Virtual Combat Training Center (V-CTC): An Intelligent Tutoring System + Tactical Simulation

William R. Murray, Michelle Sams
Teknowledge Corporation
Palo Alto, CA
wmurray@teknowledge.com, msams@teknowledge.com

ABSTRACT

PC-based tactical simulations provide war-gaming environments but do not deliver training any more than would a Combat Training Center with only engagement feedback (e.g., ammunition expended, number of kills). For maximum learning to occur, human facilitators are needed to provide tailored guidance and feedback. For simulations, intelligent tutoring systems (ITS) may perform many of the same tutoring and AAR functions of the observer/controllers. However, the time and expense to develop and incorporate an ITS into existing simulations has been a deterrent. A more promising approach is to develop a reusable ITS module that can 'plug-into' existing simulations. This approach could leverage the millions of dollars spent on developing simulations and turn them into more effective training environments with intelligent virtual coaches. The DARPA DARWARS program is funding V-CTC to demonstrate this potential.

V-CTC is designed to be plug-compatible with existing PC-based tactical simulations, requiring only minimal developer modifications. V-CTC 'observes' user GUI actions via software connectors. An event data stream sends internal information from the simulation via sockets. These events trigger rules that make tactical inferences, detect violations of doctrine, and determine tutorial strategies. The tutor can pause the simulation to probe user reasoning, resolve ambiguities in user intentions, or provide timely coaching and mini-lessons. V-CTC also provides individualized AARs regarding tactics, techniques, and procedures. The tutor performs deductive reasoning (not fixed decision trees) on an ontology-based knowledge representation. This approach supports deeper reasoning and promotes reusability and extensibility. Customization is supported by a Bayesian student model consisting of performance, knowledge, and self-assessment measures. The initial demonstration for the V-CTC concept is a combined-arms warfare tutor at the battalion and company level and is demonstrated with an adapted version of the high-fidelity tactical simulation, *Armored Task Force*.

ABOUT THE AUTHORS

William Murray is the Principal Investigator in Teknowledge's Training Systems Group. He has over twenty-five years of experience developing intelligent tutoring systems, performance support systems, pedagogical agents, conversational agents, and web-based guides and related technologies such as natural language and program analysis systems. His research activities have included natural language understanding, Bayesian student models, dynamic instructional planning, architectures for intelligent tutoring systems, and the use of program verification techniques for program analysis in ITS. He is a reviewer for the International Journal of Artificial Intelligence and Education. Bill received his Ph.D. in Computer Science from the University of Texas at Austin.

Michelle Sams is the Program Manager of the Training Systems Group at Teknowledge Corporation. Projects include knowledge-based intelligent agents for tutoring and performance support, web-based guides and learning assistants in virtual environments, natural language dialog agents, and computer-based training systems. She has over eighteen years of experience in the areas of training research, human factors, and advanced technologies for training and performance support systems. Prior to joining Teknowledge, Dr. Sams worked for the U.S. Army Research Institute for the Behavioral and Social Sciences (ARI). She has a Ph.D. in Experimental Psychology from New Mexico State University.

Virtual Combat Training Center (V-CTC): An Intelligent Tutoring System + Simulation

William R. Murray, Michelle Sams
Teknowledge Corporation
Palo Alto, CA
wmurray@teknowledge.com, msams@teknowledge.com

THE NEED AND OPPORTUNITY

Army training in most FORSCOM units revolves around preparing for and participating in Combat Training Centers (CTC). At these centers, the troops fight against intelligent, experienced live opponents using equipment and tactics of enemy forces. The CTCs provide invaluable live training opportunities, however these training experiences are limited due to their availability and expense (Chatham and Braddock, 2001). For example, in a typical National Training Center (NTC) rotation, there is only enough time to practice two to three missions on offense and on defense. It costs up to $10 million for a 28-day rotation (NTC, 2002). Commanders are often reassigned to new positions after a CTC rotation, leaving the new commander of the unit to learn anew what his departing predecessor has just learned.

Due to these limited resources, it is critical that the training experiences at these CTCs be optimized. The benefits of these live training experiences can be extended by providing low-cost, readily available, realistic, and relevant PC-based training prior to CTC rotations to better use the time there, and subsequent to rotations, to enhance retention and allow for in-unit follow-on training that builds on what has just been learned. This supplemental training also allows much greater time on task, enabling trainees to spend more time developing a wider range and more in-depth tactical skills with increased automaticity and confidence in applying those skills.

Simulations and games that apply to Defense needs are beginning to proliferate, some from the DoD and some from entertainment. Developers who create high-fidelity military simulations suitable for training purposes are typically domain experts who do not have the interest or resources to add significant training (if any) to their simulations beyond how to use the simulations. If the simulations are used in a training context, they are typically employed in a similar manner as the live training centers. Basic aggregate data (e.g., number of kills) is collected and diagnostic evaluations are conducted by human observers.

A segment of the technical training community focuses on development of intelligent tutoring systems (ITS). An intelligent tutoring system is a knowledge-based system that models knowledge of the domain, the instructor, and the student. An ITS can be used to model aspects of the tactical knowledge of an experienced commander and the training strategies of observer/controllers and instructors. The modeling of the trainee's knowledge, called student modeling in an ITS, allows instruction to be tailored to the user.

A tactical simulation and ITS provide complementary capabilities. The simulation provides an engaging and realistic war-gaming environment that motivates users to spend time honing their skills. The ITS provides a continuously available virtual observer/controller who provides judicious guidance and instruction during the simulation and a diagnostic AAR tailored to the individual's strengths and weaknesses. As a virtual coach, the ITS could improve the training effectiveness of tactical simulations. Furthermore, if the tutor were designed to 'hook into' rather than integrate into simulations then it would offer considerable cost savings. The tutor could be reused for other similar simulations (see Figure 1).

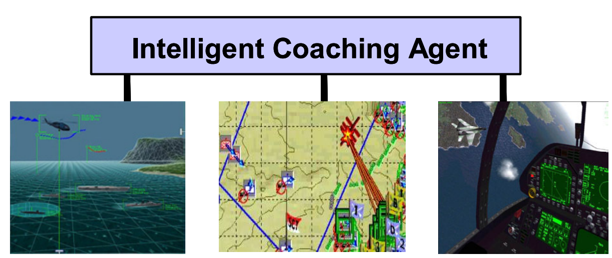

Figure 1. A virtual tutor that plugs-into simulations.

VIRTUAL COMBAT TRAINING CENTER

The goal of the DARPA DARWARS-sponsored Virtual Combat Training Center (V-CTC) is to increase the depth of knowledge and expertise acquired by commanders and staff officers. It does this by promoting expert ways of thinking. It teaches the leader to model the battlefield in his mind, analyze the situation against doctrine, make doctrinally sound decisions, and rehearse these skills in a variety of situations until they become automatic.

This approach will be demonstrated with an intelligent tutor that plugs into an existing high-fidelity tactical simulation of combined-arms warfare at the battalion and company level, called *Armored Task Force* (ATF) (Proctor, 2004).

The tutor system will be hooked into the simulation using sockets and software connectors, rather than requiring extensive simulation modifications. The tutor's interpretation rules infer user and higher-level tactical events from the lower-level simulation event stream. Constraint rules detect violations of Army doctrine. V-CTC may query the user if there is ambiguity about the user's intentions. The user, in turn, can ask certain kinds of questions about the domain and receive answers from the tutor.

The tutoring system is separate from the simulation and portable. The main components (student model, domain knowledge, and tutor strategies) are modular. We expect that the V-CTC could be used to interpret other similar tactical simulations. This may require some changes in the ontology and knowledge base to support the new domain and minor developer modifications in the simulation.

Tutoring System

Most computer-based tutoring systems build student models based on recognition exercises, such as multiple-choice, drag-and-drop, and fill-in-the-blank exercises. While these types of measures are easily collected, they do not provide a full assessment of student state. Not only is it important to have *knowledge about* a domain, but one must also be able to *apply those skills* necessary to perform the tasks and do so with confidence. This tutoring system will build a skills and knowledge student model within the context of realistic simulations.

The user plays the role of a task force (BN) commander in a tactical simulation at NTC while the tutoring component emulates an observer/controller (O/C). This virtual coach provides guidance during the simulation. It may point out violations of Army doctrine, provide guidance to improve understanding of the battlefield situation and making better tactical decisions. The frequency and type of guidance is based on the user's self-assessments and the evolving user model based on performance and knowledge measures (see Figure 2). The tutorial strategy attempts to balance the benefits of discovery learning (committing errors and seeing the consequences) with directed learning (judiciously timed guidance to avoid extensively error-filled practice).

The tutor builds a user model based on a variety of measures. The user model is initialized by user profile information (provided by user or instructor) and user self-assessments regarding key skill areas. The model evolves as performance measures are collected and interpreted based on actions during the simulation. Knowledge measures are gathered from user responses to tutor questioning at critical decision points and events. A Bayesian analysis takes these various measures and forms a student state model consisting of knowledge, skills, and confidence.

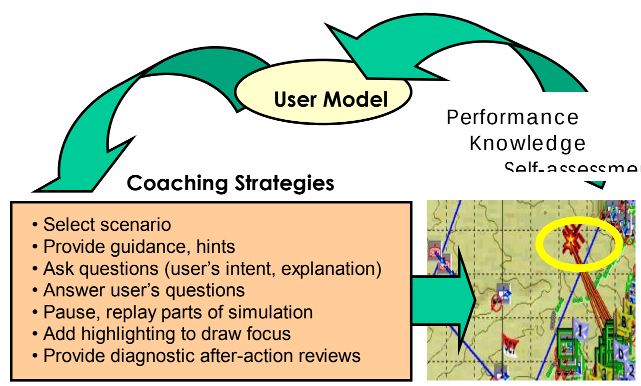

Figure 2. Tutor interactions are tailored to individual.

At the end of the simulated mission, the tutor provides a diagnostic AAR about the individual's performance regarding tactics and doctrine – not just simply a recap of kills and ammo expended. The tutor also provides a comparison of the user's initial assessment of his skills versus the tutor's assessment based on the user's performance. The objective is to calibrate the user's self-assessments so that confidence accurately reflects ability. After the AAR, users can opt to see an expert's solution (a playback of the mission), along with commentary.

The tutor will 'know' what is happening in the simulation by parsing and interpreting low-level event stream data passed to it by the simulation. Evaluation of user actions in the simulation is performed by deductive reasoning over a domain knowledge representation. The knowledge representation is backed by a standard upper ontology to support deep, common sense reasoning about general concepts (e.g., time, terrain). The domain-specific knowledge base provides the tutor with an expert level of active knowledge of tactical concepts and rules rather than fixed decision trees or hard-coded procedures.

Communication is facilitated by a natural language dialog capability. The tutor can ask probing questions and evaluate the trainee's answer. The user can directly ask certain kinds of questions (e.g., about weapons, enemy tactics, and battlefield calculus) and get specific answers.

The tutoring system will be designed with a generic, modular architecture to enable re-use of the key components for other applications. Re-usable components include the student state model, tutoring strategies, domain knowledge representation, a methodology for connecting the tutor and simulation, and the natural language understanding system. The extent of re-usability will, of course, depend on the degree of similarity to the original domain. For example, a tactical simulation at a similar echelon level with similar weaponry would have high overlap and a sea-based simulation would require extensive changes for the knowledge bases.

Simulation

The Virtual Combat Training Center concept will be demonstrated with a high-fidelity tactical simulation of combined-arms warfare at the battalion and company level, called *Armored Task Force* (ATF). ATF is the successor to *Brigade Combat Team* (BCT), both developed by ProSIM Company. BCT was an innovation in that it provided most of the fidelity of JANUS (a simulation used extensively at the Command and General Staff College) but eliminated the need for high-end workstations or controllers to interpret simulation commands. BCT has been used for training at the Joint Readiness Training Center (JRTC), in their Leader Training Program (LTP).

The ATF game pits a friendly force of up to battalion/company size against an enemy force of up to brigade/regimental size in simulated combat. ATF allows a user to take the role of the friendly forces while it plays the opposing force (OPFOR). In each scenario, the user must operate using specified assets in line with the mission objectives and constraints. ATF randomly selects from multiple enemy course of actions (COAs) stored with each scenario. However, with V-CTC, the COA will be selected by the tutor to target an individual's learning needs.

The user manipulates NATO-standard icons that represent companies, platoons, or sections (see Figure 3 for a sample screen shot). Commands can be given from the company-level on down to the platoon-level and specify paths and orders for individual vehicles. Just as in modern land warfare, the user fights with and against units consisting of a wide variety of assets. These include armor, infantry, artillery, engineers, air defense, and aircraft.

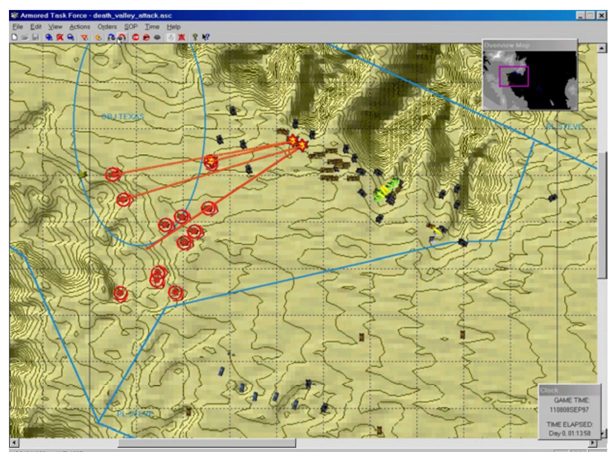

Figure 3. ATF simulation: A tank company/team assaults through the objective.

ATF does not assign specific duty positions, such as, battalion fire support officer (BN FSO) to the player. Instead, the player assumes the role of a task force (BN) commander, and controls all assets (armor, infantry, close air support, air defense, engineering and artillery). To perform well in a variety of missions requires a thorough understanding of friendly and enemy vehicle types, friendly and enemy tactics, the use of each different BOS, and an intuitive ability to focus and synchronize forces of different kinds at a decisive point in the battle. The goal of V-CTC is to provide the user with guidance and feedback specific to their normal duty position in light of this larger battlefield picture.

A subset of duty positions and missions will be selected to illustrate the tutor capabilities. The first year focus is on a player whose normal duty position is the Battalion Fire Support Officer (BN FSO). Acting as a task force commander in the simulation, he will have to move armor columns, call in CAS, use engineering units, use artillery, and synchronize all these assets. The tutor will be paying particular attention to mistakes that pertain to his normal job, such as the use of artillery and CAS for the BN FSO position. There will be six operational scenarios, four at NTC and two situated in Death Valley.

In the second year, two more duty positions will be added, but in lesser detail due to project scope. This means the tutor would provide perhaps 20% of the amount of tips and after-action diagnosis compared to the more fully developed BN FSO guidance. These additional positions will be applied for one or two scenarios. The two positions currently being considered are either an Armor or Mech Infantry company commander and an Armor or Mech Infantry S-3 (operations officer/planner).

The ATF cybernetic battlefield is a digitized elevation map of actual terrain and uses UTM coordinates. The maps are not hexes, but continuous terrain features including trees, buildings, and roads in contour-map representations. ATF includes scenarios from National Training Center, the Fulda Gap in Europe, the first Gulf War and a hypothetical Gulf War. Actual National Training Center (NTC) maps (e.g., of Crash Hill) are used in the NTC scenarios.

ATF is a real-time simulation (1X, 2X, 4X, or 8X of battle real-time). Note that this real-time aspect is very important in helping trainees acquire an intuitive feel of how fast the battlefield changes and in learning how to synchronize different battle operating systems such as artillery and armor.

TECHNICAL APPROACH

Tutor-Simulation Communication

The goal of the project is to develop a tutor that plugs-into existing PC-based simulations and interprets the simulation event stream data to derive an understanding of unfolding events and to develop a user model. The tutor will 'control' the simulation in minor ways, such as pausing the simulation and asking users to indicate items on the tactical map (e.g., the intended point-of-penetration for an obstacle belt). A major objective is to require minimal modifications from the simulation developer. The tutor-simulation connection is designed to be as generic as possible so that it can be extended to other PC-based simulations. Achieving these goals will maximize tutor re-use and keep overall costs down.

Sockets (TCP-IP) are used for inter-process communication. Through these sockets, the simulation passes its event stream to the tutor and the tutor sends back requests for information or commands to the simulation. To manage its many operations, the simulation has more information in its event stream than the tutor really needs to know about. The tutor can direct the DLL to filter events from the simulation.

The tactical events that the tutor receives are encoded in the simulation event stream as ASCII strings. The tutor needs to be able to understand what this data means. First, words and meanings are identified using natural language parsing techniques to recover syntactic and semantic structure. With this technique the tutor can identify simulation tactical events (e.g., fire orders, enemy destroyed events). Once identified, the events are time-stamped, placed on a time-line, and tied into the ontology to trigger any related knowledge sources. A knowledge source is a very general kind of rule. It uses terms from the ontology to explicitly represent constraint violation rules, tactical inference rules, and tutorial meta-rules.

The event stream provides information about what actions are occurring within the simulation. The tutor is also interested in capturing data that is not provided by the event stream, such as actions *not* taken and certain timing data (e.g., time spent planning an action). Some of this data can be captured via user actions on the graphical user interface (GUI). For example, users pause the simulation clock for planning paths and fire missions and then activate the clock again to resume execution.

Internal monitors and probes for GUI actions require simulation code modifications, so we chose to implement additional, easier-to-place monitors external to the simulation, called software connectors or wrappers (Marcelo and Balzer, 2001). Software connectors trap the simulation code calls out to the system library (operating system) code. All GUI events can be trapped without interrupting or modifying the simulation code.

The GUI events data will include activation of functions, but more importantly will also track what functions have been used infrequently or not at all. For example, the user activates toolbar buttons to enter order modes (e.g, fire mission, path planning). Through the software connectors, the tutor can track the time spent in these modes, and whether certain actions are never taken. For example, V-CTC could observe that a commander never plans multiple-volley fire missions, never uses smoke, never places units in defilade, and so on. This kind of data helps to build an often neglected part of the user model —actions not taken, currently and chronically.

The data stream and GUI actions/inactions provide the basic data for the tutor to build the user model. But how does the tutor *understand* this data and evaluate it?

Tutor Knowledge and Reasoning

The VCTC tutor uses natural language parsing techniques on the simulation event stream. At the simplest level, it identifies actions (e.g., fire orders) and events (e.g., taking fire), and makes basic inferences. For example, the simulation data stream provides fire mission data in terms of grid points, not targets. Rather than require the simulation code to change, the tutor employs heuristics to evaluate a list a targets within 1km of the aim point to identify the intended target.

For more sophisticated analyses, V-CTC uses dynamic, deductive reasoning rather than fixed decision trees or hard-coded procedures specific to a particular simulation. This reasoning is performed over a knowledge base with models of the domain, student, and pedagogy.

Developing knowledge bases is a very time consuming and expensive process. A standard approach has been to build systems consisting of hundreds to thousands of rules. Such systems are specialized and brittle, and difficult to maintain. They do not handle questions outside of their restricted knowledge areas due to a lack of common sense (basic world knowledge). More importantly, they are hard to reuse in new applications.

Our knowledge representation is a tiered, modular structure built around an upper ontology representing basic (common) concepts, such as time and spatial relations (see Figure 4). This common ontology is being developed by an international group vetted by IEEE and is in the public domain (Niles and Pease, 2001). This modular structure enables re-use of the ontology to new domains, while allowing more focused effort to develop the domain-specific knowledge at Service- and task-specific levels.

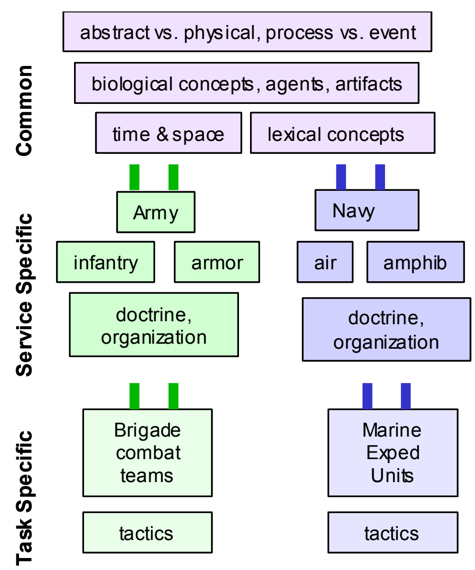

Figure 4. A modular, composable ontology.

V-CTC contains a model of the tactical domain, such as knowing about friendly and enemy vehicle types (see Figure 5). This is not a simple taxonomy, but an ontology with concepts and axioms (rules) upon which various types of reasoning can be performed. For example, simple reasoning by inheritance allows basic inferences such as a T-80U is a maneuver armored

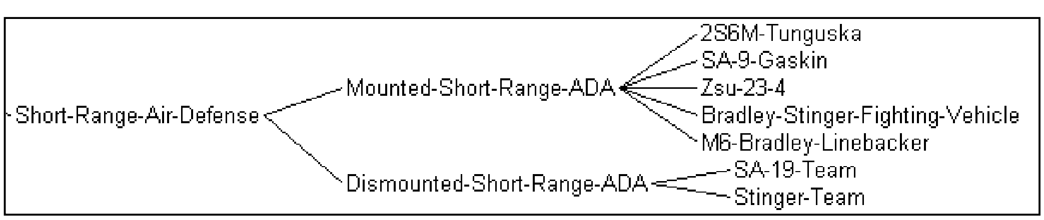

Figure 5. Part of a weapons classification, by BOS, in the domain ontology.

vehicle and reasoning-by-parts determines the maximum anti-tank range of an armored vehicle from the AT weapons it carries.

In turn, these basic inferences are used in the domain rules for detecting constraint violations. For example, "enemy air defense artillery (ADA) should be suppressed prior to close air support (CAS) missions." This rule could be triggered by friendly CAS entering the scenario air space. The virtual O/C would check each identified enemy ADA site for suppression. If a 2S6M Tunguska were detected, it would identify it as ADA and query the simulation to see if it was suppressed. If it were not, then a doctrinal constraint violation would be noted and might trigger a warning:

> *You should suppress all enemy ADA before your CAS arrives. CAS aircraft are vulnerable to an enemy 2S6M Tunguska now*!

The range of possible rules to encode in V-CTC is quite large, from frequently quoted maxims (e.g, always maintain a reserve) to arcane heuristics of tactical experts (e.g., never roll up on a T-90). More nuanced rules are also possible and most will be of this nature. For example, normally one battery high explosive (1 BT HE) is sufficient ammunition to take out an enemy infantry observer team. But in certain scenarios (e.g., Crash Hill) there is only time for a few volleys before the possibility of being overrun by two successive waves of OPFOR BNs. The commander may choose to 'overkill' with ammo (e.g., 1 BN DPICM). Therefore, the tutor does not just uniformly apply rules — it also considers METT-TC factors. Ultimately we expect V-CTC to have hundreds of tactical rules relevant to its critiquing and advising role.

It is important to emphasize that due to scope, the V-CTC tutor does not cover everything that an experienced commander knows nor handle his full reasoning capabilities. It only models certain tactical aspects relevant to the selected job roles and scenarios.

Student Model

VCTC will provide a real-time assessment of student state that is richer than current approaches. It includes performance-based measures of actions and choices during a realistic simulation, as well as knowledge-based measures of student plans, perceptions (e.g., of enemy threat), and explanations for actions taken or not taken. It includes latency (time to respond) and self-assessment measures to provide information for a model of the user's confidence in their different areas of knowledge.

Since these are complex measures in a simulated environment, rather than simple multiple-choice responses, a more sophisticated approach is needed for determining what a student knows and does not know. Furthermore, users do not respond consistently. Instead, they may forget material or make correct decisions for the wrong reasons. A Bayesian approach can handle such inconsistencies, and will analyze and integrate all the student data (performance, latency, and self-assessments) and develop a fine-grained model of student knowledge, skills, and confidence.

Bayesian analysis can be viewed as a kind of constraint propagation, although the mathematics are much more complex (Pearl, 1988). The laws of probability express constraints on likely and unlikely outcomes over a large number of trials. A Bayesian reasoning system applies the laws of probability to this kind of model (the Bayes network) to update probability distributions given knowledge of actual events (see Figure 6).

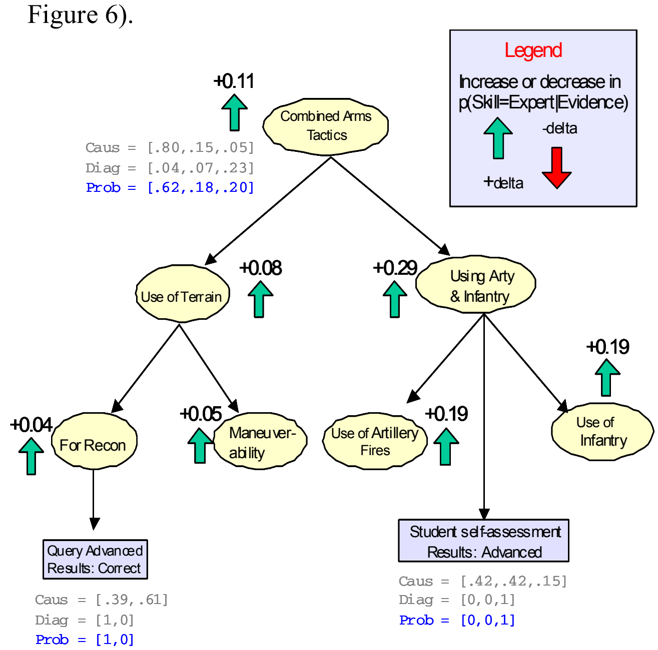

Figure 6. Bayesian student model (simplified).

One of the key properties of a Bayesian model is that it can be used to combine the different kinds of student data, and handle uncertainty in the data. Bayesian reasoning simulates certain kinds of reasoning that humans perform that are currently difficult to simulate in purely logical formalisms. It has proven to be very useful in student modeling in our previous projects (Murray, 1999).

Tutor-User Dialog

The advantage of human tutors and coaches is that they can ask questions to gain a better understanding of the user's plans and knowledge. In turn, the user can ask clarification questions and seek assistance when needed. Enabling a computer to converse as a human has been a research goal for many years. The common technical approaches are prone to error as they do not understand the full meaning of the natural language exchange — they simply respond to key words, pre-scripted patterns or statistical models.

While full natural language understanding (NLU) and generation has not been achieved, we have made significant progress on the technology in other projects (Murray, et.al, 2003; Sams & Murray, 2004). A limited, but useful, capability is included in the V-CTC. The approach is to understand a learner's natural language (NL) input by parsing the words and identifying the concepts, transforming the input to a logic format, performing reasoning on a knowledge base, and generating an appropriate response.

The ontology, domain knowledge base and the domain reasoning rules that interpret event stream data and evaluate student actions are also the foundation of the knowledge and reasoning needed to support dialog about the domain. Additional lexical information is also required, along with identification of how word senses are commonly used in this domain.

Triggered by key events, the tutor may pose a question to the user. The user types in a freely formed answer. The tutor parses the answer, models its meaning with a logical formalism, and compares this formal representation to a stored reference solution. If an incomplete or inaccurate response is given, the tutor can probe for more specificity or missing items.

The user will be able to ask some kinds of questions to the tutor. The tutor will analyze the question and search for an answer in its knowledge base. If the corresponding knowledge is found, it will formulate an appropriate NL response to the user.

While our NLU approach has deeper understanding than other approaches, it is still under development and is not error proof. Initially it will need to limit the scope of questions that the tutor can answer and may not be able to parse all user responses. We are working toward a long-term goal of supporting complex, mixed-initiative dialog. For example, we envision a time when the user can ask questions such as: "Could I use FASCAM to separate the second echelon from the first echelon to de-synchronize the enemy's attack?" and V-CTC would explain the pros and cons of the plan versus an expert's solution.

Architecture

General control for V-CTC is provided by a blackboard architecture. This software architecture allows meta-level reasoning, real-time reasoning, an integration of multiple kinds of knowledge sources, and evolving solutions to be constructed from lower-level solutions (Hayes-Roth, 1985).

The blackboard also provides a data structure that allows building a larger picture of the tactical and tutorial situation from lower-level events. Finally, the blackboard architecture can also be used to control filtering in the communications link (a DLL) between the tutor and simulation. The filtering determines the kind of events the simulation sends to the tutor. Figure 7 shows a simplified version of the overall system.

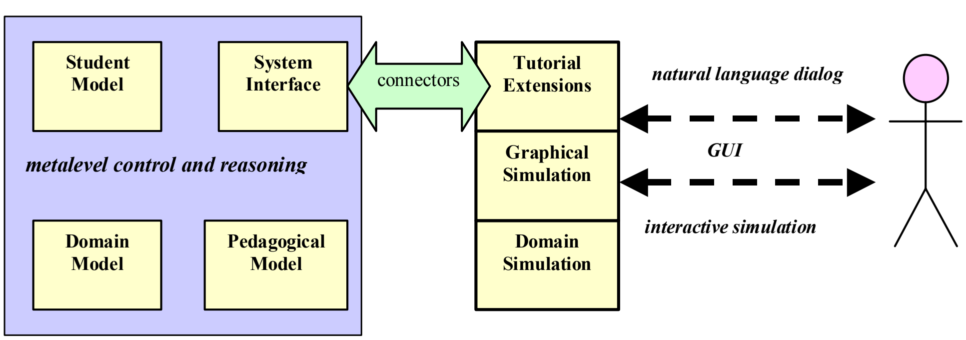

Figure 7. V-CTC architecture (simplified).

LESSONS LEARNED

A primary goal of this project was to keep modifications by simulation developers to a minimum. Any required modifications should be primarily focused on data communication formats and processes. For example, the simulation was modified by adding probes to dump state information at the end of each discrete state simulation update such as at the end of a time slice.

While we understood that we would have to filter and interpret low-level simulation event stream data, we did not fully appreciate that the simulation would be missing critical low-level data that the tutor needed to assess a trainee's performance. The tactical simulation creates detailed data to determine simulated physical world events (e.g., the impact angle of a round on the armor of a vehicle) that the tutor simply does not need. On the other hand, the simulation does not compute some basic event data that the tutor does need.

Consequently, additional changes to the simulation were required solely to support the capability to tutor. These included items such as determining what vehicles could be seen by the friendly or enemy side, the lead maneuver vehicles for either side, the location of smoke, and. status of vehicles by type (e.g., status of all engineering vehicles).

While these coding changes to the simulation are still minimal (no extensive recoding), it does present additional requirements for the simulation developer to make the tutor-simulation hybrid work. We are keeping track of the changes and the simulation developer time spent on these so that we can derive projected estimates for this approach.

EXPECTED BENEFITS

Intelligent tutoring systems (ITS) coupled with high fidelity simulations can provide supplemental training with the potential 2 sigma (standard deviation) improvement that good human tutors can accomplish. Through the use of simulations, trainees can acquire the extensive training time required to develop expertise without incurring the time, expense, and risk inherent in live training and real combat. Therefore expected benefits of a system that combines a simulation with an ITS, such as V-CTC, are improved training and reduced training costs.

Improved Training

V-CTC combines a simulation component and an ITS component. Each provides complementary training benefits.

1) The ITS improves training effectiveness by:

- Customizing training to the individual.
- Providing judicious guidance during the simulation and a diagnostic AAR regarding tactics, techniques, and procedures.
- Providing dialog-based exchanges that clarify trainee's intentions and knowledge, and answers trainee questions.
- Helping to calibrate a trainee's self-assessment with actual performance and knowledge.
- Providing exemplars of expert performance for comparison and illustration.

2) The high-fidelity simulation provides:

- Engaging job-relevant practice with exposure to many kinds of missions.
- Many hours of practice time required to build expertise and automaticity.
- A safe training environment.

Reduced Cost of Training

V-CTC reduces the development and operational costs of training.

1) Developments costs are reduced by providing an innovative method and components for an ITS to 'plug-into' existing PC-based simulation, thus

- Leveraging the millions of dollars already spent developing existing PC-based simulations.
- Promoting re-use of modular ITS components (ITS software, ontologies, tutorial strategies, and knowledge bases) for other agent-based applications.

2) Reduces operational costs of training with:

- Effective advance and refresher training, thus optimizing live training time and expense.
- Low-cost software and portable hardware platforms to run V-CTC.

CONCLUSION

In conclusion, V-CTC combines the training effectiveness of an ITS acting as a virtual O/C with the engaging realism of a high-fidelity tactical simulation. This combined training approach is a vast improvement over stand-alone simulations that provide a practice environment but have no explicit training guidance. V-CTC can benefit live CTC training experiences by providing effective training prior to CTC rotations to better use the time there, and subsequent to rotations to enhance retention.

V-CTC provides a method for the ITS to 'plug-into' existing high-fidelity simulations, without requiring major simulation modifications. This approach is being demonstrated for training combined arms warfare tactics, but the method and components are designed to be extensible for a variety of domains and simulations. With the increasing use of simulations for training, the addition of virtual, intelligent tutors and coaches will help to maximize training and provide continuously available, on-demand mission-level training for all forces at all echelons.

ACKNOWLEDGEMENTS

The V-CTC project is a Phase II SBIR (Contract No. DAAH01-03-C-R302) funded by DARPA's Training Superiority (DARWARS) program. We would like to thank Dr. Ralph Chatham, Program Manager, for his visionary concepts and commitment to revolutionizing training. We would also like to thank Pat Proctor, President of ProSIM Company, for assisting us by sharing his expertise in the areas of tactics, combined arms warfare, Army training, and the operation of the ATF tactical simulation.

REFERENCES

Chatham, R., and Braddock, J. (2001). *Training superiority and training surprise* (Defense Science Board Task Force Final Report). Avail on-line http://www.acq.osd.mil/dsb/trainingsuperiority.pdf

Hayes-Roth, B. (1985). A blackboard architecture for control. *Artificial Intelligence*, *26*, 251 – 321.

Marcelo T., and Balzer, R.M. (2001, June). Document integrity through mediated interfaces. *Proceedings of the Second DARPA Information Survivability Conference and Exposition (DISCEX-II 2001).* Available on-line at http://www.computer.org/ cspress/CATALOG/ pr01212.htm

Murray, W.R. (1999). An easily implemented, linear-time algorithm for Bayesian student modeling in multi-level trees. In *Lajoie, S.P.* and V*ivet, M. (Eds.), Proceedings of the 9th International Conference on Artificial Intelligence in Education, 50.* Amsterdam: IOS Publications. Avail on-line at http://www.iospress.nl/html/boek-309205249.html

Murray, W.R., Pease, A., and Sams, M. (2003). Applying formal methods and representations in a natural language tutor to teach tactical reasoning. *Proceedings of the 11th International Conference on Artificial Intelligence in Education.* Brighton, UK: International Artificial Intelligence in Education Society.

National Training Center (2002). *Army Nomination for FY2002 Secretary of Defense Environmental Awards*. Available at https://www.denix.osd.mil/ denix/Public/News/OSD/SecDef02/P2NonIndustrial/ army-fortirwin.pdf

Niles, I., and Pease, A. (2001, October). Towards a standard upper ontology. In Welty, C. and Smith, B. (Eds.), *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001*). NY: Association for Computing Machinery.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Mateo, CA: Morgan Kaufmann.

Proctor, P. (2004). Armored Task Force. Available on-line at www.prosimco.com

Sams, M, and Murray, W. (2004). *Dialog agent system*. SBIR OSD-CR01 Phase II Final Report. Orlando, FL: Naval Air Warfare, Training Systems Division.