INTELLIGENT TUTORING SYSTEMS:

IF THEY ARE SUCH GOOD IDEAS, WHY AREN'T THERE MORE OF THEM?

William B. Johnson
Principal Scientist
Search Technology, Inc.
4725 Peachtree Corners Circle
Suite 200
Norcross, Georgia 30092

ABSTRACT

Intelligent Tutoring Systems (ITSs) have emerged as a potential enhancement to training for the military, industry, business, and schools. While the ITS concept sounds good, the transition from laboratories to training centers has been slow. However, changes in software, hardware, and development approaches are making it possible to develop and deliver ITSs in a relatively short time using a reasonable amount of personnel and fiscal resources. An example of one such system is Microcomputer Intelligence for Technical Training (MITT). MITT, developed for the Air Force Human Resources Laboratory in cooperation with the NASA Johnson Space Center, provides intelligent tutoring on the diagnosis of problems with the space shuttle fuel cell system.

INTRODUCTION

Intelligent Tutoring Systems (ITSs) are at the top of the line in computer-based instruction (CBI). ITSs hold the promise of maintaining expertise in specific instructional domains and of possessing the pedagogical knowledge and techniques of a master teacher. ITSs also promise the ability to patiently monitor student actions, infer student intent, assess current knowledge states, and sequence the continuing instruction. Therefore, ITSs' potential to affect training and education, in a variety of applications, is tremendous.

Yes, the above introduction plays like an old record. Since the late-50s and early-60s, the innovators of training technology have promised that innovations, such as programmed learning and CBI, will revolutionize training and education. While CBI is certainly an important component of training, particularly in the military, it has not become the panacea that some researchers had promised [1]. Leading researchers of intelligent computing for training and education see that in many ITS areas we are "not too much better off than we were twenty years ago" [2]. Of course, ITSs are not the only artificial intelligence (AI) subset that has come under fire for not moving fast enough to fulfill promises [3,4,5]. The ultimate questions, regarding the computer's ability to think like a human, are articulated by Hubert and Stuart Dreyfus [6].

There are many reasons why conventional CBI has been slowly acclimated in training and education. The lack of quality software, the lack of quality instructional courseware, the lack of computers, the rapid rate of change in hardware technology, the unfamiliarity by instructional personnel, and the high development costs of CBI programs are but a few reasons why CBI has not fulfilled its promise. The reasons conventional CBI has not reached its full potential are further amplified by the complexity associated with the development and delivery of ITSs. When potential problems are known, the problems can be addressed during the early stages of ITS conceptualization, design, development, implementation, and evaluation. Therefore, the purpose of this paper is to identify potential obstacles for ITS's successful use in the military, in industry, in business, and in schools.

First, this paper defines intelligent tutoring systems and provides a list of exemplary systems including their domains. Second, potential problems regarding ITS development are discussed. Finally, a successful new ITS, Microcomputer Intelligence for Technical Training (MITT), is briefly described. The description demonstrates that ITSs' potential problems are not insurmountable.

IF THEY ARE SUCH GOOD IDEAS...

The literature provides numerous, consistent definitions of ITS [7,8,9,10]. ITSs are instructional systems that deliver training in a manner comparable to that of a human tutor. To provide such instruction, ITSs must contain an understanding of a specific domain, a means to model student understanding of that domain, and a component containing pedagogical guidelines for providing feedback and remediation.

Figure 1 depicts the modules of a generic ITS. At the center of the diagram is the instructional environment. The environment may be designed as a dialog between the student and the computer, a system that presents information and asks questions, a system that offers problems for the student to solve, a simulation of a technical system, or other instructional scenarios. The student interacts with the instructional environment through the interface. The interface may be a keyboard, touch screen, mouse, voice, or a combination of these devices. The instructional environment and interface are not unique to ITSs; all CBI must have them.

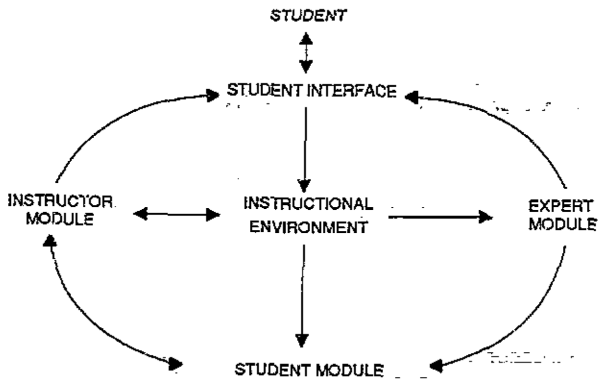


Figure 1. The modules of an intelligent tutoring system

The components that differentiate ITS from conventional CBI are the expert, the student, and the instructor modules. The expert module represents the technical system with a set of rules regarding the system's functions and/or operating procedures. For example, the expert math tutor may contain rules about carrying and/or borrowing in addition and subtraction programs. The expert module for a technical training domain would include rules regarding the operation and relationships of components in the system. The expert module must contain much of the same knowledge a human expert would possess.

The student module can be based on prerequisite student knowledge and on critical actions the student makes during interaction with the instructional environment. The student module can contain a current file of the student's actions, as well as an historical file containing the student's preferred learning style, the student's previously mastered lessons, and the types of errors to which the student is prone.

The instructor module must monitor the student's performance and compare it to the expert's performance. Like a proficient human tutor, the instructor must be able to decide the acceptable level of student performance, the appropriate remediation, and the sequence for subsequent learning.

The complexity of the instructor, student, and expert modules among existing ITSs vary greatly. While some researchers/developers concentrate on the expert, others focus on the instructor or the student module. Few researchers/developers have paid sufficient attention to all modules in order to create an ITS that can be used in instructional environments. This leads to the question *"If ITSs are such good ideas, why aren't there more of them?"*

Example Systems

AI has influenced many fields: medicine (INTERNIST, CADUCEUS, MYCIN, EMYCIN, and PUFF), chemistry (DENDRAL), biology (MOLGEN), geology (PROSPECTOR and DRILLING ADVISOR), communications diagnosis (ACE), and locomotive repair (DELTA/CATSI). These systems were designed to be job decision aids and attempted to bring an expert to the job site, laboratory, or clinic. However, these AI applications were not designed for instruction and tutoring. The Air Force Integrated Maintenance Information System (IMIS) is another AI project that emphasizes aiding rather than training.

Various state of the art ITSs were discussed during two recent conferences sponsored by the United States Air Force Human Resources Laboratory (AFHRL) [11] and the Army Research Institute for the Behavioral and Social Sciences [9]. The most comprehensive historical overview of ITSs available is provided by Wenger [12]. Table 1 lists examples of more noted ITSs. The systems are listed in approximate chronological order. Each system is not described in this paper, but references are provided for those systems listed. In general, these systems were developed as laboratory tools and remained in the laboratory to test various hypotheses related to ITSs. Exceptions are Anderson's LISP Tutor, commercially available, and the MITT system. The MITT system, designed for space shuttle fuel cell diagnostic training, is described later in this paper. The next section of this paper discusses reasons why many ITSs never leave the laboratory.

... WHY AREN'T THERE MORE OF THEM?

ITSs have not made a smooth transition from laboratories to training centers. This section offers four categorical reasons why few ITSs survive this transition: resources, personnel, equipment, and attitudes.

Resources

ITS development is expensive. An adequate system requires at least two expert systems: a content expert and a pedagogical expert. The system must be able to construct a dynamic student model and instructional environment. An interface to facilitate user interaction with the instructional environment is also necessary. Combining the

SYSTEM	SUBJECT	DEVELOPERS/DATES
SCHOLAR	Geography	Carbonell & Collins, 1968-1972 [13]
SOPHIE	Electronics	Brown, Burton & deKleer, 1974-1984 [14]
BUGGY/DEBUGGY	Math	Burton & Brown, 1979-1981 [15]
WEST	Solving a Math Game	Burton & Brown, 1978-1984 [15]
STEAMER	Ship Steamplant Operation	Hollan, Hutchins & Weitzman, 1978-1986 [16]
GUIDON	Medicine	Clancey, 1980-1986 [17]
Geometry Tutor	Geometry	Anderson, 1983-1986 [18]
LISP Tutor	LISP Programming	Anderson, 1983-1986 [19]
QUEST	Electronics	White & Fredriksen, 1984-1988 [20]
IMTS	Navy Helicopter	Behavioral Technology & Search Technology, 1984-1988 [21]
MITT	Space Shuttle	Johnson, Norton, Duncan & Hunt, 1988 [22]

Table 1. Examples of expert systems for training

instructional environment, pedagogical expert, context expert, student modeling, interface development, and total system integration can become overwhelming. Maital and Morgan [5] suggest that even a modest expert system can cost $1 million. The expert system DENDRAL took more than 40 person-years to construct, and INTERNIST took more than 20 person-years to develop [5]. These two examples contain only the expert module, and the integration of the other modules adds further to the cost.

However, increasing emphasis is being placed on the ability to generalize ITS software as successful modules are developed. Developers will not have to recreate the wheel (i.e., build each module from scratch) each time an ITS is constructed. The cost of developing an expert module can be greatly reduced by using expert system software development packages (shells). The shells permit the developer to build a knowledge base without having to develop the software that processes the rules needed to reach conclusions. The shells are designed to communicate with the rest of the computer environment [23]. Such software tools can be used to build not only the domain expert module but also the instructional expert module.

The cost of ITS development can be decreased by using a proven CBI system as a basis for the instructional environment. The ITS modules can be used to enhance a CBI system that has withstood continuing instructional use. The Intelligent Maintenance Training System (IMTS) [21] was built using the Generalized Maintenance Training System. The Microcomputer Intelligence for Technical Training Systems [22] was constructed using the CBI simulation Framework for Aiding the Understanding of Logical Troubleshooting [24]. The task of student modeling can be made more manageable by also using existing CBI systems as a basis for ITS development.

Another resource related problem that may stifle the implementation of ITSs is the origin of the development dollar. In most cases, ITS funding comes from "research" dollars, thus, operational entities do not have the vested interest to make the ITS solve immediate training problems. When ITS development finds increased funding from operational budgets, rather than research budgets, more workable ITSs are likely to emerge.

Personnel

ITS development has been the exclusive domain of cognitive psychologists, computer scientists, and educational technologists who have focused on specific aspects of intelligent tutoring. The psychologists have attended to the modeling of the student, the expert, and the instructor. The computer scientists have concentrated on improving methods of knowledge representation and computer inference. They have also specialized in building better programming environments for ITS development. Educational technologists, with interests toward pedagogy and the user, have been more likely to concentrate on interface design and knowledge base development.

As ITS development has evolved, increasing emphasis has been placed on multi-disciplined teams consisting of cognitive psychologists, computer scientists, engineers, and educators. Additionally, job experience and graduate programs are producing individuals with appropriate multi-discipline backgrounds and training.

Knowledge engineering captures the knowledge of a human expert so that it can be translated to the rules for the knowledge base. Early knowledge engineering efforts in ITS development were fairly unstructured, therefore, inefficient. Knowledge engineering researchers worked with human subject matter experts to model the ways in which experts understood the domain being studied. The knowledge engineers also observed the way experts decide what information is necessary to make intelligent decisions within the domain. After lengthy discussions with the experts, the researchers experimented with different ways to model the results. The knowledge engineering process was long and iterative. Therefore, most of the software that was developed was *on the fly* as it was needed. This process was necessary and appropriate in the beginning stages of ITS research and development, but this is no longer the case.

The knowledge engineering process is becoming more efficient because several ITSs have been developed. Technical instructors are the most appropriate subject matter experts [25,26]. Technical instructors not only provide context expertise but also provide the necessary insights to develop the rules for the instructional expert. It is important that technical instructors be made equal partners on any ITS development team.

Equipment

Traditionally, ITS research and development was conducted on AI workstations. Generally, the dedicated workstations had large memory and storage capabilities, and the computing environment was tailored to AI software development. But, the high cost of the dedicated workstations limited their availability in training centers. Furthermore, the computing environment interface, suited to AI programmers, did not adapt well to novice computer users who were using the system for training.

Off-the-shelf microcomputers may be a potential solution to the high cost problem of AI workstations. Microcomputers, with their speed; storage; memory; interfaces; software; and affordability, are a reasonable alternative for ITS development and delivery.

Attitudes

Some AI and/or ITS researchers have the attitude that their accountability to their sponsoring agencies is minimal because their science is so complex. Missing development and delivery deadlines and blaming AI workstations has become a standard mode of operation. Sponsors, who have already committed extensive resources, find themselves accepting less than what they were promised. Providing more funding for the researchers to continue or complete projects has been the standard, unacceptable solution.

ITSs' problems stemming from unfulfilled promises are perpetuated by the attitude that good work on one or two of the modules (Figure 1) is enough. Therefore a trade-off is made. Extensive resources may be used in the development of the expert module or student modeling routine causing insufficient attention to be paid to the interface or instructional environment. The result is *vaporware*: ITSs that work fine for journal articles, technical reports, conference proceedings, and limited demonstrations orchestrated by the developers. However, the resulting ITSs lack the necessary integration of all the modules and are not suited to actual instructional use by students.

On a positive note, ITS developers' and sponsors' attitudes are changing. Due to the experience developers have accumulated, they can accurately estimate the amount of time and money needed for ITS development. Sponsors have more confidence in the final product, because they have more experience writing specifications for ITS projects. Developers and sponsors now respect the importance of in-process reviews, on-going formative evaluations, and close working relationships throughout ITS research and development efforts.

The attitude discussed above is not meant to suggest that all ITS research must result in operational training systems. The importance of basic research cannot be discounted. Research in such areas as learning, cognition, memory, natural language interface, human modeling, human factors, etc. insures that the operational systems are based on sound scientific principles.

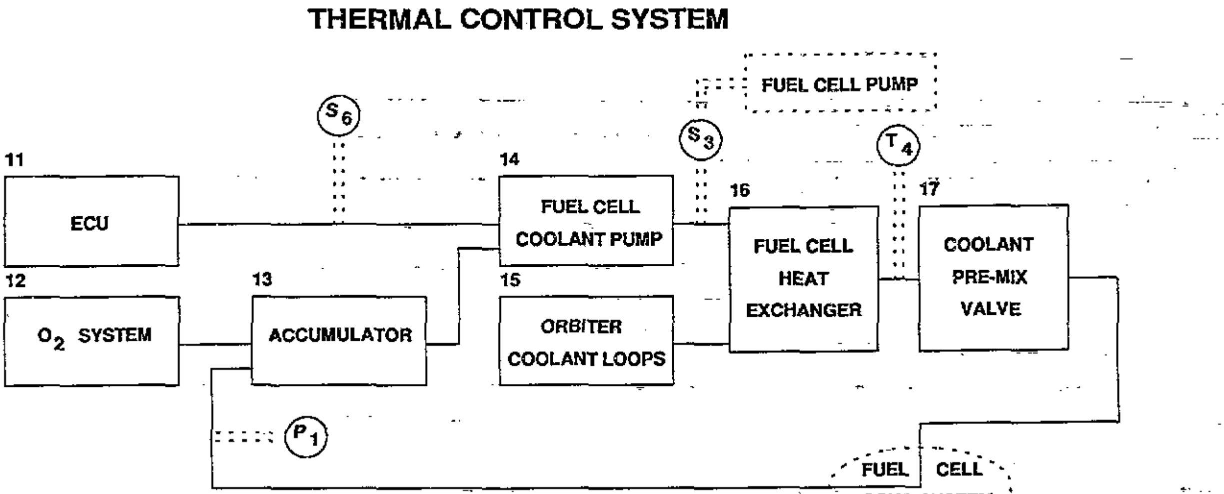
A GOOD IDEA...

This section describes a successful ITS project that overcame the four potential obstacles to ITS development: Microcomputer Intelligence for Technical Training (MITT) [22]. The description follows the ITS modules shown in Figure 1. MITT contains all five modules.

MITT was developed in cooperation with the Air Force Human Resources Laboratory at Brooks Air Force Base, Texas working in conjunction with personnel at the NASA Johnson Space Center in Houston. The system provides space shuttle fuel cell diagnostic training for astronauts and flight controllers. Flight controllers are engineering personnel that provide flight crew support from ground stations.

Instructional Environment

MITT's instructional environment is based on a proven diagnostic training simulation that Search Technology, Inc. has been involved with for more than 10 years. The program, Framework for Aiding the Understanding of Troubleshooting (FAULT) [24], contains an inference engine tailored to technical training and fault diagnosis. This CBI simulation software made it possible to quickly create an acceptable simulation of the fuel cell's subsystem. Furthermore, the interface and expert modules could be easily integrated with the FAULT program. Figure 2 shows a hardcopy functional flow chart that was developed for the FAULT fuel cell knowledge base.



GAUGE	NAME	LOCATION(S)	EFFECT
S_3	Coolant Pump Tb	R1C, SysSumm1, & Spec 69	T14 - 16
S_6	Rdy for LD Tb	R1C	T11 - 14
T_4	Coolant Temp	Spec 69	T16 - 17
P_1	Coolant Pressure	Spec 69	T13 - 17

Figure 2. A subsystem of the space shuttle fuel cell

Expert Module

MITT's expert module contains two representations of the fuel cell system. Figure 2 shows the diagram that is the basis for the functional expert. Written in C, this expert uses a connectivity matrix representing the functional relationship among the parts. The second expert, using CLIPS (the NASA expert system shell), is a series of rule-based statements that contains information regarding specific NASA fuel cell troubleshooting procedures. The two experts can be independently accessed by the student, however, the advice given by the experts is integrated to insure consistency.

Student Module

MITT's student module is also written in C. It is predominantly a tally of the student's actions throughout the simulation. The student module is updated by the FAULT simulation and by the two experts. The student module provides feedback upon completion of each problem as shown in Figure 3.

Congratulations! You have corrected the problem.

Number of students completing this problem:	78
Number of students who quit before solving:	2
Number of students who received a time out:	6

	Yours	Average
Number of minutes for diagnosis:	7	8.4
Number of errors made:	2	1.6
Number of displays accessed:	14	17.3
Number of times procedural advice used:	1	1.8
Number of times functional advice used:	2	1.3

Figure 3. Student feedback from MITT

Instructor Module

The instructional expert is a rule-based C routine that notes certain errors and intervenes when these errors are made. Examples of these errors include student actions that result from the student's misunderstanding of the MITT system or errors in logic and procedure during troubleshooting.

Student Interface

Although every module of an ITS is critically important, the user interface ultimately delivers the instruction. MITT is designed to deliver instruction so that even novices can use the system. It is replete with help for using the MITT system and also with technical information and diagnostic advice on the fuel cell domain. The primary user input device is a mouse, but a keyboard can also be used.

...THAT WORKS.

The MITT system was independently evaluated by AFHRL and NASA during the spring of 1988. It received favorable reviews by instructors and incumbent flight controllers at the Johnson Space Center. A second phase of the MITT research is planned. The work will embellish the fuel cell knowledge base and will enhance MITT capability. In addition, a MITT authoring system will also be developed.

This section discusses how the MITT project overcame the four potential problems faced by ITS developers: attitudes, equipment, personnel, and resources.

Attitude

One goal of the developers of MITT was to create a microcomputer-based ITS in less than six months for less than one person-year of effort. To accomplish this, all of the project's participants had to adopt the attitude that they would not accept anything less than a fully operational ITS by the contract's deadline. It was understood that it would be necessary to capitalize on existing software and CBI courseware to complete the project on time. The project was as much software integration as it was software development. The timetable made it impossible to develop radically new approaches to expert modules, instructional modules, or student modeling. There was no time for false starts or mid-course changes.

Equipment

A second goal was to prove that ITSs can be developed on relatively inexpensive, off-the-shelf microcomputers. Therefore, MITT was designed on an IBM/AT for compatibility with AFHRL's Zenith computers and NASA's AT&T hardware. MITT requires 640K of RAM and 1.1 Mb of hard disk storage. The graphics are compatible with the IBM Color Graphics Adapter insuring that the graphics can be transported across a variety of hardware systems.

The instructional environment is written in C to facilitate communication with the expert system shell, CLIPS. MITT relies heavily on color graphics for information display. The graphics were developed using a combination of PC Paintbrush and In-A-Vision. Both software packages are commercially available at a reasonable price. Using IBM equipment, the C language, and CLIPS insured that the final product can be used at any Air Force or NASA training center.

Personnel

The team that developed MITT had a good mix of disciplines. The AFHRL participants were much more than project managers due to their formal education in computer science and psychology. Additionally, AFHRL had considerable experience funding ITS projects which helped them clearly define their expectations and foresee possible problems and delays.

NASA's primary participant was a subject matter expert on the space shuttle fuel cell. He was also a technical and simulator instructor at NASA Johnson Space Center. The NASA personnel understood the six month schedule restrictions and cooperated accordingly.

Search Technology's team consisted of a senior instructional technologist and a senior systems engineer. Both participants had extensive CBI, simulation, and human factors experience. An instructional psychologist, with extensive ITS development experience, and a computer scientist, with experience in C and software integration, were also on the Search Technology team. The computer graphics artist had experience with the software packages that were used. All participants knew that the development timetable was a high priority item.

Resources

The final cost of an ITS is often the best kept secret of the entire project. This is not the case with the MITT project. The project was completed on time for slightly more than $100,000. This figure does not include the time spent by the NASA subject matter expert. If the expert's time was included, an additional person-month would be added to the total cost. The reasonable cost was achieved by early and precise specifications of the final ITS, off-the-shelf hardware and software selection, experienced government and contractor cooperation, and a determined attitude toward the delivery timetable and cost.

CONCLUSION

Intelligent Tutoring Systems are indeed good ideas. As it becomes increasingly possible to develop and deliver ITSs at reasonable costs, there will be many more successfully produced. ITSs will become more popular when developed by multi-disciplined personnel using affordable hardware and software systems who have a pragmatic attitude toward timetables and delivery dates. MITT has proven that ITSs can be developed within the time and money constraints set by training organizations within schools, businesses, industries, and the armed forces.

ABOUT THE AUTHOR

William B. Johnson is a Principal Scientist with Search Technology, Inc. of Atlanta. Search Technology specializes in contract research and development related to human factors, particularly, applications of artificial intelligence to job aiding and training across a variety of domains.

Johnson received his Ph.D. in Education from the University of Illinois. His research at the university focused on the transfer of training from computer-based simulation to live system diagnostic performance in the aviation maintenance domain.

Johnson has taught at the secondary, community college, and university levels. He has developed computer-based training for the Army, Navy, Air Force, NASA, and industrial customers including nuclear power companies. Johnson has numerous publications related to the use of computer-based simulation for technical training. His most recent publications appear as chapters in two 1988 books published by Lawrence Erlbaum Publishers: *Intelligent Tutoring Systems: Lessons Learned* and *Foundations of Intelligent Tutoring Systems*.

ACKNOWLEDGMENTS

The MITT project was sponsored by the U.S. Air Force Human Resources Laboratory at Brooks Air Force Base, Texas in cooperation with the NASA Johnson Space Center, Texas. The author acknowledges his colleagues at Search Technology, including Jeffrey E. Norton, Phillip C. Duncan, and Ruston M. Hunt for their continuing contributions to the MITT research and development.

REFERENCES

(1) Bulkeley, W.M. (1988, June 6). Computers failing as teaching aids. *The Wall Street Journal*, p. 25.

(2) Suppes, P. (1988). The future of intelligent tutoring systems: Problem and potential. *Intelligent Tutoring Systems - 1988 Conference*. Montreal, Canada: University of Montreal, p. 5.

(3) Gellman, H. (1986). Is artificial intelligence real? *CMA Magazine*, September-October, p. 60.

(4) Jasany, L.C., & Kolsch, J.R. (1986). Artificial intelligence: Facts and fantasies. *Production Engineering*, November, pp. 62-71.

(5) Maital, S., & Morgan, K. (1985). Expert systems: Where's the beef? *Across the Board*, October, pp. 42-49.

(6) Dreyfus, H., & Dreyfus, S. (1986). Why computers may never think like people. *Technology Review*, January, pp. 42-61.

(7) Sleeman, D., & Brown, J.S. (1982). *Intelligent Tutoring Systems*. New York, NY: Academic Press.

(8) Yazdani, M. (1987). Intelligent tutoring systems: An overview. In R.W. Lawler and M. Yazdani (Eds.), *Artificial intelligence and education volume 1: Learning environments and tutoring systems* (pp. 183-201). Norwood, NJ: Ablex Publishing.

(9) Psotka, J., Massey, R., & Mutter, S. (1988). *Intelligent tutoring systems: Lessons learned.* Hillsdale, NJ: Lawrence Erlbaum Associates.

(10) Burns, H.L, & Capps, C.G. (1988). Foundations of intelligent tutoring systems: An introduction. In M.D. Polson and J.J. Richardson (Eds.), *Foundations of Intelligent Tutoring Systems* (pp. 1-19). Hillsdale, NJ: Lawrence Erlbaum Associates.

(11) Polson, M.C., & Richardson, J.J. (1988). *Foundations of Intelligent Tutoring Systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

(12) Wenger, E. (1987). *Artificial intelligence and tutoring systems*. Los Altos, CA: Morgan Kaufmann.

(13) Carbonell, J.R., & Collins, A. (1973). Natural semantics in artificial intelligence. *Proceedings of the Third International Joint Conference on Artificial Intelligence* (pp. 344-351). Los Altos, CA: Morgan Kaufmann.

(14) Brown, J.S., Burton, R.R., & deKleer, J. (1982). Pedagogical, natural language, and knowledge engineering techniques in SOPHIE I, II and III. In D.H. Sleeman and J.S. Brown (Eds.), *Intelligent Tutoring Systems* (pp. 227-282). New York, NY: Academic Press.

(15) Burton, R.B., & Brown, J.S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J.S. Brown (Eds.), *Intelligent tutoring systems* (pp. 79-98). New York, NY: Academic Press.

(16) Hollan, J.D., Hutchins, E.L., & Weitzman, L. (1984). STEAMER: An interactive inspectable simulation-based training system. *AI Magazine*, *5*(2), pp. 15-27.

(17) Clancey, W.J. (1987). *Knowledge-based tutoring: The GUIDON program*. Cambridge, MA: MIT Press.

(18) Anderson, J.R., Boyle, C.F., & Yost, G. (1985). The geometry tutor. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 1-7). Los Altos, CA: Morgan Kaufmann.

(19) Anderson, J.R., & Skwarecki, E. (1986). The automated tutoring of introductory computer programming. *Communications of the ACM, 29*, 842-849.

(20) White, B.Y., & Frederiksen, J.R. (1986). Qualitative models and intelligent learning environments. In R. Lawler and M. Yazdani (Eds.), *AI and education: Learning environments and intelligent tutoring systems* (pp. 281-304). Norwood, NJ: Ablex Publishing.

(21) Towne, D.M. (1988). The generalized maintenance trainer: Evolution and revolution. In W.B. Rouse (Ed.), *Advances in man-machine systems research: Vol. 3* (pp. 1-63). Greenwich, CT: JAI Press.

(22) Johnson, W.B., Norton, J.E., Duncan, P.C., & Hunt, R.M. (1988). Development and demonstration of microcomputer intelligence for technical training (MITT): (AFHRL-TP-88-8). Brooks AFB, TX: Air Force Human Resources Laboratory.

(23) Olsen, B., Pumplin, B., & Williamson, M. (1987). The getting of wisdom: PC expert system shells. *Computer Language, Vol. 4*(3), pp. 117-150.

(24) Johnson, W.B. (1987). Development and evaluation of simulation-oriented computer-based instruction for diagnostic training. In W.B. Rouse (Ed.), *Advances in man-machine systems research: Vol. 3* (pp. 99-125). Greenwich, CT: JAI Press.

(25) Johnson, W.B. (1988). Developing expert system knowledge bases for technical training. In L.D. Massey, J. Psotka, and S.A. Mutter (Eds.), *Intelligent tutoring publishing systems: Lessons learned* (pp. 21-33). Hillsdale, NJ: Lawrence Erlbaum Associates.

(26) Johnson, W.B. (1988). Pragmatic considerations in development and implementation of intelligent tutoring systems. In M. Polson and R.J. Richardson (Eds.), *Foundations of intelligent tutoring systems* (pp. 191-207). Hillsdale, NJ: Lawrence Erlbaum Associates.