N91-24042

Intelligent Computer-Aided Training and Tutoring

R. Bowen Loftin, Ph.D.
University of Houston-Downtown
One Main Street
Houston, Texas

Robert T. Savely, Chief
PT4/Software Technology Branch
NASA/Johnson Space Center
Houston, Texas

INTRODUCTION

Motivation

Training is a major endeavor in all modern societies: new personnel must be trained to perform the task(s) which they were hired to perform, continuing personnel must be trained to upgrade or update their ability to perform assigned tasks, and continuing personnel must be trained to tackle new tasks. Common methods include training manuals, formal classes, procedural computer programs, simulations, and on-the-job training. The latter method is particularly effective in complex tasks where a great deal of independence is granted to the task performer. Of course, this training method is also the most expensive and may be impractical when there are many trainees and few experienced personnel to conduct on-the-job training.

NASA's training approach has focussed primarily on on-the-job training in a simulation environment for both crew and ground-based personnel. This process worked relatively well for both the Apollo and Space Shuttle programs. Space Station Freedom and other long range space exploration programs coupled with limited resources dictate that NASA explore new approaches to training for the 1990s and beyond.

This report describes specific autonomous training systems based on artificial intelligence technology for use by NASA astronauts, flight controllers, and ground-based support personnel that demonstrate an an alternative to current training systems. In addition to these specific systems, the evolution of a general architecture for autonomous intelligent training systems that integrates many of the features of "traditional" training programs with artificial intelligence techniques is presented. These Intelligent Computer-Aided Training (ICAT) systems would provide, for the trainee, much of the same experience that could be gained from the best on-the-job training. By integrating domain expertise with a knowledge of appropriate training methods, an ICAT session should duplicate, as closely as possible, the trainee undergoing on-the-job training in the task environment, benefiting from the full attention of a task expert who is also an expert trainer. Thus, the philosophy of the ICAT system is to emulate the behavior of an experienced individual devoting his full time and attention to the training of a novice—proposing challenging training scenarios, monitoring and evaluating the actions of the trainee, providing meaningful comments in response to trainee errors, responding to trainee requests for information, giving hints (if appropriate), and remembering the strengths and weaknesses displayed by the trainee so that appropriate future exercises can be designed.

Background

Since the 1970s a number of academic and industrial researchers have explored the application of artificial intelligence concepts to the task of teaching a variety of subjects [1, 2, 3] (*e.g.*, computer programming in Lisp [4, 5] and Pascal [6], economics [7], and, geography [8]). The earliest published reports which suggested the applications of artificial intelligence concepts to teaching tasks appeared in the early 1970s [8, 9]. Hartley and Sleeman [9] actually proposed an architecture for an intelligent tutoring system. However, it is interesting to note that, in the sixteen years which have passed since the appearance of the Hartley and Sleeman proposal, no agreement has been reached among researchers on a general architecture for intelligent tutoring systems [2].

Along with the extensive work on intelligent tutoring systems for academic settings has come the development of systems directed at training. Among these are Recovery Boiler Tutor [10], SOPHIE [11], and STEAMER [12]. These differ from the tutoring systems mentioned above in providing a simulation model with which the student or trainee interacts. Although these intelligent training systems each use the interactive simulation approach, they each have very different internal architectures. Further, there appears to be no agreement, at present, on a general architecture for such simulation training systems. The work reported here builds on these previous efforts and our own work [13, 14, 15, 16] to develop specific intelligent training systems as well as a general approach to the design of intelligent training systems which will permit the production of such systems for a variety of tasks and task environments with significantly less effort that is now required to "craft" such a system for each application.

APPLICATIONS

The ICAT architecture was originally applied to a training system for NASA flight controllers learning to deploy satellites from the Space Shuttle. The same architecture has been used in the construction of ICAT systems for training astronauts for Spacelab missions and engineers who test the Space Shuttle main propulsion system. Although these tasks are quite different and are performed in very dissimilar environments, the same system architecture has proven to be adaptable to each. Below is a brief summary of the specific systems that have been built or are currently under development:

PD/ICAT: [Payload-assist module Deploys/ICAT System]

A comprehensive intelligent computer-aided training system for use by Flight Dynamics Officers in learning to deploy PAM (Payload-Assist Module) satellites from the Space Shuttle. PD/ICAT contains four expert systems that cooperate via a blackboard architecture.

VVL/ICAT: [Vacuum Vent Line/ICAT System]

A PC-based intelligent computer-aided training system for use by mission and payload specialists in learning to perform fault detection, isolation, and reconfiguration on the Spacelab VVL system. VVL/ICAT consists of an integrated expert system and graphical user interface.

MPP/ICAT: [Main Propulsion Pneumatics/ICAT System]

A comprehensive intelligent computer-aided training system for use by test engineers at NASA/Kennedy Space Center in learning to perform testing of the Space Shuttle Main Propulsion Pneumatics system. MPP/ICAT is currently under development and makes use of the same architecture as PD/ICAT.

IPS/ICAT: [Instrument Pointing System/ICAT System]

A comprehensive intelligent computer-aided training system for use by payload and mission specialists at NASA/Johnson Space Center and Marshall Space Flight Center in learning to utilize the IPS on Spacelab missions. IPS/ICAT is currently under development and makes use of the same architecture as PD/ICAT.

A GENERAL ARCHITECTURE FOR INTELLIGENT TRAINING SYSTEMS

The projects described in the previous section have served as vehicles to aid in the design and refinement of an architecture for intelligent training systems that has significant domain-independent elements and is generally applicable to training in procedural tasks common to the NASA environment. The ICAT system architecture is modular and consists of five basic components:

- A user interface that permits the trainee to access the same information available to him in the the task environment and serves as a means for the trainee to take actions and communicate with the intelligent training system.
- A domain expert which can carry out the task using the same information that is available to the trainee and which also contains a list of "mal-rules" (explicitly identified errors that novice trainees commonly make).
- A training session manager which examines the actions taken by the domain expert (of both correct and incorrect actions in a particular context) and by the trainee and takes appropriate action(s). [17]

- A trainee model which contains a history of the individual trainee's interactions with the system together with summary evaluative data.
- A training scenario generator that designs increasingly-complex training exercises based on the knowledge of the domain expert, the current skill level contained in the trainee's model, and any weaknesses or deficiencies that the trainee has exhibited in previous interactions. [18, 19]

Figure 1 contains a schematic diagram of the ICAT system. Note that provision is made for the user to interact with the system in two distinct ways and that a supervisor may also query the system for evaluative data on each trainee. The blackboard serves as a common repository of facts for all five system components. With the exception of the trainee model, each component makes assertions to the blackboard, and the expert system components look to the blackboard for facts against which their rules pattern match. A comprehensive effort has been made to clearly segregate domain-dependent from domain-independent components.

The ICAT architecture described above was originally implemented in a Symbolics 3600 Lisp environment using Inference Corporation's ART for the rule-based components. The architecture is currently available for unix workstations. The user interface is implemented in X-Windows, the rule-based components in CLIPS [CLIPS is the acronym for a NASA-developed expert system shell written in C], and supporting code in C.

TRAINING PERFORMANCE

The original system developed with this architecture (PD/ICAT) has been used by both expert and novice flight controllers at NASA/Johnson Space Center. An extensive investigation of the performance of novices using the system has been conducted. Figure 2 shows two measures of performance: (1) the time required to perform the nominal task as a function of the number of training experiences and (2) the number of errors made during the performance of the nominal task as a function of the number of training experiences. It is interesting to note that, although the novices used in this investigation had very different levels of prior experience related to the task, all novices rapidly approached the same level of proficiency.

A TECHNOLOGY SPINOFF: THE INTELLIGENT PHYSICS TUTOR

Introduction

The integration of the computer into the K-12 instructional program began in the 1960s and has accelerated with the availability of inexpensive computing hardware and a growing amount of useful instructional software [20]. The bulk of the computer-aided instruction (CAI) available today is limited to rather simple programs that are useful for drill-and-practice, automated "page-turning", and the administration of objective examinations. Only a small percentage of the educational software available today for grades K-12 uses simulation, extensive branching to diagnose and remediate, and/or artificial intelligence (AI) technology [21].

Objectives

The ICAT technology described above has been brought to bear on a specific tutoring task of interest to the nation's educational institutions: the development of an intelligent tutoring system (ITS) for use in a high school or introductory college physics course. The goal of this ITS is not the conveyance of facts and concepts but rather the transfer of problem solving skills to the student. Ultimately, this project will not only produce a useful teaching aid for students enrolled in high school or introductory college physics courses, but will also provide a development structure suitable for building additional intelligent tutors for other academic subjects which require the application of problem solving skills (*e.g.*, mathematics, chemistry, and engineering).

Technology Transfer

The work described above led the NASA/JSC Office of Technology Utilization to suggest, in February, 1988, that this technology be applied to the development of an intelligent tutoring system suitable for use in the nation's educational institutions. The outcome of this technology transfer project would be a highly interactive and "intelligent" program for tutoring high school and college students in solving physics problems. This product could be mass produced and delivered economically to high schools and colleges nationwide. In addition, the methodologies employed and much of the software developed could be used to produce intelligent tutors for other problem solving domains. The effect of this technology transfer would clearly be profound and is entirely consistent with NASA's charter.

Project Description

Background. By the beginning of this decade a number of researchers had attempted to develop intelligent tutors for selected tutoring or training tasks. Specific systems such as GUIDON (medical diagnosis) [22, 23], SOPHIE (electronic troubleshooting) [11], STEAMER (naval steam propulsion systems) [12, 24], WUSOR (reasoning) [25], PROUST (Pascal programming) [6, 26], the GEOMETRY Tutor (plane geometry) [27, 28], and the LISP Tutor (LISP Programming) [4] have been built and tested.

Perhaps the most successful of these, the LISP Tutor, has been shown to have a significant effect on student mastery of a skill. For example, in one controlled experiment, two groups of students attended the same lectures on LISP programming, and one group completed the exercises in the traditional manner while the other group used the LISP Tutor. The students using the LISP Tutor spent 30% less time on the exercises and scored 43% better on a post-test than those not using the LISP Tutor [5]. Further classroom use of the LISP Tutor at Carnegie-Mellon University for a one-semester course in LISP showed that students were able, as a whole, to achieve a full letter grade improvement in their final course grade by using the LISP Tutor when compared with previous classes that did not use the tutor [29]. Interestingly, poor students demonstrated the most significant performance improvement [5].

Approach. An ITS containing at least one semester of exercises for the first course in high school and introductory college physics is now under development. If time is available, additional exercises appropriate for the second semester will also be developed. The ITS will be designed for use by students concurrently enrolled in the standard high school or introductory college physics course and will be used to tutor them in physics problem solving. In order to provide an interactive environment suitable for detailed simulations of physical phenomena, the ITS will be delivered as a part of a workstation built around the Apple Macintosh II computer. The interface will make full use of the Macintosh II's capabilities—using high-resolution graphics, color and sound to deliver a sophisticated real-time simulation as a means of enhancing the students' ability to relate the tutorial environment to that of the laboratory and the "real world". The strategy (adapted, in part, from the LISP Tutor) of examining user input continuously and providing immediate feedback upon detecting errors is utilized. The intelligent physics tutor is not intended to replace a human instructor or to replace the existing program of physics instruction. Instead, the tutor that is under development will provide an interactive environment for the application of physics concepts to the solution of problems. The principal goal of the tutor will be to enable the average student to efficiently acquire problem solving skills necessary for successful mastery of high school or introductory college physics. It is anticipated that a large fraction of such physics classes could be served by the tutor, freeing the human instructor to work more closely with the slower students as well as with the more advanced students. The tutorial lessons will be integrated with the lecture/laboratory portions of the typical course, and students will have independent access to the tutor for completion of their homework.

Since the intent of the tutor is to complement an existing textbook and course in which concepts will be logically introduced, it is essential that the tutor be adaptable to a variety of texts. To this end the tutor lessons (here "lesson" refers to a sequence of student-tutor interactions that lead to student mastery of problem solving skills that pertain to a specific concept) are keyed to physics concepts. The instructor will be provided the capability of assembling the lessons into an order consistent with the curriculum followed in the classroom and of establishing appropriate dependencies among the lessons. In addition, the instructor will have the means of choosing terms and symbols compatible with their textbook and/or preferences. Each instance of the physics tutor will contain a "global" strategy to govern the student's progression from lesson to lesson, but the tutor strategy and the semantics used in a given lesson may be altered by the instructor. The lessons for these topics will vary in length according to the complexity of problems associated with each topic. For example, the lesson for uniform acceleration in one-dimension will contain many more exercises, of varying difficulty, for the student than the lesson related to kinetic energy for rotational motion.

For each tutor lesson the student progresses from the solution of a "one-step" problem in which the available facts and the required item(s) are clearly delineated and a single relation allows the student to obtain the required result, to multi-step problems in which intermediate results or interdependencies are required to obtain the information necessary for the final solution, to problems in which the student must, given the required item(s), obtain the necessary data through the observation of a simulation.

In the "text-only" problems the student, in initial tutoring sessions, may be provided with a structured environment within which to solve the problem. The listing of given information and the identification of the required item(s) can be enforced by the tutor. The student is provided regions in which to assemble lists of given and required data obtained from the problem or implied by the problem. By using a natural and powerful interface, a student can use assemble his or her solution on the computer screen as if working the problem on a piece of notebook paper. At any time during the interaction the tutor can intervene by "popping-up" a window and/or providing audible feedback. In addition, the student can obtain assistance (examples, hints, etc.) by using the help menu item.

For the most advanced exercises in many lessons the tutor can lead the student in abstracting real-world problems in a manner that facilitates their solution via the methodologies conveyed in the preceding exercises. Experience has shown that most students have great difficulty in reading text describing a problem (or even viewing a photograph, detailed drawing, or video image) and producing an abstract representation of the problem that fits the problem solving patterns they have been shown by an instructor or in a textbook. Consider, for example, a video segment of a skier on a uniform slope. Most novice problem solvers have difficulty in abstracting such a scene by drawing a free-body diagram showing the skier as a point mass with the three forces (gravity, friction, and the normal force) acting on him or her. Such a step, however, is essential in order to apply Newton's Second Law and and determine the skier's acceleration. The tutorial strategy in this case is accomplished by providing video stills or video segments of a physical situation and overlaying a graphical abstraction. By causing the video image to fade and the graphical abstraction to remain, students can rapidly gain experience in abstracting the essential elements of a physical description. Student mastery can be tested by allowing the student to construct the appropriate graphical abstraction from a video image and, eventually, from a written problem statement.

Error detection and remediation occur at a local level with each student action compared to expected correct and incorrect actions. Based on the nature of each student error, appropriate feedback is given to enable the student to understand and correct his or her error. A global tutor strategy draws on a student model which identifies lacks of skills and knowledge demonstrated by the student in attempting to solve problems within the tutor environment. The number, type, and order of the exercises encountered by the student is determined by heuristic rules which examine the student model and the performance of the student on the exercises completed in the current lesson. The student model is also used to determine the way in which assistance is provided to the student. That is, the type, length, and tenor of messages can be tailored to the student based on the history of their experience with the tutor as contained in the student model.

As the tutor's development has proceeded, a number of components have been created which can serve as the basis for the production of tutors for problem-solving domains other than physics. The interface development has produced a set of basic objects that can also be used to build interfaces for tutors in domains other than physics. An environment for encoding the domain knowledge and the global tutoring strategy has been developed to facilitate the creation of the rule-based portions of the tutor. This environment will also be adaptable to the creation of rule-based components for other tutoring systems.

The first year of this project was dedicated to the development of prototypes in cooperation with a physics teacher from a local school district. The prototypes have been in use in a local high school since April, 1989, and a evaluation/ development/refinement feedback loop has been established. Beginning in September, 1990, a semester of physics exercises is being used to begin formal testing with a students in League City, Texas and Columbus, Ohio. Finally, during the third year of the project, refinement of the ITS and its further extension will occur.

The ITS design and evaluation has been carried out by Dr. R. Bowen Loftin (University of Houston-Downtown), Dr. Steve Brown (Cognitive Systems Technologies), and Ms. Beverly Lee (a local high school physics teacher). Computer Sciences Corporation has provided personnel for the principal code development activity. Gary Riley and Brian Donnell of the Software Technology Branch at NASA/Johnson Space Center have also contributed to the design and coding of some tutor elements.

CONCLUSIONS

A general architecture for ICAT systems has been developed and applied to the construction of three ICAT systems for very different tasks. Use by novices of an ICAT application built upon this architecture has shown impressive trainee performance improvements. With further refinement and extension, this architecture promises to provide a common foundation upon which to build intelligent training systems for many tasks of interest to the government, military, and industry. The availability of a robust architecture that contains many domain-independent components serves to greatly reduce the time and cost of developing new ICAT applications. As an added benefit to the nation, a technology spinoff project has emerged from this activity and promises to make a significant contribution to the secondary and post-secondary education.

REFERENCES

[1] Sleeman, D. and Brown, J.S., eds., *Intelligent Tutoring Systems*, Academic Press, London, 1982.

[2] Yazdani, M. "Intelligent Tutoring Systems Survey," *Artificial Intelligence Review*, Vol. 1, p. 43, 1986.

[3] Wenger, E., *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmann Publishers, Los Altos, CA, 1987.

[4] Anderson, J. R. and Reiser, B. J., "The LISP Tutor," *BYTE*, 159 (April, 1985).

[5] Anderson, J. R., Boyle, C. F., and Reiser, B. J., "Intelligent Tutoring Systems," *Science 228*, 456 (1985).

[6] Johnson, W. L. and Soloway, E., "PROUST," *Byte 10* (4), 179 (April, 1985).

[7] Shute, V. and Bonar, J.G., An Intelligent Tutoring System for Scientific Inquiry Skills, *Proceedings of the Eighth Cognitive Science Society Conference* Amherst, MA, 1986, pp. 353-370.

[8] Carbonell, J.R., "AI in CAI: An Artificial Intelligence Approach to CAI," *IEEE Transactions on Man-Machine Systems 11*(4), 190 (1970).

[9] Hartley, J. R. and Sleeman, D. H., "Towards Intelligent Teaching Systems," *International Journal of Man-Machine Studies 5*, 215 (1973).

[10] Woolf B.P., Blegen, D., Jansen, J.H., and Verloop, A., Teaching a Complex Industrial Process, *Proceedings of the National Conference on Artificial Intelligence*, Vol. II, Austin, TX, 1986, pp. 722-728.

[11] Brown, J. S., Burton, R. R., and de Kleer, J., "Pedagogical, Natural Language and Knowledge Engineering Techniques in SOPHIE I, II, and III," in D. Sleeman and J. S. Brown (eds.), *Intelligent Tutoring Systems* (London: Academic Press, 1982), p. 227.

[12] Hollan, H. D., Hutchins, E. L., and Weitzman, L., "Steamer: An Interactive Inspectable Simulation-based Training System," *AI Magazine*, Summer, 1984, pp. 15-27.

[13] Loftin, R.B., Wang, L., Baffes, P., and Rua, M., An Intelligent Computer-Aided Training System for Payload-Assist Module Deploys, *Proceedings of the First Annual Workshop on Space Operations,*

Automation & Robotics (SOAR '87), NASA/Johnson Space Center, Houston, TX, August 4-6, 1987.

[14] Loftin, R.B., Wang, L., Baffes, P., and Hua, G., An Intelligent Training System for Space Shuttle Flight Controllers, *Telematics and Informatics* 5 (3), 151-161 (1988).

[15] Loftin, R.B., Wang, L., Baffes, P., and Hua, G., "An Intelligent System for Training Space Shuttle Flight Controllers in Satellite Deployment Procedures," *Machine Mediated Learning* 3(1), 41 (1989).

[16] Loftin, R.B., Wang, L., Baffes, P., and Hua, G., "An Intelligent Training System for Space Shuttle Flight Controllers" in *Innovative Applications of Artificial Intelligence*, edited by H. Schorr and A. Rappaport (Menlo Park, CA: AAAI Press, 1989), pp. 15-24.

[17] Loftin, R.B., Baffes, P., and Wang, L, An Approach to the Diagnosis of Trainee Errors in an Intelligent Training System, unpublished report.

[18] Loftin, R.B., Wang, L., and Baffes, P., "Simulation Scenario Generation for Intelligent Training Systems," *Proceedings of the Third Artificial Intelligence and Simulation Workshop*, August 22, 1988, St. Paul, MN, pp. 69-74.

[19] Loftin, R.B., Wang, L., and Baffes,P., "Intelligent Scenario Generation for Simulation-Based Training," *Proceedings of the American Institute for Aeronautics and Astronautics Computers in Aerospace VII Conference*, Monterey, CA, October 3-5, 1989, pp. 581-588.

[20] Bork, A., *Learning with Computers*, Digital Press, Bedford, MA, 1981.

[21] Bork, A., "Nontrivial, Nonintelligent Computer-Based Learning," Technical Report, Educational Technology Center, University of California, Irvine, CA, 1986.

[22] Clancey, W. J., "GUIDON," *J. Computer-Based Instruction 10*(1), 8 (1983).

[23] Clancey, W. J., Shortliffe, E. H., and Buchanan, B. G., "Intelligent Computer-Aided Instruction for Medical Diagnosis," in W. J. Clancey and E. H. Shortliffe (eds.), *Readings in Medical Artificial Intelligence: The First Decade* (Reading, MA: Addison-Wesley, 1984).

[24] Williams, M. D., Hollan, J. D., and Stevens, A. L., "An Overview of STEAMER: an Advanced Computer-assisted Instruction System for Propulsion Engineering," *Behavior Research Methods and Instrumentation 13*(2), 85 (1981).

[25] Carr, B., "Wusor-II: a Computer-aided Instruction Program with Student Modeling Capabilities," AI Lab Memo 417, Massachusetts Institute of Technology, Cambridge, MA, 1977.

[26] Johnson, W. L. and Soloway, E., "PROUST: Knowledge-based Program Debugging," *Proceedings of the Seventh International Software Engineering Conference* , Orlando, Florida, 1984, pp. 369-380.

[27] Farrell, R. G., Anderson, J. R., and Reiser, B., "An Interactive Computer-Based Tutor for LISP," *Proceedings of the American Association for Artificial Intelligence* Conference (Los Altos, CA: Morgan-Kaufman, 1984).

[28] Anderson, J. R., Boyle, C. F., and Yost, G., "The Geometry Tutor," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (Los Altos, CA: Morgan-Kaufman, 1985), pp. 1-7.

[29] Anderson, J. R. and Skwarecki, E., "The Automated Tutoring of Introductory Programming," *Communications of the ACM 29*(9), 842 (1986).

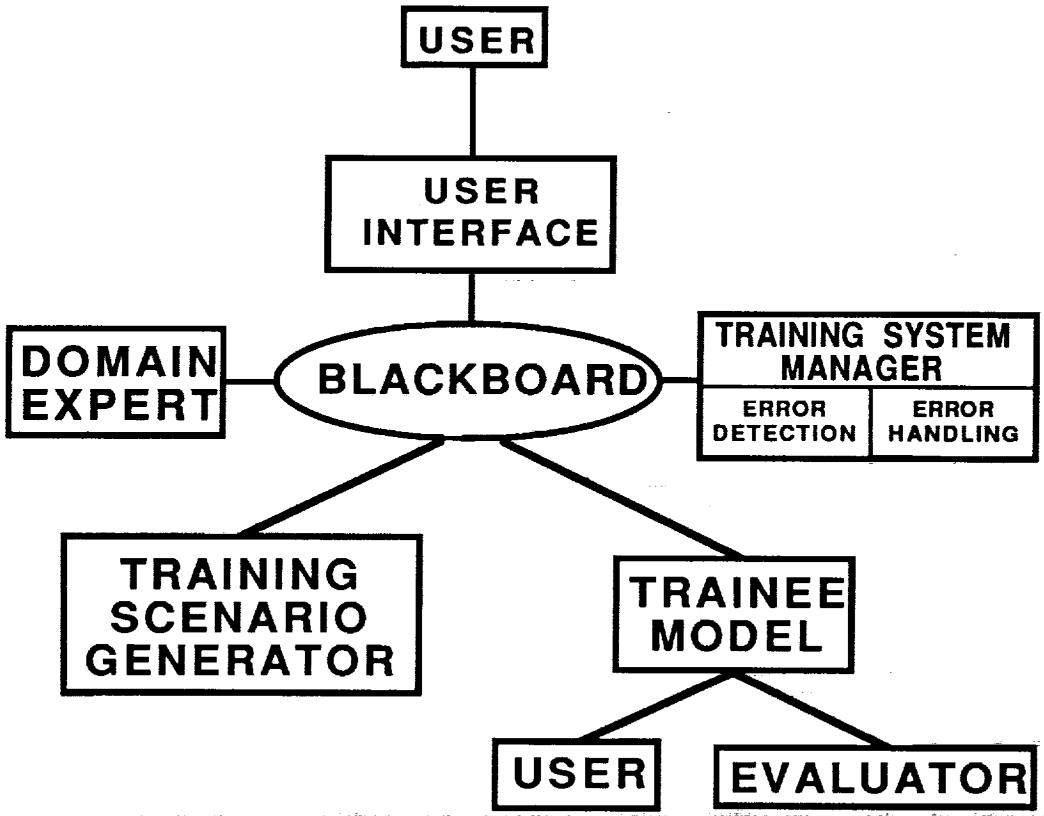

Figure 1. A Schematic Diagram of the General ICAT Architecture

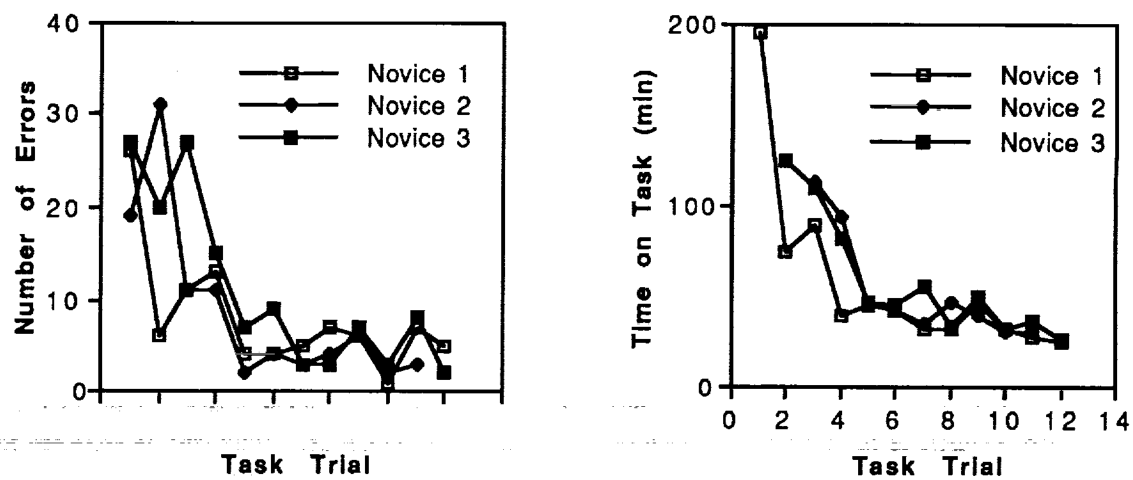

Figure 2. Performance of Novices Using the PD/ICAT System