An Intelligent Training System for Space Shuttle Flight Controllers

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

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

Grace Hua
Computer Sciences Corp.
16511 Space Center Blvd.
Houston, TX 77058

Abstract

An autonomous intelligent training system which integrates expert system technology with training/teaching methodologies is described. The system was designed to train Mission Control Center Flight Dynamics Officers at NASA/Johnson Space Center to deploy a certain type of satellite from the Space Shuttle. The purpose of the system is to provide a bridge between the fundamental knowledge acquired from textbooks and the procedural skills necessary for successful performance in large-scale integrated simulations. Trainees are exposed to scenarios that evolve to higher levels of difficulty and are provided with assistance tailored to their demonstrated skill level. Such training devices can serve to augment on-the-job training, insure uniformity in training, and allow scarce training expertise to be delivered to a large number of trainees. The architecture of the system is modular and can be adapted to a wide variety of training tasks.

Problem Definition

The Mission Operations Directorate (MOD) at NASA/Johnson Space Center (JSC) is responsible for the ground control of all Space Shuttle operations. Those operations which involve alterations in the characteristics of the Space Shuttle's orbit are guided by a flight controller, known as a Flight Dynamics Officer (FDO), sitting at a console in the front room of the Mission Control Center (MCC). Currently, the training of the FDOs in flight operations is principally accomplished 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 or she is responsible during Space 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 Artificial Intelligence Section (AIS) at NASA/JSC has developed an autonomous intelligent computer-aided training system. The system trains inexperienced flight controllers in the deployment of a Payload-Assist Module (PAM) satellite from the Space 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 or her responsibility.

Description of the Application

Since the first proposals to apply artificial intelligence to the tutoring or training task (Carbonell,1970; Hartley and Sleeman, 1973), a large number of systems have been developed for both academic tutoring environments and industrial/government training environments (Sleeman and Brown, 1982; Yazdani, 1986; Kearsley, 1987; Wenger, 1987). In spite of these efforts, few completed systems have come into widespread use for routine training.

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 Payload-assist module Deploys/Intelligent Computer-Aided Training (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.

The PD/ICAT system architecture consists of five components and is organized around a common blackboard to facilitate communication among the different components (Loftin, Wang, Baffes, and Hua, 1988).

User Interface

The user interface permits the trainee to access the same information available to him or her in the MCC and serves as a means for the trainee to take actions and receive feedback from the training session manager. The interface mimics, to the degree possible, on a single screen, the environment of the domain in which the trainee will eventually work. Figure 1 shows a typical screen display.

Domain Expert

A domain expert, referred to as the Deploy Expert (DeplEx) is in the form of a production rule system that is capable of carrying out the satellite deployment process using the same information that is available to the trainee. DeplEx also contains a list of "mal-rules" (explicitly identified errors that novice trainees commonly make; Sleeman, 1982) so that the trainee can be provided with feedback specifically designed to help him or her overcome any anticipated conceptual or procedural problems.

Training Session Manager

The training session manager (TSM) consists of two expert systems: an error detection component which compares the assertions made by DeplEx (of both correct and incorrect actions in a particular context) with those made by the trainee and an error-handling component that decides on the appropriate method of guidance based on the trainee's skill level.

Trainee Model

The trainee model is an object-oriented data structure which contains a history of the individual trainee's interactions together with summary evaluative data. The model also has a report-generation feature that produces a formatted trace of each trainee session and provides the trainee's supervisor with a high-level description of the trainee's current skill level and progress.

Training Scenario Generator

The training scenario generator is an expert system and database 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. The

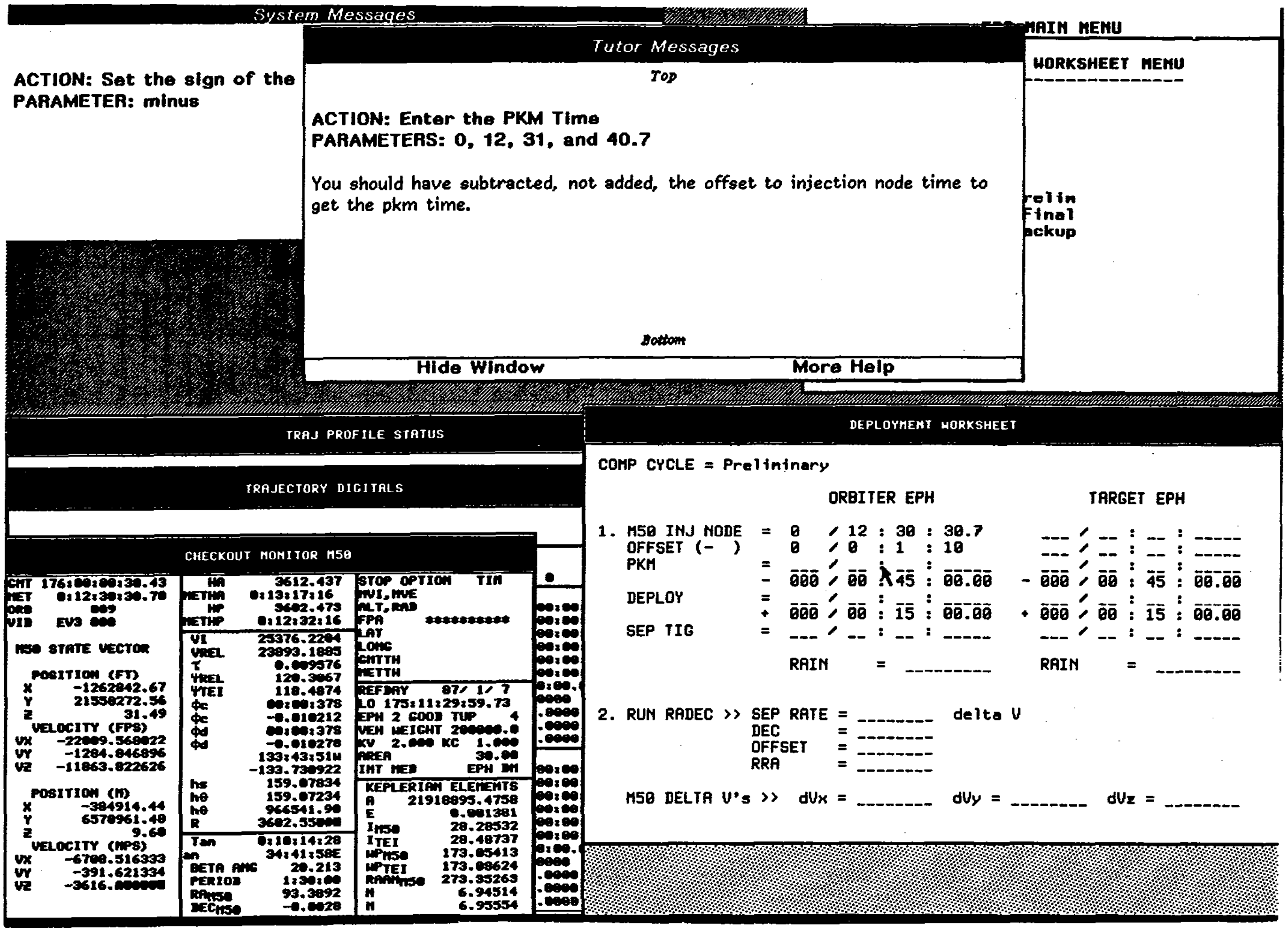

Figure 1. A typical screen as seen by a user of the PD/ICAT system. Menus at the upper right allow the trainee to interact with the system. Displays, identical to those used in the MCC, appear in the lower left. A worksheet used by the FDO in completing the task is shown in the lower right. System messages are displayed in the upper left, and error/help messges are provided in a pop-up window near the center of the screen.

database serves as a repository for all parameters needed to define a training scenario and includes problems or abnormalities of graded difficulty. The nature of this component of the PD/ICAT architecture is more fully developed elsewhere (Loftin, Wang and Baffes, 1988).

System Integration

All of the expert system components of PD/ICAT (DeplEx, TSM, and TSG) communicate via a common blackboard. The blackboard also contains a representation of the current trainee action(s) and is a source of data for the updating of the trainee model.

Innovative Features of the Application

Although a number of intelligent training systems have been developed [for example, SOPHIE (Brown, Burton and de Kleer, 1982) and STEAMER (Hollan, Hutchins and Weitzman, 1984)], few, if any, can be said to have been deployed. The PD/ICAT system has been deployed and is in use for the training of novice flight controllers and for practice and refreshment by experienced personnel. Many features of the PD/ICAT system are innovative in their own right or are innovative in their application to intelligent training systems:

- PD/ICAT is composed, in part, of four expert systems which cooperate through and communicate by means of a common blackboard. This approach was used to permit the segregation of domain-independent knowledge so that the system architecture could be adapted to different training tasks.
- Unlike most intelligent tutoring/training systems, PD/ICAT does not require the trainee to follow a single correct path to the solution of a problem. Rather, a trainee is permitted to select any correct path, as determined by the scenario context. The method used to accomplish this flexibility, without generating a combinatorial explosion of solution paths, is believed to be unique.
- Error detection occurs through the comparison of the trainee's actions with those of an expert. In the case of complex actions, the error detection is made at the highest level to avoid confusing the trainee by detecting all errors which propagated from the one deemed most significant.
- Error handling may be accomplished through the matching of trainee actions with mal-rules containing errors that are commonly made by novices. In addition, the TSM error-handling component decides, based on the trainee model, what type of feedback to give the trainee. Explanations or hints may be detailed for novices and quite terse for more experienced personnel. In some cases, the TSM may decide not to call attention to the error if there is a reasonable probability that the trainee will catch his or her own mistake.
- The training scenario generator examines the trainee model and creates a unique scenario for each trainee whenever a new session begins. This scenario is built from a database containing a range of typical parameters describing the training context as well as problems of graded difficulty. Scenarios evolve to greater difficulty as the trainee demonstrates the acquisition of greater skills in solving the training problems.
- At the conclusion of each session, the trainee is provided with a formatted trace of the session which highlights the correct and incorrect actions taken, the time required to complete the exercise, and the type of assistance provided by the system. In addition, the trainee's supervisor may view a global history of each trainee's interaction with the system and even generate graphs of trainee performance measured against a number of variables.

Criteria for a Successfully Deployed Application

From the inception of the PD/ICAT project, a number of factors were recognized as essential for success at meeting its objectives and acceptance by its intended audience:

1. The involvement of the ultimate users (or "customers") throughout the development process was certainly the most important factor in PD/ICAT's successful deployment. This involvement allowed the development team to have at hand the experts in the targeted procedure and to quickly test competing approaches to the solution of specific development problems.
2. The intended audience for the training system was asked to provide commentary at each stage of the interface development. Thus, at the project's conclusion, the inter-

face had already achieved tacit acceptance by those it was intended to serve.

3. The coding, in a production-rule system, of those components of PD/ICAT that would require alteration as the procedures it was designed to teach are altered provides built-in maintainability. The AIS has found, after fielding a number of production rule systems, that such systems may be easily altered and verified as the task they were designed to accomplish changes.
4. Sufficient training in production-rule coding and detailed documentation were provided to the system's users so that they were able to provide their own long-term support for PD/ICAT.
5. Although PD/ICAT was developed on a Symbolics Lisp machine using ART and Lisp, it has been ported to a unix-based workstation using C and CLIPS (a production-rule system, written in C, and developed by the AIS at JSC). Such workstations were already in use by PD/ICAT's intended users, and they were not required to purchase and learn to use a different hardware platform.
6. An important motivation for managers in the intended user community was the ability of PD/ICAT to capture the expertise of personnel who were to be transferred to other areas. This was a key factor in the ready availability of the experts and in the management's support of their dedication of time to this project.

Benefits to NASA

Training of astronauts and ground-based flight controllers and system engineers is a massive task. The best training and the mechanism for certifying that personnel have met training objectives occurs through large-scale, integrated simulations. Unfortunately, these simulations require the support of hundreds of people but deliver training to only one person in each position. The ability of a given trainee to get significant exposure to a particular process is, therefore, quite limited. The PD/ICAT system, on the other hand, can provide a trainee with virtually unlimited access to training in a specific procedure and insure that the integrated simulation environment can be used to maximum effect.

This deployed system has demonstrated the capability of intelligent training systems in NASA's operational environment. As a result, a number of similar systems are under development at JSC and other NASA operational centers (Marshall Space Flight Center and Kennedy Space Center). In addition to the impact of this technology on Space Shuttle training, NASA is supporting its application to future Space Station training. Since Space Station training may be a task at least an order of magnitude larger than current Space Shuttle training, the use of intelligent training systems may be the only way to meet Space Station training objectives with the available resources. To this end, the Space Station program is supporting the refinement of the PD/ICAT architecture into a generic training architecture and the creation of a general-purpose development environment for the rapid creation of and adaptation of intelligent training systems. This latter activity will have a profound impact on the nature of training, not only within NASA, but within other government agencies, industry, and the education establishment.

Deployment Process

The conception and initial planning for the PD/ICAT project began in July 1986, and initial knowledge acquisition was complete by December 1986. The code development began in January 1987 and the system was demonstrated in March 1988. During this latter period three complete versions of the system were developed before PD/ICAT was accepted. PD/ICAT has been used by both novice FDOs for training and by experienced FDOs for practice and refreshment of skills since March 1988. The rehost of PD/ICAT to a unix-based workstation was accomplished early in 1989.

The development of PD/ICAT required approximately 3.5 man-years and was accomplished by a mixed team from academia (RBL), NASA civil service (LW and PB), and private industry (GH). In addition, a number of students contributed to the project during its life. The actual direct cost of the project was approximately $120,000 (for the academic and private sector portions). The indirect costs of civil service manpower are more difficult to calculate, but were approximately $100,000.

Acknowledgments

The authors wish to acknowledge the invaluable contributions of expertise from three FDOs: Capt. Wes Jones, USAF; Maj. Doug Rask, USAF (ret.); 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, the NASA Office of Space Flight, and, for RBL, by NASA/American Society for Engineering Education Summer Faculty Fellowships and a NASA/National Research Council Senior Resident Research Associateship.

References

Brown, Burton and de Kleer, 1982. Brown, J.S., Burton, R.R., and de Kleer, J., Pedagogical, Natural Language and Knowledge Engineering Techniques in SOPHIE I, II, and III, in *Intelligent Tutoring Systems*, 227-282, eds. Sleeman, D. and Brown, J.S. London Academic Press.

Carbonell,1970. Carbonell, J.R. AI in CAI: An Artificial Intelligence Approach to CAI, *IEEE Transactions on Man-Machine Systems*, 11(4): 190-202.

Hartley and Sleeman, 1973. Hartley, J.R. and Sleeman, D.H., Towards Intelligent Teaching Systems, International Journal of Man-Machine Studies, 5: 215-236.

Hollan, Hutchins and Weitzman, 1984. Hollan, H.D., Hutchins, E.L., and Weitzman, L. Steamer: An Interactive Inspectable Simulation-based Training System, *A I Magazine*, 5(2): 15-27.

Kearsley, 1987. *Artificial Intelligence and Instruction*, ed. Kearsley, G. Reading. MA: Addison-Wesley.

Loftin, Wang and Baffes, 1988. Loftin, R.B., Wang, L, and Baffes, P., Simulation Scenario Generation for Intelligent Training Systems. In Proceedings of the Third Artificial Intelligence and Simulation Workshop, 69-73. Menlo Park, CA: American Association for Artificial Intelligence.

Loftin, Wang, Baffes and Hua, 1988. Loftin, R.B., Wang, L., Baffes, P. and Hua, G. An Intelligent Training System for Space Shuttle Flight Controllers, Telematics and Informatics, 5: 151-161.

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

Sleeman, 1982. Sleeman, D.H. Inferring (mal) Rules from Pupils' Protocols. In Proceedings of the European Conference on Artificial Intelligence, 160-164. Orsay, France.

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

Yazdani, 1986. Yazdani, M., Intelligent Tutoring Systems Survey, *Artificial Intelligence Review*, 1: 43-52.