N88-17214

AN INTELLIGENT TRAINING SYSTEM FOR PAYLOAD-ASSIST MODULE DEPLOYS

R. Bowen Loftin
University of Houston-Downtown
One Main Street
Houston, TX 77002

Lui Wang, Paul Baffes, and Monica Rua
Artificial Intelligence Section, FM72
NASA/Johnson Space Center
Houston, TX 77058

ABSTRACT

An autonomous intelligent training system which integrates expert system technology with training/teaching methodologies is described. The system was designed for use by Mission Control Center (MCC) Flight Dynamics Officers (FDOs) training to perform payload-assist module (PAM) deploys from the orbiter. The system (termed PD/ICAT for Payload-assist module Deploys/Intelligent Computer-Aided Training system) is composed of five distinct components: a user interface, a domain expert, a training session manager, a trainee model, and a training scenario generator. A user interface has been developed which permits the trainee to access data in the same format as it is presented on console displays in the MCC. The interface also permits the trainee to take actions in much the same manner as a FDO in the MCC and provides the trainee with information on the current training environment and with on-line help (if permitted by the training session manager). The domain expert (DeplEx for Deploy Expert) contains the rules and procedural knowledge needed by a FDO to carry out a PAM deploy. DeplEx also contains "mal-rules" which permit the identification and diagnosis of common errors made by the trainee. The training session manager (TSM) examines the actions of the trainee and compares them with the actions of DeplEx in order to determine appropriate responses. A unique feature of the TSM is its ability to grant the trainee the freedom to follow any valid path between two stages of the deploy process. A trainee model is developed for each individual using the system. The model includes a history of the trainee's interactions with the training system and provides evaluative data on the trainee's current skill level. Following each trainee action, evaluative assertions are made by the TSM and used to update the trainee model. A training scenario generator designs appropriate training exercises for each trainee based on the trainee model and the training goals. PD/ICAT is currently being tested by both experienced and novice FDOs in order to refine the system and determine its efficacy as a training tool. Ultimately, this project will provide a vehicle for developing a general architecture for intelligent training systems together with a software environment for creating such systems.

INTRODUCTION

The Mission Operations Directorate (MOD) at NASA/Johnson Space Center is responsible for the ground control of all space shuttle operations. Those operations which involve alterations in the shuttle's orbit characteristics are guided by a FDO who sits at a console in the "front room" of the MCC. Currently, the training of the FDOs (called "fidos") in flight operations is carried out principally through the study of flight rules, training manuals, and "on-the-job training" (OJT) in integrated simulations. From two to four years is normally required for a trainee FDO to be certified for many of the tasks for which he is responsible during shuttle missions. OJT is highly labor intensive and presupposes the availability of experienced personnel with both the time and ability to train novices. As the number of experienced FDOs has been reduced through retirement, transfer (especially of Air Force personnel), and promotion and as the preparation for and actual control of missions occupies most of the MCC's available schedule, OJT has become increasingly difficult to deliver to novice FDOs. As a supplement to the existing modes of training, the Orbit Design Section (ODS) of the MOD has requested that the Artificial Intelligence Section (AIS) of the Mission Support Directorate assist in developing an autonomous intelligent computer-aided training system. After extensive consultation with ODS personnel, a particular task was chosen to serve as a proof of concept: the deployment of a PAM satellite from the shuttle. This task is complex, mission-critical and requires skills used by the experienced FDO in performing many of the other operations which are his responsibility.

The training system is designed to aid novice FDOs in acquiring the experience necessary to carry out a PAM deploy in an integrated simulation. It is intended to permit extensive practice with both nominal deploy exercises and others containing typical problems. After successfully completing training exercises which contain the most difficult problems, together with realistic time constraints and distractions, the trainee should be able to successfully complete an integrated simulation of a PAM deploy without aid from an experienced FDO. The philosophy of the PD/ICAT system is to emulate, to the extent possible, the behavior of an experienced FDO 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 and hints (if appropriate), and remembering the strengths and weaknesses displayed by the trainee so that appropriate future exercises can be designed.

BACKGROUND

Since the 1970's a number of academic and industrial researchers have explored the application of artificial intelligence concepts to the task of teaching a variety of subjects (e.g., geometry, computer programming, medical diagnosis, and electronic troubleshooting). A body of literature is now extant on student models and teaching/tutoring methodologies adapted to intelligent tutoring systems in the academic environment[1]. The earliest published reports which suggested the applications of artificial intelligence concepts to teaching tasks appeared in the early 1970's.[2,3] Hartley and Sleeman[3] actually proposed an architecture for an intelligent tutoring system. However, it is interesting to note that, in the fourteen 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[4]. Nonetheless, a study of the literature on intelligent tutoring systems is an essential starting point for the development of the elements of an intelligent training system.

Among the more notable intelligent tutoring systems reported to date are SOPHIE[5], PROUST[6] and the LISP Tutor[7]. The first of these systems, SOPHIE, was developed in response to a U.S. Air Force interest in a computer-based training course in electronic troubleshooting. SOPHIE contains three major components: an electronics expert with a general knowledge of electronic circuits, together with detailed knowledge about a particular type of circuit (in SOPHIE this was an IP-28 regulated power supply); a coach which examines student inputs and decides if it is appropriate to stop the student and offer advice; and a troubleshooting expert that uses the electronics expert to determine which possible measurements are most useful in a particular context. Three versions of SOPHIE were produced and used for a time but none was ever viewed as a "finished" product. One of the major lacks of the SOPHIE systems was a user model. It is interesting to note that the development of a natural language interface for SOPHIE represented a large portion of the total task.

PROUST and the LISP Tutor are two well-known intelligent tutoring systems that have left the laboratory and found wider applications. PROUST (and its offspring, Micro-PROUST) serves as a "debugger" for finding nonsyntactical errors in Pascal programs written by student programmers. The developers of PROUST claim that it is capable of finding all of the bugs in at least seventy percent of the "moderately complex" programming assignments that its examines. PROUST contains an expert Pascal programmer that can write "good" programs for the assignments given to students. Bugs are found by matching the expert's program with that of the student; mismatches are identified as "bugs" in the student program. This ability is contained in the PROUST "bug rule" component. After finding a bug, PROUST provides an English-language description of the bug to the student, enabling the student to correct his error. The system cannot handle student programs that depart radically from the programming "style" of the expert. The LISP Tutor is currently used to teach the introductory Lisp course offered at Carnegie-Mellon University. This system is based on the ACT (historically, Adaptive Control of Thought) theory and consists of four elements: a structured editor which serves as an interface to the system for students, an expert Lisp programmer that provides an "ideal" solution to a programming problem, a bug catalog that contains errors made by novice programmers, and a tutoring component that provides both immediate feedback and guidance to the student. Evaluations of the LISP Tutor show that it can achieve results similar to those obtained by human tutors. One of its primary features is its enforcement of what its authors regard as a "good" programming style.

TRAINING VERSUS TUTORING

The PD/ICAT system was developed with a clear understanding that training is not the same as teaching or tutoring[8]. The NASA training environment differs in many ways from an academic teaching environment. These differences are important in the design of an architecture for an intelligent training system:

a. Assigned tasks are often mission-critical, placing the responsibility for lives and property in the hands of those who have been trained.
b. Personnel already have significant academic and practical experience to bring to bear on their assigned task.
c. Trainees make use of a wide variety of training techniques, ranging from the study of comprehensive training manuals to simulations to actual on-the-job training under the supervision of more experienced personnel.
d. Many of the tasks offer considerable freedom in the exact manner in which they may be accomplished.

FDO trainees are well aware of the importance of their job and the probable consequences of failure. While students are often motivated by the fear of receiving a low grade, FDO trainees know that human lives, a billion dollar shuttle, and a $100+ million satellite depend on their skill in performing assigned tasks. This means that trainees are highly motivated, but it also imposes on the trainer the responsibility for the accuracy of the training content (i.e., verification of the domain expertise encoded in the system) and the ability of the trainer to correctly evaluate trainee actions. PD/ICAT is intended, not to impart basic knowledge of mathematics and physics, but to aid the trainee in developing skills for which he already has the basic or "theoretical" knowledge. In short, this training system is designed to help a trainee put into practice that which he already intellectually understands. The system must take into account the type of training that both precedes and follows--building on the knowledge gained from training manuals and rule books while preparing the trainee for and complementing the on-the-job training which will follow. Perhaps most critical of all, trainees must be allowed to carry out an assigned task by any valid means. Such flexibility is essential so that trainees are able to retain and even hone an independence of thought and develop confidence in their ability to respond to problems, even problems which they have never encountered and which their trainers never anticipated.

SYSTEM DESIGN

The PD/ICAT system is modular and consists of five basic components:

1. A user interface that permits the trainee to access the same information available to him in the MCC and serves as a means for the trainee to assert actions and communicate with the intelligent training system
2. A domain expert (DeplEx) which can carry out the deployment process 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).
3. A training session manager TSM) which examines the assertions made

by the DeplEx (of both correct and incorrect actions in a particular context) and by the trainee. Evaluative assertions are made following each trainee action. In addition, guidance can be provided to the trainee if appropriate for his skill level.

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

Figure 1 contains a schematic diagram of the PD/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 "factbase" for all five system components. With the exception of the trainee model, each component makes assertions to the blackboard, and the rule-based components look to the blackboard for facts against which to pattern match the left-hand sides of their rules.

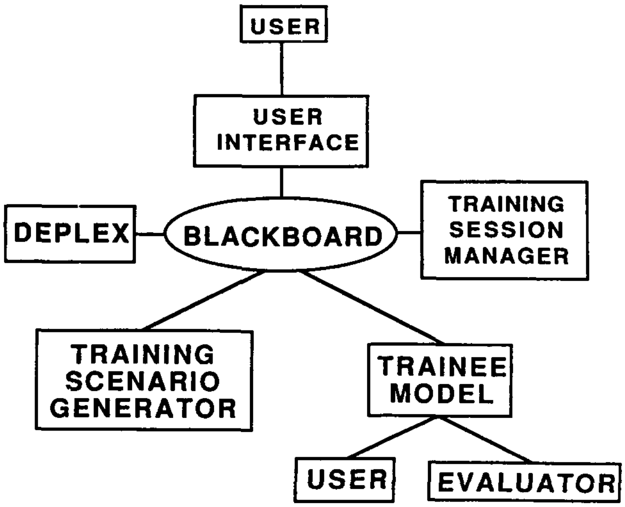
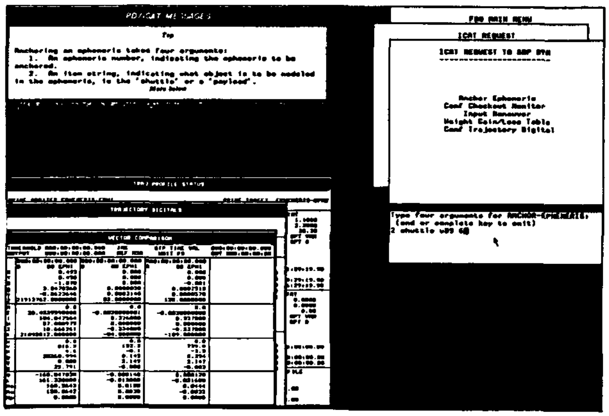


FIGURE 1 - PD/ICAT ARCHITECTURE

User Interface



Top

Anchoring an ephemeris takes four arguments:
1. An ephemeris number, indicating the ephemeris to be anchored.
2. An item string, indicating what object is to be modeled in the ephemeris, ie the 'shuttle' or a 'payload'.
More below

Type four arguments for ANCHOR-EPHEMERIS:
(end or complete key to exit)
2 shuttle u89 6

FIGURE 2 - Typical Screen Seen by PD/ICAT User

The primary factor influencing the interface design was fidelity to the task environment. To avoid negative training, it was deemed essential that the functionality and, to the extent possible, the actual appearance of the training environment duplicate that in which the task is performed. Figure 2 contains a view of the typical display seen by a trainee on a Symbolics 3600 series LISP machine. The upper right corner of the display contains menus that allow the trainee to make requests of other flight controllers, respond to requests from other controllers, call up displays, obtain information about the current or previous step in the deploy process, request help from the training system, and return to a previous step in the process. This menu has as many as three levels depending on the nature of the action taken by the trainee. Some actions are completely menu driven while others require the input of one or more "arguments". All actions taken by the trainee through these menus and the arguments that they may require become assertions to the blackboard. All requests directed to the trainee and all messages sent to the trainee in response to his requests or actions appear in a window in the upper left corner of the screen. These two portions of the screen serve to functionally represent the voice loop interactions that characterize the current FDO task environment. Any displays requested by the trainee appear in the lower portion of the screen,

ORIGINAL PAGE IS
OF POOR QUALITY

overlapped, if more than one is requested. Clicking the mouse on any exposed portion of a background display will bring it to the foreground. The displays replicate those seen by a FDO on console in the MCC. During development nominal data was supplied to these displays (from a dedicated ephemeris-generating program or from "dummy" data sets) so that negative training would not occur. Experienced FDOs using PD/ICAT have expressed satisfaction with the user interface.

DeplEx

The Deploy Expert is a "traditional" expert system in that it contains if-then rules which access data describing the deploy environment and is capable of executing the PAM deploy process and arriving at the correct "answers". In addition to "knowing" the right way to conduct the PAM deploy, DeplEx also contains knowledge of the typical errors that are made by novice FDOs. In this way, PD/ICAT can not only detect an erroneous action made by a trainee, but also, through these so-called "mal-rules", it can diagnose the nature of the error and provide an error message to the trainee specifically designed to inform the trainee about the exact error made and correct the misconception or lack of knowledge which led to the commission of that error. Another of the interesting features of the PD/ICAT system is its continual awareness of the environment (the external constraints dictated by the training exercise) and the context of the exercise. Rather than having DeplEx generate a complete and correct solution to the deployment problem, only those actions which are germane to the current context are asserted. In this way the expert "adapts" to alternate, but correct, paths that the trainee might choose to follow. Figure 3 shows schematically how DeplEx operates. This strategy was adopted because the human experts that perform PAM deploys recognize that many steps in the deploy process may be accomplished by two or more equally valid sequences of actions. To grant freedom of choice to the FDO trainee and to encourage independence on his part, the experts felt that it was essential to build this type of flexibility into the PD/ICAT system.

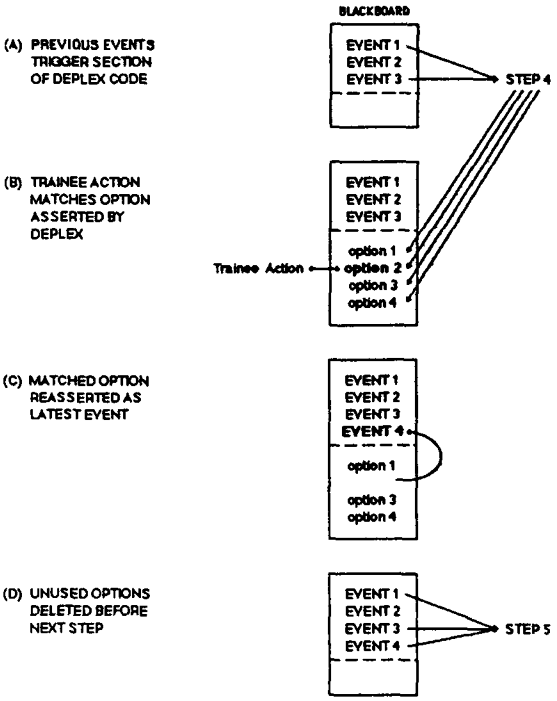

BLACKBOARD
(A) PREVIOUS EVENTS TRIGGER SECTION OF DEPLEX CODE
EVENT 1
EVENT 2
EVENT 3
STEP 4
(B) TRAINEE ACTION MATCHES OPTION ASSERTED BY DEPLEX
EVENT 1
EVENT 2
EVENT 3
option 1
option 2
option 3
option 4
Trainee Action
(C) MATCHED OPTION REASSERTED AS LATEST EVENT
EVENT 1
EVENT 2
EVENT 3
EVENT 4
option 1
option 3
option 4
(D) UNUSED OPTIONS DELETED BEFORE NEXT STEP
EVENT 1
EVENT 2
EVENT 3
EVENT 4
STEP 5

FIGURE 3 - DeplEx Operation, Illustrating Its Adaptability to Valid, User-Selected Alternatives

Training Session Manager

The training session manager is dedicated principally to error-handling. Its rules compare the assertions of DeplEx with those of the trainee to detect errors. Subsequently, DeplEx asserts facts that allow the TSM to write appropriate error messages to the trainee through the user interface. In addition, TSM is sensitive to the skill level of the trainee as represented by the trainee model. As a result, the detail and "tone" of error messages is chosen to match the current trainee. For example, an error made by a first time user of the training system may require a verbose explanation so that the system can be certain the trainee will have all of the knowledge and concepts needed to proceed. On the other hand, an experienced trainee may have momentarily forgotten a particular procedure or may have "lost his place". In this latter case a terse error message would be adequate to allow the trainee to resume the exercise. The TSM also encodes all trainee actions, both correct and incorrect, and passes them to the trainee model.

Trainee Model

Successful intelligent tutors incorporate student models to aid in error diagnosis and to guide the student's progress through the tutor's curriculum[9]. The trainee model in the PD/ICAT system stores assertions made by the TSM as a result of trainee actions. Thus, at its most fundamental level, the trainee model contains, for the current session, a complete record of the correct and incorrect actions taken by the trainee. At the conclusion of each training session, the model updates a training summary which contains information about the trainee's progress such as a skill level designator, number of sessions completed, number of errors made (by error type and session), and the time taken to complete sessions. After completing a session, the trainee can obtain a report of that session which contains a comprehensive list of correct and incorrect actions together with an evaluative commentary. A supervisor can access each trainee's model to obtain this same report or to obtain summary data, at a higher level, on the trainee's progress. Finally, the training scenario generator uses the trainee model to produce new training exercises.

Training Scenario Generator

The training scenario generator relies upon a database of task "problems" to structure unique exercises for a trainee each time he interacts with the system. The initial exercises provided to a new trainee are based on variants of a purely nominal PAM deploy with no time constraints, distractions or "problems". Once the trainee has demonstrated an acceptable level of competence with the nominal deploy, the generator draws upon its database to insert selected problems into the training environment (e.g., a propellant leak which renders the thrusters used for the nominal separation maneuver inoperable and requires the FDO to utilize a more complicated process for computing the maneuver). In addition, time constraints are "tightened" as the trainee gains more experience and distractions, in the form of requests for information from other MCC personnel, are presented at "inconvenient" points during the task. The generator also examines the trainee model for particular types of errors committed by the trainee in previous (and the current) sessions. The trainee is then given the opportunity to demonstrate that he will not make that error again. Ultimately, the trainee is presented with exercises which embody the most difficult problems together with time constraints and distractions comparable to those encountered during integrated simulations or actual missions.

SYSTEM INTEGRATION

The PD/ICAT system is currently operational on a Symbolics 3600 series Lisp machine. The user interface and trainee model are written in common Lisp while the rules of DeplEx, TSM, and the training scenario generator are written in ART 3.0. The system will ultimately be delivered to MOD in a Unix workstation environment. To accomplish this delivery, the ART rules were written to facilitate translation into CLIPS[10] and the Lisp code will be converted into C. It is uncertain, until the exact delivery environment is specified, how well the user interface can be ported.

CONCLUSIONS

The PD/ICAT system has, so far, proven to be a potentially valuable addition to the training tools available for training Flight Dynamics Officers in shuttle ground control. The authors are convinced that the basic structure of PD/ICAT can be extended to form a general architecture for intelligent training systems for training flight controllers and crew members in the performance of complex, mission-critical tasks. It may ultimately be effective in training personnel for a wide variety of tasks in governmental, academic, and industrial settings.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the invaluable contributions of expertise from three FDOs: Capt. Wes Jones, USAF; Major Doug Rask, USAF, and Kerry Soileau. Various students assisted with the knowledge engineering and coding of portions of the user interface and TSM: Tom Blinn, Joe Franz, Bebe Ly, Wayne Parrott, and Chou Pham. Finally, the encouragement and guidance of Chirold Epp (Head, ODS) and Bob Savely (Head, AIS) are gratefully acknowledged. Financial support for this endeavor has been provided by the Mission

Planning and Analysis Division, NASA/Johnson Space Center and (for RBL) by an American Society for Engineering Education/NASA Summer Faculty Fellowship.

REFERENCES

1. See, for example, Sleeman, D. and Brown, J.S. (eds.), Intelligent Tutoring Systems (London: Academic Press, 1982) and Yazdani, M. "Intelligent Tutoring Systems Survey," Artificial Intelligence Review **1**, 43 (1986).

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

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

4. Yazdani, M. "Intelligent Tutoring Systems Survey," Artificial Intelligence Review **1**, 43 (1986).

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

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

7. Anderson, J.R., Boyle, C.F., and Reiser, B.J., "Intelligent Tutoring Systems," Science **228**, 456 (1985) and Anderson, J.R. and Reiser, B.J., "The LISP Tutor," Byte **10**(4), 159 (April, 1985).

8. Harmon, P. "Intelligent Job Aids: How AI Will Chang Training in the Next Five Years," in Kearsley, G., ed., Artificial Intelligence and Instruction: Applications and Methods (Reading, MA: Addison Wesley Publishing Co., 1987).

9. See, for example, a number of papers on student models in Sleeman, D. and Brown, J.S., (eds.), Intelligent Tutoring Systems (London: Academic Press, 1982)

10. "CLIPS" is an acronym for "C-Language Integrated Production System" and was developed by the Artificial Intelligence Section, Mail Code FM72, NASA/Johnson Space Center, Houston, TX 77058. Its advantages as a delivery vehicle for expert systems are discussed in Giarratano, J., Culbert, C., Riley, G., and Savely, R.T., " A Solution of the Expert System Delivery Problem," submitted for publication in IEEE Expert. For additional information on CLIPS, write to the AI Section at NASA/JSC.

ACRONYMS

AIS	Artificial Intelligence Section
DeplEx	Deploy Expert
FDO(s)	Flight Dynamics Officer(s)
MCC	Mission Control Center
MOD	Mission Operations Directorate
ODS	Orbit Design Section
OJT	on-the-job training
PAM	Payload-Assist Module
PD/ICAT	Payload-Assist Module Deploy/Intelligent Computer-Aided Training
TSM	Training Session Manager