

**Novice and expert perspectives.** The peculiar difficulties of needing to satisfy both novices (students) and experts (trainers) requires that ITS have multiple levels of representation for both structural and functional components. Part of the enormous load of individualizing instruction and of developing elaborate student models can be supported by varying viewpoints on the knowledge representation schemes and by varying the level of abstraction and the grain of this knowledge representation. Thus an expert may want only a top-level overview of the system for some purposes and a very detailed, elaborate view of components for other purposes. A novice might well be overwhelmed by too much detail in the beginning and needs a graduated system of perspectives to lead into the complexities.

**Qualitative representations.** Qualitative simulations provide convenient vehicles for creating systems with both meaningful structural and functional components. Instead of a textual description of terminology and its interrelationships, the structure is described visually. Functional relations can also be described by using animation, color, arrows, and textual descriptions. However, visual descriptions of both structure and function are notoriously concrete: It is difficult to obtain the right level of abstraction without the use of conceptual descriptions in a textual and hierarchical form. Furthermore, it is difficult to create visual descriptions at different levels of abstractions in a truly hierarchical format. Functional and structural relations must be compressed or eliminated in arbitrary ways that defy accurate concrete representations. So, it is only when conceptual (textual) and concrete (visual) representations complement each other that adequately faithful models can be created.

In Psotka, J., Massey, L. D., & Mutter, S. A., (eds.), Intelligent Tutoring Systems: Lessons Learned. Lawrence Erlbaum, Inc., 1988.

Ritter, F. E., & Feurzeig, W. (1988). Teaching real-time tactical thinking. In J. Psotka, L. D. Massey & S. A. Mutter (Eds.), Intelligent tutoring systems: Lessons learned (pp. 285-301). Hillsdale, NJ: Erlbaum.

## Teaching Real-Time Tactical Thinking

Frank Ritter<sup>1</sup>  
Wallace Feurzeig  
BBN Laboratories

### TEACHING THINKING IN REAL-TIME

Virtually all AI system development to date has concerned tasks that do not involve real-time decisions in time-critical situations. Medical diagnosis, electronic troubleshooting, geological exploration, and computer systems configuration are representative of the kinds of task domains addressed in AI applications. In these kinds of tasks, the world within which the task is carried out (the task environment) does not significantly change while the user contemplates his next action. The user has essentially unlimited time between successive "moves." The AI system also benefits by having a great deal of time to process the user's inputs and make informed responses.

It is not surprising that considerably less AI development has been done in domains where interactions are complicated by the intrinsic need for real-time processing and response on the part of the system and its users. There is an important class of military tasks, those involving strategic or tactical decision-making in rapidly changing environments, in which real-time operations are essential, for example, air traffic control or radar intercept operations. AI systems for real-time operational tasks, for example, the pilot's associate, are only beginning to be developed. The AI methods and the instructional principles relevant to the design of the related Intelligent Computer

<sup>1</sup>This work was performed while the author was at BBN Laboratories. He is currently with the Department of Psychology, Carnegie Mellon.

Aided Instruction (ICAI) systems for real-time training are beginning to emerge from current work in progress.

The problems of instructional design for ICAI applications to real-time tactical decision-making tasks are significantly different from those involved in complex systems maintenance and troubleshooting tasks. The knowledge acquisition problem is more complicated for real-time tasks. Expert performers tend to "compile" their knowledge to enable efficient and rapid execution. They cannot explicitly or precisely describe how they do what they do. The development of indirect knowledge acquisition methods to characterize expert performance is essential here.

Similarly, the problem of diagnosing student errors and performance behaviors is severely exacerbated by real-time constraints. Diagnostic methods like those used in QUEST (Feurzeig & Ritter, chapter 15, this volume), where the student is queried for his or her reasons and hypotheses, can be very useful. However, in a real-time task, it is not possible to stop the world and interrupt the student to elicit the reasons for his or her actions in the way shown in QUEST.

The diagnosis of real-time performance, particularly in rapidly changing tactical decision-making tasks, has to be carried out after task completion. Even if the student's performance can be analyzed in real-time, it is not usually possible to engage the student until his or her problem-solving session is over. The system can then replay the student's recorded run in a debriefing session, critiquing his or her task performance along the way. Real-time tasks call for distinctly different methods, such as those being investigated currently in a system called TRIO, described in the subsection that follows.

## AN OVERVIEW OF TRIO

TRIO (Trainer for Radar Intercept Officers) is an expert instructional system for training F-14 interceptor pilots and radar officers in dynamic spatial reasoning and the basic tactics of high-speed air intercepts (Feurzeig, Ash, & Ricard, 1984). As of the fall of 1987, it is in place at Pensacola N.A.S., and is undergoing advanced test. The TRIO task environment supports simulations of airborne radars, interceptor and target aircraft operations, and weapons models. It provides dynamic displays of heading, bearing and displacement vectors, radar screens, flight instruments, intercept parameters, radar and missile envelopes, and interceptor and target aircraft ground tracks. It incorporates real-time speech recognition and synthesis subsystems including advanced

capabilities for recognition of naturally articulated utterances from an extensive lexicon. TRIO supports three instructional modes: (a) demonstrations by the TRIO expert program, (b) student practice with optional guidance, and (c) performance analysis and student debriefing following student practice.

### The Radar Intercept Task

The F-14's primary mission is air defense of the carrier. This is carried out by intercepting approaching hostile aircraft (bogey) as directed by the carrier. Because the speeds are so great and the missile envelopes so small, there is little margin for error. The intercept has to be right the first time.

The RIO (Radar Intercept Officer) interprets the radar displays and other flight instruments from the back seat of the F-14. He verbally directs the pilot based on his electronic (radar) view of the world until the pilot has visual contact with the bogey. Only after visual contact has been established does the pilot take command.

TRIO trains proficiency in a standard intercept tactic called a *gouge*. In the typical gouge, the RIO attempts to get off three coordinated missile shots: a long- and a medium-range forward-quarter missile shot, and a rear-quarter missile shot, ending up behind the nonmaneuvering bogey. To maximize performance, the student takes all three shots as if none of them were successful. Even though the bogey may not deviate from its straight flight path, the situation geometry is constantly changing. Each position requires a different set of maneuvers for successful positioning, culminating in a series of very precise turns to maneuver into a position behind the bogey.

The conceptual intercept space, shown in Figure 10.1, classifies each airplane position in three dimensions: (a) Lateral Separation: the distance between the fighter's and the bogey's flight path—along with the F-14's turning radius, this determines whether the fighter has room to turn in behind the bogey; (b) Range: the distance between the two planes—each missile has an optimum firing range; and (c) Target Aspect: this angle between the fighter bearing and the bogey's heading—in the gouge, this angle is optimized to provide the least radar cross-section to the bogey, while enabling the interceptor to stay in front of the bogey and to get a good radar picture of it. The RIO must also adjust his airspeed and altitude to gain the maximum advantage in intercept performance over the bogey. Finally, the RIO must maneuver the plane to achieve optimal missile shot parameters.

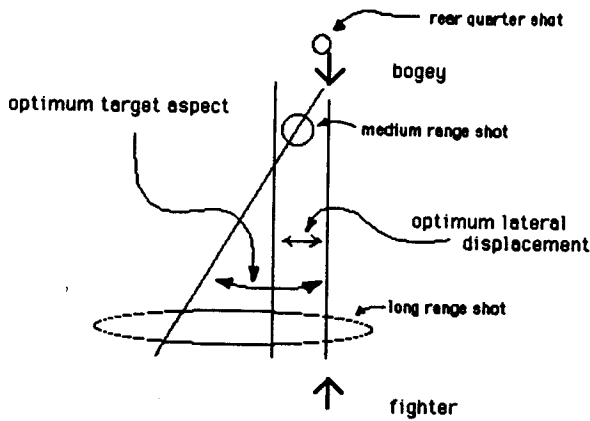


FIGURE 10.1 The intercept solution space.

As Figure 10.1 indicates, the missile envelopes and maneuvering space decrease as the intercept unfolds and the plane approaches the bogey.

Student RIOs tend to have difficulty with the central aspect of intercept work, that is, spatial mapping from a two-dimensional radar screen to the three-dimensional world. The radar screen shows the motion of the bogey and other objects relative to the interceptor symbol, not the more familiar bird's-eye view. The coordinate mapping requires students to use dimensions on the radar screens that are orthogonal to their real-world axis, or to use translations that compare different units. (For example, up and down, the Z dimension on both radars, is range in the X-Y plane. The lateral, or X dimension, on the DDD (Detailed Data Display) is not a measurement of displacement, but of angular offset in degrees.) To gain a real understanding of the radar screens requires the development of dynamic reasoning skills as well as considerable practice. The acquisition and assimilation of these skills is further complicated by the need to employ them while under stress in the rapidly changing air battle environment.

### TRIO Radar Displays and Flight Instruments

Figure 10.2 shows the instruments on TRIO's control panel. In the top left is an altitude readout in feet—in our example, 20,000. Next to that is an airspeed indicator reading .6 mach. Beneath both of them is the compass.

The window in the top center position is the DDD. It is an image

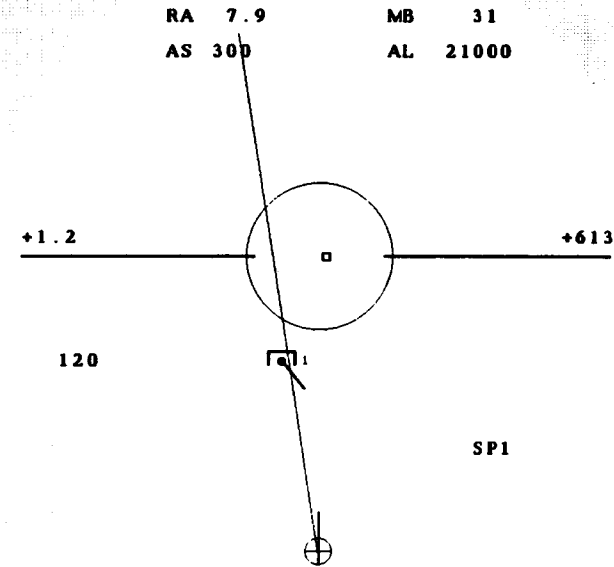
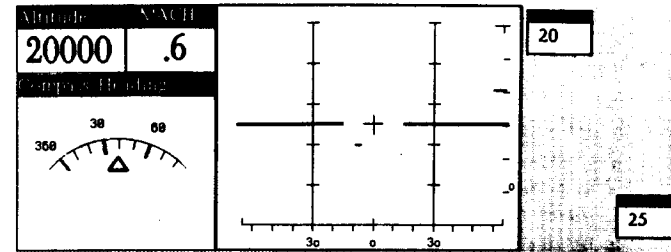


FIGURE 10.2 The student's situation displays.

of the raw analog radar data. The more easily read Tactical Information Display (TID) is the large round window taking up the bottom two-thirds of the display, beneath the DDD.

The TID displays the bogey's position relative to the fighter, and several flight constants in digital form. The fighter's position is represented by the circle on the bottom with a line pointing straight up it. This line represents the fighter's velocity vector. Because the fighter is always moving forward with respect to itself, this vector always points in the same direction. The thin dashed line from the fighter to the upside-down box is the antenna train angle (ATA)

between the fighter and the bogey. Each dash is 20 miles—this makes range estimation easy.

The bogey is represented by the open-box symbol with a vector pointing in the direction of the bogey's flight. In Figure 10.2 the bogey is headed across the fighter's flight path. The closing velocity is 613 knots, which is displayed on the right side midway down the TID. The closure velocity can also be estimated from the vector lengths. The heavy horizontal bar is the artificial horizon. A circle appears when a missile is selected, indicating the allowable firing parameters for that type of missile. The missile steering dot, visible in the circle, indicates the current firing parameters. When the dot is inside the circle, the missile is correctly positioned for firing. In addition to the circle, the two-letter abbreviation for the missile and the number of those missiles on board are displayed on the lower right side; our example screen shows that a Sparrow missile is selected, with one remaining.

The two-letter combinations on the top of the TID are abbreviated names for the flight parameters displayed to their right. RA is range in nautical miles. MB is magnetic bearing from the fighter to the bogey. AS, bogey airspeed in knots, and MH, magnetic heading of the bogey, alternate being displayed every 2 seconds. AL is the bogey's altitude in feet.

The vertical angle between the fighter and the bogey is printed below and to the left of the block of bogey flight parameters, and above the artificial horizon. Below this is a timer that serves as an indicator that the TID is functioning correctly. On the top left there are two windows for displaying the radar range scales selected.

### REAL-TIME INSTRUCTIONAL METHODS IN TRIO

Students learning a complex real-time procedural task generally need to be instructed before, during, and after practicing the task. Before attempting the task, the student needs instruction on how to do it. A dynamic explanation, such as an actual demonstration, is a great deal more desirable than a static description. While performing the task the student may need help to focus his attention on critical features at appropriate times, or to guide him through difficult situations. An instructor looking over the student's shoulder can provide enlightening and productive practice sessions. Instructors like to coach students learning procedural skills in the dynamic environment as they are learning. This enables students to be taught the features that experts use to understand the situation while in the context of the

ongoing task. Such coaching also allows exercises to continue that would otherwise have been lost due to early, correctable errors. The student will not have time for discussions of alternative strategies, or for detailed fault analysis while performing the task. Postpractice debriefing is a natural place for these discussions. We have implemented and integrated in TRIO instructional facilities for all three aspects of real-time training—demonstrations, guided practice, and critical debriefings.

### The TRIO Articulate Expert

On current training simulators, instructors of tactical tasks must spend time executing sample runs for the students to observe, emulate, and learn from. Automating this capability with an expert system is highly attractive to instructors and students. It can provide more viewing time for the students. When the expert performs at an appropriate level of performance accessible to a student, it provides a standard for the students to measure themselves against, while teaching them to use a common terminology. Instructors like an expert facility because it standardizes the initial training steps and frees them to spend more time teaching advanced topics.

Designing and building expert systems to explain what experts are doing while performing a procedural task is difficult; human experts often have trouble explaining while doing. One would like the explanations not just to inform, but also to gradually educate the student to think like an expert. Human experts think in terms of goals; and goal-driven actions; so should an expert program. The expert must also run quickly and efficiently because it sits on top of a simulation.

The TRIO expert program is capable of performing the same intercept tasks that it trains. TRIO provides an articulate goal-directed flying expert (FEXPERT) as an exemplar for student RIOs. The TRIO expert is articulate—as it performs air-intercept engagements it explains its performance along the way. Each time it takes an action (e.g., calls for a change in heading, altitude, or airspeed, selects or fires a weapon, or changes the radar display presentation) it can state the reason for the action, not only in terms of what the action is intended to accomplish, but also why this is desirable in terms of the RIO's goals.

The goal structures of the tactics employed in performing intercepts are explicitly represented in the rules that drive the FEXPERT. The use of a goal-based rule hierarchy enables rapid evaluation and execution of the rules and facilitates real-time intercept

performance in rapidly changing air battle situations. It also aids in the generation and presentation of tactically based explanations of the expert's actions to better motivate the sense and purpose of the strategic thinking and spatial reasoning involved.

When each goal is seen as desirable or necessary to achieve, it is introduced. Each action taken to achieve this goal then has a context to which it can refer. The use of a goal-based expert system by providing rule structure also provides the means to keep the number of active rules low. This significantly speeds up the expert system. If the expert can think quickly, it can be checked more often, which implies a higher level of expertise and of control over the exposition.

The expert's actions and explanations are given as spoken utterances, from a real-time speech-synthesis device. The use of speech as the output mode is necessary because the RIO trainee has to attend closely to the radar scope and tactical information displays generated during the rapidly changing air battle. This poses a heavy visual workload. The use of another modality that does not distract the trainee's monitoring of the displays or otherwise hamper his visual performance is essential.

TRIO also supports speech-recognition capabilities to simulate the RIO's voice communications with the F-14 pilot. The student RIO directs the intercept through commands to a simulated pilot provided in TRIO. The TRIO speech-recognition facility is capable of real-time recognition of naturally spoken English messages from a specified lexicon of allowable RIO utterances, consisting of flight directives such as "Come starboard hard as possible to a heading of two four zero degrees." The TRIO "pilot" interacts with the flight simulation to carry out the flight directives spoken by the RIO.

The articulate expert capability is central to TRIO's capability for instructional demonstrations. In this mode of instruction the expert performs an intercept in very much the same way the trainee is expected to perform it. The intercept problem is usually assigned by an instructor or generated by TRIO, but problems may also be posed to the expert by the trainee. The expert explains its actions and the underlying reasons for them in terms of its current goal structure. The knowledge is represented using a special type of production rule system—continuous running, interrupt-driven, goal-directed rules—to operate the articulate expert program. The expert performs intercepts in real-time and explains its actions and reasoning along the way. It uses the identical information that the student sees and drives the simulation through the same interface. The intent is to provide the trainee with concrete models that prepare him for his own attempt to do similar intercepts.

The expert system has to be efficient to work in this environment.

The simulation generates and displays considerable information to provide real-time response from the radar and aircraft models. Maintaining current information at the rates imposed by the high closing speeds of the two aircraft requires rapid processing. The student is polled continuously for new intercept control instructions. (The expert's performance must be attainable by the student, so the expert is only polled every few seconds.) The correct and rapid performance of the expert is achieved through the shaping of the rules into a goal tree and the use of a clean control structure.

The goal hierarchy that is taught to the student also guides the expert in running an intercept. The expert's task is represented in the rules as a series of high-level goals together with associated subgoals or actions for fulfilling each goal under any contingency. The top-level rules expand to access the rule sets that are appropriate for the current situation. Each rule has several components, including a test to check if this rule is to be fired now, an action to be taken (none for goals), a rule set to access next (none for actions), rules to remove from current consideration, and a rationale for why the current goal was established or the current action taken that becomes part of the explanation given to the student.

To illustrate the way rules interact, consider the following. When running an intercept, one would like to have a slight (.1 mach) speed advantage on the bogey. One of the first goals is to achieve this. Its test is set to be always true, because it is always desirable to achieve a speed advantage. When it fires it tells the student "Get co-speed plus." (Explanations must be terse because additional actions and explanations will quickly follow.) The co-speed goal then brings in three action rules to achieve itself. One of them will match the current situation of being either too fast, too slow, or at the desired speed. When one of them fires, it removes the others which cannot apply. It also gives a command to the simulated pilot instructing how to set the throttle. (All commands to the pilot are printed out on the screen or spoken for the student.)

The expert is called periodically by the simulation. If a rule has recently fired, the expert may suspend action for an appropriate time set by that rule. (The designated time represents the time between successive actions that a good student would require.) Without this delay, FEXPERT could run much faster during certain periods of the intercept, but it would fail as an exemplar because a student would not be able to achieve its speed.

Figure 10.3 shows the display of the expert's command to the pilot (on the top screen labeled FEXPERT's Commands), with the rationale that is normally spoken to the student printed below it.

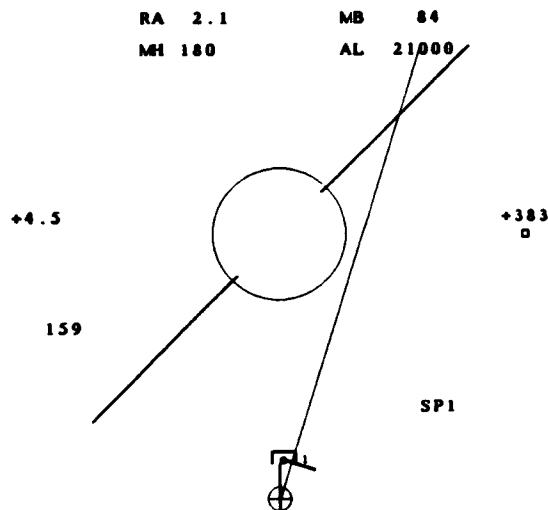
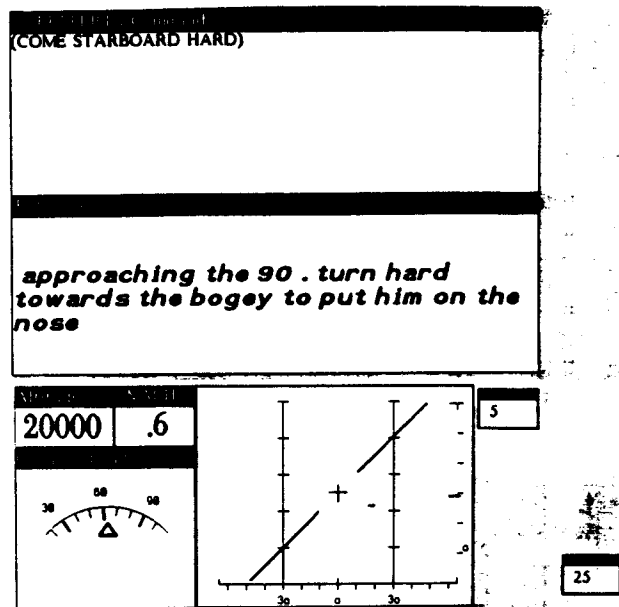


FIGURE 10.3 Example of FEXPERT output.

### Student Practice and TRIO Daemons

After a trainee has seen the articulate expert fly an intercept to demonstrate a new tactical procedure or the application of a familiar tactical procedure to a new situation, he typically tries to do it on his own using TRIO's guided practice facility. The trainee's performance is monitored and recorded for subsequent analysis. The trainee may choose to attempt the intercept without help. Otherwise, TRIO is available throughout the run to provide specific guidance to aid his understanding, and to help the trainee notice and avert major errors that threaten the success of the intercept. This is accomplished in TRIO by the use of daemons.

A daemon is a continuously active, rapidly executing program that monitors the state of task-critical parameters to detect a specific event, such as the imminent loss of radar contact (also called radar lock) or missile threshold. Daemons are used in TRIO primarily to detect and report imminent errors in time for correction by the trainee. For example, the missile envelope daemon's test checks to see if the current range is less than the maximum firing range. If its event occurs, a daemon takes two actions. It records the event on the history list for use in the post-flight-performance analysis debriefing narrative. It also alerts the trainee so that he can try to take corrective action. The alert is communicated as a short message in a daemon display window, possibly accompanied by flashing, speech, or alerting sounds generated by the speech-output device. This warning may just be a static warning like the missile daemon's warning to "Load the phoenix missile" (if the student has failed to do so), but it may also be customized to the current situation. For example, another daemon warns the student when he asks the simulated pilot to go above the maximum ceiling by computing the current maximum ceiling and reminding the student what it is. After the warning is generated, an action may also be taken by the daemon to correct the error. For example, some daemons reset a student-specified parameter, such as the plane's speed, to be within the safe range.

The daemons in TRIO are invoked either individually or in toto by the instructor, the student, or by other programs in TRIO. Their use is particularly valuable during the early phases of training when it is instructionally useful to develop a trainee's awareness of critical events that need attention and must be acted on.

Rapidly changing tactical situations such as those occurring in air battle engagements impose intense attentional demands. For real-time tasks with high cognitive loads, guidance must be presented in a way that allows a trainee to maintain his attention on the tactical situation while noticing and assimilating instructional communica-

tions. Interrupting real-time problem solving in an intrusive way can actually impair training (Munro, Towne, Cody, & Abromowski, 1982). So the guidance offered by TRIO, which may come when the trainee is absorbed in the intercept, is communicated in a clear and nonintrusive manner without stopping or slowing the action or breaking the trainee's concentration. This is accomplished by the use of terse spoken warnings about correctable inflight errors. Their use does not further task the visual workload of the student, and it provides all the desired guidance.

The Daemon System (Daemons) provides this capability in TRIO. It checks all the daemons on a frequent periodic basis for applicability to the current situation. If one of them applies, its warning is given to the student through the speech-generation device. Only one daemon is allowed to fire at a time—this ensures that the student is not overwhelmed with advice.

Figure 10.4 shows a student about to lose radar contact with the bogey because the student has allowed the bogey to drift too far to the right. The system warns the student, who may be attending to other instruments, of this imminent event. The daemon's warning that is spoken to the student is displayed here below the radar screens. If the student does not heed the warning by taking corrective action, he will lose radar contact, as shown in Figure 10.5. It is very difficult for novices to complete an intercept after losing radar contact. This warning may enable the student to complete the intercept and continue to learn.

Daemons help make the instruction in TRIO flexible. The student can select with the main menu which groups of daemons he would like available when running in guided practice mode. Daemons allow the student to control the amount of advice that he receives, and the areas that the daemons cover, such as safety-of-flight rules, or the use of radar-range scales. The debriefer is also able to activate daemons. This opens up a rich world of possibilities. For example, when the student makes an error, the debriefer can specify daemons to help the student on the next guided-practice flight. If the debriefer-selected daemon fires again, the debriefer can point out and emphasize this chronic error to the student.

Daemons do have limitations. Because they are being used in a real-time simulation that has its own computational requirements, there is a limit to the resources that are available. This limits the number that can be active, and imposes the restriction that the test be simple. The student's ability to process a limited number of corrections also limits the number that one would want active. The faults and the daemon's explanations must be easily understood by the student. Complicated faults must wait for the more leisurely pace

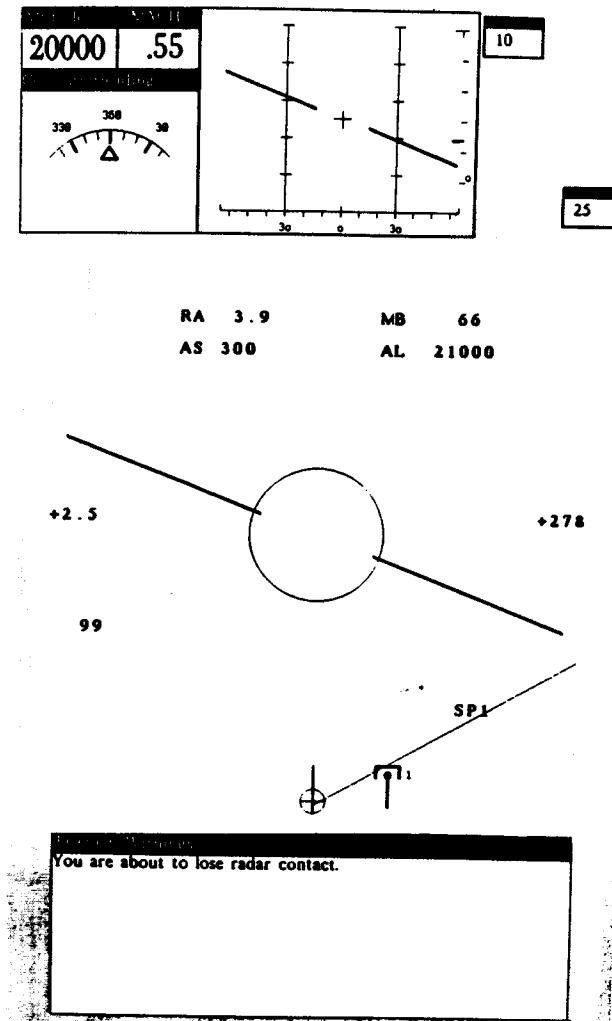


FIGURE 10.4 Display at radar lock daemon warning.

and clarity of expression provided by the debriefer. The debriefer, which has more time and resources, should also deal with faults that cannot be corrected in flight.

#### Performance Analysis and Student Debriefing

Following the practice run, TRIO analyzes and debriefs the trainee's performance. The trainee's intercept control actions are mapped into the solution space defined by tactical doctrine. The analysis of the

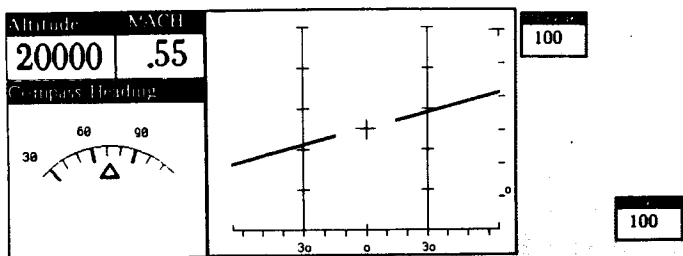


FIGURE 10.5 Display after radar lock lost.

trainee's performance is based on the use of pattern matching methods that translate and then compare the trainee's actions in a solution space to allowable performance behaviors. The solution space represents alternative solution paths during each phase of the intercept as permitted by the prescribed engagement rules and procedures. These paths allow considerable variation in the kind, number, and timing of trainee actions over those demonstrated by the expert program in its execution of an intercept.

The trainee's actions are examined by the performance analysis program. The solution space analysis identifies faulty action sequences. This is, those that could not be effective in realizing the

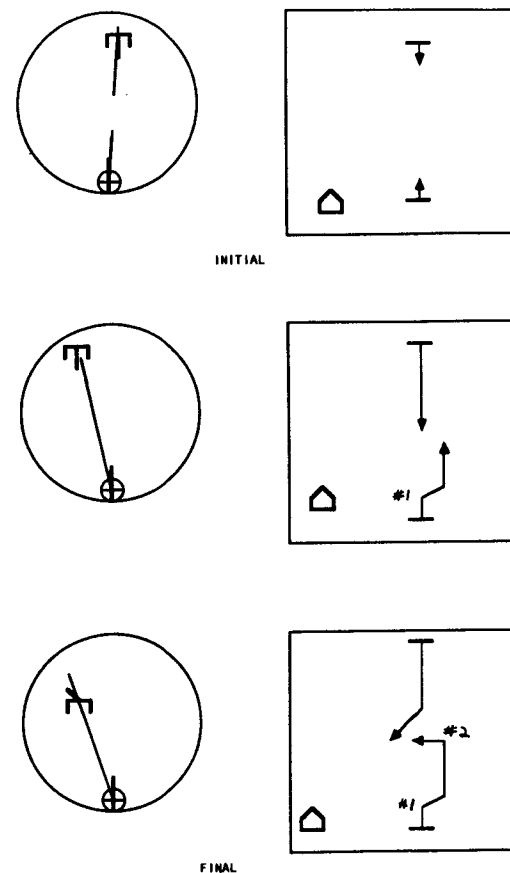


FIGURE 10.6 Ground track and associated radar displays.

appropriate subgoals in the intercept solution space, and determines very specific reasons for their unacceptability in terms of their adverse effects on the intercept. The analysis enables TRIO to generate explanations of what the trainee did wrong, where it happened, why it was wrong, and what he should have done. The explanations are given in terms of the top-level goal structure.

A debriefing program generates the corresponding narrative. The narrative is driven by two major sources of performance-history data—the solution space analysis and the procedural errors detected by daemons during the run. During the trainee debriefing, TRIO replays the relevant segments of the intercept together with the accompanying narrative. The trainee's actions and omissions, and relevant related



events in the intercept situation, are identified during the presentation through the use of ground tracks of the interceptor and bogey, the associated radar displays, and related displays showing the state of the flight instruments and key intercept parameters.

Figure 10.6 shows the series of ground tracks and radar displays that the student saw while he was flying and which are redisplayed by the debriefer. The debriefer's output is shown in Figure 10.7.

## CONCLUSION

The methods used in TRIO—goal-directed rules for expert presentations, daemons for guided practice, and analysis based on solution spaces for critiquing student performance—provide a rich autonomous learning environment for real-time procedural tasks. Providing an autonomous learning environment frees instructors to spend more time on advanced topics. This training paradigm has potential relevance for other real-time training applications in a variety of operational environments. Examples include ship navigation, GCI (ground controlled intercept), and flight-controller training. The methods generated by TRIO are now being employed as the central framework of an intelligent instructional system under development for training Patriot air defense operators.

More complex real-time training (involving coordinated intercept operations, for example) will require more sophisticated methods than

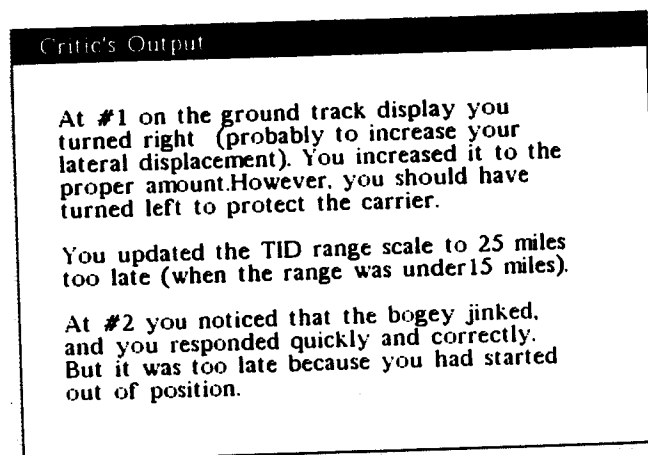


FIGURE 10.7 Critic's output.

those currently used in TRIO. We are beginning to explore extensions of TRIO methods toward these ends.

## ACKNOWLEDGMENTS

This work is being supported by the Human Factors Laboratory, Naval Training Systems Center, Orlando, Florida, under contract N61339-82-C-0143. We would like to acknowledge the invaluable technical guidance provided by our project consultant, Mr. Edward P. Harvey. Mr. Harvey was a Senior Instructor in the F-14A Fleet Replacement Squadron at Naval Air Station Oceana, responsible for training pilots and RIOs in tactical employment of the F-14A Tomcat.

## REFERENCES

- Feurzeig, W., Ash, W. L., & Ricard, G. L. (1984, October). TRIO, an expert system for air intercept training. *Proceedings, Interservice/Industry Training Equipment Conference and Exhibition*. ITEC.
- Munro, A., Towne, D. M., Cody, J. A., & Abromowski, H. (1982, October). *ONR final report: Techniques for computer-based training of air-intercept decision-making skills* (Tech. Rep. No. 101). Redondo Beach, CA: Behavioral Technologies Laboratories, University of Southern California.