THE RADAR SYSTEM CONTROLLER INTELLIGENT TRAINING AID

James E. McCarthy, Stephen Pacheco, H. George Banta, John L. Wayne, and David S. Coleman
Sonalysts, Inc.
Waterford, CT

ABSTRACT

The AN/SPY-1 is a phased array radar system that functions as part of the AEGIS combat system aboard modern U.S. Navy Cruisers and Destroyers. Enlisted personnel, known as radar system controllers (RSCs) operate and maintain the radar system. The RSC must optimize radar performance in a number of disparate environments. In order to enhance a new operator's ability to maintain this optimization, the AEGIS Training Center contracted for the development of a training aid. The resultant Radar System Controller Intelligent Training Aid (RSC ITA) is a PC-based training aid that makes use of a master/apprentice training paradigm. We describe it below.

ABOUT THE AUTHORS

Dr. James E. McCarthy has a B.S. in Mathematics Education from Winona State University, as well as an M.A. and a Ph.D. in Experimental Psychology from Miami University. He joined Sonalysts, Inc. in January 1993. Since that time he has designed and developed intelligent instructional technology for military and civilian uses.

Mr. Stephen Pacheco has a B.S. in Mechanical Engineering from the University of Rhode Island. He has over twenty years of experience as a lead software engineer. His work includes extensive experience with the design and development of simulation-based training systems.

Mr. H. George Banta has a B.E. in Civil Engineering from the Steven's Institute of Technology. He has a wealth of experience in the design and development of intelligent technology. Examples of his work include the development of intelligent systems to aid in resource allocation as well as tactical decision making.

Mr. John L. Wayne has a B.S. in Biology from Tulane University, and an M.B.A. in Management from Bryant College. He joined Sonalysts, Inc. in 1987 after 9 years as a surface warfare officer in the U.S. Navy. During his tenure at Sonalysts, he has assisted in the design, development, and utilization of a range of training systems.

Mr. David S. Coleman has a B.S. in International Security Affairs from the U.S. Naval Academy. Following seven years as a U.S. Navy surface warfare officer, he joined Sonalysts, Inc.. Since that time, he has had 13 years of experience managing a number of multimedia training system developmentefforts.

THE RADAR SYSTEM CONTROLLER INTELLIGENT TRAINING AID

James E. McCarthy, Stephen Pacheco, H. George Banta, John L. Wayne, and David S. Coleman
Sonalysts, Inc.
Waterford, CT

INTRODUCTION

Consider the various ways in which instruction could proceed. The most basic approach would be to disseminate the appropriate instruction in the form of a book. The book approach has many advantages. For example, books allow for self-paced instruction, support easy review of previously encountered content, and place the student in control of the content and order of instruction (see Laurillard, 1987 for a discussion of the benefits of student control). Unfortunately, books also have many disadvantages. A book presents information in only one style, and therefore is appropriate only for students with a compatible style of learning. Books cannot adapt to or interact with students. Books can only present static representations of the world and therefore are ill-suited to concepts that are fundamentally dynamic.

Another option would be to use classroom instruction. Classroom instruction is moderately adaptive and interactive. Students can exert some control over content, presentation method, and pacing through their questions and other interactions. Classroom instruction, however, has limits. The size of the class and (perhaps more importantly) the skill of the instructor determines, to a large extent, the amount of flexibility in areas such as instructional tactics and pacing. As class size increases, the likelihood of student or instructor initiated discussion decreases and it becomes more and more difficult for instructors to attend to individual students. Further, even if instructors were able to attend to individual students, they could only use those instructional tactics in which they had been trained and with which they felt comfortable. Student and teacher have little recourse if the instructor's repertoire of tactics does not include those which are most suitable for teaching a given concept to a given student. With large class sizes or with teachers with limited instructional skill or experience, classroom instruction suffers from all the negative attributes associated with book-based instruction without the advantages of self-pacing, content and order selection, or ready review.

Given the limitations of book- and classroom-based instruction, it may seem as if the ideal solution would be to assign to each student a tutor trained to teach the subject-matter in the style most appropriate for the learning style of the student. The tutor would be maximally adaptive to the student in that the pacing of instruction would reflect the student's ability to absorb the material and the method of instruction could be adapted to meet the moment-to-moment needs of the student. Even this seemingly ideal situation has limits, however. Besides the obvious cost disadvantage of the approach, human tutors, like books, are extremely limited in their ability to simulate complex dynamic events.

The preceeding discussion implies that the ideal training aid is a tutor that is pedagogically appropriate for each student, that is inexpensive to employ, and that can allow the student to interact with complex, reality-based, simulations. A simulation-based Intelligent Training Aid (ITA) is such a tool. Through careful construction, one can create a tutor that makes appropriate use of a wide range of instructional tactics, monitors the student, detects when he or she is experiencing difficulty, diagnoses the nature of that difficulty, and responds with remediation appropriate for both the difficulty encountered and the current state of the learner. This can be done in a complex, dynamic, and realistic problem-solving context that will allow the students to monitor the effect their appropriate and inappropriate actions have. The Radar System Controller Intelligent Training Aid (RSC ITA) described below is an example of such a system.

Reigeluth and Schwartz (1989) state that, "Computer-based simulations can provide efficient, effective, and highly motivational instruction that can readily serve the need for individualization. Simulations also enhance the transfer of learning by teaching complex tasks in an environment that approximates the real world setting in certain important ways." Others (Anderson et al., 1985; Burger, and DeSoi, 1992) echo the importance of hosting instruction within a problem solving context. Too often in contemporary education, theory is artificially separated from application. Simulation-based ITAs marry thought to deed and allow students to discover some truths on their own as they are coached to recognize others.

Before considering in detail the design of the RSC ITA, let's consider the context in which it will be used.

THE RADAR SYSTEM CONTROLLER: TASKS AND TRAINING

The AEGIS combat system is part of virtually all modern U.S. Navy Cruisers and an increasing proportion of U.S. Navy Destroyers (CG 47- and DDG 51-Class ships). The AN/SPY-1 radar system serves as the eyes of that system. The AN/SPY-1 is a phased-array radar system that automatically detects and tracks surface and air contacts. It then transfers the data to other portions of the combat system.

Enlisted personnel, known as radar system controllers (RSCs), are responsible for operating and maintaining the radar system. RSCs must manage the radar system to minimize the temporal or spatial blindness facing the combat system. Temporal blindness refers to the length of time it takes the radar system to search the battle-space. Spatial blindness refers to the range at which radar detects and tracks a contact of a given size. Increasing spatial blindness reduces temporal blindness. Conversely, decreasing spatial blindness generally results in an increase in temporal blindness. The RSC must maintain a balance between spatial and temporal blindness that is appropriate for the existing tactical environment.

RSC training takes place during a 24-week radar system operations and maintenance course. Maintenance training occupies the first 23 weeks of that course. Operations training occurs during the last week of the course. During the laboratory portion of operations training, each student operates a functioning radar system for approximately 3 hours.

Feedback and lessons-learned from Operation Desert Shield/Storm and other fleet operations revealed that novice RSCs who completed the training course were, in general, ill-equipped to deal with the complex management task facing them in the fleet. Consequently, the AEGIS Training Center sought to improve the level of operational training provided to RSCs by contracting for the development of a training aid that would allow RSC-trainees to practice their operational skills in an operationally realistic training environment. The Radar System Controller Intelligent Training Aid (RSC ITA) is the result of that development effort.

INTELLIGENT TRAINING AIDS: OVERVIEW

As a rule, all intelligent training aids share four common components: a learning environment/student-device interface, a domain expert, a student model, and an instructional expert. Let's consider each of the components in turn.

Learning Environment/Student-Device Interface

The learning environment defines the context in which learning takes place. The learning environment defines the tasks facing the student as well as the tools the student can use. The student-device interface is the medium of communication between the student and the ITA.

ITA developers are increasingly taking the perspective that students learn from doing. The perspective stems from an instructional philosophy that holds that learning is a process of construction not absorption, and that new ideas must be tied to each other and to old ideas (Burton, 1988, Burger & DeSoi, 1992). Learning environments now more than ever require students to apply and even discover knowledge.

A critical dimension of the learning environment and student-device interface is the degree of similarity or fidelity between the learning environment and the real world. Burton (1988) identified four kinds of fidelity: *physical, display, mechanistic,* and *conceptual.* It has been noted that it is important to match the level of fidelity to the training task at hand. For example, if students are learning sensorimotor tasks, then it is likely that the ITA needs high levels of physical fidelity. On the other hand, if we are teaching reasoning skills, conceptual fidelity is probably more important than physical fidelity.

Together, the choice of an appropriate learning environment and student-device interface scheme are critical for the success of an ITA. An effective scheme can, in fact, enhance learning before the intelligent components of the software ever come into play.

Domain Expert

The domain expert is a software module that represents the knowledge or performance of someone expert in the domain of instruction. For example, if we were to build a medical diagnosis ITA, the domain expert would represent the knowledge of an expert diagnostician. In the present case, the domain expert reflects the performance of an expert RSC.

There are three general classes of domain experts. They are: black box, glass box, and process models of expert decision making (Anderson, 1988). Black box models of expert decision making are usually extremely efficient algorithmic processors that produce the correct input/output behavior in the instructional domain. Black box domain experts produce the correct solution and therefore they can judge *the correctness of the student's actions.* However, because they use processes that are unlike those used by human experts, they cannot produce instructionally useful explanations of their behavior. Moreover, although the black box expert can judge the correctness of a student's action, the judgment cannot be extended to include diagnosis of the student's difficulty or misconception.

Although black box domain experts have acknowledged weaknesses, they can support lower-cost development of intelligent training aids (e.g., Gugerty & Hicks, 1993; Gugerty, Hicks, and Walsh, 1993). As these examples point out, black box expert systems can be re-purposed from other tasks to support a form of intelligent instruction. Black box models eliminate the need for extensive knowledge engineering efforts thus reducing the time and cost of development. They do so, however, with a price of reduced capability.

The second class of domain experts are glass box models. Glass box models solve problems in the domain by using reasoning heuristics that are similar to those employed by human experts. Glass box domain experts can judge the correctness of a student's actions, they can diagnose a student's difficulty, and they can provide explanations of the expert's decision. Constructing glass box domain experts require more extensive knowledge engineering efforts. As a result, they are generally more capable, but more expensive.

The third class of domain experts are process models (or cognitive models). Process models attempt to encode and employ knowledge in human-like ways. The benefit to this approach is that the knowledge is in a form that the system can most easily and completely communicate to the student. The cost is that process models are relatively more difficult and expensive to construct. Process models must commonly take the form of production systems (e.g., Anderson, et al., 1985) or semantic networks (e.g., Carbonell, 1970). Process models can judge the correctness of student actions, explain the expert's solution to a problem, identify related issues, generate real-time remediation, and support mixed-initiative dialogues with the student.

In considering the class of domain expert to include in an intelligent training aid, the developer must weigh the cost of development against the cost of sacrificing power and elegance. As always, the correct solution is a function of the constraints of the problem.

Student Model

The student model is the intelligent training aid's conception of who the student is. The student model is the repository of data that allows the ITA to adapt to the particular student using the system. Generally, each student has his or her own student model and each model is updated as the students interact with the ITA.

There are four classes of student models: Performance models, overlay models, error models, and simulation models (Ohlsson, 1986). The fidelity of the associated domain expert constrains the selection of a student model.

Performance models focus on *how much* the student knows, not *what* the student knows (Ohlsson, 1986). Performance models are the only type of student model that is available when we use a black box model domain expert. Performance models allow us to assess the global level of understanding of the student. With them, we can adjust problem difficulty or pacing. They do not, however, contain enough data to permit us to tailor instruction to the particular difficulties facing the student.

Overlay models represent the student's knowledge as a sub-set of an expert's. Overlay models present an expert's knowledge as a collection of concepts. They then record the student's mastery, or lack thereof, of each concept. Overlay models are most consistent with glass box domain experts. The overlay model allows us to determine specific areas of strength and weakness for each student. This knowledge allows the ITA to use the students' strengths to overcome their weaknesses.

Error models represent the student as possessing some number of common misconceptions. The misconceptions are often called "bugs". Therefore, error models are often called "bug catalogues". Error models are most consistent with glass box and process models of the domain expert. Because they point out specific shortcomings in student performance the ITA can adapt instruction to combat those weaknesses.

Simulation models represent the student as following a more or less appropriate problem solving script. In some sense, simulation models represent a combination of overlay and error models. The student's correct and incorrect actions are combined to form a simulation of that student's performance. Simulation models are most consistent with process models. As with overlay and error models, simulation models permit instruction tailored to specific areas of strength and weakness.

Note that each of the classes discussed above focuses on student knowledge, not on student cognitive style. A complete student model should include information on both. Two factors have hampered efforts in this area. The first is that there are very few tests of specific cognitive style dimensions. The second is that educational psychology has made very few prescriptions regarding how to teach a given concept to a student with a particular cognitive style (Ohlsson, 1986). In the absence of such prescriptions, cognitive style data is of little use to the ITA and therefore student models have not incorporated it. For ITAs to approach their potential, developers and educational psychologists must address these issues.

Instructional Expert

Just as the domain expert is a software module that represents the knowledge of an expert in the instructional domain, the instructional expert is a software module that represents the knowledge of one skilled in instructional practice.

Of the all ITA components, the instructional expert is the least formalized. The situation stems from the observation made in the preceding section: educational psychology has not yet been able to suggest how to teach a particular student a given concept in a defined context. In the absence of such guidelines, the design of instructional experts has tended to be ad hoc and quite variable across systems.

Instructional experts use data from the student model together with data from the most recent domain expert assessment of student actions to make instructional decisions. If the student *model is performance based, the instructional* expert must decide whether to change the pacing of instruction and/or the difficulty of the *problems facing the student.* When a richer student model is available, the situation becomes more complex.

When the ITA has an overlay, error, or simulation student model, there are three decisions an instructional expert must make. First, the expert must decide whether or not to intervene in an instructional situation. Many times skilled instructors observe a student making a mistake but remain silent so the student can discover the error for himself or herself. Similarly, if the student is performing correctly, the expert must decide whether more is gained through positive reinforcement than is lost by interrupting the student.

If the instructional expert decides that the present situation warrants intervention, it must then decide which issue or concept to discuss. Often *a single student action will reveal* multiple instances of student knowledge and misconception. The instructional expert must *choose one to address.*

Finally, the instructional expert must choose *the form of the instructional intervention.* All ITAs have a number of ways to present remediation on a given topic. The instructional expert must choose the form of remediation that is most beneficial for the particular student at that time.

Let's now consider the RSC ITA in terms of each of these components.

RADAR SYSTEM CONTROLLER INTELLIGENT TRAINING AID

Overview

The RSC ITA is a master/apprentice (Burger and DeSoi, 1992) training system that uses the pedagogical principle of fading. The RSC ITA attempts to foster a synthetic master and apprentice relationship similar to the one that exists in a shipboard environment. When a novice RSCs arrive on board, they must satisfy personnel qualification standards before they are allowed to stand watch alone. The *standards are satisfied under the watchful eye* of an experienced RSC who monitors the novices' actions and provides instruction as needed. The RSC ITA functions in a similar manner. Figure 1 depicts the general functioning of the RSC ITA.

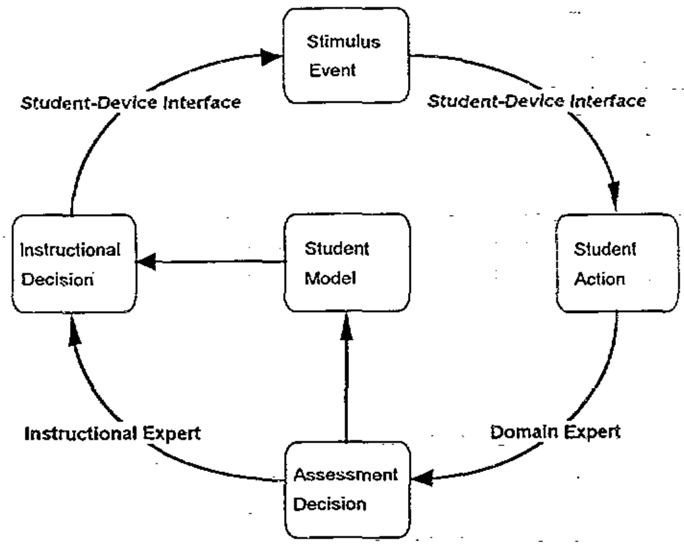

Figure 1: A Cycle of Interaction

For the most part, the RSC ITA is a simulation-based training aid. Therefore, most interaction begins with some sort of simulation-based stimulus event. The event must then be conveyed to the student. That is the job of the student-device interface. Next, the student takes some action in response to the stimulus event. The master, or domain expert, that is figuratively looking over the student's shoulder evaluates the student's action. The evaluation leads to an assessment decision which, in turn, is used to update the student model. The instructional expert then uses data from the revised student model and the assessment decision to reach an instructional decision. The student-device interface then conveys the decision to the student and the cycle repeats itself.

Now, let's consider each component of the RSC ITA in some detail.

Learning Environment/Student-Device Interface

The learning environment and student-device interface is an extremely important component of the ITA. We made every effort to create an operationally realistic learning environment. We paid special attention to attaining high levels of cognitive fidelity in the design of the student-device interface and the learning environment.

The learning environment is a functional replication of the RSC watchstation. It includes a model of the world, a model of the radar system, and a simplified replication of the operator's console. The world and radar models interact to produce symbology on the console's PPI and a dynamic amplitude spectrum on the console's A-Scope. Clutter, jamming, and hostile and friendly surface and air tracks are all recreated. As the student changes the radar configuration, the appearance of these entities is affected. For example, increasing sensitivity will tend to increase the detection range of a given target.

The student-device interface, depicted in Figure 2, is a Windows™-based reproduction of the OJ-451 console display and controls used by the RSC. Controls are activated by standard point-and-click operations.

In response to simulation events, the student could take a range of actions. For example, the student could hook tracks (i.e., select them for closer inspection), build sectors and sub-sectors, impose radar doctrine, observe the effects of doctrine, and communicate with other combat information center (CIC) team members. Watching all student activity is a

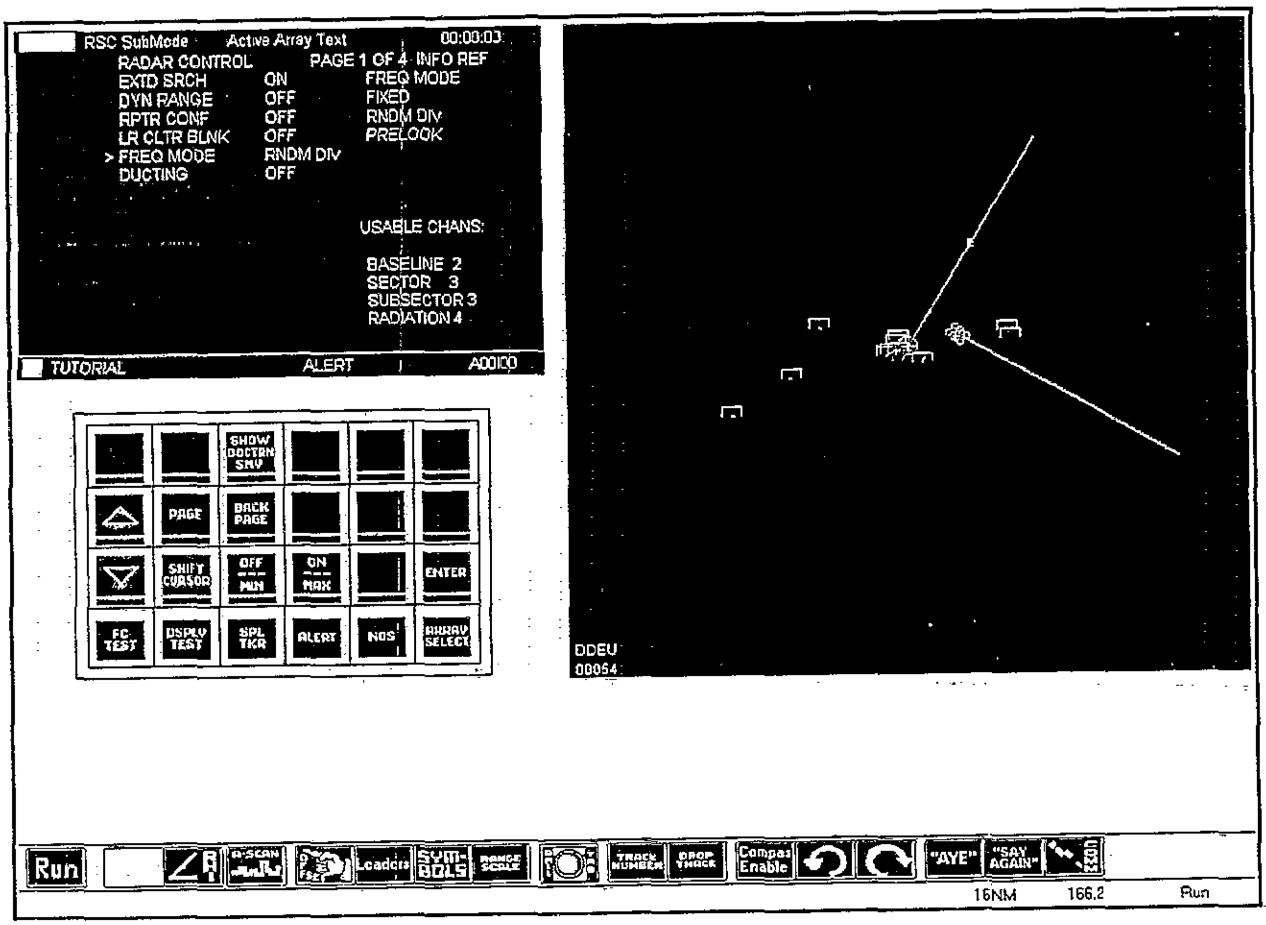

Figure 2: RSC *ITA Student Device Interface*

domain expert module in the guise of a master RSC.

Domain Expert

The domain expert in the RSC ITA is based on a process model. One segment of the domain expert observes the console and reaches conclusions about the proper course of action. For example, the domain expert might observe that the time required to search the battle-space was approaching unacceptable levels. At the same time, it might note an isolated region of false (clutter) tracks. The domain expert might then decide that it was necessary to impose doctrine in the clutter region. The domain expert would also know in which order to try various doctrine impositions (i.e., it would know which "fixes" to try first and which to try last), as well as how long to observe each imposition to see if it was successful. Finally, the domain expert would be able to recognize when it had acheived a satisfactory radar configuration.

The domain expert's decisions set up a number of expectations. As the student performs, the domain expert gathers the actions and compares them to the existing expectations. Three classes of actions might result:

1. The student's actions might meet the domain expert's expectations.
2. The student's actions might represent a mutation of the expected action (i.e., a bug). For example, the domain expert might expect the student to build a particular sector, and the student might build the sector, but enter the bearing limits incorrectly.
3. The student might take an action that is totally unexpected, that is, it does not conform to any of the expectations nor the anticipated mutations of those expectations.

The comparison of the expectations and actions leads to an assessment decision and the revising of the student model.

Student Model

The student model assesses mastery of three types of knowledge: declarative (i.e., knowledge of facts), procedural (i.e., knowledge of actions), and contextual (i.e., knowledge of timing). An endorsement tree (Murray, 1991) serves as the underlying formalism.

The endorsement tree formalism recognizes that often there are multiple sources of information about a student's mastery of a given concept, and that the sources differ in their reliability. For example, a student's claim of knowledge and his or her demonstration of knowledge both are informative, but we are more likely to believe the demonstration than the unsubstantiated claim. Further, the endorsement tree formalism recognizes that knowledge of a student's mastery of one concept can tell us things about the student's mastery of other concepts. For example, if a student has demonstrated mastery of fraction multiplication, then it is likely that the student has mastered integer multiplication. Finally, the endorsement tree formalism rejects numeric methods (i.e., confidence levels or weighted averages) in favor of symbolic methods (i.e., counting within ordinal equivalence classes).

To construct the student model, we first decomposed the knowledge domain into a hierarchy of learning objectives. For example, a learning objective such as, "The student can build a low-power sector", has as a component, "The student can build a sector", which, in turn, has as a component, "The student can enter sector bearing limits". The decomposition led to the construction of a learning objective tree. Next the sources of evidence for and against mastery were defined. Examples include default beliefs, inherited or propagated beliefs, answers to single questions, data trends, and activities which reflect knowledge. The evidence sources are used to define classes of evidence that are ordered by their assumed reliability.

The student model places each evidence datum that the domain expert passes to it in the appropriate evidence class. When a decision based on the student's mastery of a concept is required, the student model applies the following procedures. Beginning with the most reliable evidence category, the student model pairs positive and negative endorsements. If there are more positive endorsements, the student model assumes that the student has mastered the objective. *If there are more* negative endorsements, the student model assumes that the student has not mastered the objective. If there are an equal number of positive and negative endorsements, the student model moves on to the next most reliable evidence class. If at the end of the process, *all positive evidence is balanced by an* equal amount of negative evidence, the student model labels mastery uncertain. If no endorsements exist, the student model labels mastery unknown.

If the instructional expert needs a finer grain of data on which to base its decision, the endorsement model can provide a strength of belief value as well. The strength of a belief is based on the evidence class on which it is based, as well as the number of positive or negative arguments that remain after pairing. In addition, the instructional expert can, conceivably, make use of the full pattern of data in all evidence categories.

An *endorsement-based* student model can support an instructional expert as it matures and becomes more complex.

Instructional Expert

In light of any clear prescriptions on its *construction, we decided to design the* instructional expert with an eye towards simplicity and ease of expansion.

The first decision the instructional expert must reach is whether or not to intervene. When the instructional expert makes the decision, it must balance its need to be *unobtrusive* with the need to forestall poor operating procedures. The RSC ITA instructional expert only intervenes when the student model posts a new endorsement. The student model only posts endorsements after the student has completed a meaningful block of sections (e.g., after the student has pressed the "Enter" key to enter all the settings pertaining to a radar sector). If the endorsement is positive and the *student has a high level of mastery on that* objective, the instructional expert intervenes (with positive feedback) only if it has not made a recent positive feedback comment to the student. If the endorsement is positive and the student has demonstrated a lack of mastery, the *instructional* expert always intervenes. If the endorsement is negative, the instructional expert always intervenes.

Next, if the student model records more than one new endorsement, the instructional expert must decide which learning objective to *address. If all the endorsements are positive* then the instructional expert picks the most general learning objective for positive reinforcement. If multiple positive endorsements exist at the same level, the instructional expert chooses the one discussed least recently. If there are negative endorsements, *the instructional expert chooses* to discuss those. If there are multiple negative endorsements, the instructional expert chooses the most specific learning objective. If there are multiple negative endorsements at the same level, the instructional expert chooses the one discussed most recently.

Finally, the instructional expert must select the form of the intervention. The ITA uses the instructional philosophy of "fading". The instructional expert instantiates the philosophy by providing more information to students whose level of mastery is low and less information to those students whose level of mastery is high. For example, assume that two students, one with a low level of mastery and *the other with a high level*, both performed the same correct action. On one hand, the instructional expert might say something like "Good Job!" to the high level of mastery student. On the other hand, the instructional expert would probably tell the low level of mastery student what he did, that it was *correct, and why it was correct.* The low level of mastery student needs this level of information to tune his performance, but the high level of mastery student would probably find it intrusive.

Conclusion

Intelligent training technology provides a powerful training alternative in both the military and civilian sectors. Intelligent training aids such as the RSC ITA allow students to apply skills in realistic ways as they acquire them. Application-based instruction is entirely consistent with constructionist learning theory and, in all likelihood, results in greater retention and transfer of training.

The master/apprentice training paradigm used within the RSC ITA is extremely flexible and can be applied to a number of training problems. Simply put, the master/apprentice paradigm is appropriate whenever the training task could best be accomplished through direct *work-related tutoring*. *There are countless* examples in both the military and civilian sectors of such settings. The examples include learning to diagnose system faults in power plant operations, learning to operate military sensor and weapons systems, and learning engineering principles in an academic setting.

Although a number of research issues still exist in this field, none is more striking than the need for educational models that support the development of instructional expert modules. Developers of intelligent trainers must work together with educational psychologists to insure that intelligent training systems reach their full potential.

Bibliography

Anderson, J.R. (1988). The expert module. In M.C. Polson, & J.J. Richardson (Eds), Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum Associates Publishers.

Anderson, J.R., Boyle, C.F., and Reiser, B.J. (1985). Intelligent tutoring systems. Science, 228, 4698, 456-462.

Burger, M.L. and DeSoi, J.F. (1992). The cognitive apprenticeship analogue: A strategy for using ITA technologyfor the delivery of instruction and as a research tool for the study of teaching and learning. Int. J. Man-Machine Studies, 36, 775-795.

Burton, R.R. (1988). The environment module of intelligent tutoring systems. In M.C. Polson, & J.J. Richardson (Eds), Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum Associates Publishers.

Carbonell, J.R. (1982). An investigation of computer coaching for informal learning activities. IEEE Transactions of Man-Machine Systems, 11, 190-202.

Clancey, W.J. (1982). Tutoring rules for *guiding a case method dialogue*. In D. Sleeman & J.S. Brown (Eds.), Intelligent Tutoring Systems. New York: Academic Press.

Gugerty, L., & Hick, K. (1993). Non-diagnostic intelligent tutoring systems. Proceedings of the 15th Interservice/Industry Training Systems and Education Conference, November, Orlando, FL.

Gugerty, L. Hicks, K., & Walsh, W. (1993). From an intelligent job aid to an intelligent-computer-aided training system: Training applications of the Integrated Maintenance Information System (IMIS). Proceedings of the 15th Interservice/Industry Training Systems and Education Conference, November, Orlando, FL.

Laurillard, D. (1987). Computers and the emancipation of students: Giving control to the learner. Instructional Science, 16, 1, 3-18.

Murray, W.R. (1991). An endorsement-based approach to student modelling for planner-controlled intelligent tutoring systems. *Technical Paper, AL-TP-1991-0030.* Air Force Systems Command, Brooks Air Force Base, Texas.

Ohlsson, S. (1986). Some principle of intelligent tutoring. Special Issue: Artificial intelligence and education. Instructional Science, 14, 3-4, 293-326.

Reigeluth, C.M., and Schwartz, E. (1989). An instructional theory for the design of computer-based simulations. Journal of Computer-Based Instruction, 16, 1, 1-10.