An Intelligent Tutoring System (ITS) for Future Combat Systems (FCS) Robotic Vehicle Command

Randy Jensen, Richard Stottler
Stottler Henke Associates, Inc.
San Mateo, CA
Jensen@StottlerHenke.com
Stottler@StottlerHenke.com

Henry Marshall, Jeffrey Stahl
Simulation Technology Center, US Army Research, Development and Engineering Command
Orlando, FL
Henry.A.Marshall@us.army.mil
Jeffrey_Stahl@peostri.army.mil

ABSTRACT

Under the Army's Future Combat Systems (FCS) concept, the warfighter manning a Control Vehicle (CV) crewstation must maintain situational awareness and apply tactical decision-making principles in a heightened information-rich setting with distributed vehicles and sensors under his command. This paper discusses a proof-of-concept Intelligent Tutoring System (ITS) to provide scenario-based practice for the FCS soldier. In this context, a limited principle hierarchy serves as the instructional basis for the training system and the automated evaluation of student actions in an FCS scenario. Embedded training systems for this domain must be integrated with a variety of software packages using a common protocol. This system communicates with the OneSAF Test Bed (OTB) simulation environment, and the control interface for networked robotic vehicles under the student's command. In addition to the fundamental tactical principles, students are also monitored for their mastery with the task of translating tactical intentions to robotic commands correctly executed in the control interface. The ITS observes the student's actions and performance in a simulated scenario and produces specifically tailored feedback on principles executed correctly and incorrectly. Design issues for the development of an ITS for the FCS domain also include the need to facilitate scenario authoring, and the objective of providing a flexible architecture that can switch between real-time feedback during scenario execution versus strictly after action review. This proof-of-concept system aims to provide a foundation for future training systems based on the same architecture, but supporting team training on multiple scenarios with multiple simultaneous participants.

ABOUT THE AUTHORS

Randy Jensen is a project manager and software engineer at Stottler Henke Associates, Inc. He has developed numerous intelligent tutoring systems for Stottler Henke, as well as authoring tools, simulation controls, and assessment logic routines. He holds a B.S. with Honors in symbolic systems from Stanford University.

Henry Marshall is a Principal Investigator on Mounted Embedded Simulation Technology at the Research, Development & Engineering Command (RDECOM) Simulation Technology Center (STC). Prior to this assignment he worked at the Simulation, Training and Instrumentation Command (STRICOM) where he spent 11 years as lead for the CGF/SAF, HLA and Linux Porting developments on the Close Combat Tactical Trainer (CCTT) system in addition to being a OneSAF team member. His twenty years with the Government have been mainly in CGF and Software acquisition. He received a BSE in Electrical Engineering and an MS in Systems Simulation from the University of Central Florida.

Jeffrey Stahl is a Principal Investigator on Mounted Embedded Simulation Technology at the RDECOM-STC. Prior to this assignment, he worked for 2 years on the Advanced Robotics Simulation STO. He holds a B.S. in Computer Engineering and is working on his M.S. in Engineering Management at the University of Central Florida.

Richard Stottler co-founded Stottler Henke Associates, Inc., an artificial intelligence consulting firm in San Mateo, California in 1988 and has been the president of the company since then. He has been principal investigator on a large number of tactical decision-making intelligent tutoring system projects conducted by Stottler Henke, including projects for the Navy, Army, and Air Force. Currently, he is working on an intelligent tutoring prototype for the future combat system control vehicle, funded by the US Army STRICOM. He has a master's degree in computer science from Stanford University.

An Intelligent Tutoring System (ITS) for Future Combat Systems (FCS) Robotic Vehicle Command

Randy Jensen, Dick Stottler
Stottler Henke Associates, Inc.
San Mateo, CA
Jensen@StottlerHenke.com
Stottler@StottlerHenke.com

Henry Marshall, Jeffrey Stahl
Simulation Technology Center, US Army Research, Development and Engineering Command
Orlando, FL
Henry.A.Marshall@us.army.mil
Jeffrey_Stahl@peostri.army.mil

INTRODUCTION

The Future Combat System (FCS) has identified training to be a Key Performance Parameter (KPP). The approach to providing training has been identified to be a fully embedded approach, making the system operations and training modes interchangeable. Since the training facilities of the present will be replaced by troops training in motor pools and doing mission rehearsal in assembly areas, the need to provide robust training packages becomes much more important. Much of the utilization of major virtual training devices such as the Close Combat Tactical Trainer (CCTT) is centered on platoon level task training. Much of this training is performed in CCTT using structured training packages. In a domain with small focused task based exercises, there are a limited number of outcomes in scenario execution. This reduces the development cost to construct and annotate scenarios to be used by automated after action review mechanisms in an Intelligent Tutoring System (ITS). In addition, performing these exercises in an embedded ITS would greatly enhance and standardize training, making it available anywhere a unit is deployed. It could also increase the use of Advanced Distributed Learning (ADL) reach back capabilities for exercises and logging of training performance.

To explore the use of an ITS system in this context, a task area under the Army's Future Combat Systems (FCS) concept was selected as the implementation candidate. The FCS warfighter manning a Control Vehicle (CV) crewstation must remotely control robotic platforms to perform reconnaissance and engage the enemy. The concept of remote control of robotic assets is a major technology addition over the current legacy capabilities that will be provided by FCS. The operator must maintain situational awareness and apply tactical decision-making principles in a heightened information-rich setting with distributed vehicles and sensors under his command. This paper discusses a proof-of-concept ITS developed to provide scenario-based practice for the FCS soldier. The ITS is integrated with an FCS Embedded Training / Mission Rehearsal (ET/MR) Testbed, configured to represent an FCS CV.

TRAINING REQUIREMENTS

The broad objective for the ITS is to approximate the experience that a student would receive in working one-on-one with an instructor who uses sound teaching strategies. In a review of several ITS projects evaluated for training effectiveness, Dede and Lewis (1995) found that "there is ample experimental evidence that well designed, well developed ITSs can train very effectively." In a scenario-based domain, an ITS enables high-fidelity simulations to be used toward training objectives without a human instructor present, by automating the process of monitoring student actions and providing feedback, either in real-time or in after-action review. This proof-of-concept provides a demonstration of the use of ITS technology to provide students with individualized experience in employing the FCS concept of operations. Although FCS doctrine has not been completed, it is advantageous to design the embedded training systems in parallel with the advanced tactical concepts they will exercise. As a network-centric concept, FCS makes use of unmanned platforms which can provide an array of significant battlefield advantages for assault, reconnaissance and logistics, but only if the human warfighters have the ability to maximize the use of these assets.

This is echoed by Enhanced Embedded Training (Faber, 2001), "The training of Warfighters, responsible for using complex weapon systems in combat, is increasingly challenging. The knowledge required to operate these systems effectively is very complex and changes very rapidly. Complexity is driven by several factors: a growing richness of features, combinations of interactions among a growing number of system components, and a growing range of operational scenarios that must be handled." With practice in the simulated crewstation environment provided in the ET/MR Testbed, students can open a scenario, go through an exercise, and receive customized feedback on their application of the FCS concept of operations.

In this context, a limited principle hierarchy serves as the instructional basis for the training system and the automated evaluation of student actions in an FCS scenario. In addition to the fundamental tactical principles, students are also monitored for their mastery with the task of translating tactical intentions to robotic commands correctly executed in the control interface. The ITS observes the student's actions and performance in a simulated scenario and produces specifically tailored feedback on principles the student has executed correctly and incorrectly.

The ITS tutors on three types of principles or skills:

A. Tactical Decision-Making
These principles involve the ability to interpret the tactical situation and commander's intent, and decide what should be done. This can include decisions such as the choice of a vehicle formation, the correct interpretation of sensor data in maintaining accurate situational awareness, and the determination of correct responses to different forms of enemy threats encountered in the course of an exercise.

B. Command Formulation
Principles in this category involve translating tactical decisions (category A) into commands or orders that can be issued within the simulation via the control software. For this domain, the category B principles primarily involve the formulation of commands for controlling unmanned robotic vehicles.

C. Execution
Execution principles are roughly equivalent to an understanding of the "buttonology" of the control software used in the embedded setting and in the Testbed. The student must be able to correctly use the operator interface of the software to implement commands formulated as a result of category A and B decisions.

For the proof-of-concept system, the training emphasis is primarily focused on the use of unmanned platforms for reconnaissance. Since there is no completed existing doctrine for the FCS concept, there was no readily available set of principles or scenarios with well-defined evaluation criteria. In order to provide a proof-of-concept with a meaningful demonstration of technology, an existing training scenario from a previous experiment was selected for adaptation in this context. Thus, the specific principles and concepts embodied in the proof-of-concept are similar to what may be developed as FCS tactical doctrine, but do not represent an attempt at instructional accuracy with respect to the development of new tactics for the FCS concept. Rather, the tactical principles applied in the proof-of-concept serve to illustrate capabilities that can be implemented in a full-scale system to be developed in parallel with the FCS doctrine itself. Given this context, the ITS provides students an opportunity to gain practice selecting a formation suited to the task and terrain, identifying threats and targets through extended sensor capabilities, and responding to different tactical situations.

The ITS requires two primary sets of data during an exercise. First, the ITS must get real-time information about the state of the simulated exercise in order to assess the conditions under which the student performs different actions. Data in this category includes information about vehicle locations, headings, control status and sensor input, as well as the outcomes of contact with enemy forces. Outcomes can provide useful feedback to the student as to the appropriateness of decisions during an exercise. However, simulation outcomes are not always relevant to student feedback, as a free-play simulation allows for a degree of flexibility where negative outcomes may occur even if the student has performed well on all or most relevant principles. The second category of data involves the student's activities within the software. This second category is particularly helpful in diagnosing mistakes that the student has made, since it reveals useful information about the student's intentions.

TESTBED INTEGRATION

The ET/MR Testbed consists of two crewstations representing robotic control stations in an FCS vehicle. The embedded simulation component includes the OneSAF Test Bed (OTB) as the driver for computer generated forces. The primary control environment that the student uses is an Operator Control Unit (OCU) that functions as the control interface for networked robotic vehicles under the student's command. The OCU operates directly with OTB to control and monitor status for robotic entities under the trainee's control. Other modules in the ET/MR Testbed include a variety of visualization and situational awareness interface tools, which are essential to the high-fidelity simulation experience, but which the ITS does not need to monitor or interface with.

The ITS is tightly integrated with the simulation and control elements of the Testbed, not only to monitor events as they occur during an exercise, but also to publish messages which the Testbed uses to display feedback in its native user interface, both during and after execution as appropriate. The majority of feedback comes in after-action review at the completion of the scenario, but occasionally it is necessary to provide a hint or immediate feedback, in order to prevent the student from continuing too far with an incorrect decision. Examples of this are given in the following section.

ZONE RECONNAISSANCE SCENARIO

As discussed earlier, the absence of existing tactical doctrine for the FCS concept should not prevent the initiation of design work on the embedded training systems that will be developed in more depth when doctrine for the domain becomes more mature. However, this presents a challenge for an early stage proof-of-concept ITS which attempts to demonstrate embedded training functionality as it can be developed for the FCS. This challenge was addressed by adapting an existing training scenario which involves tasks and decision making skills which are similar to those that apply in the FCS domain. The scenario selected as the basis for this system was an Unmanned Combat Demonstration (UCD) scenario previously developed as part of a combined operational experiment. The UCD scenario involves a zone reconnaissance task to be performed with two robotic vehicles, with the purpose of measuring student times, accuracy, and multitasking abilities. Although this scenario was developed with realism as an objective, the tactical decision making requirements are not based on any specific existing doctrine. A zone reconnaissance scenario is an effective candidate for adaptation with the demonstration ITS for a number of reasons. The reconnaissance task is well suited to the employment of robotic vehicles under the FCS concept, as maneuvers with greater exposure to enemy threats represent a good use for unmanned vehicles, and therefore will be a common activity for the FCS soldier in a command vehicle. Also, in the performance of this task, it is possible to clearly distinguish the three skill categories – tactical (A), command formulation (B), and execution (C) – which are relevant for this domain.

The scenario is designed for FCS command and control vehicle crew members, who must use their distributed assets to gain detailed information about routes, terrain, and enemy forces within a zone with well-defined boundaries. This is a common preparatory task assigned by a commander before sending main body forces through a zone. For the purposes of this scenario, the primary objective is to determine the locations of enemy threats, and possibly neutralize a threat if the situation warrants it. Before beginning the scenario, the student is given an initial briefing which describes the zone boundaries, terrain, expected enemy locations if any, any critical features such as crossings or obstacles, and additional instructions reflecting the commander's intent. In this exercise, the student has two Armed Robotic Vehicles – Reconnaissance (ARV-R) under his control, which he is tasked with directing through the Operator Control Unit software.

Upon entering the zone, reconnaissance vehicles typically are employed in a dispersed formation, to maximize the area of surveillance, while still maintaining the ability to provide fire support to each other. The formation may potentially involve respective positions outside of direct line of sight contact, as long as their sensor packages overlap near their maximum ranges. The ARV-Rs are able to operate with a degree of autonomy in terms of specific route selection through terrain, but human oversight is still necessary, especially when contact is made with an enemy.

To illustrate the ITS observation and evaluation mechanism, it is helpful to discuss a specific task that the student must perform in more detail, and step through the logic that the system applies. To take a simple example, a common task is to employ the robotic vehicles in an appropriate formation. In this scenario, the student should maintain a V-formation with his ARV-Rs. This task embodies three kinds of principles corresponding to the three categories described earlier. In a tactical sense (category A), the student should be able to determine that a V-formation is fitting for a zone reconnaissance task to correctly carry out the commander's intent as described in the introductory briefing. In the command formulation sense (category B), the student must understand how to construct missions that can be assigned to individual vehicles so that they will assume a V-formation. In the execution sense (category C), the student must be able to correctly issue appropriate commands in the OCU so that they result in a V-formation.

In practice, the student should set up the V-formation by establishing defaults for each robotic vehicle. This takes advantage of their autonomous navigation capabilities and minimizes the real-time control demands on the C2 crew members. Figure 1 shows a V-Formation correctly configured using the OCU software.

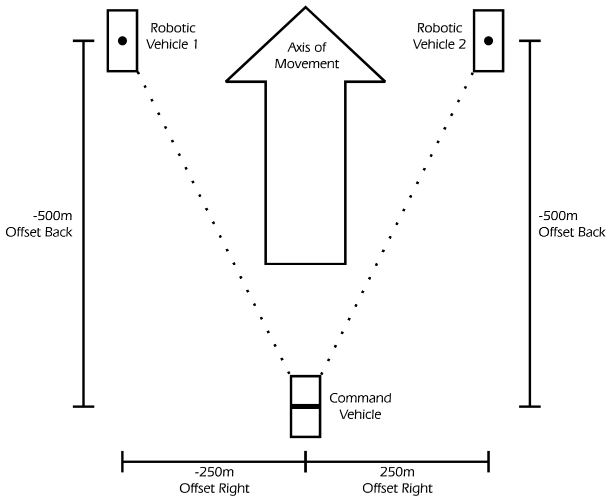

Figure 1: Correct V-Formation

The OCU commands are to specify the Offset Back of each vehicle to -500m, and to specify Offset Right of 250m for one vehicle and -250m for the other. With this sample task, there are three likely cases:

Case 1: The student doesn't set any defaults. If this is his first experience with this principle, the ITS proactively suggests he set defaults and gives a brief explanation of how this is done. If the student has previous experience setting these types of defaults and is usually successful, this latter advice is not presented. If he has previously had problems setting defaults, then a more detailed step by step tutorial is presented, because the student is exhibiting problems with the execution (category C) principle of using the software user interface to set defaults. Continued failure would cause the ITS to schedule the student for specialized practice, outside of any scenario, after the scenario completes. If the student sets the defaults correctly after prompting, then he understands the tactical decision making (category A) principles of using a V-formation for zone reconnaissance in accordance with the commander's intent, and understands the principles relating to formulating commands (category B) with parameters that are correct within some error tolerances, as well as the corresponding execution (category C) principles. Alternatively he might set the defaults with incorrect values, which triggers the second major case.

Case 2: The student sets default offset values for the robotic vehicles using the correct procedure, but sets them to the wrong values. This implies that one or more of the following conclusions can be made about the student's conceptual understanding of the principles underlying this task.

i. The student has a poor understanding of the tactical benefit of a V-formation in performing the reconnaissance task.

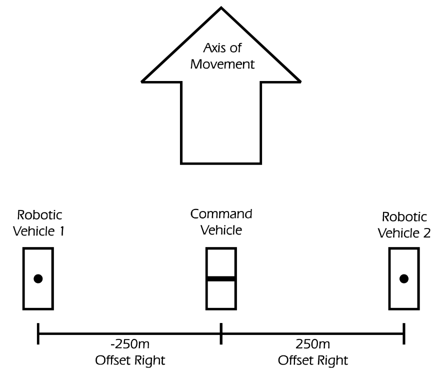

Figure 2: Incorrect Formation

In Figure 2 above, the student has chosen a simple line formation, which is not the most effective tactical approach for this scenario. With the line formation, the Command Vehicle is overly exposed and does not make sufficient use of the robotic vehicles in a forward position for reconnaissance.

ii. The student made the correct tactical decision to order the ARV-Rs to maintain a V-formation, but either did not understand what this entails, or did not understand how to construct orders that would accomplish this.

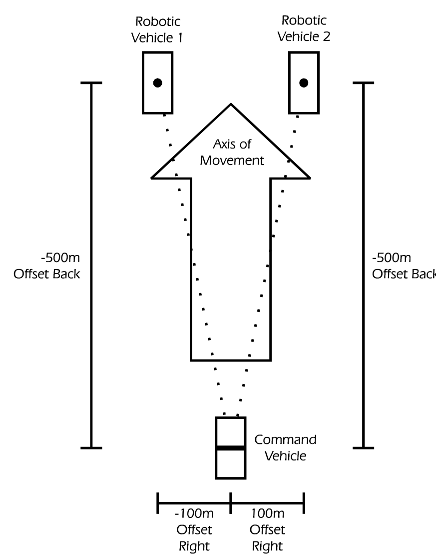

Figure 3: Poorly Constructed V-Formation

In Figure 3 above, the student has made the right tactical decision by employing a V-formation. However, the student has constructed this poorly, resulting in a narrow V. With the two robotic vehicles in close proximity to each other in their default offset positions, the student is wasting sensor resources by having too much overlap in the area of surveillance for each. Instead of 100m offsets, the student should have chosen values of 250m.

iii. The student understands both the tactical and command formulation elements of the task, but failed to correctly carry out the execution of appropriate orders in the OCU software environment. For example, a student may not understand that an "in front" distance is represented by a negative "behind" distance.

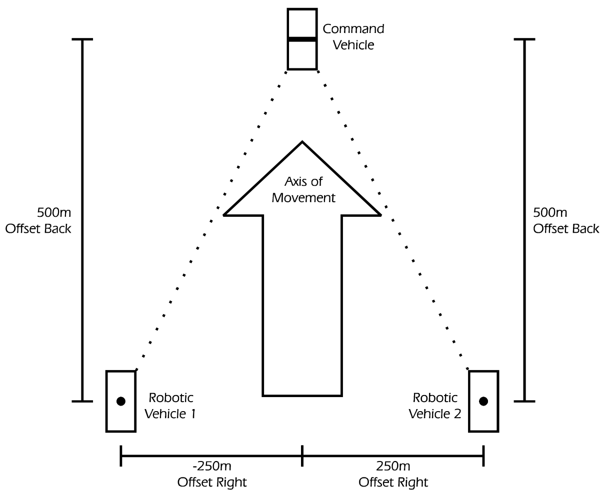

Figure 4: Incorrect Parameter Entry

In Figure 4 above, the student has chosen the correct formation and offset distances, but has made a small operator error in forgetting to use a negative value for the Offset Back field. This results in an inverted V-formation, which has the opposite of the desired effect for the reconnaissance maneuver, making the Command Vehicle the lead element instead of the robotic vehicles.

In the debriefing, the ITS reacts to this case with a simple interactive dialog which aims to determine the source of the problem by questioning the student. In the first question about the student's tactical plans, if the ITS determines that the student was trying to set up the wrong formation, it explains the commander's intent and rationale behind the V-formation. If the ITS determines that the student intended to issue orders for a V-formation, but doesn't have the correct concept of how this formation is expressed in terms of instructions to units, then the ITS gives feedback about proper distances in an effective V-formation. If the student had the right formation and command parameters in mind, but failed to correctly issue orders in the software accordingly, then he receives a description of the relevant procedure for doing this.

Case 3: The student sets defaults correctly. The scenario debriefing includes an entry noting the student's successful decisions and actions. The ITS records the student's successful experience with all of the principles involved (which include principles from categories A, B, and C defined above) and the ITS would increase its estimate of the student's mastery of these principles.

This discussion of a portion of the scenario illustrates the kinds of principles and the forms of customized performance feedback that the ITS applies in this domain. There are many examples of more complicated performance assessment associated with other principles such as the student's real-time decision making in situations where enemy contact is made. The student may respond to enemy contact by attacking with one or more vehicles, calling for indirect fire, changing the maneuver parameters (offsets, speed, heading), seeking defilade positions, withdrawing one or more vehicles, or numerous other approaches. The tactical principles that apply all fit into one of the three categories described for this system. The conditions for determining how these principles apply in a specific situation are therefore represented in the ITS evaluation mechanisms, with the objective of maintaining the ability to reconcile multiple possible approaches to a situation which may all satisfy a given principle. As the underlying simulation is a free-play environment, the student is free to do things that the ITS does not anticipate. This motivates the structured dependence on the principle hierarchy at the core of the ITS architecture, so that the ITS can watch for conditions in the simulated exercise that are directly relevant to training objectives, and essentially ignore simulation events that are not relevant.

Along with issues related directly to actions performed by the student or events in the scenario, the tactical principles also incorporate factors related to terrain, enemy force levels, and so forth. For example, in carrying out the reconnaissance task, one principle is to make maximum use of terrain concealment in the determination of routes. This principle has a corresponding evaluation mechanism, which assesses the terrain in addition to the locations of vehicles and scenario events.

EVALUATION MECHANISMS

Given the flow of data from the ET/MR Testbed to the ITS, the ITS must have a complex evaluation mechanism which can properly process events and situational data to reach conclusions about the student's understanding of different principles. This is especially challenging in a real-time free-play scenario, because even with just one scenario, different students executing different commands and actions, or executing similar commands at different times, can arrive at dramatically different kinds of situations. The more structured a domain is, the simpler it is for an automated system to identify what steps are tactically correct for a given situation. With a free-play simulation, the converse becomes the reality: it is much more difficult to define abstracted methods to recognize instructionally significant states in the simulation without using scenario-specific inputs. A finite state machine (FSM) based approach is highly effective for evaluating student decisions in free-play simulations, and thus this is the core of all student evaluation in the proof-of-concept system.

Consider an example in the scenario described in the previous section. In the course of the zone reconnaissance task, one of the robotic vehicles (ARV-R2) detects an enemy, consisting of two Russian BTR80s, which open fire. The fact that they open fire reveals that they have also detected the ARV-R2, and the correct response developed in this scenario is to counter-attack with direct and indirect fire, using ARV-R2 to lay down direct fire while the other robotic vehicle ARV-R1 makes a flanking maneuver to kill the enemy. There is a corresponding principle in the principle hierarchy, corresponding to the tactical decision about whether or not to attack in a given situation. This principle has a corresponding FSM for evaluation purposes. It should be noted that this tactical decision making rule is an artifact of the UCD scenario, and may not apply in the eventual FCS doctrine. The objective of the proof-of-concept ITS is not to attempt to conclusively develop or demonstrate FCS tactics, but rather to demonstrate a functioning automated trainer with a representative set of principles that can be modified and augmented in a full-scale ITS for the FCS. Also, there are other relevant tasks that are significant in this example, and which are reflected in other principles with their own FSMs for handling the assessment of whether the student correctly understands their application. However, for this example, we will limit the discussion to a walk-through of a simplified version of the "Understand when to attack" principle and evaluation FSM.

The evaluation FSM is indexed to the relevant principle through a scenario definition interface. This is a generalizable task, which can easily be delegated to an instructor or subject matter expert for training systems with a plurality of exercise scenarios. Thus, when the exercise is executed in the simulation, each evaluation FSM for each principle relevant to the scenario is activated. The proof-of-concept system uses an FSM-based approach which consists of a number of enhancements to the traditional FSM model. In particular, the evaluation machines are composed using a visual authoring environment for this kind of purpose (Fu, Houlette, and Jensen, 2003), which supports multiple chained conditions between states, local and global variables, and the organization of different FSMs in a hierarchically nested structure.

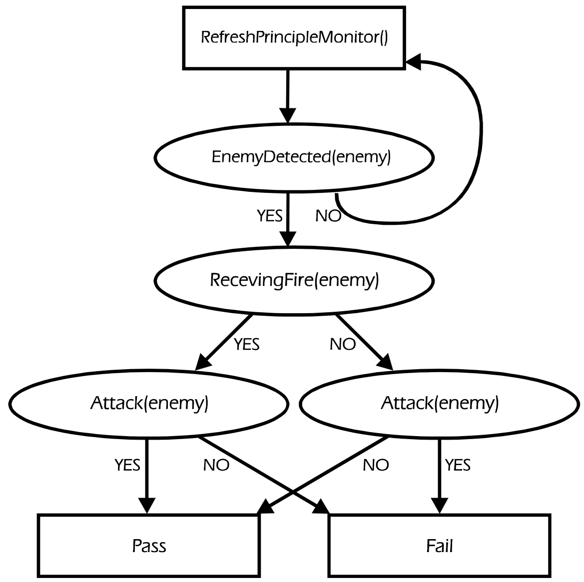

Figure 5: Simplified Evaluation FSM

Figure 5 shows a simplified high level representation of the FSM logic for the evaluation of the "Understand when to attack" principle. The initial condition of the FSM is a persistent monitoring state. As soon as the condition becomes true that one of the vehicles under the student's command detects an enemy, then the FSM initiates the logic for determining whether it is appropriate or not to attack. If the friendly vehicle is receiving fire from the enemy that it has detected, then this means that the enemy is also aware of the friendly vehicle. Therefore, the student should respond by counter-attacking the enemy, and the Pass / Fail determination for this principle is based on the student's decision in this situation. Likewise, the converse is represented as well in the FSM diagram above. In the actual system there is a host of other factors that are relevant to the tactical decision about whether to attack, such as the determination of what kind of enemy has been detected, what indirect fire assets are available, whether any other threats with potentially different priority levels have been identified in the vicinity by other vehicles, and so on. These factors are omitted in this example for the sake of illustration.

When the Pass or Fail outcome is reached in an evaluation FSM, the ITS records this with the student model, along with the situational data about the circumstances where this occurred, so that feedback may be constructed accordingly. This includes a record of specific transitions that were activated within FSMs in the course of events that led up to the current Pass/Fail outcome. Because the ITS evaluation FSMs update states and trigger transitions as events occur in the Testbed, they support both a real-time feedback mechanism and an after-action mechanism that uses logging capabilities. When instantaneous feedback is warranted or requested, especially in the case of certain procedure-related principles, then the evaluation FSM itself may trigger an immediate message to be published out to the student via the Testbed user interface. This applies on a case by case basis with different kinds of principles. An example of a procedural principle for which real-time feedback is appropriate would be to remember to use both robotic vehicles. If the simulation execution has continued for some time without the student issuing any commands to one of the ARV-Rs, then an FSM triggers a message to remind the student that he has not made use of all his robotic vehicle assets. This is accomplished by the use of messaging capabilities developed within the Testbed user interface and exported via an API as callable primitives that can be used in FSMs constructed on the ITS side of the interface. In other cases, the instructional conclusions from the FSMs are recorded in the student model and used to construct the after-action review.

FUTURE WORK

Typically a full-scale simulation-based ITS contains a case-base of predefined scenarios which are indexed to a principle hierarchy. When the student engages in exercises, the record of his performance is maintained in a student model. This is used to make instructional decisions for the automated selection of appropriate future exercises based on the tutoring system's estimation of the student's mastery of individual principles. Remediation through practice in exercises that require the skills associated with key principles is an important component of embedded training, and thus a primary design goal for a full-scale tutoring system. The demonstration ITS is designed to be compatible with the addition of further scenarios to be developed in the future and indexed with the same principle hierarchy. Thus, with an extensible architecture, the demonstration ITS both illustrates the concept for the FCS domain and also can be directly built upon for future implementation. This extensible architecture comes as a result of the use of an ITS authoring tool to streamline the process of constructing a principle hierarchy, associating principles with both general and scenario-specific evaluation routines, associating scenarios with principles, and applying instructional logic for remediation.

With the development of additional exercise scenarios, an important area of future research involves the simplification of the scenario authoring process. In particular with the evaluation mechanisms associated with principles in a free-play scenario, we expect to facilitate the process for both subject matter experts and knowledge engineers by employing a visual authoring tool for constructing evaluation FSMs. However, we anticipate considerably more upfront design effort in developing abstracted evaluation machines which can be applied in different situations in different scenarios, and still draw appropriate conclusions about student strengths, weaknesses, and decision-making concept understanding. But the result of this effort would be a set of reusable FSMs which can be applied to new scenarios either automatically or with parameterization or with minor customization.

This ITS proof-of-concept will be useful not only for the next step of full development of a trainer for the domain with a complete set of training exercises, but also for future extensions. One relevant extension is the transition to team and leader training applications. The FCS concept applies both at an individual and team level, and indeed in the battlefield this concept of operations will allow for different FCS control vehicles to hand off control of robotic vehicles as necessary, which introduces a further set of tactical and coordination principles that will be new for trainees. The OTB simulation, Testbed and control software will support the team training context with relatively small transition cost, and thus the automated instruction capabilities of the ITS can be leveraged even further through the identification of performance standards.

THE ROAD AHEAD

Successful demonstration and evaluation of the proof-of-concept ITS technology will result in integration into the Tank and Automotive Research and Development Command (TARDEC) Crew instrumentation and Automation Testbed which has an embedded simulation component as part of their Vehicle Electronics (VETRONICS). This effort will also provide the One SAF Objective system (OOS) valuable information on how to support ITS interfaces, and provide guidance for a composition of OOS that supports ITS. Future ITS work could also focus on the injection of a command intelligent agent that would role-play the company or platoon commanders. This is very valuable since these persons are typically not available for many training sessions. This would provide the trainee commands based on their actions and progress.

In conclusion, the FCS Lead Systems Integrator (LSI) has shown significant interest in making ITS a part of the embedded training packages. With the concurrent LSI contract and the continued development of the Unit of Action operational concept, standards for performance can be identified and applied to an Intelligent Tutoring System. The ITS will allow embedded training to reach its maximum potential as a technology for the Objective Force. Having soldiers who can make effective, rapid decisions on the battlefield will increase the survivability and the lethality of the Objective Force.

REFERENCES

Burmester, G., Stottler, R., & Hart, J. L. (2002) "*Embedded Intelligent Tutoring Systems (ITS) for the Future Combat Systems (FCS) Command and Control (C2) Vehicle*", In proceedings of I/ITSEC 2002.

Dede, C., & Lewis, M. (1995) "*Assessment of Emerging Technologies That Might Assist and Enhance School-to-Work Transitions*." Washington, DC: National Technical Information Service.

Faber, Terry D., (2001) "*Enhanced Embedded Training (EET), What is it? What is it good for? How do we get it?*", prepared for the 3rd Annual Testing and Training: A National Partnership Symposium and Exhibition, Army Training Support Center (ATSC), 22 October, 2001.

Fu, D., Houlette, R., & Jensen, R. (2003) "*A Visual Environment for Rapid Behavior Definition*", In proceedings of the 2003 Conference on Behavior Representation in Modeling and Simulation, Scottsdale, AZ.