THE IMPACT OF ARTIFICIAL INTELLIGENCE ON MAINTENANCE TRAINING

Charles E. Thomas III
Honeywell Systems and Research Center
Minneapolis, MN

David J. Sykes
Albert Scsigulinsky
Honeywell Training and Control Systems Operations
West Covina, CA

Summary

The increasing complexity of military systems, reduced quality and availability of personnel, and reduced resources have made Weapons System Support and Readiness (WSSR) more and more difficult to maintain. This paper discusses surrounding issues and proposes a system concept for developing, combining, and integrating advanced training, job performance aiding, and Artificial Intelligence (AI) technologies in order to reduce the time and cost of maintenance actions and their instruction. In particular, expert systems coupled with video disk and other presentation and I/O technology will allow expert problem solving skills and knowledge to be made available to relatively inexperienced technicians, embodied in an integrated maintenance Job Performance Aiding/On-the-Job Training (JPA/OJT) system. A key component of the system will be an "explanation facility" through which the underlying reasoning of the system can be imparted to the technician. The basic objective of the OJT component is to build the conceptual knowledge of the technician rather than have him/her simply execute instructions. Since the expert system will handle the dual role of job performance aid and intelligent tutor, it is anticipated that the separation between maintenance actions and maintenance training will eventually become less distinct. Consequently, maintenance training equipment as we know it today can be expected to be gradually superseded by some form of "intelligent maintenance assistant."

The Problem

In recent years, the sophistication of military systems has increased rapidly, and the attendant maintenance and diagnostics requirements have become correspondingly more complex. At the same time, the quality and availability of incoming personnel has decreased. The resulting gap between the desired level of personnel expertise and the characteristics of incoming personnel have placed technical knowledge and skill in chronically short supply. As a result, Weapons System Support and Readiness (WSSR) is and will continue to be more and more difficult to maintain.

Current trends will exacerbate the situation: the population of eligible youths for military training is declining, the reading and skill level of military recruits is not increasing at the same rate as equipment innovation, technical documentation is becoming more complex and voluminous, turnover rates are high, and job competition from private industry is strong. This places tremendous pressure upon military training requirements.

In this paper, we will argue that a different approach to maintenance and training will be necessary to alleviate these pressures. Based primarily upon expert system technology and cognitive science research, we outline a system for combining job performance aiding (JPA) and on-the-job training (OJT) to optimize resource use for both maintenance and training.

Current Training Methods

Our systems concept is based upon experience with maintenance trainer design, interaction with Air Force subject matter experts (SME), and observations of aircraft maintenance training procedures at the organizational level, that is, flight-line maintenance at an Air Force Aircraft Maintenance Unit (AMU). A brief overview will motivate later discussions.

Having had some basic schooling in electronics, a new recruit begins to learn his or her Air Force Specialty Code (AFSC), with the expectation of reaching the apprentice-trainee "3" level at the end of initial training. Formal training is conducted in a dual learning mode--both classroom and OJT. Classroom instruction typically includes the use of simulated aircraft maintenance trainers or part-task trainers, as well as traditional lecture methods. These training aids provide familiarity with aircraft system locations and Technical Order (T.O.) procedures while avoiding use of flight line equipment and resources. Safe simulation of malfunctions without damage to real equipment is therefore possible.

Training on the actual aircraft employs the "buddy system" wherein a "5" level apprentice, qualified on a particular system, would supervise maintenance activities of the "3" level trainee. This apprentice training is meant to impart minimum basic concepts on a particular aircraft system: operational checks, the use of T.O.'s, diagnosis and troubleshooting, and removal and replacement of Line Replaceable Units (LRU). Typically, an additional year of OJT at the AMU is required before a 5-level or apprentice rating is achieved. Further experience and testing, as well as considerations of position and rank, are necessary before the supervisory "7" level of technician-specialist is achieved.

Shortcomings of Today's Training

Though classroom training at the apprentice trainee level has improved with the introduction of more sophisticated simulation devices, training on the flight line has suffered due to increasing burdens and changing expectations at the 5-level. The apprentice, while fully qualified to work on his system specialization, must also supervise 3-level trainees and impart those skills critical to mission readiness. Typically, the majority of maintenance tasks involve troubleshooting as opposed to operational checks and removal/replacement. Additional tasks such as keeping up with T.O. changes and new aircraft equipment also contribute to job complexity and impact efficient time management.

New directives, such as RIVET Force, are attempting to combine certain skill codes and specialties, thus increasing the knowledge base required and forcing additional emphasis on troubleshooting experience and training. As a result, the frequency of undesirable troubleshooting strategies (such as "shotgun" removal and replacement) may increase.

Expert Systems for Maintenance JPA/OJT: Application Philosophy

We believe that an expert system-based tool for integrated JPA and OJT offers great promise in aiding achievement of WSSR objectives. Military programs for development of expert diagnostic system prototypes are already well underway; these systems could be expanded to the type of intelligent machine assistant and tutor combination that we envision. An intelligent JPA/OJT tool could aid personnel on the job using a combination of expert knowledge of fault diagnosis and repair, advanced interface techniques, and most importantly, provide mechanisms to explain its reasoning and knowledge to the user in a tutorial fashion. The idea is attractive for its two-for-one use of personnel time: real maintenance work and individual training both get done simultaneously. If the funding of expert system technology continues at the current level, we view the development of a JPA/OJT tool as a potential near-term solution.

The central concept *in both traditional* training and advanced expert system-based JPA is to put available knowledge to work when and where it is needed. In training, the emphasis is on putting knowledge in the head; in AI, it is on putting it in highly usable form in a machine. But we strongly believe that <u>expert systems will not and should not be seen as eliminating the need to develop human experts</u>. First, advanced JPA's should be cooperative problem solving environments, where both man and machine contribute according to their respective strengths. Man is not just a sensor/effector. Second, without a dedication to training we will be left with no true human experts. Machine-based knowledge should be thought of as a *supplement*, *not a* replacement.

The distribution of human expertise in a typical domain is shown in Figure 1. Note that very high expertise is rare. Figure 2 illustrates the relative capabilities of the best human experts, state-of-the-art expert systems, and new trainees. Though there are claims to the contrary, it is highly unlikely that *practical, intermediate-term ES's* will be able to handle all rare and obscure faults; there will always be some expertise that cannot be incorporated into the knowledge base (e.g., using subtle perceptual cues, or postulating hypotheses based on "deep," or causally-based knowledge and experience). Consequently, it always will be necessary to rely on the best human experts to solve problems which require human flexibility.

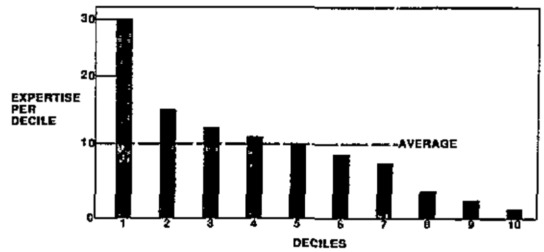

Figure 1. Distribution of Human Expertise

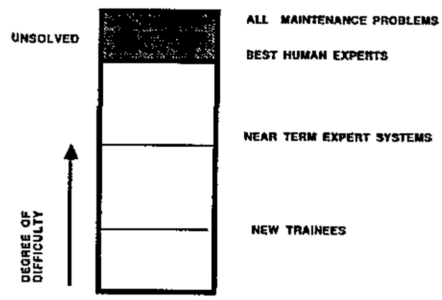

Figure 2. Relative Problem Solving Capabilities

There is an obvious need to ensure a continuing supply of expert individuals in the training pipeline. The use of expert systems strictly as a problem solving JPA will do nothing to develop the next generation of human experts. <u>Instead of adopting a smart machine/dumb man philosophy we need to focus on a smart machine/smart man approach</u>. In other words, we need to employ expert systems in building human expertise as well as in getting the job done.

Ideally, this would be accomplished by creating an OJT environment in which the function of the maintenance support system could vary as shown in Figure 3. Under crisis conditions it would act as a expert JPA, *minimizing* time to completion. Under conditions of light workload it would function as an off-line Intelligent Training System (ITS) using advanced explanation facilities and tutoring techniques (discussed later). Under normal conditions it would perform a mix of JPA and ITS which would help to improve the efficiency of OJT. For the maintenance domain, an expert OJT system could 1) improve the ability of the low end performers to the average level 2) support the high end performers when they are under time pressure by forcing a rational approach and 3) preserve eroding expertise in a usable form. The intention is not to replace man but to support him by integration of expert system knowledge with his own experience. For advanced JPA, this calls for a mixed initiative mode of operation in which either man or machine can take the lead depending on the particular circumstances. The basic architecture of a system intended for this dual role is illustrated in Figure 4.

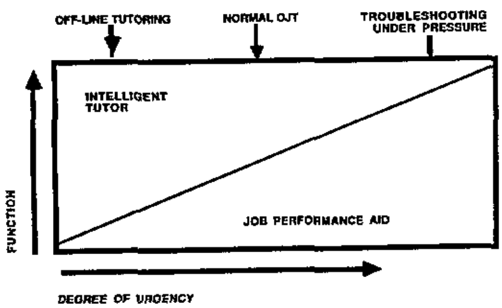

Figure 3. Varying Mix of JPA and Tutor

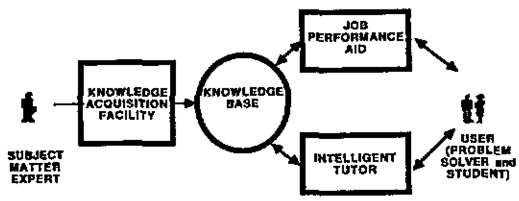

Figure 4. Combined JPA and Intelligent Tutor

Expert System Technology: An Overview

An expert system is designed to solve (or assist in solving) complex real-world problems by using knowledge derived from human experts. Expert systems are the most mature and commercially predominant implementation of AI technology today. Consequently, there are currently hundreds of expert systems in existence (of widely varying quality, complexity, and utility), an assortment of software "tools" for building systems large and small, and an abundance of introductory literature[1]. Table 1 lists several "classical" systems.

Although promising, today's ES's have shortcomings that have thus far limited their widespread and routine use. For instance, ES behavior is "robust" only within a very narrow domain of expertise; the system will fail to yield valid results (and not know it) if that range is exceeded. For similar reasons, *choosing a good problem* turns out to be the real key to a successful system because not all problems are amenable to ES treatment. Also, user-system interaction is generally very limited and user acceptance has been poor due to insufficient support for system conclusions and inability to answer questions as a human consultant would. In sum, there are few large and routinely used ES's due to either technical or human factors problems. This situation is changing, however, as these issues are addressed in research laboratories and commercial prototypes.

Table 1. Examples of Expert Systems

System	Domain of Expertise
MYCIN	Diagnosis of Infectious Diseases
PROSPECTOR	Geological Exploration
DENDRAL	Molecular Structures
MACSYMA	Mathematics
XCON	Computer Configuration
DIPMETER	Oil Well Drilling
CATS-1	Diesel-Electric Locomotive Maintenance

Figure 5 illustrates the basic architecture of an expert system. The knowledge base contains the human expert's knowledge, typically in the form of rules and facts. Rules generally take the form "IF (some conditions) THEN (some action)." The inference engine controls the order in which the rules in the knowledge base are invoked to solve a specific problem submitted by the user. The knowledge acquisition (KA) facility allows the subject matter expert, who may not be very conversant with computers, to enter his knowledge directly, with prompts from the system. In practice, assistance from a "knowledge engineer"--the AI equivalent of a software engineer--is usually needed to integrate the new knowledge into the existing system. The user interface allows a dialog between the user and the system, including requests for user-supplied data.

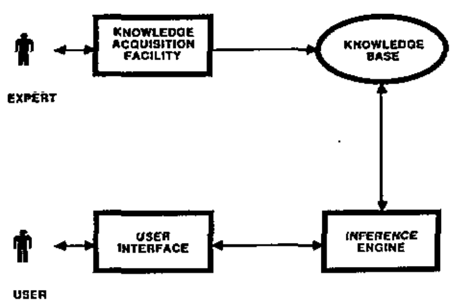

Figure 5. Basic Expert System

Although most "vanilla" ES's look the same at this level, in reality there are major *differences* in the design of nontrivial systems for different types of problems, particularly in the inferencing mechanism. For instance, an ES for diagnostic purposes works very differently from an ES for planning or design. MYCIN, a diagnosis ES, uses backward chaining to derive the probability of a certain diagnosis based on limited input data. XCON, an ES for design, uses a generate-and-test method to produce a hardware configuration based on a customer's order. There is no single mode of reasoning across all systems.

Why Traditional Expert Systems Do Not Make Good Trainers

On first inspection, it looks like an expert system's knowledge base would be just what is needed for training. After all, all the knowledge necessary for expert problem solving is there, and additional rules for teaching might be added easily enough. Unfortunately, using an ES knowledge base for teaching turns out to be not as easy as it looks, due to some subtle problems. In a nutshell, the the real-world equivalent is that many of the best human experts do not turn out to be effective teachers.

When designing a conventional expert system the primary or only objective is to create a successful problem solving tool. The architectures of most current expert systems reflect this principle. However, since it may be necessary to convince the user of the validity of the solution to the problem, some explanation capability is usually incorporated. In most cases this is achieved by "playing back" the appropriate rules to the user. While this explanation technique may sometimes be adequate for justification of the system's conclusions, it is far from satisfactory for instructional purposes. There are several reasons for this:

1) The rule's English equivalent may not be easily interpreted. For example, consider the following MYCIN rule:

```
IF:   1) THE INFECTION IS
         MENINGITIS
      2) THE SUBTYPE OF MENINGI-
         TIS IS BACTERIAL
      3) ONLY CIRCUMSTANTIAL
         EVIDENCE IS AVAILABLE
      4) THE PATIENT IS AT LEAST
         17 YEARS OLD
      5) THE PATIENT IS AN
         ALCOHOLIC

THEN:    THERE IS SUGGESTIVE
         EVIDENCE THAT
         DIPLOCOCCOS-PNEUMONIAE
         IS AN ORGANISM CAUSING
         THE MENINGITIS
```

In this case the underlying causal process to be explained is:

> ALCOHOLICS BREATHE IN THEIR OWN SECRETIONS, SO ORGANISMS FOUND IN THE MOUTH CAN TRAVEL TO THE LUNGS, CAUSING PNEUMONIA.

This explanation is not clearly evident from the above rule because the <u>real</u> reasons for why the premises lead to the conclusion are not present in the rule; causality is implicit rather than explicit. This means that the system can perform well, but doesn't "know" enough to be able to explain why what it does works.

2) The underlying knowledge is usually embedded in many rules and there is implicit knowledge contained in the interaction between the rules. This means that a single rule might only make sense in the context of other rules that work with it, either in terms of programming tricks, problem solving strategies, inference structure (i.e., control), or meaning.

3) The inferencing techniques used in an ES rarely even attempt to emulate the same form of reasoning as a human expert. The uniform control structures, logic, and/or probabilistic techniques that are used by an ES to arrive at the same conclusion as a human expert are usually very different than the human reasoning process. Though this is not important if a single (correct) answer is acceptable output, the surface behavior of the system (e.g., a series of data requests at the user interface) and resulting explanation of a system's "line of reasoning" can be difficult or impossible to follow or emulate, can be inefficient, or frustrating. In applications where understanding system recommendations is important, user acceptance is greatly reduced.

An expert system designed primarily for problem solving has proved to be of limited benefit as a device for teaching, without major overhaul. To meet the training objective the system must have explanation as its principal thrust. This requires an architecture and knowledge base that differs from the conventional "performance"-oriented approach. In the next section, we discuss AI as applied to classroom instruction, as background to the JPA/OJT system that incorporates many ICAI techniques.

Intelligent Computer Aided Instruction (ICAI)

The effectiveness of a traditional CAI system is dependent upon the ability of the courseware author(s) to anticipate every possible incorrect response by the student. The program can then branch to instructional material intended to remedy the postulated deficiency in the student's understanding or knowledge. In CAI, all responses are "canned" (i.e., written word by word and stored in the program verbatim) and are invoked at specific program branches.

In an ICAI system, responses may be at least partially <u>constructed</u> by the program as it executes, and can be invoked in a more flexible fashion. Ideally, the program can use this "intelligence" to adapt to instructional situations not specifically foreseen by the courseware author. Also, since an ICAI system includes an "expert tutor," the (programmable) teaching strategies of the best instructors in a particular domain may be captured and made available to many students via machinery. (Note that no ICAI system to date claims to replace instructors, only supplement them.)

Table 2 shows several examples of ICAI systems that have been successfully implemented and documented[2]. The basic architecture of an ICAI system is depicted in Figure 6. There are three main components: an expert knowledge base (essentially an embedded ES), the tutoring module and the student model. The expert knowledge base contains the specific task knowledge and, as will be described later, must be organized in a form suitable for instruction and explanation, as well as problem solving performance.

Table 2. Examples of ICAI Systems

System	Subject
GUIDON	Medical Diagnosis
SOPHIE	Electronic Troubleshooting
WEST	Elementary Math
BUGGY	Arithmetic Skills
SCHOLAR	Geography
MENO	PASCAL Programming
STEAMER	Steam Propulsion Plant Operation

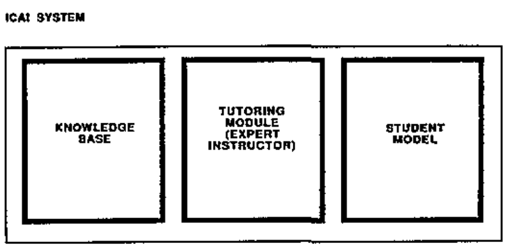

Figure 6. Basic Components of an ICAI System

The Student Model

The student model is a "snapshot" representation of the student's knowledge and understanding of the subject matter under instruction. Its function is to provide input to the tutoring module (e.g., gaps in a student's knowledge of the domain, problem areas, interaction history). In addition the student model may contain hypotheses about a student's possible misconceptions. Of necessity the number of recognized misconceptions must be limited to a common subset due to the virtually unbounded number of possible misconceptions that could exist about most topics.

The two most common methods of implementing a student model are the "overlay" and "bug" approaches. In the overlay method the student's knowledge is represented as a subset of the instructor's knowledge. In the bug approach the student's knowledge is characterized as a series of bugs (i.e., deviations) from an acceptable understanding of the subject matter. For example, a system called Buggy can recognize 130 different possible bugs in the performance of subtraction of two multiple digit numbers.

The student model is created and maintained using several techniques. These include direct questioning of the student, implicit evaluation of the student behavior, and historical observation of the student's learning experience. Some form of student model is key to a successful ICAI system.

The Tutoring Module

Good instruction requires knowledge about how to teach a particular subject in addition to a mastery of the subject matter itself. The tutoring module incorporates instructional theory into teaching strategies. It communicates with the student, selects course material, critiques his actions, prescribes remedial material and provides assistance as appropriate.

There are several teaching strategies used in ICAI systems:

- o Constructive debugging: characterize the bug and focus on correcting it.
- o Reactive learning: give feedback to the student on his ideas.
- o Socratic tutoring: allow the student to formulate his own beliefs.
- o Coaching: observe the student and interrupt when appropriate to suggest new ideas.

The last method, coaching, is the one that shows the most promise for job performance aids and OJT since it most resembles the master-apprentice scenario.

What is Needed for Maintenance OJT?

Maintenance Technician's Role

Currently, when T.O.'s are incorrect, incomplete, or ineffective in locating a fault, or when automatic test equipment itself fails or is also ineffective, a maintenance technician must originate his own procedures for troubleshooting and repair. We have argued in a previous section that this will continue to be the case even with the advent of expert systems. The training problem does not go away: at minimum, there will be a need for competent human backup to automated systems. Our basic argument is that some number of technicians must have the skills and knowledge to construct rather than just follow technical procedures for troubleshooting and repair. Given that expert system technology matures sufficiently to be used routinely by maintenance technicians as JPA's, how can we extend them for master-apprentice OJT, so they can serve double duty?

Making Minimum Knowledge and Skills Explicit

In any training setting, training costs should be contained by first identifying the minimum knowledge and skills necessary to perform tasks associated with a particular job, and then provide training which promotes good retention and the ability to apply what is learned to real problems[3]. It is generally recognized that teaching trainees to follow T.O. procedures alone does not promote flexible troubleshooting skills. Other conceptual knowledge and cognitive skills necessary for proficiency must be identified.

Identifying this minimum set is particularly important for using expert systems in training since typical systems contain only the knowledge necessary to perform well to solve the problem. They do not contain conceptual knowledge per se; it is said to be "compiled into" the performance knowledge in the same way that rationales for T.O. procedures are implicit in the procedures themselves and unavailable for inspection. Symptom-fault associations of the type contained in a rule base are the "compiled" version of the deeper causal chains and inferencing knowledge possessed by an expert.

One of the advantages of the targeted master-apprentice training paradigm is that a good journeyman can give a running explanation of what is going through his head as he constructs his procedures: he makes his rationale explicit for the learner. "Thinking aloud" protocols of expert problem solvers demonstrate clustering of closely associated concepts, use of "competitor sets" (sets of competing hypotheses consistent with accumulated evidence), and "chunking" of evidence and clues in ways that evoke particular hypotheses or a data-collection activity. Thus, it becomes imperative that we identify the types and uses of expert knowledge that will be necessary for good OJT commentary, because it will otherwise not be in the system when it is needed.

Recent research suggests that troubleshooting proficiency has at least three basic prerequisites:

- device-specific conceptual knowledge
- an understanding of how to use device knowledge to direct the application of available troubleshooting techniques
- skilled performance of basic troubleshooting techniques (e.g., resistance checks, voltage checks)

<u>Device Knowledge</u>

It is obvious that the technician must know something about the device he must troubleshoot. What is a source of radical disagreement, from the standpoint of impact on resultant training requirements, is how much and what type of knowledge is necessary for a technician to understand and reason about complex systems. If a knowledge of theory is deemed necessary (e.g., the ubiquitous Ohm's Law) then training will be much more costly. Our position is that for most troubleshooting, there is a middle ground between learning strictly procedural material and learning in depth theoretical material. That middle ground is described in terms of device function and role, outlined below.

Engineered devices and their components can be understood using four highly interrelated concepts[4]:

- The "role" of a device is the function that the device plays within the context of other structures (e.g., the role of a power train is to transmit the rotary motion that is produced by the engine to the rear wheels).
- The "function" of a device is its behavior as a black box, but unlike its "role" it is context-free (e.g., the function of a wheel is to spin, but its role must be defined in terms of what spinning does for the system as a whole, as in a gyroscope, or a steering wheel).
- The "structure" of a device describes what components it is made of and how they are connected. Each component can in turn be described in terms of its own role, function, etc.
- A device's "mechanism" is how its structure produces its function. This is usually a physical causal chain (e.g., in a Rube Goldberg machine, the swinging boot kicks the bucket that contains the ball that rolls down the chute and knocks over the first domino...).

A synthesis of recent research leads us to believe that it is knowledge of the functions and roles of individual system components and their functional interconnections that is crucial, but only to the particular component level that the technician can (or is expected to) affect. If a technician is responsible for repair at the LRU level, he need only know enough about each component's inputs, conditional processes, resultant outputs, and functional connections to other LRU's to be able to apply general troubleshooting principles to construct an efficient set of procedures. To apply these principles, he need only know how an LRU will act given a certain set of inputs. Specifically, what he does not need to know is the theory (mechanism) behind an LRU's operation or what subcomponents it consists of (structure) if the LRU is the lowest level of his repair responsibility.

It is only necessary to promote enough comprehension of each component so that its effects on other components can be understood. For example, in the case of a simple electrical battery it is not important to understand the basic electrochemical process taking place within the battery. It is sufficient to know the effect of various loads at the terminals and what the impact is when the parameters go outside normal limits. The only exception that is allowed to this general rule is if a component's processing is more easily explained and remembered by reference to its subcomponents (structure and/or mechanism) than by a description of its overall processing function.

We have dubbed this level of understanding of a device as a "minimum mental model," emphasizing an understanding of the interaction between system components. It has been experimentally demonstrated that a minimum mental model of a device is necessary and sufficient to promote faster learning, more accurate retention, and faster execution of operations procedures than strict memorization of those same operations procedures without the model[5]. In addition, it allows inferences to be drawn about faults and their possible locations based on expectations about proper and improper device functioning.

Strategic Knowledge

The other critical component of proficiency is more elusive. It is strategic knowledge, and concerns the use of a device model in constructing a set of procedures which will isolate a fault. Strategic knowledge can be viewed as a body of decision rules that are invoked by particular aspects of a conceptual view of the domain. This knowledge has all the hallmarks of expert problem solving: use of perceptual cues, heuristics (hit-or-miss rules of thumb), attentional focus, hierarchical goal structures (e.g., divide-and-conquer troubleshooting strategy), and flexibility. This is the knowledge that is buried in T.O. procedures, and that is implicit rather than explicit in most expert system knowledge bases.

Research has demonstrated that there are large differences in skilled and unskilled airmen in their ability to construct an understanding of the problem space, pursue hierarchical goals, and systematically focus their hypotheses[6]. It has been said that the task goal structure of some unskilled airmen rarely becomes more elaborated than "get the supervisor off my back." Without guiding strategies, troubleshooting can and often does turn into "swaptronics," in which a technician replaces each replaceable part with a new part which is believed to be fault-free. In problem solving parlance, this is called "blind search," and is one of the most inefficient methods of problem solving. Because one keeps swapping until the test equipment indicates normal unit functioning, this blind search usually works if there is a single fault located in a swappable part. For this reason, it is effective, but extremely inefficient and costly.

The training problem is this: in extended practice sessions, it is the number of correct performances of a task that determines the adequacy of learning and subsequent skill proficiency, not just the number of trials[7]. This is true of cognitive as well as motor skills. The obvious lesson is that swaptronics is not valuable experience; "doing the job" in this way *doesn't provide good practice.*

A training system should therefore emphasize strategic problem solving, since there are large gains to be had for a more structured approach to the troubleshooting activity. The training should include a large number of practice sessions where the job is done "right," that is, efficiently and expertly. This is the role that could be played by JPA/OJT expert systems.

System Concept for JPA/OJT Expert System

System Concept

An expert JPA system designed for troubleshooting could and should also be designed to be articulate, augmented with explicit device and strategic knowledge for use in training. Equipped with this knowledge and, optionally, other media extensions, an articulate expert system could serve in a master-apprentice, learning-by-observing, learning-by-doing paradigm.

We envision this system as a kind of "Intelligent Technical Order" system, with the system using and explaining its diagnostic expertise, and supplying streamlined technical instructions to the technician. Actual troubleshooting procedures would be carried out and results reported back by the human technician.

Videodisc imagery (stills and moving), graphics, voice synthesis and perhaps voice recognition, and on-line database interface serve as add-ons to this basic concept, allowing the trainee to observe correct performance of basic troubleshooting techniques that have not been mastered (e.g., measurements), illustrations of device models (e.g., highlighted schematics of components and their functional relationships), and other JPA-oriented information display techniques (e.g, no-fault values). Though it will not be pursued further in this paper, combining media and "intelligence" creates a highly desirable information delivery synergy, particularly when the goals of job performance aiding and training become temporarily indistinguishable (as in the case of basic measuring methods).

As discussed in prior sections, existing written troubleshooting and maintenance procedures can reflect hidden rationales that are necessary to develop a full understanding of why the procedure is reasonable. What is needed for training is a system that displays expert cognitive skills as a model for the trainee, explaining and justifying its problem solving behaviors in terms that make explicit the underlying concepts, strategies, and device models. For instance, a T.O. procedure to test a certain component before another may have been motivated by an historical data base that shows the first as the most likely fault candidate. The JPA/OJT system should point this out:

"At this point, the fault could be caused by one of two things. Either component A is not Xing correctly or component B is not Zing correctly.

[Optional highlighted video of component functions.]

Component A is usually the problem. Test it first.

[Optional video segment on testing procedure.]"

In our studies of explanation techniques, we have identified other implicit knowledge and meta-knowledge that should be explicitly stated in an explanation. For instance, heuristics (rules of thumb) should be identified as such, just as a master/teacher would:

"Usually, jiggling this knob may reveal a dirty contact, but this doesn't always work."

Other examples of crucial embedded knowledge that should be called out in explanation of procedures is:

- o relating data requests to pursuit or change of current subgoals
- o focus of attention on cues triggering alternative hypotheses
- o distinguishing causal shortcuts (A-- D) from full causal chains (A-- B-- C-- D)
- o justification of elimination of alternatives (e.g., "the fault is not in component A because...")

Note that this JPA/OJT combination makes a tradeoff; the job gets done a little slower than with the pure man-as-effector paradigm, but the man gets "smarter." There would be less instructional intervention than with an off-duty tutoring system, but on the other hand, there is useful work getting done during learning.

This system concept also has an important efficiency going for it: work is learning. In a nutshell, this is because the procedures take on meaning: "what to do," "when to do." and "how to do" get related to "why to do." By making procedures meaningful--and successful-- we satisfy the requisite criteria for making practice useful, and get better use of maintenance time for both maintenance and training for those situations when expert systems will fail and expertise must be available elsewhere.

Concept Expansion

As it stands now, our system concept (Figure 7) is not a complete, classroom-type ICAI system. This is intentional, primarily because a really robust ICAI system which does socratic tutoring and testing is very difficult to build, particularly with respect to user modeling. We have chosen instead a limited "overlay" user model (which views novice knowledge as a subset of expert knowledge and ignores possible misconceptions) and developed a model of explanation with associated algorithms. The primary emphasis is on effective explanation of expert performance on-the-job.

The Explainer block contains the knowledge and algorithms needed to prescribe and create explanations and justifications of the actions and knowledge of the expert system. The Metaknowledge Base contains supporting knowledge about the expert system's "performance"-oriented Knowledge Base and its control strategies. Embodying some basic theories of understanding in dialogue, the Explanation Model within the Explainer block is relatively independent of complex user states except in terms of past dialogue content and structure. It is primarily a set of algorithms that calculate what might be missing from the user's understanding of expert system behavior, using the limited User Model as input. It applies some "metalevel" algorithms that operate on the knowledge bases to construct explanations consistent with high-level models of the components of satisfactory explanation for several generic subject matters (engineered devices, goal-oriented diagnostic actions). Explanations are constructed according to principles of natural structure[8].

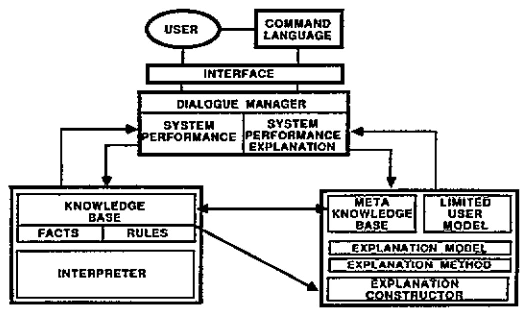

Figure 7. Simplified System Concept

This approach is roughly equivalent to asking "how would I explain this device or line of reasoning to a person who I don't know personally?." A great deal can be done with a system of this type without becoming entangled in a lot of expensive complexity of the type faced in elaborate ICAI systems. A key assumption is that the user can ask for more detail if he doesn't understand, and that there are conversational ways of testing a user's approximate level of comprehension. Emphasis is placed on explaining rather than interactive tutoring.

Knowledge Representation and Control Structures

Using an AI-based system for training places certain demands upon its modes of problem solving and the form of its knowledge base. As was stated above, expert systems are usually oriented toward performance rather than explanation; they tend to be black boxes rather than glass boxes. At least two very basic characteristics of typical expert systems will have to be modified before they can be acceptable substrates for training applications:

First, the problem solving method used by the system must be capable of emulation by humans. For instance, "brute force" computational methods employed by some expert chess programs make their problem solving method impossible for humans to emulate. Backward chaining is a popular control structure for an expert system, but systems that use it (e.g., MYCIN) are psychologically invalid and virtually impossible to learn. Much good descriptive work has been done on human diagnostic problem solving, in which forward and backward reasoning is mixed to accommodate working memory and other human factors. This work should be consulted as a model for future control structures that are in fact emulative. For example, in NEOMYCIN progress has been made toward an extensible representation of domain-independent strategic knowledge for instructional purposes[8].

Second, the most critical strategic and supporting conceptual knowledge underlying diagnostic procedures must be explicitly represented and either integrated with or cross-referenced to the "performance" knowledge base (see Figure 8). This is necessary to allow the system's inferences and lines of reasoning to be justified in a manner compatible with "natural" explanation. Use of knowledge organizations which are hierarchical or heterarchical and multiply-indexed (i.e., through multiple relations) is desirable for explanation since systems of objects can be represented from various relational viewpoints (e.g., structural, functional, mechanistic). For example, operational equipment can be represented a hierarchical "partology" describing the structural composition of a class of equipment. Using the same objects with a different relational perspective, equipment can be represented as a "functionology," with each part associated only with its functional role in the system as a whole. (Incidentally, we believe that this kind of knowledge will eventually be necessary anyway, for purposes of updating the system by way of intelligent knowledge acquisition and maintenance.)

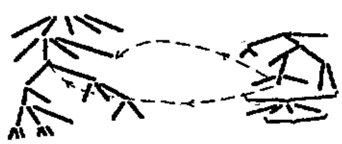

Figure 8. Problem Taxonomy is Separate From Diagnostic Strategy

For instructional purposes, rules turn out to be a poor way of representing chunks of interrelated knowledge. A frame-based or semantic network-based representation currently best fills this need (note that the two are formally equivalent, just different mechanisms). A frame-based system actually incorporates a number of generic rules (e.g., for inheritance) that allow the representation to more efficiently encode highly interrelated knowledge chunks.

It is this knowledge chunking that we are after, since it allows us to talk about knowledge in a more connected way (e.g., a bigger chunk than a single rule). Attempts to use MYCIN rules as the to-be-learned material have shown that it is exceedingly difficult to learn rule systems per se[9]. The rule forms in expert systems are apparently not the learning "primitives" they were thought to be. Experience has also shown that an explanation/training facility cannot be retrofitted to a traditional rule-based system--the knowledge requirements are too different. Thus, the time to think about design requirements for a JPA/OJT system of the type discussed in this paper is before a strictly performance oriented system is built, not after.

Conclusions

Artificial Intelligence can be expected to have a significant impact on maintenance training in the future. Expert job performance aids are expected to improve maintenance effectiveness, especially for moderately difficult diagnostic tasks. However, human experts possessing a sound conceptual knowledge of systems and troubleshooting strategies will still be necessary to handle the rare and exceptional faults.

In order to provide cost-effective training necessary for maintenance personnel to understand and reason about increasingly complex systems, we have proposed a system concept for an "intelligent" JPA/OJT system. This system would provide explanation and multimedia illustration of the device models and troubleshooting strategies underlying expert system and human performance. This high quality "articulate expert system" would provide a means of combining JPA and OJT without sacrificing the conditions necessary to make both activities effective and efficient.

References

1. Hayes-Roth, F., Waterman, D.A., Lenat, D.B., Building Expert Systems, Reading, Mass.: Addison-Wesley, 1983.

2. Sleeman, D. & Brown, J.S. Intelligent Tutoring Systems. London: Academic Press, 1982.

3. Hollan, J. D., Hutchins, E. L., Weitzmann, L., "STEAMER; An Interactive Inspectable Simulation-Based Training System," AI Magazine, Summer 1984, pp. 15-27.

4. Weld, D.S., "Explaining Complex Engineered Devices." (Tech Report 5489) Bolt, Bernanek & Newman, 1983.

5. Kieras, D.E., Bovair, S., "The Role of a Mental Model in Learning to Operate a Device," Cognitive Science, 8, 1984, pp. 255-273.

6. Gitomer, D., A Cognitive Analysis of a Complex Troubleshooting Task. Doctoral Thesis, University of Pittsburg, 1984.

7. Fisk, A., Scerbo, M. & Schneider, W. "Issues in training for skilled performance." Proceedings of the Human Factors Society, 1983.

8. Goguen, J.A. & Wiener, J.L. "Reasoning in Natural Explanation." International Journal of Man-Machine Studies, 19, 1983, pp.479-496.

9. Clancey, W.J. and Letsinger, R. NEOMYCIN: Reconfiguring a rule-based expert system for application to teaching, in Clancey and Shortliffe (editors), Readings in medical Artificial Intelligence: The first decade, Reading, Mass.: Addison-Wesley, 1983.

About the Authors

Charles E. Thomas III

Mr. Thomas is a Research Scientist at Honeywell Systems Research Center, Man-Machine Sciences Group, Training and Simulation Section in Minneapolis, MN. Mr. Thomas is presently pursuing applied research in explanation/presentation facilities for expert systems. This program is part of an effort to use existing research in cognitive models, both diagnosis and planning, dialog engineering, structure of "natural" explanation, and the explicit representation of strategic control knowledge: the objective is to develop software architectures that allow sophisticated explanation capabilities in an applied, cooperative,man-machine problem solving environment. Mr. Thomas has also worked on a variety of SRC projects involving technology assessment, training system concept development and training requirements analysis. He holds a BS degree in Psychology from Duke University, a MS in Ed. Psychology from the University of Minnesota, and a MBA from the University of Arkansas.

David J. Sykes

Mr. Sykes is a Staff Engineer with the Training and Control Systems Operations of Honeywell in West Covina, CA. His responsibilities include planning AI programs for training systems and development of advanced visual simulation systems. Prior to this, he was with Honeywell Information Systems in Phoenix and Boston where he designed distributed processing systems and computer networks. Mr. Sykes has published fifteen papers on computer networks, system security and simulation of distributed systems. He holds a BS in Electrical Engineering, a MS in Computer Science and is currently completing his Ph.D. in Industrial Engineering at Arizona State University.

Albert Scsigulinsky

Mr. Scsigulinsky is a Senior Systems Training Analyst at Training and Control Systems Operations of Honeywell in West Covina, CA. His responsibilities include training and mission analysis, system requirements and Human Factors design for Advanced Programs in visual simulation and maintenance trainers. He has had prior experience in training analysis and system testing of Honeywell's F-15 Simulated Aircraft Maintenance Trainer (SAMT) program. Mr. Scsigulinsky has published three papers in the area of memory and cognitive psychology. He holds a BS in Psychology from St. Lawrence University, a MS in Experimental Psychology from California State University, Fullerton and is currently a Ph.D. candidate in Cognitive Psychology at the Claremont Graduate School, California.