programming and design skills.

CONCLUSIONS AND FUTURE DIRECTIONS

This article has described first steps in the process of developing an Integrated Virtual Learning System for programmable logic controller education. So far, our evaluation results have been very encouraging. In addition, the process of developing and evaluating the prototype has provided the opportunity to develop and evaluate our evaluation procedures. In future evaluations, for example, we plan to gather data not only about the lesson as a whole, but about specific instructional activities. We also plan to monitor the time required to complete various activities more closely.

We are currently in the process of developing more lessons. Future lessons will incorporate games and hardware emulators in addition to ITS and animations, and will focus on teaching program structure, ladder logic, and programming applications. Ultimately we hope to have a complete system that will be evaluated not only with undergraduate students, but also with high school students and industry professionals.

REFERENCES

1. C. Ray Asfahl, *Robotics and Manufacturing Automation*, John Wiley & Sons, Inc. New York, NY (1992).
2. *World Programmable Logic Controller Market*, Frost & Sullivan Research Publication 7450-01-00-00 (2001).
3. R. Filer, and G. Leinonen, *Programmable Controllers Using Allen-Bradley SLC 500 and ControlLogix*, Pearson Education, Inc., Upper Saddle River, NJ (2002).

4. E. Wenger, *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*, Morgan Kaufmann Publishers, Inc., Los Altos, CA (1987).
5. S. Hsieh, and P. Y. Hsieh, Intelligent tutoring system authoring tool for manufacturing engineering education, *Int. J. Eng. Educ.*, **17**(6), 2001, pp. 569–579.
6. V. J. Shute, and J. Psotka, Intelligent tutoring systems: past, present, and future. In D. Jonassen (Ed.), *Handbook of Research for Educational Communications and Technology*, New York, NY: Macmillan (1996) pp. 570–600.
7. A. Gramopadhye, S. Bhagwat, D. Kimbler, and J. Greenstein, Use of advanced technology for visual inspection training, *Applied Ergonomics*, **29**(5), October 1998, pp. 361–375.
8. E. Rogers, Y. Kennedy, T. Walton, P. Nelms, and I. Sherry, Intelligent multimedia tutoring for manufacturing education, in D. Budny and B. Herrick (eds.), *Proceedings of Frontiers in Education Conference, 1–4 Nov 1995, Vol. 2*, pp. 4d1.5–4d1.8.
9. D. Shin, E. S. Yoon, S. J. Park, and E. S. Lee, Web-based interactive virtual laboratory system for unit operations and process systems engineering, in *Computers and Chemical Engineering—7th International Symposium on Process Systems Engineering, July 16–21 2000*, **24**(2), pp. 1381–1385.
10. B. P. Woolf, C. Poli, and I. Grosse, Multimedia systems and intelligent tutors for teaching Design for Manufacturing, in *Proc. 1997 27th Annual Conference on Frontiers in Education, Nov 5–8 1997, Part 2 (of 3)*, pp. 711–716.
11. B. T. Cheock, and A. Y. C. Nee, Developing a design system into an intelligent tutoring system, *Int. J. Eng. Educ.*, **13**(5), 1997, pp. 341–346.
12. M. M. Dessouky, S. Verma, D. E. Bailey, and J. Rickel, A methodology for developing a web-based factory simulator for manufacturing education, *IIE Transactions*, **33**, 2001.
13. J. R. Anderson, *The Architecture of Cognition*, Cambridge, MA: Harvard University Press (1983).
14. T. Murray, Authoring intelligent tutoring systems: an analysis of the state of the art, *Int. J. Artificial Intelligence in Education*, **10**(1), pp. 98–129, 1999.
15. P. Y. Hsieh, H. M. Halff, and C. L. Redfield, Four easy pieces: development systems for knowledge-based generative instruction, *Int. J. Artificial Intelligence in Education*, **10**(1), pp. 1–45, 1999.

APPENDIX

Student comments

- ‘Very well put together. Aided in my understanding of counters. The Q&A at end were difficult to understand at times, but were also helpful. Excellent supplement to lecture.’
- ‘The simulations were a very big help in explaining the timers. Seeing which bits were energized in the simulation helps you remember better.’
- ‘Computer-based lessons well supplemented course material. In fact, they made more clear some information on timers.’
- ‘I thought the lesson was helpful, but could use some improvements—maybe it could be explained better to how it pertains to me—real world.’
- ‘Web-based lessons were very helpful because of interactive aspects. The TON, TOF, RTO lesson was very helpful.’
- ‘It was very simple to follow, but it just became too repetitive. The visual of the separate timers was very helpful in understanding the timers.’
- ‘The computer-based virtual stuff was a big help in understanding many of the TON, TOF, RTO applications and it may be a good idea to use more of them.’
- ‘I liked it a lot. To be able to work on homework and visually see it helped a lot. The online course let me go at my own pace. . . Overall I think that the online stuff is very helpful and should be expanded.’
- ‘I feel this tutorial truly helped with my understanding of timers. The only suggestion I could make is to have more of these tutorials for other topics discussed in the class.’
- ‘Instead of just switches and lights have it available with an actual application, like an assembly line [so we] can see how timers and so forth interact with each other. Also, ADD and SUB functions need to be added.’
- ‘The quiz at the end was somewhat difficult to follow and answer. Quiz should be stated clear and list all multiple answers when being asked for more than one answer.’
- ‘In the future much more information would be good. This was one of the best computer learning devices around. It was easy to follow and understand.’
- ‘The quiz could be bit vague at times, for example when there was more than one correct answer that was to be given. Besides this, I feel that the overall presentation of material is *excellent* and possibly could be used in other applications such as an animated PLC operating with other components.’

Sheng-Jen ('Tony') Hsieh received his Ph. D. in Industrial Engineering from Texas Tech University, Lubbock, TX. He is currently an Assistant Professor in the Department of Engineering Technology and Industrial Distribution and a member of the Graduate Faculty at Texas A&M University. His research interests include intelligent manufacturing

system design, thermal signatures for process and product characterization, and simulation and modeling. He is also the Technical Director for an industry grant to establish the Rockwell Automation Center at Texas A&M University. The Center will support courses related to automation, control, and production systems, and provide a vehicle for education, outreach, and research.

Patricia Yee Hsieh is currently a Senior Training Specialist and web-based training designer for the Texas A&M University System. This research took place while she was a Senior Instructional Designer at MATCOM. She has 13 years of instructional technology R&D and multimedia courseware development experience, including seven years of experience in designing and developing authoring tools for intelligent tutoring systems. Ms. Hsieh received her MA in Curriculum and Instruction from the University of Texas at Austin and her undergraduate degree with honors from Harvard University.