International Journal of Artificial Intelligence in Education (1999), 10, 257-277

The Advanced Embedded Training System (AETS): An Intelligent Embedded Tutoring System for Tactical Team Training

W. Zachary *CHI Systems, Inc., 716 N. Bethlehem Pike, Suite 300,Lower Gwynedd, PA wayne_zachary@chiinc.com , http://www.chiinc.com, Phone: 215-542-1400, Fax: 215 542-1412*

J. Cannon-Bowers *Naval Air Warfare Center-Training Systems Division, Orlando, FL, Cannon-BowJA@navair.navy.mil , http://www.ntsc.navy.mil*

P. Bilazarian, D. Krecker, and P. Lardieri, *Lockheed Martin Advanced Technology Laboratories, Camden, N.J. {pbilazar, dkrecker, plardier} @atl.lmco.com , http://www.atl.lmco.com*

J. Burns *, Sonalysts, Inc., Orlando, FL burns@sonalysts.com , http://www.sonalysts.com*

Abstract. The Advanced Embedded Training System (AETS) applies intelligent tutoring systems technology to improving tactical training quality and reducing manpower needs in simulation-based shipboard team training. AETS provides layers of performance assessment, cognitive diagnosis, and team-training support on top of the existing embedded mission simulation capability in the Navy's Aegis-class ships. Detailed cognitive models of trainee task performance are used to drive the assessment, diagnosis and instructional functions of the system. AETS' goal is not to replace human instructors, but to allow one instructor to perform the work of several, and in a more consistent and efficient manner than possible today.

INTRODUCTION

The development of automated instruction has proceeded through several stages, dating back to simple computer-aided instruction systems in the late 1950s which provided strictly didactic instructional material in rigid instructional sequences. Dominant themes emphasized in the last two decades have been dynamic environments for applying and practicing problem-solving knowledge (problem-based learning), diagnosis based on the underlying knowledge state of the student rather than observed behavior alone (student modeling and cognitive diagnosis), and adaptation of instruction to the student's evolving knowledge state (adaptive tutoring). In recent years, the dominant paradigm has been what Collins, Brons and Newman, (1989), called "cognitive apprenticeship" systems, in which the computer acts as an adaptive coach to the student (the apprentice) who works through a series of problem-solving exercises The current generation of intelligent tutoring systems (ITS) all fall into this category, including the widely cited LISP tutor (Anderson, Corbett, Koedinger, Pelletier, 1995), the SHERLOCK maintenance tutor (Lesgold, Lajoie, Bunzo & Eggan, 1992), the physics tutors (Van Lehn, 1996), among others.

All of these research-oriented intelligent tutoring systems were built in problem domains which share several features. The domains involved little indeterminacy (cases in which multiple knowledge paths can lead to the same action), the relevant knowledge was comparatively closed and could be represented by relatively few rules, and the problem-solving activities were individually-based (rather than team-based) and involved non real-time problems that could be easily stopped and re-started. Unfortunately, tactical domains possess none of these features. There is great indeterminacy, the problems and required knowledge are complex and open, the problem solving is team-based, and the problem environment is fast-paced. Perhaps because of these reasons, classical intelligent tutoring systems technology has been slow to emerge into the tactical world.

A major attempt to address these issues, however, has been made in the Advanced Embedded Training System (AETS), an Advanced Technology Demonstration project undertaken by the U.S. Navy to apply intelligent tutoring technology to shipboard team training. The program is described in Zachary and Cannon-Bowers, (1997), and Zachary, Bilazarian, Burns and Cannon-Bowers, (1997).[1] AETS sought to apply the concepts of intelligent tutoring—problem-based learning, cognitive diagnosis, student modeling, and focused, adaptive tutoring—but found that new approaches had to be developed to deal with the indeterminacy, open knowledge spaces, team problem solving, and real-time nature of tactical domains. This paper reviews the AETS architecture and operation, with focus on these new approaches and the underlying issues they addressed. The best place to begin, however, is with an overview of the specific task domain addressed by AETS. This domain is similar to many other tactical domains, and provides the background needed to frame the problems addressed by AETS and the solution strategies generated.

The Task Domain

The AETS focuses on the Air Defense Team, which is one of several (albeit one of the most important) teams functioning in a ship's Combat Information Center. AETS particularly focuses (for reasons discussed shortly) on ships utilizing an advanced command and control system called Aegis. The general job of the Air Defense Team is to protect own-ship and other assets (e.g., an aircraft carrier) from air attack. This job can be particularly difficult under ambiguous conditions, such as in the Persian Gulf, where the skies are filled with commercial aircraft, industrial aircraft (e.g., helicopters moving to/from oil drilling platforms), and military aircraft from multiple countries, many of which are hostile to one another and to US forces, and where the threat of terrorist attack is also omnipresent.

The activities of the air defense team revolve around representations of airborne objects (which may be fixed-wing aircraft, rotorcraft, or missiles) in the space around the ship. These objects are generically called 'tracks', and are displayed as icons on a geographical display at each person's workstation. The major source of track data is the powerful radar on-board the Aegis ship, but there are other sources, too. These include data from radars on surveillance aircraft that are relayed back to the ship; from other shipboard sensors (such as ones that listen for other electronic emissions from an object); and from sensors on other ships, ground forces, and other intelligence sources. A track may be known from just one or from multiple data types. In some cases, different data sources may create different tracks that, in 'ground truth', represent the same object. At the same time, single tracks can represent multiple objects, such as close flying aircraft. The air defense team as a whole is responsible for:

- disambiguating and creating as much additional information about each air track as is possible or needed;
- determining if a track represents any potential threat to the own-ship or something the own-ship is defending; and
- taking a variety of actions to neutralize or minimize any potential threat.

The space of actions that can be taken is large. The air defense team may, for example, send verbal warnings calling for a track to change course. In some cases, the team may direct friendly aircraft to try to get visual identification on a track, to challenge the track, or to escort it away from the own-ship. And in some rare cases, the team may launch a missile to attack

1 Portions of this paper were presented at the ITS-98 (Intelligent Tutoring Systems Conference, 1998) in San Antonio, Texas, USA, and were published in the proceedings in Zachary, Cannon-Bowers, Burns, Bilazarian, and Krecker (1998).

and destroy the track. Internally, members of the team will verbally pass information they have inferred or received to other members of the team. These internal messages are important in correlating data across sources and helping the team share a common mental model of the airspace.

The Aegis command and control system has a unique capability to run in an embedded simulation mode, in which a mission simulation is run on the Aegis computers and 'worked' by the crew using the actual tactical workstations on the ship. This embedded simulation capability is called ACTS (Aegis Combat Training System, as its purpose is to support training), and was the foundation on which the AETS system was built. Currently, however, when a tactical team receives team training with this embedded simulation, all performance measurement, diagnosis, feedback, and instruction is performed by human instructors, often in a labor-intensive manner (up to one instructor per student), and with a great deal of inconsistency across instructors and ships. Typical embedded training simulation problems (which are indicative of real-world complexity) will involve up to 100 tracks, many of which are in the area simultaneously. While on duty, an operator (called a watchstander) will repeatedly choose a track to analyze, select it on the geographical display, and interact with various functions in the software system to gain data on the track and to analyze that data. For example, a watchstander might select (or 'hook') a track that is unidentified, and interrogate the system to see what electronic emissions have been identified from it. This watchstander may, after some interactive analysis (and remembering earlier verbal messages from the electronic warfare supervisor, who focuses on electronic emissions), conclude that an emission from a radar of type 'x' was associated with the aircraft. The watchstander would use knowledge that radar x is typically used to lock onto targets by missiles of type 'y', which are carried only on aircraft of type A or B. The watchstander would then reason about those two aircraft, and remember that one way in which they can be differentiated is by their flight speed. The watchstander might then interrogate the system to determine the speed of the track, and from this data conclude that it was consistent with aircraft type B, and thus identify the track (perhaps tentatively) as a type 'B' aircraft carrying a type 'y' missile. All this information would then be entered into the system, and (sometimes) verbally shared with other team members. All watchstanders have to share attention across many tracks, making sure that they do not persevere on one track too long, allowing another potentially dangerous track to 'slip through' unnoticed. In the course of an hour, most watchstanders in the team will hook hundreds of track symbols (many of them dozens of different times), enter thousands of keystrokes, and make dozens of verbal announcements.

Design Goals, Rationale, and Principles

The existing Aegis embedded training system is called ACTS (Aegis Combat Training System) and is intended to be used 'afloat', i.e., while the ship is at sea or in port.[2] The crew does not leave the ship to receive training, but rather trains by using the ACTS system to work training scenarios, while on-board trainers observe and provide feedback on their performance during the simulation. Because the basic embedded training system provides no support for real-time monitoring of the trainees by the trainers, there are nearly as many trainers observing performance as there are trainees (i.e., one per watchstation), making this form of training labor-intensive. In addition, there is substantial inconsistency among trainers in the data they record, the feedback they provide, and the focus of their training (e.g., workstation 'knobology' versus tactical knowledge versus team performance). This is not the fault of the trainers. They are not professional trainers, but rather other (albeit more experienced) members of the tactical team, and there is no standardized training curriculum or instructional materials available to them. Moreover, the simulations are fast paced and the ACTS system provides no support for

2 However, ACTS is also used in shore-based training sites, to give new crews realistic practice in working with Aegis equipment and tactics.

these instructors to capture or record actions at the workstations or over the voice communication channels. Given these constraints, the human trainers do remarkably well.

The AETS was undertaken to address some of these limitations of the basic embedded training process by building intelligent instructional elements on top of the ACTS embedded simulation capability. The approach and design goals were driven by two overall system objectives. The first was to provide higher levels of trainee performance for a given amount of training. The second was to reduce costs, largely by requiring fewer human instructors to support the training process. Specifically, AETS was based on the following design goals:

- require fewer instructors -- to allow the number of human trainers involved to be reduced from a 1:1 ratio (in the current case) to a 1:4 ratio;
- support more consistent training -- to provide tools that would allow the human trainers to observe and provide feedback on trainee performance in a more consistent manner that would reflect underlying training and performance objectives;
- enable better recording and analysis -- to provide tools that would render trainee performance during the embedded simulation machine-accessible and make these performance records available to automated performance assessment and diagnosis as well as human-trainer manipulation; and
- provide improved instruction and performance feedback -- to support the human instructor with objectives-based automated and semi-automated instruction and curricular materials.

The task environment imposed three major constraints on the achievement of these design goals with existing intelligent tutoring technology. The first was 'teamness.' The air defense problem is solved in the Aegis system with a team of human operators, and the embedded simulation basis for AETS meant that all members of the team had to work the problem simultaneously and collaboratively. As a result, the conventional intelligent tutoring approach of stopping the problem to provide timely feedback or instruction was problematic at best. It was impossible to stop one operator without stopping all others simultaneously, and this would totally destroy the cognitive flow of the overall problem solving process for the rest of the team. Thus, teamness meant that any feedback or instruction given during the problem simulation could not involve stopping the underlying simulation.

The second constraint was the underlying level of operator workload. In general, the air defense problem has an almost video-game-like pacing, with all team members needing virtually all of their attention at all times during the simulation. This left little residual mental, sensory, or motor resources available to process or interact with instructional material or performance feedback during the problem simulation. Thus, the underlying workload level meant that feedback and instruction during the problem had to be minimized and/or deferred until the simulation was over for a more detailed treatment. This, in turn, placed additional emphasis on the ability of AETS to capture and recreate the situation in which the feedback was originally triggered, so that it could be contextualized for the trainee when the feedback was eventually provided.

The third major constraint was the indeterminacy of the solution space in the Air Defense task. At the lowest level, the human-computer interfaces for each watchstander are such that many distinct sequences of interaction can be used to accomplish the same human-computer interaction goal, e.g., display a specific piece of information about a given track. At higher levels, there are many paths of reasoning that can be used to accomplish mission-level goals with regard to tracks, e.g., classify the intent of a specific track. And at the attentional level, the fact that the watchstander must be maintaining many threads of reasoning at the same time, each about a different track, means that there are many ways in which the flow of processing of tracks may proceed. A corollary of this constraint is the openness and complexity of the knowledge that must be used by the various watchstanders. There is substantial knowledge about interaction with the (highly-complex) workstation, knowledge about the current situation, knowledge about background facts and relationships (e.g., aircraft types, speeds, characteristics), and knowledge of tactics, standard procedures, and so on. Thus, it was virtually impossible to define any specific action as the correct action (to the exclusion of others) or to define any specific reasoning sequence (to the exclusion of others) that was needed to generate the action.

The above constraints and goals framed the overall design rationale for AETS. The underlying philosophy was a cognitive apprentice approach as discussed earlier. However, a key design principle from the outset was that there would be dual instructional roles, i.e., that the role of tutoring and instruction was to be explicitly divided between the human instructor and the tutoring system itself. The teamness and workload constraints further refined this notion and led to a second key design principle, that the flow of instruction and feedback would be systematically partitioned into four separate streams:

- during the simulation, generated automatically and directed to the trainee;
- during the simulation, generated automatically and directed to the instructor for inclusion in subsequent post-simulation comments or to prompt the instructor to observe some on-going or upcoming process;
- after the simulation, generated by the instructor and delivered via the individual and team debriefing on the results of the simulation; and
- between training sessions, generated both automatically and manually and delivered similarly, in the form of individualized training and/or remedial instruction.

Together, these two principles of dual instructional roles and feedback partitioning of the instructional stream structured the design for the ‘back end’ of AETS, the part that provided instruction and feedback (in contrast to the ‘front end’ that observed and diagnosed trainee behavior). These principles allowed specific focused information to be provided to the individual watchstanders by the system, but at a low level that would not deleteriously affect their task performance, and during the simulation so as to not disrupt teamness. Other information beyond this low level would effectively be buffered by the human instructor, who would sort and organize the assessment and instructional material and determine what to present during the problem and what to present for the after-simulation review. This organization of the instructional stream also identified the need to incorporate a theory and method for the instructor’s post-problem debrief and instruction at the team level. As discussed later, the theory and method selected was that of Team Dimensional Training (Smith-Jentsch, Zeisig, Acton, and McPherson, 1998). The front end of AETS -- the part that provided observation, assessment, and diagnosis -- was designed around two additional principles that were responses to the constraints of indeterminacy of the solution and knowledge space. These principles can be termed situational relevance and focus toward correct action.

The principle of situational relevance provided a criterion for defining the action space to be considered in the processes of observation, assessment, and diagnosis of the trainee. While actions at the lowest or atomic level (single keystrokes, eye movements, etc.) were highly indeterminate in the task domain, there was another, more abstracted level at which actions could be more crisply evaluated. This was the level of appropriateness or relevance of the action in the current situation (defined to include the watchstander’s environment both within the Combat Information Center and the external tactical situation of own-ship). In any given situation (or more precisely any time slice within an evolving situation), human commanders and instructors generally agreed on what it was appropriate for that watchstander to do. For example, it was generally required for an air defense coordinator to give verbal warnings (to change course) to certain categories of aircraft approaching own-ship at specific distances, and it was considered inappropriate for the coordinator NOT to do so. Similarly, within a bounded segment of a situation, it was possible to determine that it was appropriate to make or attempt to make identification of certain tracks, to make certain kinds of communications to colleagues and superiors, and so on. The appropriateness of these actions (and incorrectness of omitting them) was defined by the applicability of doctrine, standard ship procedures, and other explicit sources.

However, the kinds of actions to which this notion of situational relevance applied were not the atomic actions discussed above. Rather, these situationally relevant actions were composites of many separate atomic actions. They represented abstracted actions, and in fact represented the outcome of (indeterminately) many reasoning paths and atomic action sequences. It was also the case that multiple actions became situationally relevant in essentially the same context, although the order in which they needed to be undertaken within this context might be indeterminate. Nonetheless, these situationally relevant actions created a level of analysis at which there was very little indeterminacy, because it became possible to determine that it was 'correct' for the action to be taken in that context, and 'incorrect' for the action not to be taken or taken in some other form. This was particularly true if one could assign a 'window' around each of these abstracted actions that defined the temporal or situational boundaries in which the abstracted action was situationally appropriate. Thus, the AETS performance observation, assessment and diagnosis processes were designed to operate at the level of these abstracted actions, also called High Level Actions or HLAs. In addition to providing a strategy for dealing with the indeterminacy of action, the use of HLAs also provided a strategy for dealing with the complexity of the action space. Because each HLA encapsulated a fairly large chunk of atomic actions, using HLAs as the basis for observing, assessing and diagnosing trainee performance dramatically reduced the number and frequency of actions that would have to be considered by the system. Furthermore, this reduction in the granularity of the action space further contributed to a solution to the workload problem, in that providing feedback to trainees at the level of HLAs would result in a smaller number of instructional transactions that would have to be processed by each watchstander.

The last design principle addressed the nature of the diagnosis and instructional process. A common approach in intelligent tutoring systems (Wenger, 1987) is to focus on diagnosing the presence of incorrect or 'buggy' knowledge. If, for example, a student is observed noting that "2+2=5", the tutor might seek to determine why the student made such an error -- did the student not understand addition?; not understand the quantity '2'?; or did the student memorize the number fact incorrectly? This approach works best when the knowledge space is well-defined and where there are commonly occurring misconceptions. However, the indeterminacy of knowledge and of action (at least at the atomic level) made both of these conditions invalid. It was very difficult to distinguish potentially buggy reasoning paths from non-buggy ones that had not been observed before, and the problem was complex enough that there was no commonly occurring buggy knowledge[3]. This suggested a design principle that took an opposite approach from the 'buggy knowledge detection" strategy. Rather than identify why watchstander trainees made mistakes, AETS would try to determine how to get them to do the right (i.e., situationally relevant) thing. This allowed the observation, assessment, and diagnosis processes to:

- focus on identifying when situationally relevant actions were not taken (or not taken correctly),
- focus on immediate feedback to get them to take the correct action (if key to the team's overall problem flow), and
- diagnose the cognitive process by identifying what knowledge the trainee needed to take the 'right' action, and determining which elements the trainee was exhibiting possession or mastery of.

This approach simplified the tutoring strategy in a task environment with tremendous complexity and indeterminacy and focused it on task performance at the level of the high level action units.

3 At least as reported by experienced human instructors.

AETS ARCHITECTURE

The AETS architecture, was defined and implemented based on the design principles and goals discussed above. Figure 1 depicts the overall functional architecture of the AETS. The center row of the figure highlights four generic components that work in parallel. Three fully automated components supplement and support a fourth interactive component (far right) that allows a human instructor to record and organize observations and to initiate feedback to one or more trainees. The instructor receives information from the automated components, makes notes on his direct observations, and communicates with trainees, all using a hand-held device called ShipMATE (Shipboard Mobile Aid to Training and Evaluation, discussed in more detail later).

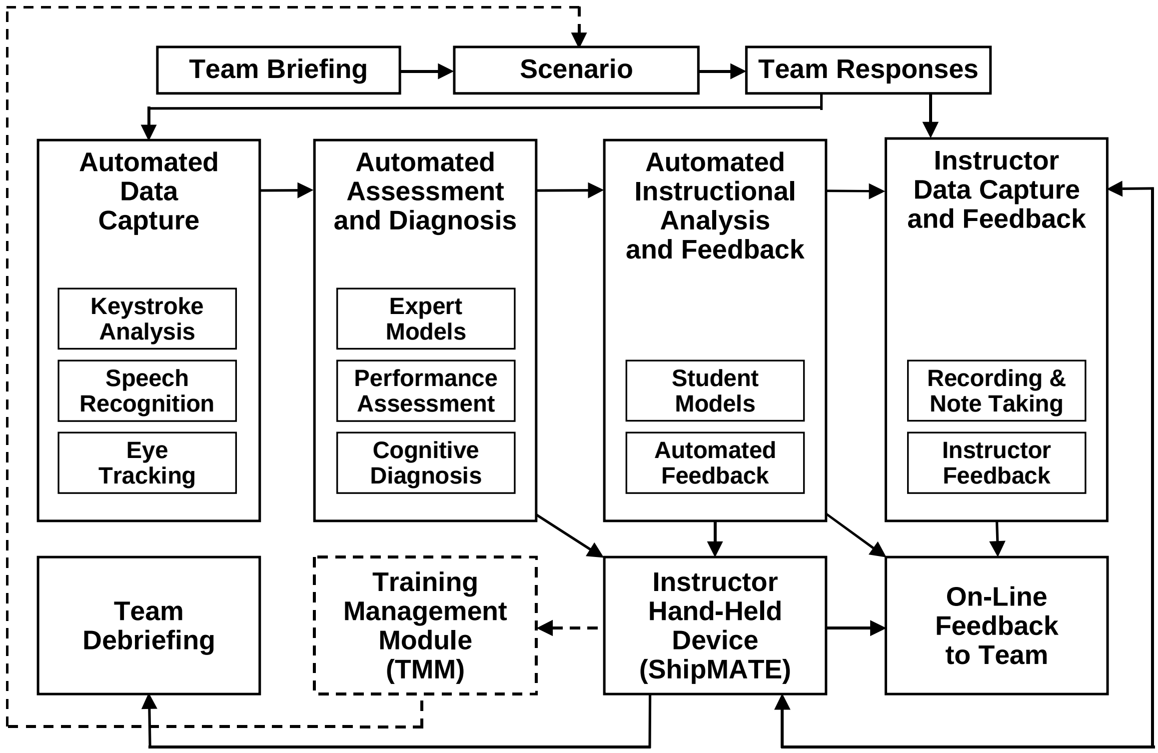

Figure 1. AETS functional architecture.

The primary interactions with the team being trained are shown at the four corners of Figure 1. A training session begins with a pre-problem briefing on the up-coming training problem called a scenario. After the briefing, the existing ACTS simulation capability of the Aegis-class ship plays out the scenario on the team's workstation consoles, where the team reacts to and works through the scenario just as they would a real operational mission. The team's responses are observed both by the automated system and by the instructor. During the scenario, real-time automated and instructor-initiated feedback is sent to the trainees. Shortly after the end of the scenario, the instructor uses ShipMATE to present a post-problem debriefing, using automatically generated reports and his own computer-assisted performance assessment.

The automated data capture component observes, analyzes, and abstracts the actions of each trainee in multiple modalities of human-system and human-human interaction. It captures all keystroke sequences and aggregates them automatically into higher level units that represent the operator's functional interactions with the workstation. In parallel, the system records and processes all trainee speech communications with other team members, recognizing and analyzing them into semantic components that define the source, destination, message type, and the objects and relationships mentioned. Also, in parallel, each trainee's eyes are tracked, and the dynamic eye-movement data are analyzed to assess what the trainee viewed, when, and for how long. The output of the automated data capture component is three streams of observed high level actions (HLAs) for each trainee. The combination of keystroke, speech, and eye HLAs provides a coherent record of what the trainee is doing during the simulated problem.

The automated assessment and diagnosis component dynamically compares this picture of what the trainee is doing with a model-based specification of what the trainee should be doing. An executable cognitive model of each trainee position observes and analyzes the information provided to the trainee and identifies HLAs that an experienced operator would be expected to take at that point in the problem[4]. The cognitive models also identify the knowledge and skill elements needed to generate the expected behavior plus any training objectives associated with the expected HLAs.Performance assessment and cognitive diagnosis are two separate stages of automated analysis of the trainee's behavior. Performance assessment compares observed HLAs of the trainee to expected HLAs generated by the model to determine if the recommended behavior occurred. Results of the performance comparison (both correct actions and deviations) are summarized by the calculation of overall numeric and qualitative scores, which are used to evaluate the trainee's performance relative to the training objectives. This comparison of expected to observed HLAs constitutes a behavioral diagnosis of the actions of the individual trainees and of the team as a whole. In AETS this behavioral diagnosis is called the performance assessment process. The second stage, cognitive diagnosis, relates performance assessment results to knowledge and skill elements identified by the cognitive model and makes inferences to determine what knowledge and skills are (and are not) being demonstrated in the observed behavior. The overall flow of processing is shown in Figure 2.

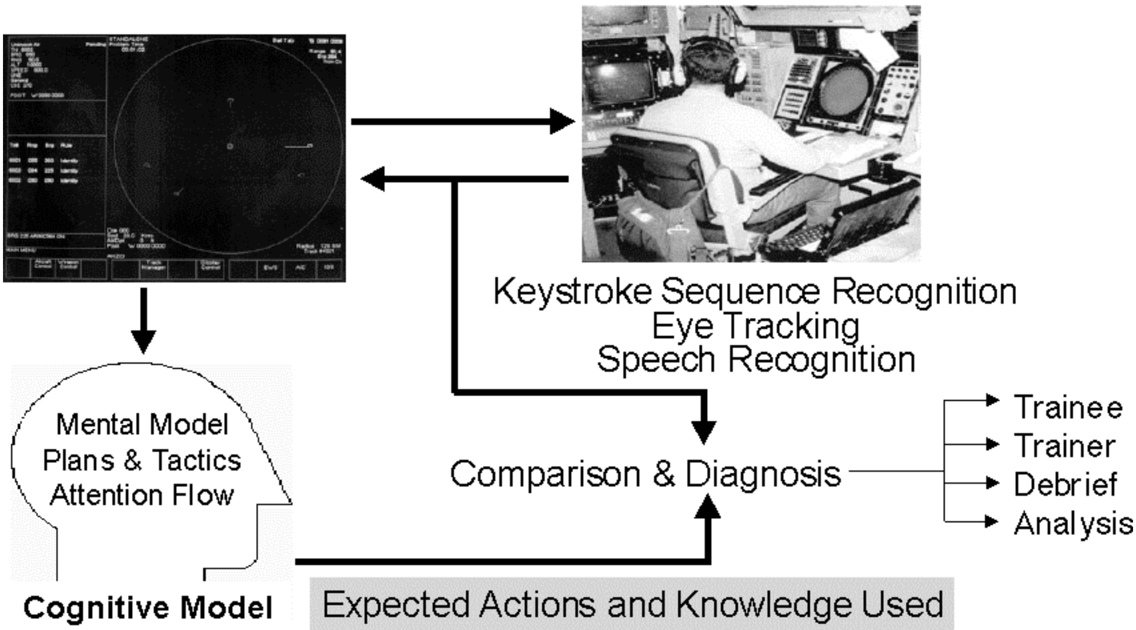

Figure 2. Flow of processing in AETS.

The automated instructional analysis and feedback component has two interrelated functions. First, it maintains a dynamic student model for each trainee that constrains and guides the instruction and feedback process. The student model records inferences about the trainee's mastery of training objectives, as evidenced by performance assessment results. The second function is the generation of real-time automated instructional feedback to the trainees. Feedback is triggered by automated performance assessment results and involves first selecting the instructional content and then selecting a feedback template. Instructional content selection

4 As discussed below, the generation of (desired) high level actions often requires cognitive modeling and simulation down to a much finer grain of detail than that abstracted level of High Level Actions.

depends on the qualitative result of a current performance comparison, how recently the trainee received feedback on the associated training objective, and the priority of the training objective. The structure and modality of the feedback depend on a feedback template, which is selected according to the training objective involved and its mastery by the trainee. Templates vary with respect to high or low information content and directive or reflective instructional approach. Feedback, which is provided through a limited display window and/or highlighting of console display elements, is judiciously limited to avoid intrusion onto task performance. This component also provides the instructor with summaries of trainee mastery of important training objectives for use in the post-problem debriefing.

Instructor data capture and feedback is an interactive software component supporting the instructor and running on a lightweight, pen-centered, hand-held computer called ShipMATE. This system's infrared and radio-frequency communications capability allows the instructor to move freely about the ship, untethered to any physical equipment. ShipMATE serves two main functions. First, it provides a medium for the automated components to communicate performance, diagnosis, and instructional information to the instructor, who can decide how (and if) to use it. Second, it allows the instructor to make records and notes concerning trainee performance and to construct and present real-time feedback and post-problem debriefings. Three ShipMATE features facilitate the instructor's data capture during the scenario. Digital ink allows the instructor to write notes with a digital pen and to recall them for later use. Voice annotation allows the instructor to record verbal notes through a microphone and to replay them later. Finally, team communications capture allows the instructor to record for later use/analysis any trainee voice communications over local voice networks. ShipMATE facilitates instructor-generated feedback to trainees through a database of feedback templates that the instructor can call up and transmit. For post-problem debriefing, ShipMATE constructs a variety of reports from automated performance, diagnosis, and instructional information and allows the instructor to select, organize, and present or replay captured information.

The following simple example, representing just one brief thread of activity within AETS, demonstrates how the automated components work together. The processing sequence is charted in Figure 3. A tactically significant event occurs at 2 minutes, 30 seconds into a training scenario when shipboard sensors detect a new track, corresponding in scenario 'ground truth' to an unknown F-4, at a distance of 50 nautical miles from own-ship. It is the responsibility of the Anti Air Warfare Coordinator (AAWC) watchstander to inspect new tracks and attempt to identify them. An expert watchstander should make initial responses within 30 seconds of the time the track symbol appears on the AAWC display screen.

Scenario Time (min:sec)	Critical Event Description	Allowed Response Time	AAWC Expert Model Expected Actions (with Training Objectives)	AAWC Trainee Observed Actions
02:30	Track 1125 (Unknown F-4, Touch and Go) Initializes at Bearing = 72°, Range = 50 NM	30 sec	02:36 - Hook Track 1125 for Evaluation 02:37 - View Track Kinematic & ID Data 02:39 - Change Track 1125 ID from "Pending" to "Suspect" 02:43 - Hook Potential Threat Track 1125 02:44 - View Track Kinematic & ID Data	02:38 - Hook Track 1125 02:40 - View Track Data 02:50 - Hook Track 1125 02:52 - View Track Data

AAWC Automated Performance Assessment	AAWC Automated On-Line Feedback
02:38 - COMPLETED Hook Track 1125 02:40 - COMPLETED View Track Data 02:50 - COMPLETED Hook Track 1125 02:52 - COMPLETED View Track Data 03:00 - OMITTED ID Change from "Pending" to "Suspect"	03:10 - Corrective / Directive Feedback to AAWC "Change ID Track 1125"

Figure 3. Example of automated instruction for the AAWC operator.

Reflecting this, the cognitive model for the AAWC identifies a series of expected high level actions (HLAs) as situationally relevant to the new track, in the cognitive processing that follows the simulated visual detection of the new track symbol by the model. The model also specifies that within a 30-second time window the AAWC watchstander should (1) hook the new track, (2) view its current kinematic and identification data, (3) change its identification from "pending" to "suspect", (4) once again hook the track, and (5) review its updated kinematic and identification data. The cognitive model also associates the expected HLAs with specific training objectives to give the instructional component their situational context. (The reader should note that any given HLA can and does occur in different contexts. For example, the Hook Track HLA occurs twice in this sequence -- the first hook is for evaluation of a new track, while the second is to maintain awareness of the behavior of a potential threat track.) The expected HLAs and associated training objectives are then sent to the performance comparison process.

Meanwhile, the automated data capture component is simultaneously observing the actions that the AAWC watchstander is performing and abstracting them into observed HLAs. In the example, this component reports that the trainee was observed to hook the new track twice, to view the track data both times, but not to change the track's identification. The observed HLAs are also sent to the performance comparison process. The automated performance assessment engine dynamically compares the expected HLAs with the observed HLAs and reports an action evaluation for each expected HLA. When an observed HLA matches an expected HLA, it reports a completed action. However, when no observed HLA matches an expected HLA by the end of the time window, it reports an omitted action. In the example, the identification change is reported as omitted when the 30-second response time expires.

The automated instructional analysis component reads the set of action evaluations and determines that it is appropriate to send the AAWC trainee a feedback message concerning the omitted HLA. Based on the trainee's mastery of the associated training objective (as recorded in the student model of the instructional component), the instructional system chooses a directive-type feedback template to instruct the trainee to make the identification change. Assuming that the AAWC trainee does now take the action, the expected flow of activity within the Air Defense Team can be maintained during the embedded training simulation.

Within the above architecture, there were four key components that were particularly critical to the implementation of the system. These four components were the:

- cognitive models;
- performance assessment subsystem;
- cognitive diagnosis; and
- instructional support subsystem.

These are each discussed in additional detail below, along with a consideration of the problems and issues that were involved in creating each of these components.

COGNITIVE MODELING

The AETS architecture depends on embedded cognitive models of each watchstander role addressed by the system. These cognitive models had to be able to generate, dynamically and in real time, expert-level predictions of the content and timing of each type of High Level Action defined for that watchstander role. In addition to this architectural requirement, there proved to be other constraints on the cognitive models as well. One key constraint was that the models had to be able to adapt their expert level knowledge to the often less-than-expert actions of the trainees. As shown in Figure 2 earlier, each model needed to interpret and analyze the changing problem situation, and at each appropriate moment recommend the proper HLA that should be taken. If the cognitive models were empowered to actually implement those actions (as they were in model development and testing), then they could effect a full expert-level execution of that watchstander role. However, during training scenarios, actions were taken only by the trainee. This meant that, in practice, the trainee was free to do something totally different (including nothing at all), and often did so. When this happened, the cognitive model had to be able to adapt to what was then an unexpected and suboptimal situation, and recommend, in essence, the appropriate remedial HLA given that the trainee did not take the appropriate action. This constraint meant that the cognitive models had to be much more complex than a pure expert-level model -- they had to be robust and adaptive to unexpected behaviors of the trainee.

A second constraint somewhat related to the first was the need for the same models to be able to generate both the low-level keystroke actions needed to operate the watchstation and the abstracted high level actions that were the basis for trainee observation, assessment, diagnosis, and instruction. The reasons for this are illustrative of how theoretical distinctions can get muddled in practical applications. In principle, the models should have been able to operate only at the HLA level, observing the overall situation and determining what were the appropriate HLAs within that situation (whether or not the trainee-watchstander's actions produced a desired situation or a sub-optimal one). The HLAs and the situation assessment were focused on the level of tactical knowledge of the problem domain. In theory, this task knowledge should be divorced from the lower-level tool-knowledge needed to manipulate the actual watchstation human-computer interface to gain information about the tactical situation, and thus, the cognitive models should have been cleanly partitioned between this level of task knowledge and tool knowledge (and should not have required the latter). Unfortunately, this theoretically clean distinction did not apply in practice, because the user interface to the watchstation simply did not support it. The design of the interface was such that almost constant interactions with the watchstation were necessary to extract related aspects of situational information. This meant that both the human watchstander (and the cognitive model as well) could not simply interpret the situation without constantly utilizing a great amount of detailed knowledge (and actions) about how to extract information through the human computer interface. Stated differently, tool knowledge about AEGIS was inextricably intertwined with the task knowledge about how to solve Air Defense problems. Thus, even though the models needed only to generate HLAs based on their situational relevance, the cognitive models had to contain (and constantly exercise) a virtually exhaustive knowledge about how to extract specific elements of situational information from the Aegis system in order to develop and maintain the situational knowledge needed to generate these predicted HLAs. This unfortunate fact dramatically increased the required complexity of the models.

A third constraint was that the models had to be able to trace out the knowledge that needed to be used between one HLA and another, to support the model-tracing approach to cognitive diagnosis (detailed in a subsequent section). The use of the HLA level as the basis for diagnosis, meant that there were long underlying sequences of internal processing and knowledge states in-between one HLA and the next. The main problem in creating this knowledge trace was computational. Each perceptual stimulus received by the model could lead to an unknown number of possible future behaviors; the cognitive architecture had to somehow build and maintain each of these possible threads in a manageable real-time manner, and in a way that provided a valid trace of the cognitive processes involved. To solve this problem, multiple possible means of recording and tracing the reasoning threads were developed, and assessed both for theoretical reasonableness (in human information processing terms) and computational efficiency.

Creating models that met all these constraints, the first two of which did not become completely clear until development was well underway, proved in itself a major challenge. These embedded cognitive models were expressed in the COGNET framework (Zachary, Ryder, and Hicinbothom, 1998; Zachary, Ryder, Ross, Weiland, 1992) for cognitive modeling. This framework was selected for several reasons. First, and foremost, the framework was highly appropriate to the need. The COGNET technique had been available for some time, and had been proven highly useful in analyzing domains such as Air Traffic Control (Seamster, Redding, Cannon, Ryder and Purcell, 1993) and telephone operations (Ryder, Weiland, Szczepkowski, and Zachary, 1998). COGNET was designed specifically for modeling and analysis of real-time, multi-tasking domains such as Naval command and control. Second, and also helpful, was the fact that there were existing partial models in COGNET format that could be used as an advanced starting point for AETS modeling (Zachary, Ryder, and Hicinbothom, 1998). Third, a software tool was available to render COGNET representations executable and to allow them to interact with or be embedded into simulated or real worlds (Zachary, Le Mentec, and Ryder, 1996). This tool[5] allowed the cognitive modeling results to be directly compiled into executable code for use in AETS.

Figure 4 shows the high level cognitive architecture of the COGNET model of a given watchstander (there are currently three of these in the AETS). The architecture, discussed in more detail in Zachary, Le Mentec, and Ryder, (1996), and Zachary, Ryder, Hicinbothom, (1998), is organized into three parallel processes -- sensation/perception, cognition, and motor action -- and is based on the seminal Model Human Architecture, first proposed in (Card, Moran, and Newell, 1983). The perceptual process, which internalizes sensed information, uses a set of spontaneous-computation (i.e., self-activating) knowledge sources called perceptual demons. The perceptual demons contain the expert's knowledge about how to internalize sensed information and relate it to the evolving mental model of the situation. These demons activate themselves and operate spontaneously as relevant auditory and/or visual cues are sensed in the outside environment.

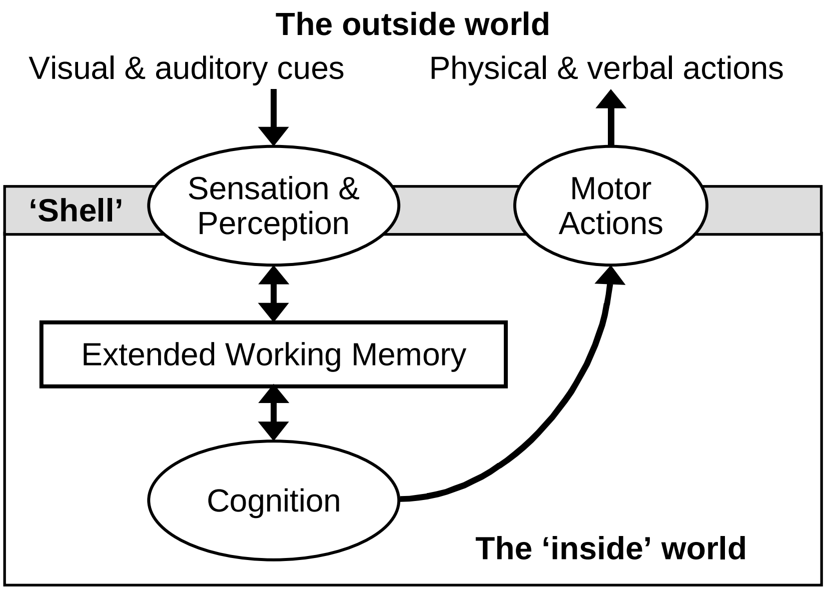

Figure 4. Processing structure in an executable COGNET model.

In parallel to the perceptual process, a cognitive process activates and executes complex chunks of procedural knowledge (called cognitive tasks in COGNET). These cognitive tasks are activated on the basis of an internal representation (maintained in extended working memory) of the problem. This problem knowledge may include knowledge of the external (tactical) situation, the work team and its situation, constraints and requirements on tactical actions (called Rules of Engagement), etc., and is expressed in declarative terms. In COGNET, the declarative problem knowledge is represented as a multi-panel blackboard structure, and is sometime referred to as the mental model of the problem. Each panel is subdivided into levels, or categories, on which hypotheses (representing individual knowledge elements) are dynamically posted and unposted (as the situation changes). The cognitive tasks are activated in a context sensitive manner, based on the current pattern of facts and knowledge elements on

5 This is now commercially available under the name iGEN™ from Agenix Corp. The AETS models were built using a customized version of this tool.

the blackboard. The cognitive tasks are represented as GOMS-like (Card, Moran, and Newell, 1983) goal hierarchies, but with a formally defined operator set that includes generic cognitive operators to manipulate information in the mental model and domain-specific action operators. Often, the cognitive tasks must compete for the person's limited attention, according to a built-in attention model, and may interrupt one another as the situational dynamics dictate. As a cognitive task executes, the simulated watchstander develops and executes plans for solving the problem and/or pieces of the problem. The cognitive tasks contain low-level cognitive operations that can modify the mental model, as well as other operators that activate one or more motor-action processes that result in physical or verbal actions being taken in the outside environment.

For all the reasons discussed above, the AETS models were relatively complex. For example, one of these, the model of the Anti-Air Warfare Coordinator or AAWC who coordinates the activity of the Air Defense team, includes:

- 25 cognitive tasks, with more than 500 lower level goals and subgoals, and several thousand instances of individual cognitive or action operators;
- 15 blackboard panels, with more than 90 individual levels and more than 500 hypotheses active on this blackboard at a typical moment;
- more than 100 perceptual demons, and
- more than 300 domain-specific action types.

The models are discussed in more detail in Zachary, Ryder, Hicinbothom, Bracken, (1997) and Zachary, Ryder, Hicinbothom, Santarelli, Scolaro, and Szczepkowski, (1998).

PERFORMANCE ASSESSMENT

Automated performance assessment plays a central role in the Advanced Embedded Training System (AETS). As shown in Figure 5, the AETS conducts automated performance assessment both at the (high level) action level and at the (scenario) event level. At the action level, it compares automatically captured observed HLAs with model generated expected HLAs and outputs the results as action evaluations. At the event level, it aggregates the observed and expected HLAs taken in response to each critical scenario event and calculates composite response scores. As the figure indicates, the action evaluations and composite response scores support the automated diagnosis of knowledge and skill deficiencies, the automated generation of on-line feedback, and the automated preparation of reports for post-problem debriefings. This section describes some of the challenges encountered in automated performance assessment and its integration with these other processes and summarizes the solutions conceived for the AETS.

The HLAs and scenario events that underlie the processes shown in Figure 5 constitute a common domain model of the operational environment. While this domain model serves to coordinate the processes, it must also take into account their particular requirements and limitations. HLAs have to be observable by automated data capture technologies as well as be predictable by the cognitive models. For action evaluations to provide meaningful instructional opportunities for on-line feedback, they require some representation of the situational context. As discussed previously above, the models supply this by associating context-specific training objectives with the expected HLAs they generate. Critical scenario events must be instructionally significant for potential debriefing reports yet sufficiently bounded in scope and response time for identifying specific HLAs as situationally relevant responses to them. Meeting these constraints required a coordinated design of the AETS data capture, cognitive modeling, performance assessment, diagnosis, and instructional processes along with the action and event structures they share.

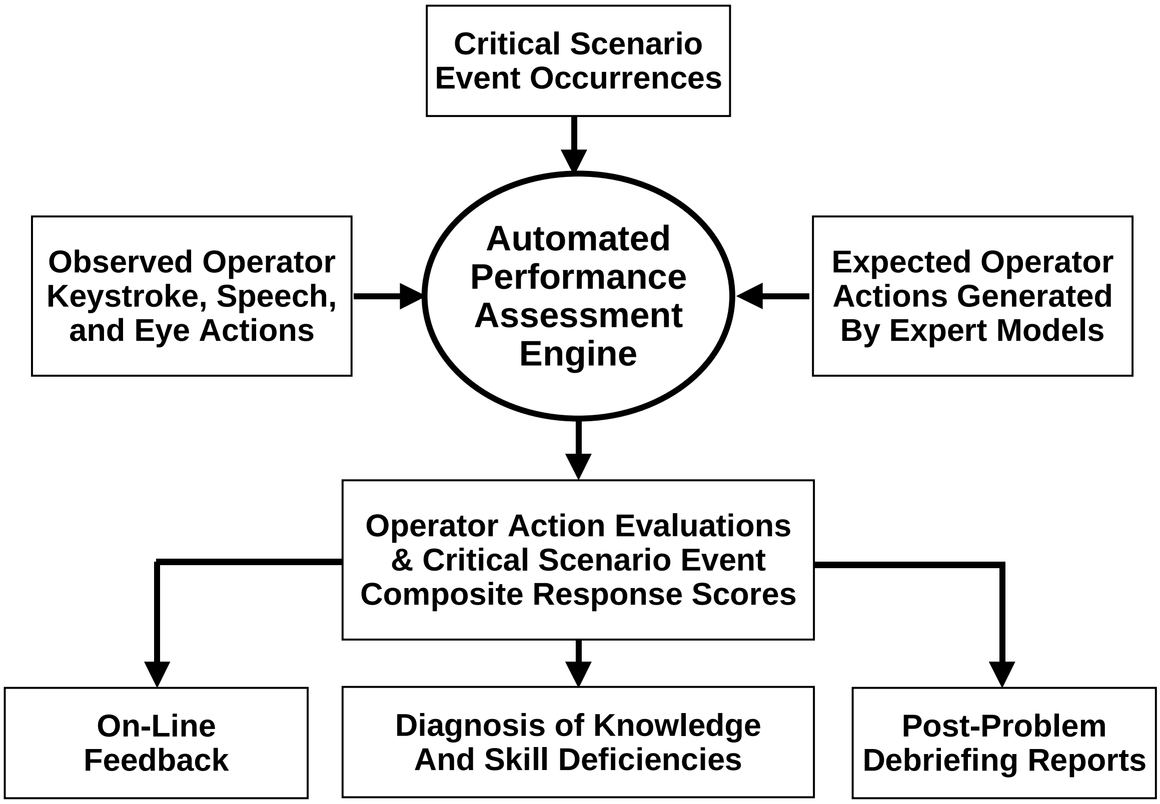

Figure 5. Overview of the role of the AETS Automated Performance Assessment Engine.

It was concluded early in the AETS design process that a meaningful and efficient automated performance assessment process depended on a common taxonomy of observed and expected actions. The granularity of these actions could not be too fine, or they would lack independent significance for analysis and require inordinate data fusion. Nor could they be too aggregated, or data capture would risk missing important actions. As discussed earlier, the AETS solution was to define, for each modality of action (keystrokes, speech, and eye dwells), a list of high level action types, along with specific attributes that might vary from one instance of the HLA to another. Keystroke HLAs defined complete keystroke sequences that accomplished a tactically meaningful function, such as entering the identity of a previously unidentified aircraft track. Similarly, speech HLAs corresponded to entire messages communicated by trainees, and eye HLAs represented dwells on certain parts of the trainee's console display. A total of 179 HLA types were defined in AETS. Of these, 11 were eye-dwell actions, 50 were keystroke/workstation-based actions, and the remainder were speech actions. Each HLA type defined a basic action, while variable attributes defined specific information-content items entered, spoken, or seen.

There were a variety of challenges in recognizing and evaluating HLAs. While keystrokes could be definitively captured, there were still many different keystroke sequences that constituted some HLAs, and all had to be unambiguously recognized. More critical was the fact that current technology could not support foolproof speech recognition or eye tracking. Consequently, a confidence measure was associated with observed speech HLAs so that low confidence recognition could be discounted in the performance evaluation process. Similarly, some eye HLAs were defined to relate to small regions of the console rather than to the precise data displayed there. Evaluation difficulties arose from the multiplicity of ways in which HLA attributes could be set (or left unset). Matches between observed and expected HLAs were therefore scored based on a classification of attributes into key and non-key categories. An overall positive or negative action evaluation depended on whether or not all key attributes of an observed HLA matched those of the expected HLA. Non-key attributes only affected a finer gradation of scoring.

The AETS employed the concept of an Event-Based Approach to Training (EBAT) (see Johnston, Cannon-Bowers, and Jentsch, 1995) by collecting and evaluating sets of expected HLAs related to critical scenario events. For complex scenarios in which many different events simultaneously demand the trainee's attention, a central difficulty for manual EBAT is determining which actions relate to which events. The AETS solution was to select from the scenario a number of critical events (either because of tactical importance or because of training significance) and to specify for each one a time window for responses and a response identifier (or condition). When the automated performance assessment process was notified of the occurrence of a critical scenario event for a watchstander, it would collect all the expected HLAs generated by the cognitive model that fall within the time window for that event and match the event's response identifier (typically a track number). When the time window expired, the performance assessment then calculated a composite response score for the critical event by combining all the action evaluations from the collected set of expected HLAs. The instructional analysis component used composite response scores to relate critical event responses to training objectives and to provide the instructor with examples for use in post-problem debriefing.

COGNITIVE DIAGNOSIS: RECOGNITION-ACTIVATED MODEL ASSESSMENT (RAMA)

AETS introduced the concept of the situationally relevant High Level Action to deal with the problem of indeterminacy in both the (detailed-level) action space and the (detailed-level) knowledge space. While the COGNET models generated expectations of the HLAs that each trainee needs to perform, many different sequences of lower-level actions could be undertaken by the trainee to achieve each of the specific HLAs. Thus, the cognitive diagnosis algorithm had to operate at the level of knowledge that was needed to produce the HLA (typically task knowledge) but not necessarily the lower-level knowledge needed to produce the lower-level actions from which an HLA was built. The RAMA (Recognition-Activated Model Assessment) was developed to address this need. RAMA is based on a model tracing approach. However, rather than comparing the model trace to every trainee action, RAMA does this only at the HLA level, when the behavioral assessment component recognizes that the trainee either has or has not taken some recommended high level action. This recognition activates a process of assessing all the knowledge that could have been used to generate the HLA (of current focus) from the likely knowledge state of the last recognized HLA. RAMA works on the knowledge traces produced by COGNET, as described above, and operates as follows:

from any given point in the problem-solving process where an expected HLA has been recognized to have been taken (or not taken) by a trainee:

- Process the cognitive model forward to (the next expected) HLA, while collecting all the knowledge elements used directly or indirectly to generate that next expected HLA;
- Wait for the performance assessment system to observe the HLA or conclude that trainee did not perform it;
- If HLA was taken correctly, then update the system's belief that trainee understands and has used correctly the intermediate knowledge states in the current knowledge trace;
- If HLA was not taken correctly, then update the system's belief that one or more of the intermediate knowledge states in the current knowledge trace was not understood or used correctly by the trainee.

Through this approach, RAMA is able to build, over time, a coherent model of the trainee's state of acquisition of each element of knowledge in the COGNET model.[6]

---

6 The system is currently utilizing two simultaneous approaches to perform the actual diagnosis, which occurs in the last two steps. One is a bayesian propagation technique (see Martin and Van Lehn, 1995;

FEEDBACK, INSTRUCTION, AND TEAM TRAINING SUPPORT

AETS mixes both individual instructional feedback and team training, and uses both automated and human delivery methods. Automated individual feedback is delivered at the trainee's workstation using objective-based templates that are dynamically tailored to the specific context. Other individual feedback is provided by human instructor(s) with the help of ShipMATE device(s). ShipMATE is a prototype software application that runs on a lightweight (approximately 3.5 lb.), off-the-shelf, pen-based, hand-held computer. Importantly, this lightweight and portable system supports wireless data communications to the rest of the AETS. Thus, an instructor is free to move about the Combat Information Center (CIC) when the team works, while maintaining a continuous link with the automated performance assessment and diagnosis system and with other shipboard instructors (if any).

Currently, ShipMATE supports the trainer in preparing the team for the problem (prebrief support), as a data collection device to be used during the problem simulation, and as a tool for preparing and conducting post-problem debriefs. The following capabilities enable these functions:

- Dynamic link to automated AETS performance assessment system
- Entry of digital ink notes
- Entry of voice annotations
- Capture of 'live' team communications
- Access to scenario support materials (e.g., maps, charts, etc.)

ShipMATE allows an instructor to move about, observe, and collect data on multiple trainees at the same time. During the simulated problem, instructors use ShipMATEs to track individual and team performance, to provide dynamic feedback to training team members through the ShipMATE interface, and to record and explicate complex individual and team behaviors with specific performance measurement tools. ShipMATE also provides tools for the instructor to formulate and deliver training to the trainee, either verbally or through the trainee's workstation.

Following the problem simulation, instructors meet to prepare individual and team-level debriefs. This process is facilitated by ShipMATE, as it allows for rapid review and reduction of instructor-based and automated measurement system data. ShipMATE is also designed to support presentation of debrief materials both for individuals and for the team. Finally, ShipMATE can be used to propagate performance measurement data collected and reduced by instructors forward in the training cycle and to update performance records in anticipation of subsequent training evolutions.

ShipMATE serves as the linchpin to AETS. It provides the instructor with access to the automated performance assessment and diagnosis elements, while also providing the instructor with performance measurement tools to record, assess, and diagnose complex performance breakdowns that require human explication. While a defining objective of the AETS system is to make team training more efficient by increasing automation, thereby reducing personnel requirements, eliminating instructors from the training cycle altogether was not. Rather, the AETS vision seeks to take advantage of the expertise of shipboard instructors by providing

---

Mislevy, 1995), in which bayesian inference is used to update the likelihood that the trainee possesses the corresponding knowledge elements used by the model. The other is a deterministic analog developed by Anderson and colleagues (see Corbett, Anderson, and O'Brien, 1995). In this second approach, the belief that the trainee possesses a specific knowledge element is incremented deterministically when an expected behavior is observed, and negative credit is propagated only as long as it does not contradict a positive credit. The bayesian update is only performed once at the end of the execution. The Anderson technique could be applied iteratively but is not, to maintain a compatibility with the bayesian approach. Ultimately, only one approach will be used, based on performance during the summative evaluation of the system.

them with a "window" into the automated performance assessment and diagnosis system. ShipMATE is the interface that provides this window.

ShipMATE supports the structured enhancement of Norman's concepts of experiential and reflective thinking (Norman, 1993). Norman contends that experiential thinking evolves from our experiences in the world and it is reflexive and automatic in nature. Reflective thinking, on the other hand, requires more careful deliberation. Experiential thinking about dynamic dependencies, between equipment, task, and system models requires on-line data capture, data reduction, and data fusion that provide the basis for feedback designed to keep the operator(s) "in the game" so that learning can take place. Reflective thinking about the strategic dynamics of CIC team performance requires technologies that support and facilitate team level diagnosis. In fact, we believe that this diagnosis is most effective when the team does it itself. Advanced technologies are certainly not sufficient to ensure that this type of thinking (learning) takes place, but as Jonassen and Reeves (1996) suggest, technologies can support reflective thinking when they enable users to compose new knowledge by adding new representations, modifying old ones and comparing the two. As both an on-line aid and as a post-problem aid, ShipMATE supports both types of thinking (learning) on the part of CIC operators.

While on-line feedback can be provided to the trainee automatically, or by an instructor using the ShipMATE device, the on-line feedback provided within AETS was not intended to engage trainees in deep analysis of their performance. The design of the AETS recognizes the wisdom of traditional approaches to training in which this type of critical examination of team performance is conducted in a post-problem setting. However, this type of instruction is susceptible to the pitfalls of unaided recall. ShipMATE was expressly developed to aid instructors in:

- collecting data during the simulation (thereby offsetting potential problems associated with subjective recall of performance), and
- organizing this performance data in a specific pedagogical framework for team training called Team Dimensional Training or TDT (Smith-Jentsch, Zeisig, Acton, and McPherson, 1998).

The TDT framework focuses on critical <u>team</u> skills (specifically, communication, information exchange, leadership, and supporting behavior). By using ShipMATE to capture and organize the data needed to deliver TDT, instructors are able to guide teams in reflective thinking. With the ShipMATE device as the window to the AETS, instructors are provided a richer and more representative picture of performance as it unfolds. Stated differently, ShipMATE supports a more objective and representative data collection process which, in turn, allows for a more well informed post-problem discussion of team performance issues.

IMPLEMENTATION STATUS AND FUTURE DEVELOPMENT

AETS is being implemented and evaluated as a three phase project. Phase one, completed in 1997, developed the system's communication and computing infrastructure and integrated an initial system prototype for only two watchstanders, using a mix of real and placeholder components. The automated data capture component and the instructor data capture component were fully functional in this initial prototype, while the automated assessment and diagnosis component was only partially implemented, with no active cognitive diagnosis. The automated instructional analysis and feedback function was represented only by a placeholder component.

Phase two expanded the AETS to a laboratory prototype system with fully functional versions of all components of the architecture and a team of four watchstanders. Figure 6 is a photograph of the laboratory prototype system as it was integrated at Lockheed Martin Advanced Technology Laboratories in the summer of 1998. In the photograph one can see a team of four trainees at simulated watchstation consoles as well as a pair of instructors each using a hand-held device (ShipMATE). A series of team-in-the-loop sessions involving four training scenarios was conducted on the laboratory prototype system in August, 1998, and demonstrated the functionality and training value of the system. Though this was only a formative evaluation, Navy instructors and operators were enthusiastic about the potential utility of the AETS for shipboard training. The primary research and development of AETS was completed with Phase two.

Phase three will upgrade and expand all components of the phase two system, but its main purpose is to establish and demonstrate the system's ability to communicate with operational equipment on-board Aegis-based ships. In addition, the system will be installed at a land-based training facility[7], where it will be further evaluated and used to support various training experiments.

Figure 6. Laboratory prototype of the AETS at Lockheed Martin Advanced Technology Laboratories.

The current AETS focuses on the individual training simulation as the unit of training. That is, it does not focus on the broader curricular issues of developing and sequencing training scenarios over time so as to meet specific training objectives or to support training needs of specific ship crews. However, these issues are being addressed in a related research effort to create a Training Management Module, or TMM. The TMM will automate the process of scenario preparation and ensure that training objectives will be linked to performance measurement tools. It also ensures that training is iterative in nature, with each subsequent training evolution utilizing performance histories from previous sessions as input.

Focusing on the scenario as the unit of training for this system enables instructors to tailor scenarios to individual and team performance strengths and weaknesses. This ensures the efficient use of training resources and supports the overall goal of enhancing readiness. Here again, while technology provides the mechanism for the implementation of objective-based training using a software-based support module, it is the objective-based framework with its focus on performance measurement that creates the need for the TMM. In the near future, the TMM will be integrated with the AETS technology to allow shipboard instructors to do the following:

- Create training sessions customized to the needs of that ship's crew;
- Initialize AETS based on mission, past performance, scenario event selection, and command-generated standards;
- Create a mechanism to integrate ship-level training via AETS into higher-level (e.g., multi-ship) joint training;
- Decrease the time and effort required to develop and manage scenario-based training using AETS; and

7 The Aegis Training and Readiness Center (ATRC) in Dahlgren, Virginia.

- Transition the AETS usage guidelines, training processes, and lessons learned into operational usage.

CONCLUSIONS

The AETS project represents a major effort to apply intelligent tutoring system (ITS) concepts to complex, real-time problems. Although it is too early to assess the operational value of the system, there are already several clear implications and lessons from this large and ambitious project:

Hybrid approaches work. There is no 'pure' ITS architecture that can or should be directly applied to create an ITS in a complex, real-world environment such as Aegis Air Defense team training. Rather, the details of the system and architecture must be fit to the constraints and opportunities of the application environment. An eclectic set of techniques were chosen, each because it was the best approach to meeting the system's specific requirements for speech recognition, cognitive modeling, instructional management, etc. A flexible architecture and state-of-the-art system integration methods proved more than sufficient to integrate these diverse pieces into a hybrid whole.

Use architectural redundancy. The fluid nature of this task domain, and the difficulties of observation in the domain dictated that AETS could not rely on only one way to get things done. Multiple data paths were built in from the start, because AETS could not rely on any one source for all its data, or for fully reliable data all the time. These multiple paths (e.g., collecting data separately from eyes, hands, and voice) allowed the data sources to be used alone if necessary, and to add value to one another when all were available. Similarly, the system was designed to provide simultaneous cognitive and behavioral diagnosis, but to work with only behavioral diagnosis if cognitive analysis could not be performed at any time. These are necessary concessions to the complexity and uncertainty in real-time and real-world domains. By anticipating them in the architecture, it is possible to make a virtue of necessity.

Embedded training is an ideal home for ITS. An increasing number of complex real-time systems are being built, like Aegis, with embedded simulation capability, and such a capability can provide an ideal platform for building an ITS. It allows the ITS to train in the actual work environment, so fidelity ceases to be an issue. It also eliminates the need to create/simulate workstations, interfaces, and/or underlying systems, often the major costs of ITS development. It focuses attention, however, on effective system integration practices, and on designing to the software interfaces provided by the host environment. Although the designer of an embedded ITS (like AETS) has ultimately little control of the training environment and must compromise constantly, the payoff of working in an embedded training context is clear -- virtually instant fieldability.

AETS concepts and technologies are enablers for ITSs in new training domains. The systems and software architectures described here are applicable to a variety of other complex, team-based, real-time training environments. The automated performance assessment, behavioral and cognitive diagnosis, and instructional management algorithms, which comprise a large portion of the AETS, are leveragable in new domains, both separately and in the context of the AETS architecture.

Embedded systems like AETS can ultimately achieve a goal of providing a continuous learning environment, one in which the ITS is integrated with embedded problem simulation and actual work experience. This can create a seamless web of training, practice, and performance, all using a common workstation and a common set of performance standards and measures. By eliminating the need for off-site training facilities and by reducing the number of instructors and workstations needed, an embedded ITS can achieve the goal of the user community, which is not just better training, but better training that is also faster and more cost efficient than current methods.

Acknowledgements

The views expressed are those of the authors and do not reflect the official positions of the organizations with which they are affiliated or of the sponsors. This work was performed under contracts N00014-96-C-0051, N61339-97-C-0405, and N61339-96-C-0055. The authors wish to acknowledge the support and contributions of RADM George Huchting and Susan Chipman, and the many members of the project team, including Kevin Bracken, Jim Hicinbothom, Joan Ryder, Tom Santarelli, Jackie Scolaro, Mike Szczepkowski, Tim Brown, Jean-Christophe Le Mentec, Bob Pesavento, and Jim Boardway at CHI Systems; Ron Koontz, Peter Rosenfeld, Tom Geigel, Mike Junod, and Carlos Henriquez at Lockheed Martin Advanced Technology Laboratories; Allen Hale and Paul Berger at Lockheed Martin Government Electronic Systems; Milt Stretton, George Banta, Jim McCarthy, Linwood Hudson, Kathy Homan, Rick Moore, and Gale Morganroth at Sonalysts; Eduardo Salas, Denise Lyons, Kim Jentsch, and Joan Johnston at the Naval Air Warfare Center, Training Systems Division; Kurt Van Lehn at the University of Pittsburgh; and Stellan Ohlsson at the University of Illinois, Chicago.

References

Anderson, J., Corbett, A., Koedinger, K., and Pelletier. R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4(2):167-207.

Card, S., Moran, T., and Newell, A. (1983). The Psychology of Human-Computer Interaction. etc.

Collins, A. (1996) Wither Technology and Schools: Collected Thoughts on the Last and Next Quarter Centuries. Paper read at *Third International Conference on ITS,* Montreal, Canada.

Collins, A., Brons, J.S., & Newman, S.E. (1989) Cognitive apprenticeship: teaching the craft of reading, writing, and mathematics. In Resnick, L.B. (Ed.), *Knowing, learning and instruction: Essays in honor of Robert Glaser*. Hillsdale, N.J.: Lawrence Erlbaum Assoc.

Corbett, A., Anderson J., and O'Brien, M. (1995) Student Modeling in the ACT Programming Tutor. In P. Nichols, S. Chipman, & R. Brennan, (Eds.). *Cognitively Diagnostic Assessment*. Hillsdale. NJ: Erlbaum.

Jonassen, D. H., & Reeves, T. C. (1996). Learning with technology: Using computers as cognitive tools. In D. H. Jonassen (Ed.), Handbook of Research on Educational Communications and Technology, (pp. 693-719), New York: Macmillian.

Johnston, J., Cannon-Bowers, J., & Jentsch, K. (1995). Event-based Performance Measurement System for Shipboard Command Teams. In *Proceedings of the First International Symposium on Command and Control Research and Technology*. Washington, DC: The Center for Advanced Command and Technology, pp. 274-276.

Lesgold, A., Lajoie, S., Bunzo, M., & Eggan, G. (1992) Sherlock: A coached practice environment for an electronics troubleshooting job. In Larkin, J. and Chabay, R. (Eds.). *Computer Assisted Instruction and Intelligent Tutoring Systems: Shard Goals and Complementary Approaches*, Hillsdale, N.J.: Lawrence Erlbaum Assoc.

Martin, J. & Van Lehn, K.. (1995) Student assessment using bayesian nets. *International Journal of Human Computer Studies,* 42:575-591.

Mislevy, R. (1995). Inference in Cognitive Diagnosis. In P. Nichols, S. Chipman, & R. Brennan, (Eds.). *Cognitively Diagnostic Assessment*. Hillsdale. NJ: Erlbaum.

Norman, D. A. (1993). Things that make us smart: Defending human attributes in the age of the machine. Reading, MA: Addison-Wesley.

Ryder, J.M., Weiland, M.Z., Szczepkowski, M.A., & Zachary, W.W. (1998). Cognitive engineering of a new telephone operator workstation using COGNET. *International Journal of Industrial Ergonomics 22, pp. 417-429* .

Seamster, T. L., Redding, R. E., Cannon, J. R., Ryder, J. M., & Purcell, J. A. (1993). Cognitive task analysis of expertise in air traffic control. *International Journal of Aviation Psychology, 3(4),* 257-283.

Smith-Jentsch, K. A., Zeisig, R., Acton, B., and McPherson, J. (1998). Team dimensional training. In J. Cannon-Bowers & E. Salas (Eds.), *Making Decisions Under Stress*, Wash., DC: APA.

Van Lehn, K. (1996). Conceptual and meta learning during coached problem solving. In C. Fasson, G. Gauthier, and A. Lesgold (Eds.) *Intelligent Tutoring Systems: Lecture Notes in Computer Science 1086*. N.Y.: Springer. pp. 29-47.

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

Zachary, W., Bilazarian, P., Burns, J., & Cannon-Bowers, J. (1997) Advanced Embedded Training Concepts for Shipboard Systems. In *Proceedings of 19th Annual Interservice/Industry Training Systems and Education Conference*. Orlando: National Training Systems Assoc. (published on CD-ROM)

Zachary, W., & Cannon-Bowers, J. (1997) Guided Practice -- a New Vision for Intelligent Embedded Training. *Proc. of Human Factors Society 41st Annual Meeting*. Santa Monica, CA: Human Factors Society, pp. 1111-1112.

Zachary, W., Cannon-Bowers, J., Burns, J., Bilazarian, P., & Krecker, D. (1998). An Advanced Embedded Training System (AETS) for Tactical Team Training. In B. Goettl, H. Halff, C. Redfield, and V. Shute (Eds.) *Intelligent Tutoring Systems: Lecture Notes in Computer Science 1452*. Berlin: Springer, pp. 544-553.

Zachary, W., Le Mentec, J-C., & Ryder, J. (1996). Interface agents in complex systems. In Ntuen, C. & Park, E.H. (Eds.), *Human interaction with complex systems: Conceptual Principles and Design Practice*. Norwell, MA: Kluwer Academic Publishers, pp. 35-52.

Zachary, W., Ryder, J., & Hicinbothom, J. (1998). Cognitive Task Analysis and Modeling of Decision Making in Complex Environments. In J. Cannon-Bowers & E. Salas (Eds.), *Making Decisions Under Stress*, Washington, DC: APA. pp. 315-344.

Zachary, W., Ryder, J., Hicinbothom, J., Bracken, K. (1997). The Use of Executable Cognitive Models in Simulation-based Intelligent Embedded Training. *Proceedings of Human Factors Society 41st Annual Meeting*. Santa Monica, CA: Human Factors Society, pp. 1118-1122.

Zachary, W, Ryder, J., Hicinbothom, H., Santarelli, S, Scolaro, S, & Szczepkowski, M. (1998) Simulating Behavior of Tactical Operators and Teams Using COGNET/GINA. *Proceedings of the Seventh Conference on Computer Generated Forces and Behavioral Representation*, Orlando, FL.

Zachary, W., Ryder, J., Ross, L., & Weiland, M. (1992) Intelligent Human-Computer Interaction in Real Time, Multi-tasking Process Control and Monitoring Systems. in M. Helander & M. Nagamachi (Eds.). *Human Factors in Design for Manufacturability*. New York: Taylor and Francis, pp. 377-402.