

Intelligent tutoring systems for Manufacturing Engineering education

Sheng-Jen (“Tony”) Hsieh¹ and Patricia Yee Hsieh²

¹College of Engineering, Texas A&M University, College Station, TX, USA, <http://entcweb.tamu.edu/hsieh>

Tel: (979) 845-4985, Fax: (979) 862-7969, hsieh@entc.tamu.edu

²MATCOM, San Antonio, TX, USA, <http://www.matcomcorp.com>

Tel: (210) 655-8911, Fax: (210) 590-3338, hsiehp@satx.meitech.com

Abstract: Modern manufacturing requires the use of expensive and sophisticated equipment. Manufacturing Engineering programs try to prepare students for industry by exposing them to the types of tools and equipment they will need to use. However, limited equipment availability and high faculty to student ratios make it difficult to provide the type of hands-on coached learning experiences necessary to become proficient. Computer-based instruction, especially intelligent tutoring system technology, has potential to provide students with individualized instruction at their convenience. However, development of computer-based instruction has traditionally been a highly labor intensive process, which few faculty have time to engage in. The recent advent of intelligent tutoring system authoring tools promises to make this process more accessible.

The objectives of this study were to 1) evaluate the feasibility of faculty course development using an intelligent tutoring system authoring tool called XAIDA; and 2) evaluate the instructional effectiveness of the developed lessons. An intelligent tutoring system authoring tool called XAIDA was used to develop a lesson on use of a computer numerical control (CNC) machine. The lesson, which takes approximately 45-60 minutes to complete, was developed in about 80 hours, including development of multimedia. The lesson includes presentation and adaptive practice questions (i.e., different students receive different practice questions, based on their performance).

The lessons were evaluated with manufacturing undergraduates, using a parallel pretest-posttest design. Students completed a pretest about CNC operation and a survey about how they felt about learning from the computer before taking the lesson, and a similar posttest and survey after taking the lesson. Students not only improved in their test performance after taking the lesson, but also enjoyed learning from the computer and wanted more computer-based training.

Keywords: intelligent tutoring systems, manufacturing engineering education, machining

1. Introduction

Modern manufacturing requires the use of expensive and sophisticated equipment. Many manufacturing engineering programs prepare students for industry by exposing them to the types of tools and equipment they will need to use. For example, a course on *Manufacturing Automation and Robotics* would use programmable logic controllers, sensors and instrumentation, robots, and machine vision systems for process automation system design. A course on *Production Systems and Computer-Aided Manufacturing* would use CAD/CAM software (e.g., FeatureCAM) and computer numerical control (CNC) machines. Courses on *Quality Assurance* and *Electronics Manufacturing* require use of expensive machines such as surface mounted placement machines (SMT) to practice the electronic manufacturing process. However, the learning experience is often hindered by such obstacles as high faculty-to-student ratios; limited access to the lab (in that students may only practice using equipment during the scheduled lab times); limited resources to support students outside labs and the classroom; and limited equipment to support lab assignments. It is not unusual for a lab class of 10-20 students to share 3-5 sets of equipment.

Computer-based instructional systems could alleviate many of the above problems. They can provide individualized instruction, much like a human tutor would. In addition, unlike human instructors, they can be available around the clock, wherever there is a computer. Although they cannot replace the experience of using actual equipment, they can teach much of the preliminary information students need to know about equipment before using it, so that lab time can be spent more productively and efficiently. In particular, a class of computer-

based instructional systems—known as *intelligent tutoring systems*—holds great promise for manufacturing engineering education in the future.

Intelligent tutoring systems, ITS authoring tools, and XAIDA

Intelligent tutoring systems (ITSs) are computer-based instructional systems that utilize artificial intelligence techniques to represent the *knowledge* needed to teach about a subject. A classic ITS consists of four components: an *expert module* that represents the knowledge to be taught, a *pedagogical module* that represents the teaching strategies employed by the system, a *student model* that represents what the student knows about the subject at any given point in the instruction; and a *student interface module* that allows the system to interact with the student [1]. The advantage of a knowledge-based approach to instructional system design is that content can be easily updated and manipulated. In addition, instructional strategies can be revised or added to the system. [In contrast, in traditional computer-based instruction, the knowledge and the instructional strategies are fixed within a predetermined instructional presentation.]

Although intelligent tutoring system approaches to computer-based instructional system design have been studied since the 1970s [2, 3], only recently have ITSs begun to be used outside the research laboratory [4]. This is probably because ITS techniques require more computational power than traditional approaches to computer-based instruction, and only in recent years has computer technology advanced to the point where the needed hardware is readily available and affordable.

In addition, many existing intelligent tutoring systems were developed for a specific instructional purpose. Only the programmers who developed them were able to make changes to the system. Furthermore, the underlying knowledge structures were often tailored to the needs of a particular research effort; they were not easily generalized to other domains or even to other types of knowledge within the same domain. However, in recent years, there has been increasing interest in developing ITS authoring tools.

ITS authoring tools generally offer a developer interface that allows developers to create a knowledge base; and a student interface that presents information from the knowledge base to the student. The content and instructional approach of the developed ITS varies from tool to tool; generally a tool is specialized for a particular class of content and approach. For example, RIDES allows instructors to develop simulations of equipment; students can then use the simulations to learn about the equipment and to practice procedures [5]. The Goal-Based Scenario Builder facilitates development of lessons that use a particular teaching strategy—namely, learning-by-doing through investigation [6].

XAIDA is a prototype of an authoring tool designed to allow expert maintenance technicians to develop ITSs for maintenance topics. XAIDA consists of four tools, each of which represents and teaches a specific aspect of maintenance knowledge—physical characteristics, theory of operation, procedures, and troubleshooting. A detailed description of XAIDA can be found elsewhere [7]. Here it is sufficient to say that XAIDA's Physical Characteristics tool, though originally intended to teach about characteristics of equipment (such as parts, functions, connections, etc.), has proved versatile enough to be used for a variety of domains and knowledge types. Some examples of the wide variety of lessons that have been developed using XAIDA can be found at <http://www.accd.edu/accd/workdev/ondemand.htm>.

XAIDA's Physical Characteristics tool utilizes a knowledge base organized as a semantic net based on the structure of the system to be taught. Thus, when used to teach about characteristics of a device, the nodes of the network are parts, and information about part characteristics (such as sub-parts, functions, and connections) is associated with each part. When used to teach about a procedure, the nodes are steps in the procedure, and information such as notes and warnings can be associated with particular steps.

To create a lesson in XAIDA, a developer (typically an instructor) uses a program called Develop to specify the knowledge to be taught using the available knowledge structure. If the content lends itself easily to being represented using a semantic network, lesson development is extremely rapid and straightforward. All a developer needs to do is specify the knowledge to be taught and link in any resources (such as multimedia files) desired to enhance the presentation. XAIDA's instructional delivery program (Deliver) then uses built-in techniques and templates for instructional presentation, question generation, and student modeling to automatically generate an interactive lesson using the content provided.

2. Method

An ITS authoring tool, XAIDA, was used to develop a lesson on how to set up a computer numerical control (CNC) machine to machine a part. This topic was selected because the first author teaches a course (*Numerical Control and Computer-Aided Manufacturing*) for which there is only one CNC machine available for lab classes of 15-20 students. Because students must take turns using the machine (and because they are often learning about it as they

use it), they do not have enough time to become proficient. The lesson was intended to familiarize students with the procedure before they actually use the machine, so that they get more out of their experience with the actual equipment. The lesson can also serve as a means of review. The following sections describe how the lesson was developed and evaluated.

2.1 Lesson Development

Steps, sub-steps, and notes related to setting up the CNC machine were identified. The process was viewed as five major steps, each of which had several sub-steps. Figure 1 contains the information for one of these major steps. Practice and test questions were developed based on the lesson content. A digital camera was used to take pictures of the equipment and controls. A screen capture program was used to capture screen shots of the CNC interface.

This information was entered into an XAIDA knowledge base using Develop. Figure 2 shows XAIDA-Develop's Outline Editor, which is used to define nodes. Figure 3 shows a fact being defined using the Fact Editor. Figure 4 shows the Resource Editor, which is used to link in multimedia files. Figure 5 shows the Question Editor, which can be used to customize XAIDA's default templates, used for presenting facts, questions, and feedback. Figures 6 and 7 show presentation and practice question screens generated by XAIDA-Deliver.

Including development of multimedia (primarily digital images), this lesson took about 80 person hours to complete. Because we were primarily interested in defining and representing the procedural information needed, we did not take advantage of XAIDA's ability to link to digital audio or video files, or to other documents or applications. This probably resulted in the lesson being less attractive than it could have been, but we felt it at least included all the information elements needed to be instructionally effective.

2.2 Lesson Evaluation Goals

The purpose of this study was to evaluate how the lesson fared with students. We were particularly interested in the following issues:

- How long did it take students to complete lessons?
- How did students feel about learning from the computer? I.e., did students prefer to learn about the lesson content in a traditional instructor-led classroom, using computer-based instruction, or a combination? Furthermore, did their preferences change after taking the computer-based lesson? Also, how comfortable were students with using computers, and did their comfort level change after taking the computer-based lesson?
- Did the lesson succeed in its goal of teaching students more about the procedure for setting up the CNC machine?
- What did students think of various aspects of the lesson, such as features, objective, instructional sequence, difficulty, repetition, emphasis on important information, question-and-answer sessions, use of multimedia, interactivity, pace, quality, and relevance to their education?

2.3 Participants, Materials, and Experimental Procedure

Twenty-five students from a Numerical Control and Computer-Aided Manufacturing class participated in this study. These students had viewed a demonstration of how to set up the CNC machine a few weeks prior to this study, but were not proficient at the task.

Paper-and-pencil evaluation instruments were designed to measure the effects of the computer-based lessons on subject knowledge and attitudes toward instructional methods and computers. Pretest and posttest instruments consisted of a 3-item survey and a subject test. The survey asked students to (1) choose their preferred method of instruction for that subject, (2) rate their degree of comfort in using computers, and (3) rate their degree of familiarity with the assigned subject. The subject test consisted of multiple choice questions on the lesson content. For example, a question on the subject test was: *When homing the axes, which menu do you use? (a) setup menu (b) opening screen menu (c) basic menu (d) jog menu.*

An 12-item opinion survey was administered after the posttest. For this 12-item survey, students used Likert scales (with anchors from 1 to 7) to rate their opinions about several characteristics of the lesson, including features (e.g. buttons, hot boxes), question-and-answer sessions, objective, use of multimedia, instructional sequence, interaction with computer, difficulty, pace, content repetition, emphasis on important information, relevance to education, and overall quality.

Students first completed the pretest materials and then began the computer-based lesson. They worked at their own pace until they completed the entire lesson, including the practice questions. Students then completed the posttest and the opinion survey.

3. Results

The results of the study in terms of attitudes about learning from computer, instruction effectiveness, and the reactions to the lesson are summarized as follows:

1. Time needed for lesson completion: mean = 27.3 minutes
2. Attitudes about learning from computers: preference for instructor-paced, computer-assisted learning (CAI) increased from 30% to 43%
3. Instructional effectiveness: mean pretest score was 33.3%; mean posttest score was 87.9%. This difference was significant [$f = 271.37$, $F(0.0500,1,46) = 4.05$, $p < .001$]
4. Lesson interface easy to use? mean rating = 2.16 (1 = easy; 7 = difficult).
5. Lesson objective clearly presented? mean rating = 2.04, (1 = clear; 7 = unclear)
6. Lesson easy to understand? Mean rating = 4.52 (1 = difficult; 7 = easy)
7. Sufficient amount of computer interaction? Mean rating = 5.28 (1 = insufficient; 7 = sufficient)
8. Was lesson motivating? Mean rating = 3.32 (1 = motivating; 7 = unmotivating)
9. Was lesson relevant to your education? Mean rating = 5.40 (1 = irrelevant; 7 = relevant)
10. Was lesson pace adequate? mean rating = 5.60 (1 = inadequate; 7 = adequate).

Some factors that may have affected the evaluation results include:

- Students had previously viewed a demonstration on how to operate a computer numerical control machine. However, we felt that the lessons were still appropriate for use as a review.
- The XAIDA software automatically generates practice questions based on individual student performance; no two practice sessions are alike. It tends to ask about things it thinks the student does not know about, and it keeps asking until it believes the student has mastered every item of instructional content. (If it were a human instructor, it would come across as being thorough to the point of obsession.) Though this approach is sometimes perceived as repetitive by students, it appears to be very instructionally effective.

4. Conclusions and Recommendations

Most of the students who participated in this evaluation learned from the lessons and changed their perspectives on learning with computers. In addition, students tended to rate the characteristics of the lessons positively, indicating that they believed the lessons were easy to use, motivating, and relevant to their education. Future directions include (1) integrating ITS lessons with domain specific software, such as programmable logic controller emulators, to provide a more complete learning experience (rather than teaching facts only), (2) combining ITS lessons with other web technology such as on-line chat to provide an interactive atmosphere for learning, and (3) incorporating a dynamic decision mechanism that can learn and adapt instruction to individual learning styles.

5. References

- [1] E. Wenger, *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge*. Morgan Kaufmann Publishers, Inc., Los Altos, CA, 1987.
- [2] J. Carbonell, "AI in CAI: An artificial-intelligence approach to computer-assisted instruction," *IEEE Transactions on Man-Machine Systems*, vol. 11, pp. 190-202, 1970.
- [3] D. Sleeman and J.S. Brown, Eds., *Intelligent tutoring systems*. Academic Press, London, 1982.
- [4] K.R. Koedinger, J.R. Anderson, W.H. Hadley and M.A. Mark, "Intelligent tutoring goes to school in the big city," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 30-43, 1997.
- [5] A. Munro, M. Johnson, Q. Pizzini, D. Surmon, D. Towne and J. Wogulis, J. "Authoring simulation-centered tutors with RIDES," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 284-316, 1997.
- [6] B. Bell and S. Kedar, "When less is more: Supporting authoring and interface building via special purpose task models," In J. Greer, Ed., *Artificial Intelligence in Education, 1995: Proceedings of AI-ED 95- 7th World Conference on Artificial Intelligence in Education*, Association for the Advancement of Computing in Education, Charlottesville, VA, 1995.
- [7] P.Y. Hsieh, H.M. Half and C.L. Redfield, "Four easy pieces: development systems for knowledge-based generative instruction," *International Journal of Artificial Intelligence in Education*, vol. 10, no. 1, pp. 1-45, 1999

Sub-Steps/Notes	Illustration
This step allows you to verify that your program behaves as you intended before you actually machine the part.	Blackboard
On the Basic menu, press F5 to preview your program.	Arrow pointing to F5 soft key on Basic menu
In the blue VIEW SIZE box, you will see the settings for the View window. Make sure the settings are a little larger than needed to display the work piece. Enter numbers as needed, hitting Enter after each line.	VIEW SIZE box
After the last line, a blank grid will appear. Press F2 (PAUSE) to preview you program step by step. You can also press and hold F2 to preview the program without a pause.	Blank grid
Watch the cutting path to make sure that it is doing what you intended. If the cutting path is not consistent with your design, check your program. When you are finished previewing your program, press the Escape (Esc) key to return to the Basic menu.	Show cutting path in preview window.

Figure 1. Information associated with the "Preview Program" step

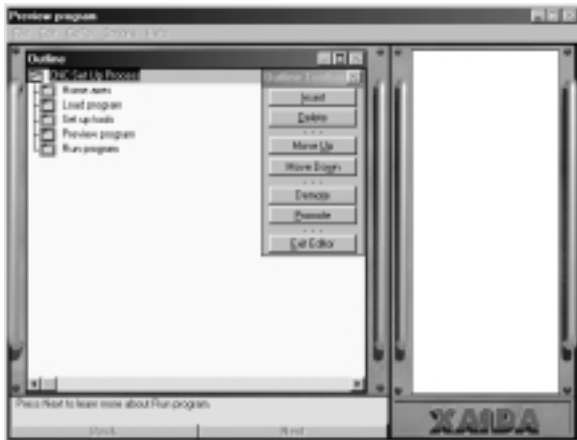


Figure 2. Outline Editor in XAIDA Develop



Figure 5. Question Editor in XAIDA Develop

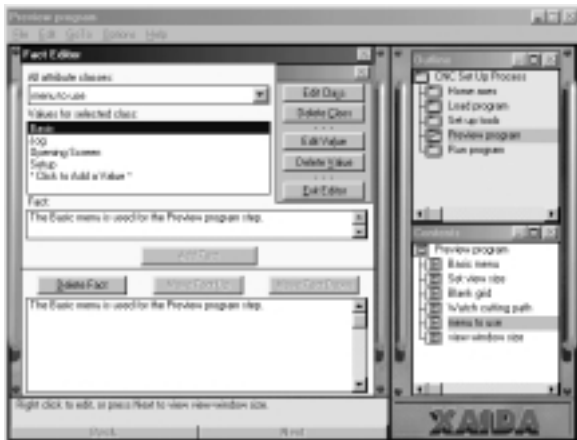


Figure 3. Fact Editor in XAIDA Develop

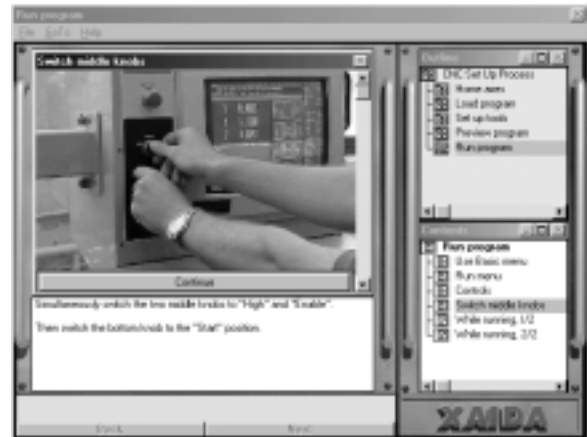


Figure 6. Sample Presentation Screen in XAIDA Deliver

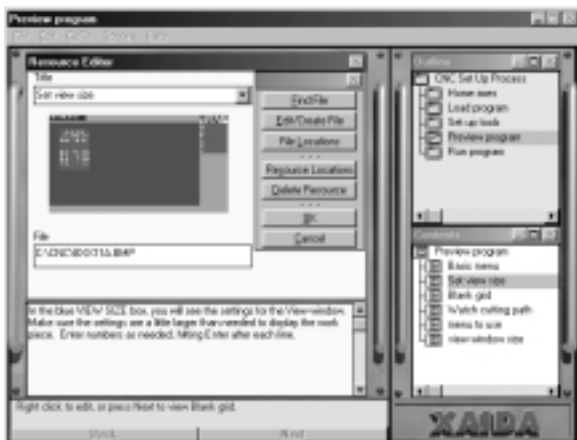


Figure 4. Resource Editor in XAIDA Develop

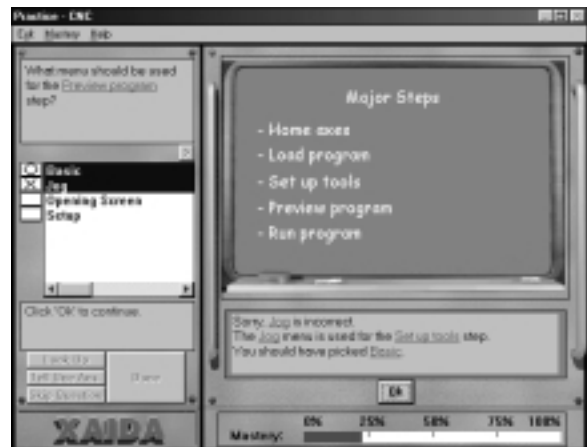


Figure 7. Sample Practice Question Screen in XAIDA Deliver