TRANSITIONING AN ITS DEVELOPED FOR SCHOOLHOUSE USE TO THE FLEET: TAO ITS, A CASE STUDY

Dick Stottler
Stottler Henke Associates, Inc.
San Mateo, CA.

Nancy Harmon
MARCORSYSCOM, PMTRASYS
HarmonNJ@navair.navy.mil, 407-380-4003

Phil Michalak
Stottler Henke Associates, Inc.
San Mateo, CA.

Abstract

This paper describes our experiences in transitioning the Tactical Action Officer Intelligent Tutoring System (TAO ITS), designed and developed specifically for use by students at the Navy's Surface Warfare Officers School (SWOS), to fleet use. PMS-430 recognized that while they were fulfilling the needs of integrated team training, the Battle Force Tactical Trainer system required a major portion of the shipboard Combat Information Center (CIC) to be manned in order for the TAO to practice tactical decision making. Experts and instructors agree that the most important factor for maintaining a TAO's tactical decision-making skill is the opportunity to practice making decisions and timely feedback. SWOS has found that the TAO ITS increased the amount of such practice by ten times. Both PMS-430 and SWOS have deemed it beneficial to transition the TAO ITS to the fleet for shipboard use. The TAO ITS and the benefits realized by students at SWOS are described in [Stottler and Vinkavich 2000].

Transitioning the TAO ITS to shipboard use would realize several benefits. Since TAO ITS is PC based and requires no extra human players nor support personnel, it enables TAOs and prospective TAOs much greater opportunities to practice their tactical decision-making skills anytime/anywhere. One of the primary limitations to free-play simulated scenario training out in the field or onboard ship is the need for evaluation of the student's actions. Tactical decision-making practice is almost worthless without knowing whether the decisions were good or bad. The TAO ITS provides automatic debriefing capabilities, giving the student the important feedback as to which decisions were made correctly versus the omitted or bad ones.

There were several considerations in planning the transition of the TAO ITS to fleet use due to the differences between SWOS and the ship's environment and mission. Individual ships would want to train the TAOs with data specific to their ship and with scenarios appropriate for their geographical area. The TAO ITS already possessed this ability, but the existing interface was built to be used by only a handful of SWOS instructors. These capabilities had to be made far more user-friendly. In the schoolhouse, both instructors and documentation were available for students if they needed additional information. The shipboard version of TAO ITS would have to include this information.

The TAO ITS was alpha-released to the Fleet in January and beta-released in April, 2001. Recommended enhancements were made, and it will be released for general fleet use in August, 2001. The results and lessons learned from this process are described in this paper.

Bibliographic Sketches

Dick Stottler co-founded Stottler Henke Associates, Inc. (SHAI), an artificial intelligence consulting firm in San Mateo, California in 1988 and has been the president and CEO of the company since then. He has been principal investigator on a number of intelligent tutoring system projects conducted by SHAI including the Tactical Action Officer Intelligent Tutoring System. Currently, he is working on the transition of the TAO ITS for fleet use and on an intelligent tutoring system to teach armored company commander decision-making for STRICOM. He has a Master's Degree in computer science with a concentration in artificial intelligence from Stanford University.

Nancy Harmon is presently a Program Officer for MARCORSYSCOM, PMTRASYS focusing on Combined Arms Staff Training and Deployable Virtual Training Environments. Prior to this assignment, she was a Program Manager with the Naval Air Warfare Center, Aviation and Surface Directorates, focusing on training systems development for Navy and Marine Corps Aviation, and shipboard/shorebased training technology migration for the Surface Navy Combat Systems. She has been employed by the US Government for 27 years, 11 years in Program Management and 16 years in the contracts department.

Phillip Michalak has worked on a number of Intelligent Tutoring System (ITS) projects as an employee of Stottler Henke Associates, Inc. (SHAI). His current ITS duties include the lead software engineer role for an adult literacy ITS, and all of the software engineering and maintenance responsibilities for the Tactical Action Officer Intelligent Tutoring System (TAO ITS). He holds a Bachelor's Degree in computer science from Carnegie Mellon University.

TRANSITIONING AN ITS DEVELOPED FOR SCHOOLHOUSE USE TO THE FLEET: TAO ITS, A CASE STUDY

(Stottler, Harmon, Michalak)

PROBLEM DESCRIPTION

Expert Tactical Action Officers (TAOs) are a high value commodity because they make high-value decisions including use of the ship's weapons systems. Expert TAOs and instructors believe that the most important parameter for gauging the expertise of a TAO is the amount of tactical decision-making practice that he has had. One expert instructor stated, “The difference between a good TAO and a great TAO is tactical experience.” To greatly increase this tactical experience requires significantly more time in tactical warfare situations. Training in tactical scenarios has typically required expensive hardware and a large number of support personnel to play various roles in the simulation and to evaluate the TAO's performance. To reduce the cost and increase the accessibility of tactical training for TAOs, SWOS required a new training system that would run on a low-cost (PC) platform and eliminate the support personnel. It had to be highly portable, personal, and standalone. It needed to be highly configurable and maintainable by the tactical experts themselves. The goal was to greatly increase the amount of tactical decision-making SWOS TAO students would experience.

BACKGROUND

General Intelligent Tutoring System (ITS) Description

The most effective way for students to learn is to work individually, face-to-face, with a qualified tutor, well equipped with instructional material, lab equipment, and so on. As shown (in Figure 1), studies have shown that individually tutored students perform two standard deviations better than students only receiving classroom instruction. When learning a vocational skill, a student would also practice with real equipment used on the job, and interact with other members of the future work team or with a realistic physical simulation of the work situation if safety forbade actual practice. The teacher could then tailor the teaching approach to the student's speed of learning and performance; the proper technique could immediately be demonstrated when the student made errors, and gaps in the student’s prerequisite knowledge could quickly be detected and remedied. Unfortunately, expert instructors are a scarce resource, and buildings and equipment are expensive, making this preferred form of instruction costly and rare.

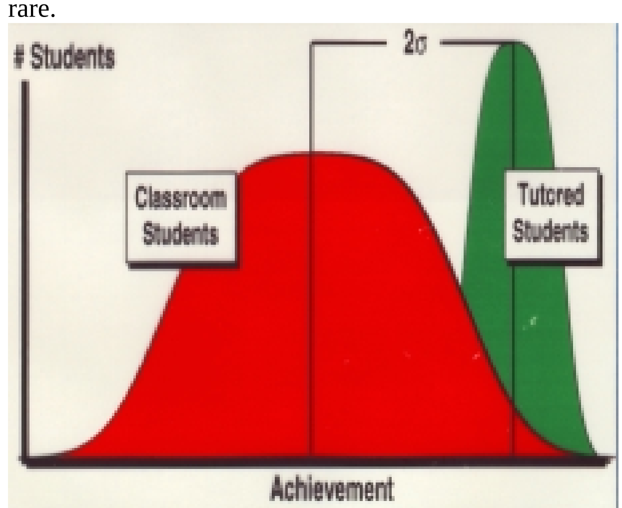

Figure 1. Tutored Student Achievement versus Classroom Student Achievement.

The goal of an intelligent tutoring system (ITS) is to provide a learning experience for each student that approaches the standard of learning received with one-on-one tutoring from an expert teacher equipped with all necessary training aids. ITSs use artificial intelligence techniques to adaptively make both "how to teach" and "what to teach" decisions appropriate to each individual student during a course.

To achieve its goal, ITS software monitors each student's interactions, and builds a "student model" for each individual. This model comprises the student's performance on training and remediation exercises; knowledge of all the information and remediation received; the knowledge mastered, failed, unknown, and misunderstood by the student; and the student's learning style. As an expert teacher who works one-on-one with a particular student would, the software develops an effective teaching style customized to each student. In other words, an ITS emulates an expert teacher teaching one-on-one in a particular subject.

While the evidence is still limited but very positive, ITSs have the potential to create a revolution in effective, low cost education and training. To date, computers have played a marginal role as compared to the central role of teachers. ITSs will ensure that computers support trainers and help students to learn much more efficiently.

ITSs are particularly effective when the preferred mode of instruction is “learning by doing.” The best ITSs embrace this philosophy, and most use simulations on PCs for exercises, avoiding the expense of physical equipment in many cases. Most traditional computer based training (CBT) systems are simply electronic page turners enhanced with hyperlinks and multimedia and multiple choice questions. Even the best CBT that provides "learning by doing" cannot provide the individual feedback students require without a high teacher/student ratio. Intelligent tutoring systems can enable adult students to quickly gain the expertise that otherwise might require years of on-the-job experience. Extensive studies supported by the U.S. government have shown clear superiority of intelligent tutoring systems over ordinary computer-based training.

Description of the TAO ITS

The following summary description is excerpted from [Stottler and Vinkavich 2000]. (See that paper for more details.) The TAO ITS is a simulation-based intelligent tutoring system designed to run on a PC. It has been used at the Navy's Surface Warfare Officer's School (SWOS) in Rhode Island since early 1999. As well as being a powerful assistant to the classroom instructor, the TAO ITS's advanced capabilities as an "electronic teacher" enable a student to use simulations for learning on his own, anytime, anyplace.

The TAO ITS was designed to provide tactical action officer students at SWOS with realistic, practice-based instruction and individualized feedback. A tactical action officer controls his ship's sensors and weapons and directs the movements of the ship and other support vessels and aircraft. The TAO also monitors the movements and actions of friendly and enemy ships, planes, missiles, and submarines in the region. The TAO integrates this information in real time to form a dynamic tactical picture, selects appropriate responses, and issues orders.

The TAO ITS allows students to act as TAOs in simulated scenarios and receive individual feedback on their performance in tactical decision-making, use of ship’s sensors and weapon systems, and reporting procedures. Unlike conventional training simulators, after a student completes each scenario, the TAO ITS also automatically evaluates the student's actions to determine tactical principles that the student has correctly applied or failed to apply. These detailed assessments of student performance are available to both the student and his instructor.

This evaluation is carried out using sophisticated pattern-matching algorithms defined by tactical experts via a graphical user interface, without programming. The student can then learn how to correct his problems by either selecting multimedia training material associated with any principle, or by replaying relevant parts of the last scenario he worked to review his mistakes.

The TAO ITS also helps the student choose the next scenario to practice with. The student can allow the software to choose a scenario that contains untested principles, or other scenarios that test principles recently failed by the student, or simply pick his or her own preferred scenario. The instructor can use a scenario generator included in the software package to create any number of additional scenarios, defining complex behaviors for each friendly and enemy ship and aircraft to create realistic, multi-agent tactical simulations.

The software has three parts: the scenario generator with which instructors, with limited assistance from a programmer, can create any number of simulated scenarios; the student interface, which presents selected scenarios to the student to practice different tactical concepts; and the instructor interface tool so the instructor can review all the students' work with the tutoring system and assess their progress in detail.

Scenario Generator

The scenarios created with the scenario generator can be set in any part of the world, and populated with different surface, air and underwater platforms (i.e., ships, planes, helicopters, missiles, and submarines.) Each individual platform is implemented as an "intelligent agent" and can be given its own performance characteristics and behaviors. For example, a hostile plane will have its own mission such as flying various patterns to search out enemy vessels, and when one is found to attack it.

Since the simulator is free-play, there is no guarantee that any particular concept will actually be tested when a student runs a scenario. For example, if the student orders his ship to head away from an enemy plane and remains concealed from it, an entirely different set of events may play out than those that would if the ship were discovered by the plane. To deal with this aspect of free-play simulators, the TAO ITS has "evaluators" associated with the concepts. An evaluator is designed to look for prescribed sequences of events and actions during a scenario. For example, if a missile is fired at the student's ship, there may be a range of appropriate actions he should take in response. A number of evaluators are set up to examine the chosen actions, and depending on what they observe, the software may recognize whether different principles are observed or not. There is not a one-to-one correspondence between evaluators and principles. That is, a combination of evaluated sequences may need to occur to trigger recognition of observance of one principle, or that one evaluated sequence may indicate that several principles were breached.

Student Interface

The heart of the intelligent tutoring system is the student interface that presents selected scenarios to the student so he can practice different tactical concepts. The software was designed to adaptively select scenarios for an individual student who needs to practice concepts (principles) he has not yet practiced or ones he has recently failed. It enables a student or instructor to pick any scenario from all of the ones available. As well as the intrinsic feedback that free-play simulations naturally provide, the TAO ITS provides detailed, useful, extrinsic feedback to the student once a scenario is finished. This feedback reviews all the concepts attempted and whether they were passed or failed. At this point, the student can review multimedia material about any concept or see a replay of the scenario to review errors.

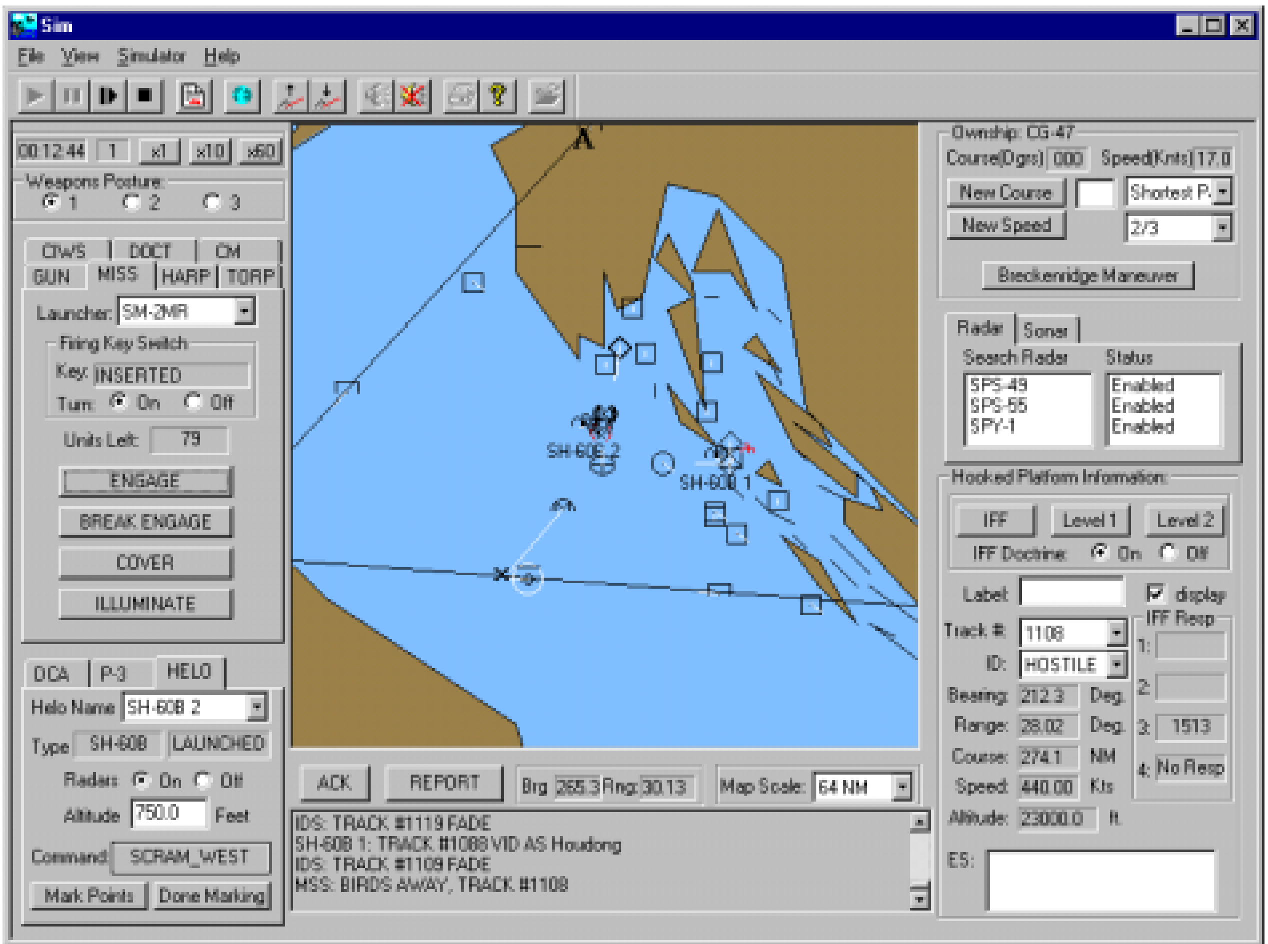

Figure 2. TAO ITS Simulator Interface.

While there is a basic physical simulation driving all the platforms in a scenario, the simulator in the TAO ITS is inherently conceptual. For example, a tactical action officer on an AEGIS cruiser works in the Combat Information Center (CIC), and is supported by a large number of individuals who provide him with information and respond to the TAO's orders. To simulate all of the commands, the TAO ITS provides the means to implement them. Thus, (as shown in Figure 2), the left uppermost section of the screen operates all the ship's weapon systems, the lower left section issues commands to any supporting aircraft, the upper right section provides control over the ship's navigation, and the lower right section operates the radar and sonar equipment and displays numerical responses to this equipment. The lower middle panel displays communications from crewmembers, for example, reports of incoming missiles, and these communications can also be heard with the voice synthesizer. The central display panel is a reasonable facsimile of the large screen display in the CIC, and it displays only the information that would be realistically available to the TAO at any time.

Once terminated, the software evaluates the student's actions by comparing his actions and the circumstances under which he made those actions against the "evaluations," and prepares an Evaluation Summary (see Figure 3).

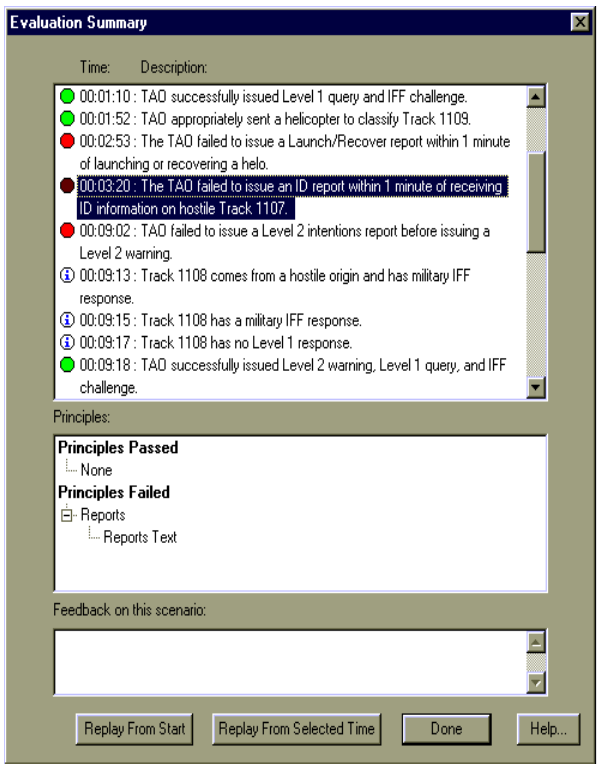

Figure 3. Evaluation Summary.

The Evaluation Summary lists the situations in which the student demonstrates understanding of concepts (principles) by their correct application. Correct decisions are marked by a green circle. Situations where understanding of concepts was not observed (either by incorrect or omitted actions) are marked by red circles. Also provided are the time and description of the action. The exact principles observed or not observed for any of these situations can be found by clicking on the particular situation. By clicking on any noted principle the student will be taken to multimedia information that explains the principle. The Evaluation Summary form also allows the student to replay the recent scenario from the start or from a selected time.

Instructor Interface Tool (IIT)

The instructor can manage the students as groups using this tool (see Figure 4). It also provides the instructor with tools to manage the hierarchy of instructional principles and the set of multimedia review content, to link specific multimedia review content to principles, and to associate principles and evaluations with specific scenarios.

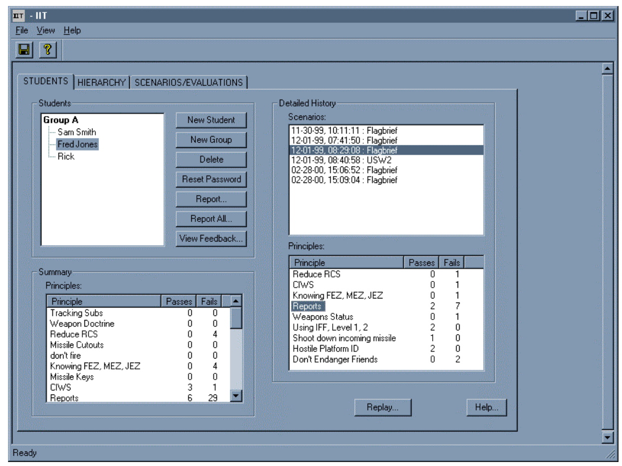

Figure 4. Instructor Interface Tool.

TAO ITS operates on Windows PCs and can be configured to run in several different ways. This is accomplished by a single installation CD with the installation script asking which of 3 installations is desired – Server, Student, or Instructor (which includes the Student version but adds the scenario generator and IIT). The most useful installation for operational use is multiple users on a network. This requires the installation of the Server version on the network file server, usually a single Instructor version on the instructor's machine and several Student versions on student machines. Since the student version is installed most (and potentially by the least informed user) the Student version is the default.

When TAO ITS was originally completed and delivered to the SWOS in 1999 it was not seen as a system to be completely maintained by the developers. Rather it was conceived as a shell that SWOS could use to enter principles, their descriptions, scenarios, and evaluation machines to create a complete training system. Since the large majority of the training system maintenance related to keeping these items up-to-date, SWOS had the capability to maintain and update TAO ITS themselves. The content that was originally delivered with TAO ITS was completely conceived by SWOS. Some of it they had entered, but most of it was entered by the software developers, owing to a lack of available time by the SWOS personnel. Independent studies of students at SWOS have found almost all students had highly favorable reactions. Instructors estimate that students get ten times more tactical decision-making practice now than before TAO ITS was used.

WHY TRANSITION TAO ITS TO THE FLEET

The Problem Description Section describes the general need for TAOs to get more tactical decision-making practice. While the TAO ITS was remarkably successful at achieving this goal at SWOS, shipboard personnel were experiencing the same lack of tactical decision-making practice that the TAO ITS was designed to solve. They had no personal, standalone tactical training capability onboard. The shipboard training systems were mostly designed to provide team training using the actual existing combat system equipment. Thus the only way for the TAO to practice was with the majority of the CIC staffed along with some support personnel, an expensive exercise which provided only limited opportunities. Meanwhile, since the TAO ITS was allowing ten times more tactical-decision-making practice at SWOS and was highly portable (it could run on a laptop), it was natural to try to transition it from a schoolhouse-only environment to additional use in the fleet.

During preliminary visits to various ships and fleet training organizations in 1999, several different fleet uses of TAO ITS were suggested by fleet personnel. Of course the most obvious was standalone TAO training. Other suggested uses were mission rehearsal, battle order development, junior officer training, battle group development and coordination (with a networked version), and head-to-head training (also with a networked version). Some of the more enthusiastic fleet personnel stated that they were ready to use TAO ITS immediately. Based on these preliminary investigations, we began planning for the transition of the TAO ITS to the fleet.

CONSIDERATIONS IN PLANNING THE TRANSITION

Since the TAO ITS would not run on the same computer hardware as the existing on-board tactical or training systems we did not encounter any technical difficulties in transitioning the software to shipboard use. In fact our early visits to ships showed that the PCs onboard the ship and the LANs used to interconnect them, were exactly like those at SWOS and those found in typical office environments.

We anticipated that the largest difference in the use by the fleet would be a greatly increased number of users and a stronger desire to customize TAO ITS to specific ships and their own types of missions. We expected that they would want to define their own ship's parameters in more detail and with classified information. They would want to author new scenarios and the accompanying intelligent behaviors to control simulated platforms in those scenarios as well as new ship-specific principles and evaluations. With the increased number of users from the fleet would come greatly decreased personal support from SHAI as compared to SWOS. At SWOS, while there were dozens of student users at a time, there were only a handful of instructor users. The use of TAO ITS as a student is very straight-forward as compared to an instructor who needs to author scenarios, specify platform behaviors in those scenarios and create evaluation machines. The few instructor users at SWOS could be easily supported by the TAO ITS software development team. Furthermore, they had the two year development period to slowly become familiar with the software. In transitioning to the fleet, dozens or hundreds of fleet users would be introduced to the tool simultaneously and would have little time becoming familiar with it. Also contact with shipboard personnel is inherently more difficult.

Our primary strategy with dealing with this decreased personal support problem, was to substantially improve the hard copy and online documentation. Most of the user interface was already about as intuitive as it could get, given the basic concepts on which TAO ITS is built. As described later, this documentation effort had little effect.

One concern was the complexity of the installation procedure. This was greatly simplified with defaults supplied as appropriate and additional explanatory documentation written. Furthermore, as described later, a special one page, very simple installation and getting started document was included as hard copy with every 4.0 Alpha and Beta CD. We wanted to make sure the evaluators had no problems with the installation.

We also needed to devise a way that TAO ITS could be delivered as both classified and unclassified versions. This was accomplished by developing two different installation CDs. The primary installation was the standard TAO ITS 4.0 Beta CD. But since all types of information in TAO ITS which might eventually need to be classified are represented in data files (including ship and weapon parameters, scenarios, evaluation machines, principles, behaviors, and descriptive information), the standard non-classified data can be easily overwritten (or edited by users). Thus the classified CD simply overwrites some of the unclassified files with classified ones. Furthermore, fleet users are free to install TAO ITS on a classified machine then edit the unclassified data with classified data (such as more accurate parameters for their own ship, for example.)

A related problem which could occur was that, for the first time, a large number of users would be editing scenarios, behaviors, principles, and especially platform descriptions, in parallel. We needed to devise a way that when future updates of TAO ITS were delivered, with updated data files, that these updated files would not overwrite the work of fleet personnel. Furthermore, it would be inefficient to have many fleet personnel entering data for a large number of enemy platforms in parallel. The most common platforms desired for scenarios needed to be pre-entered once, before the fleet version was widely released.

The version of TAO ITS developed for SWOS did not include an undersea warfare component. It was felt that the fleet version must include this component at the level of detail appropriate for the TAO. Before development began, it was decided that the concept of operations for the fleet TAO ITS should be determined based on visits to actual ships which would also be helpful in determining what enhancements were required. Time was allocated to answer questions from fleet evaluators, though little of this was ever required, except to conduct onsite observations. Effort was allocated to support SWOS's entry of content. The majority of the effort was allocated to implementing the anticipated enhancements including better documentation, clearer installation procedures, more ship type definitions, and minor user interface upgrades. Unfortunately, there was not enough budget to make some of the more far-reaching changes such as a networked version for head-to-head and battle group coordination training.

As described above, TAO ITS was intended to be a shell which contained knowledge maintained by naval personnel. But as we developed the fleet version, the purely administrative question arose, "Which naval personnel?" Heretofore, SWOS owned and maintained all the TAO ITS content; but one of the advantages for the fleet was the ability to customize the scenarios, behaviors, evaluations, and principles for individual ships. It was eventually decided that SWOS would own the primary tactics, which consisted of the principles, their evaluations, and a basic set of scenarios and behaviors. The individual ships would own the ship modeling parameters, new ship-specific scenarios, and new principles and new evaluations for those principles, without changing the SWOS principles and evaluations.

Although SWOS was extremely satisfied with the TAO ITS, this was partly because they were so involved in defining its functionality. SWOS, NAWC-TSD, and the software development team wanted TAO ITS to be successful in the fleet and all worried about fleet acceptance. NAWC-TSD pointed out very early on about the ramifications of a much larger base of users. Therefore, a cautious approach was taken that began with an Alpha release of the fleet version to NAWC-TSD, SWOS, and a few fleet users. This Alpha release included what were thought to be the minimal enhancements needed for the fleet version. Feedback from this release guided the list of enhancements to be implemented. These were implemented in a Beta version which was released to a wider, more diverse audience of fleet users who represented an unbiased cross section of the fleet as a whole. Once this feedback was received, the version for the entire fleet could be released with more confidence of its acceptance.

FINDINGS

The plan described above was followed. The first difficulty encountered was getting the quality of the feedback that we expected from the Alpha version. The primary difficulty was in getting evaluators to spend the time necessary. Few went beyond the simplest use of TAO ITS – playing the role of a student in simulated scenarios and getting debriefed. The comments were entirely positive but didn't address what we knew to be the potential problems – scenario creation and behavior editing. We did not allocate the time to make sure the evaluators were performing the necessary functions, because we wanted the evaluation to be as realistic as possible: When the TAO ITS would eventually be released, we would not be able to call all fleet users to get them using it. The end result was that the Beta version was created based more on our previous beliefs as to what was required than on the previously anticipated feedback. This primarily related to some redesign of the scenario generator user interface to make it more intuitive, allowing remediation files to be web links, and facilities to allow both classified and unclassified versions. We also were determined to be more active with the Beta version reviewers. Their experience is described below.

Feedback on the fleet usefulness of the Beta version of TAO ITS 4.0 was received from about a dozen fleet personnel ranging in rank from ensign to captain, although lieutenants and lieutenant commanders were the most represented. About half were observed in person. In general, the comments were primarily positive. Almost everyone's summary comment was something like, "This is a good training tool and it will be useful onboard." Beyond that summary, there were a large number of specific findings. The time of fleet personnel onboard ship is severely limited. This has several consequences. Personnel are conditioned to rarely read software documentation. For example, as described above, TAO ITS can be configured to run in several different ways. For evaluation purposes, the best installation is to install the Server and Instructor versions on the same machine, the one to be used to perform the evaluation. TAO ITS 4.0 Beta was accompanied by a single piece of paper which had 5 bolded headings – Overview, TAO ITS Installation, Requirements, Server Installation, and Instructor Installation. Overview consisted of two lines, the first being: "This document is designed to guide you through setup of the TAO ITS for evaluation in a standalone environment." The first two sentences of TAO ITS Installation were: "This section gives detailed instructions for installation of the TAO ITS in a standalone environment (everything is installed on one machine, and there is only one user). This involves performing a Server Installation followed by an Instructor Installation (the directions for both are below)." The server and instructor installations then each listed 6 steps, all but one of which was simply hitting the "Next" button. The other step was choosing either "Server" or "Instructor". This is also described in the on-line documentation. Yet, the majority of evaluators failed to install the Server version, on which the other versions depend. The only solution to this problem would seem to be to send a separate CDs for evaluators which, with one mouse click, installs all needed versions on a single machine. This will either force us to send a separate CD with a different installation script for operational users, or no operational version will be installable by merely accepting all of the defaults. The needs for operational users are in conflict with the needs of evaluators (even though these may be the same people, just at different times).

As described in the Transition Planning Section, the improvements in user-friendliness that we made were accomplished primarily though the on-line help and on-line Microsoft Word User documentation. These were very rarely consulted. We are devising a method for bringing relevant information to the first-time user's attention in a more proactive way, when it is relevant.

TAO ITS is highly configurable. New scenarios can be (and are intended to be) created by tactical personnel. Furthermore, the behavior of the entities in the scenario including enemy platforms, friendly platforms, warfare commanders, and team members can all be defined graphically and without programming. However, the fleet personnel were not used to having this kind of capability and the flexibility it provides. Evaluators would only attempt to edit these items when personally prompted to do so. At times there was a problem with fleet personnel understanding the degree to which TAO ITS was configurable. For example, a comment from one evaluator was that in a particular scenario, to make it more realistic, there needed to be more involvement from the warfare commanders. He couldn't be made to understand that, as an instructor, he could change that behavior himself. Certain types of users could more quickly grasp these configuration capabilities quicker than others, as described below.

The fleet users could be broken down into two different groups based on their skills. The younger members of this group, typically ensigns and lieutenants, could quickly grasp the concepts of creating scenarios and behaviors, and in fact enjoyed doing it. However, this same group tended to either be less disciplined in their use of correct protocol or became overwhelmed by the tactical requirements of playing the scenarios. They didn't have the good cognitive tactical habits of the other group. The other group tended to be older, and included older lieutenants, Lieutenant Commanders and above. They could much more easily handle the tactical requirements of playing scenarios, but had more difficulties creating scenarios and especially creating the intelligent behaviors of platforms. Since everyone could intuitively use the TAO ITS's simulation without looking at the documentation, this group's lower computer skills did not hamper their ability to perform in the simulation. This was an important design goal of the simulation because the ITS assumes mistakes are based on a lack of tactical knowledge, not a failure in knowing how to use the simulation. The existence of these two groups suggests that fleet use at the instructor level would be facilitated by pairing an older, tactical expert with a younger, computer savvy assistant, at least for creating scenarios, creating behaviors, and creating new evaluation machines.

As described above, TAO ITS was conceived as a shell with accompanying content to be maintained by SWOS. We considered SWOS our TAO ITS development partners. However, difficulties arose with this arrangement as we developed the fleet versions. SWOS is not equipped or funded for this role. It was difficult to share the responsibility (programmers responsible for software development and SWOS responsible for content) for preparing the fleet version of TAO ITS. Anyone at SWOS working on TAO ITS content had other duties as their primary responsibilities. Most damaging was the inevitable Naval staff turnover. As the main SWOS TAO ITS content authors left, they were replaced with newcomers. As the third generation was arriving, the TAO ITS software implementation was substantially complete. This meant a greatly reduced role for the software developers and a greatly reduced presence of those developers at SWOS. This made it difficult to interact with the newcomers to the degree really necessary. A similar problem, but even more difficult to deal with, is now manifesting itself in the fleet. Ship personnel experience a high degree of turnover too. For example, the 1999 XO of a particular AEGIS Destroyer was involved early in the TAO ITS project and was an enthusiastic supporter of its use onboard. He had it installed and evaluated it and determined that it would be useful. One of his assistants was trained in its use. But both have since left the ship. Further discussion below of the findings is broken down by the different software modules they relate to.

Simulator and Debriefing Findings

More fleet evaluation use was performed on TAO ITS's simulator than any other component. This is the component in which the student spends the large majority of his time. The findings were similar to those experienced at SWOS: almost universally positive, with comments like "good training tool" and "good feedback". Most evaluators tended to group their comments on the simulation and the debriefing together, since the two are so closely tied together, and thus, were very complimentary of both modules. In particular, the simulation appears to be very intuitive for most fleet users, with no need to read the documentation. One set of negative comments related to the fact that TAO ITS does not check for every possible negative act. This is actually a limitation in the current content, not in the current software. But, it is the case that an evaluation machine (or at least one transition in an evaluation machine) must be created to explicitly check for each significant incorrect or omitted action. Typically, the instructors do not create evaluations for grossly incorrect actions if they feel a serious student would never make such a mistake. Again, this particular evaluator didn't realize that he, himself, could have added an evaluation machine to check for the particular incorrect actions that he was concerned with. Computer savvy fleet evaluators playing the role of students particularly like the debriefing and in-scenario prompting provided by the automated warfare commanders relating to reporting and querying. One improvement they would like to see is the content of a proper SITREP when it is sent. Currently, the SITREP is sent when a button is pushed and the proper content is never shown to the student.

Instructor Interface Tool (IIT) Findings

Fleet evaluators created students and groups without any problems. They quickly grasped the difference between scenario files in the Scenario Generator and scenarios in the IIT, although we will need to make the documentation more proactive. Users can quickly grasp the meaning of the student performance displays. Scenario files can be easily copied, but IIT scenarios could use this capability as well.

Scenario Generator Findings

Several fleet evaluators were given the task of creating a scenario with no guidance, in order to see which items caused them problems. The user experience was very positive. There were many positive exclamations from the lieutenants as they explored the Scenario Generator (SG) interface. They had fun exploring the interface and creating scenarios. It was especially good that it was simple to change the Ownship platform type since they had the misconception that it was an AEGIS-only tool. The younger group performed significantly better than the older evaluators on most tasks, with larger differences noted on tasks that are more like programming such as the creation of Platform Behaviors in the graphical editor, which they caught on to quickly. Younger evaluators immediately placed platforms with the SG to create scenarios, with older evaluators lagging slightly behind. Evaluators effectively used online help when prompted to do so and said that the online descriptions of the behavior primitives were useful. Younger evaluators would use the simulation for quick sanity checks on scenarios, whereas older evaluators tended to do this with difficulty. Evaluators made good use of pre-existing behaviors.

There was some confusion between editing platform instance characteristics and platform type characteristics. The SG should probably warn the user when he is about to edit a platform type, since this is rarely his intent. Uncharacteristically, the SG crashed on two occasions, but in both cases, the "autosave" feature had preserved most or all of the data. Autosave really helped to keep a positive impression of the tool. The crashes may have been the result of several evaluators using one networked copy of the SG. An undo for the SG (beyond file save/restore) would be useful since some of their experimentation led down false alleys and they would have liked to have been able to undo a change. The unusual method of leaving object insertion mode using a right click (whereas left click continually adds more simulation objects) continued to cause confusion, even with the existing tip at the bottom of the screen.

One major upgrade the evaluators requested was SAG (Surface Action Group) capabilities. Many scenarios that the users were trying to create involved tasking other ships to deal with a threat. In fact, two students created a behavior whereby a friendly ship would follow and attack a platform that had fired on ownship. They wanted the capability to direct the support ships to take this action (during the simulation) rather than creating a behavior for it. They also wanted to use Sectors for assigning responsibilities and group oriented defenses.

During creation of behaviors, multiple users tried to connect states to existing transitions which is not allowed. The documentation was not clear on the fact that existing transitions can't be moved from one state to another. They also had a misunderstanding about what the term "Ownship" referred to and how to create logical Ors and Ands with the behavior editor.

Other Miscellaneous Issues

Different evaluators had different opinions as to the value of TAO ITS being classified versus unclassified for fleet use. Some felt that the convenience of being able to use it on any ship PC outweighed the slight loss in fidelity of the unclassified data. Others thought that the fidelity was more important. Fortunately, TAO ITS can be set up to run either way easily. A few minor miscellaneous enhancements were suggested, but not required. In fact, most evaluators thought it would be a useful tool for fleet users as-is, without any modifications at all.

One important capability which has been suggested several times during the last couple of years is a multiplayer version. This remains on our desired list of enhancements.

LESSONS LEARNED

The following are a list of general lessons learned that should be considered when transitioning an ITS from schoolhouse to fleet use. Many would also be applicable to any advanced training system. First, we found the PCs and LANs used onboard ship for general administrative functions (and available for use by standalone PC-based training systems) are the same as would be expected in a typical office environment and present no technical difficulties for software designed to run on generally available PCs and LANs.

Documentation, no matter how convenient, concise, or well-written is rarely read, including even on-line help. Therefore, you can't document around an awkward interface feature or difficult concepts. Bringing up needed information proactively the first time the user enters the relevant area is a good idea. Furthermore, shipboard personnel have very little time and are very hard to get hold of. It may be difficult to recruit enough of them for evaluation purposes.

Most importantly, simulation-based ITSs appear to be acceptable to (and even welcomed by) fleet users. Fleet users believe them to be beneficial, although further study will be needed to confirm this.

Software designed for schoolhouse use will often also be useful for training on board ship, with relatively minor enhancements required. The TAO ITS simulation is intuitive for fleet users even though its functionality was designed by SWOS instructors without the fleet in mind per se. Similarly, TAO ITS's student management capabilities were intuitive and easy to use, even though it was designed primarily for SWOS. The flexibility to allow SWOS instructors to customize many aspects of the system is a good capability for fleet users too. The Navy got twice the return on that particular investment, which should be considered for other schoolhouse systems.

It is hard to expect a schoolhouse to enter and maintain significant content, especially without a budget to do so. This is true even if they have the software tools (and enthusiasm) to do so. The time of Navy tactical experts must be explicitly allocated to create the necessary base scenarios. It is important to resolve early what information should be standardized and what information ships can tailor. The keeper of the standardized knowledge and information must be explicitly decided.

When working with both the fleet and schoolhouse, be prepared for a lot of staff turnover. Navy personnel cycle through many positions in about two years, perhaps even less for fleet positions. A project is likely to catch most in the middle of their rotation. The number one defense is to make the software as user-friendly and intuitive as possible.

In general, the user-friendliness versus capability trade-off (given a limited budget) works itself out differently between a schoolhouse and the fleet. Both want very user friendly interfaces for the students (i.e. the simulation). The schoolhouse would rather sacrifice user-friendliness to have more capabilities, given that they have a small number of users, compared to what the fleet users would decide given the same budget.

One solution is to drop or hide some of he capabilities in the fleet version. However, this tends to lead to separate versions for the fleet and the schoolhouse, which is impractical. To obtain the optimal user-friendliness for the fleet with the same capabilities as the schoolhouse, requires a very substantial budget, more than the software developers probably realize.

Since the software developers are involved with the system for a long time and they observe users (at the schoolhouse) who have mostly been involved with the system for a long time, they have a hard time realizing what usability problems really exist and how they should be fixed. It takes experience with a fresh batch of fleet users to shake things up adequately, which is why an early Alpha version release to fleet users is so helpful. If a user interface feature is noticed to be non-intuitive early, it probably always will be until it is redesigned. This seemingly obvious fact can get lost when dealing with a small group of users over a long period of time. For example in TAO ITS, every new user always had to be told about using the right click to turn off the object insertion feature, yet it was never reimplemented. It was never considered important since the number of new users had always been small and spread out over a long period of time.

It may take users a while to understand, appreciate and utilize new capabilities (such as the new capability to configure what used to take programming to change) they are not accustomed to, especially if they are less computer-savvy. Single click installation of an evaluation setup is a good idea for evaluation purposes. Fleet users have different concerns than schoolhouse users. But all fleet users are not the same; they are not one monolithic group, but will be diverse with different strengths and weaknesses in both tactical and software knowledge.

It is a good idea to let the individual ships decide separately between classified and unclassified versions. An added benefit of having so many user-defined model parameters and behaviors is that this information can all be kept in data files, thus keeping the software itself unclassified. It is important to keep the future training system updating process in mind in advance of the first operational fleet release to protect the investment of the users' effort relating to editing scenarios, behaviors, etc. Similarly, if multiple fleet users want to input the same data (such as a description of a particular threat platform) then that data should be pre-entered, to eliminate redundant entry work of users. Head to head and cooperative versions of training simulations are a good idea. Another alternative is a competitive scoring system. Autosave is a good feature to have in a scenario generator or other software requiring significant user input. It can even maintain a good impression of your tool in the face of unexpected (and uncharacteristic) crashes. "Undo" is also a nice feature if you can afford it, especially for new users. Most fleet users could, with some effort, create scenarios using existing behaviors on their own. More computer-savvy users could create behaviors and edit platform definitions with some effort. Fleet users have difficulty distinguishing between types and instances. (This is a very general problem that we have encountered in several domains and applications.)

Alpha and Beta testing with fleet users is important! This is because a new group of users will always try to do things (and have misunderstandings) that you did not anticipate. This will cause you to discover the need for more enhancements than you expected. We found that about half of the needed enhancements could be at least guessed at in advance, but that the other half were completely unanticipated. Therefore, a budget created before the transition project starts based on anticipated enhancements will always be short of what is really needed.

FUTURE WORK

We plan to receive maintenance and support funding for TAO ITS to continue to improve and maintain the product and aid SWOS in the maintenance of its content. We expect these improvements primarily to relate to making it easier for nonprogrammers to create behaviors and evaluation machines. We have received funding to make enhancements specifically to improve TAO ITS's adaptability to different students and evaluate the effectiveness of those improvements. We hope to get funding to make a networked version for head to head and battle group coordination training.

REFERENCES

Stottler, R. H., and Vinkavich, M. (2000). *Tactical Action Officer Intelligent Tutoring System (TAO ITS).* I/ITSEC 2000 Proceedings.