Enhancing After Action Reviews Through Transparent AI Knowledge Representations

Jared K. Snyder, Jason R. Potts, Dan Allison

Discovery Machine Inc.

Williamsport, PA

jsnyder@discoverymachine.com, jpotts@discoverymachine.com, dallison@discoverymachine.com

ABSTRACT

The effectiveness of an after action review (AAR) relies heavily on expert trainers who visually observe, recognize, and record examples of trainees' positive and negative behaviors that occur during the course of the simulation (Salter, et al., 2005). This method is not cost effective for large training events since they require a large staff of experts whose recall of events is often limited by the fact that several simulation-based training exercises are conducted before debriefing begins (Freeman, et al., 2004).

This paper presents a framework for increasing the efficiency of instructor-led AARs by helping trainees diagnose, recall, understand and generalize their own performance. This framework was applied in the development of the After Action Review Console (AARC) which provides trainees with simulation playback, a graphical representation of the decision-making processes of automated intelligent entities, and a visual representation of automated intelligent entities' perceived environments. The visual representation of the behavior adds context to help the trainer assess the trainee's performance. The representation also enables data harvested during a training exercise to be organized and filtered, allowing the instructor to focus attention on key decisions and actions performed by the trainee. We also present a mechanism by which the best practices of experts, captured through a formalized knowledge capture methodology, can be used to develop automated entities for simulation execute well-defined tactics, maneuvers, and reactions. The knowledge capture approach enables the transparent knowledge representations that are essential to the AAR approach. This approach has the potential to improve the efficiency of training by reducing instructor workload and improving feedback to trainees.

ABOUT THE AUTHORS

Jared K. Snyder is a Software Engineer for Discovery Machine Inc., whose work focuses on behavior modeling and developing core technology offerings. After graduating with honors from Bucknell University with a Bachelor's of Science degree in Computer Science, Mr. Snyder has been designing and developing software for Discovery Machine Inc. since 2009. His work has focused on implementing graphical user interface tools in Discovery Machine Inc.'s core technology offerings and creating behavior models for simulated entities within constructive military training environments.

Jason Potts is the Director of Intelligent Systems for Discovery Machine Inc., overseeing the software development of their core technology offerings, as well as the application of their technology to customer problems across a variety of domains. Since graduating with honors from WPI with a Bachelor's of Science degree in Computer Science, Mr. Potts has been designing and developing software for Discovery Machine Inc. since 2001. This work has focused on development of tools facilitating knowledge capture, and representation of human expertise in a machine executable form. In addition to behavior modeling of simulated entities within constructive military training environments, these applications also include representation and automation of business process intelligence within commercial sectors such as manufacturing or pharmaceuticals.

Dan Allison is the Director of Knowledge Engineering and Professional Services for Discovery Machine Inc. Mr. Allison has 25 years of experience in system engineering and modeling in a variety of businesses including DoD SONAR and RADAR systems, medical device design and military training systems. His technical emphasis has been system analysis, modeling, algorithm design, and requirements management. His most recent work focuses on knowledge engineering using the Discovery Machine Knowledge Capture Methodology and building deployable knowledge models using Discovery Machine's Modeler software. Mr. Allison earned a BSEE from Penn State and MSEE from Syracuse University and is six sigma certified. He has received 2 patents on the topic of deconvolution algorithms and has been an active member of both IEEE and INCOSE.

Enhancing After Action Reviews Through Transparent AI Knowledge Representations

Jared K. Snyder, Jason R. Potts, Dan Allison
Discovery Machine Inc.
Williamsport, PA
jsnyder@discoverymachine.com, jpotts@discoverymachine.com, dallison@discoverymachine.com

INTRODUCTION

Simulation-based training has been used in the military for many years as a way to supplement live training exercises. With recent advances in tools for simulation-based training, this type of training is being implemented widely across many fields including healthcare, transportation and emergency planning preparedness (Lewis, 2010). The use of automated intelligent behaviors has contributed to this increased use; they both increase training benefits and reduce cost of the training.

This paper presents an approach by which intelligent entities are developed such that their knowledge is transparent and can be used to improve the efficiency of instructor-led after action reviews (AARs). We begin by outlining the inadequacies of current AAR methods which are addressed by our proposed approach.

AFTER ACTION REVIEW

The military defines an AAR as follows:

> "An after-action review (AAR) is a professional discussion of an event, focused on performance standards, that enables soldiers to discover for themselves what happened, why it happened, and how to sustain strengths and improve on weaknesses. It is a tool leaders and units can use to get maximum benefit from every mission or task" (Department of the Army 1993, p. 1).

The military's definition of an AAR suggests that the most important features of an AAR are the trainee's involvement in the review and the trainee's ability to retain good behaviors while diminishing negative performances.

Researchers have described the central functions of an AAR in the following manner:

> "…help the learner recall from the recent training specific episodes of correct performance in context, and generalize this knowledge so that it can drive performance in similar (but not identical) future circumstances. An AAR must also help the learner recall specific instances of incorrect performance, discourage repetition of that performance in the future, cue recall of correct performance knowledge, and associate it with similar circumstances" (Wiese, Freeman, Salter, Stelzer, & Jackson, 2008, p. 4).

These functions are then decomposed into the following five cognitive functions for the trainee to accomplish during an AAR (Wiese, et al., 2008 p. 4):

1. Diagnosis of Trainee Performance
2. Recall of Trainee Performance in Training
3. Understanding of Expert Behavior
4. Generalization to Future Situations
5. Assessment and Display of Competence

Diagnosis of trainee performance requires the trainee to recognize the actions they performed that led to success or failure. This is often supported in AARs by having a serial playback of the training scenario. The serial playback allows the trainee to perform a root cause analysis to diagnose the subtleties of the events that led up to the training success or failure.

Recall of trainee performance requires the trainee to *accurately* remember the events and situations that occurred during the simulation. In long training events, the ability to recall details of the scenario decays quickly over time. Also, trainees and trainers each have cognitive biases which tend to give a false sense of confidence in their ability to recall.

Understanding of expert behavior requires that a trainee learn what an expert would have done given a similar situation. Generalization to future situations requires the trainee to simplify a situation they encountered so that they may apply it to future similar situations they may encounter. Assessment and display of competence requires the trainee to rate and discuss their performance in the AAR process.

In this paper, we focus on how AAR tools help achieve the diagnosis and recall functions of an effective AAR. Specifically, we show how using AI behavior models with an understandable knowledge representation can be used to improve the use of AAR tools.

CURRENT METHODS AND LIMITATIONS

Currently, one way training systems attempt to meet the aforementioned AAR goals is by relying heavily on expert trainers who watch the events of the training exercise. Since these experts are well versed in military tactics, they are the perfect candidates to recognize and record examples of trainees' positive and negative actions that occur during the course of the simulation (Salter, Hoch, & Freeman, 2005). Unfortunately, due to the size of some distributed simulation-based training events, this solution requires a large staff of experts. Since "teams typically [conduct] several simulator sessions before debriefing the first" (Freeman, Salter, & Hoch, 2004, p. 2577), another limitation of this type of solution is that recall of previous runs can be seriously diminished by the time that elapses until an AAR begins. Another hindrance to this solution is that, due to the nature of distributed training, trainers may not be located with the trainees, and therefore may have a less clear view of a trainee's performance (Salter, et al., 2005).

Another current method of reaching the goals of an optimal AAR is use of after simulation playback tools. Since most important data from the simulation is recorded during the training event, applications which access this data can use it to replay the simulation with controls similar to a standard video player (play, rewind, fast forward, etc.; Wiese, et al., 2008 p. 4-5). Since a serial playback tool visually replays the simulation, the playback tool

> "…should ease recall. It should reinforce memory for normative sequences of events (e.g., mission phases) and thus it should help trainees to generalize from the specific episode to its class. Finally, serial replay should help trainees recognize actions that snowball into disaster (or success)" (Wiese, et al., 2008, p. 5).

Therefore a simple AAR playback tool can only meet three of the five goals of an optimal AAR: diagnosis, recall, and generalization but only to a limited degree.

With the implementation of automated intelligent behaviors into simulation environments, certain deficiencies in meeting these goals can be alleviated to increase the effectiveness of an AAR.

There are a few issues with the current methods that limit their ability to provide an effective AAR. Those methods that use automated intelligent constructs don't have a good way for the trainee to understand why the automated entities performed certain actions in the simulation. Some of the automated constructs are essentially black boxes, making it difficult to understand their behavior. Along with this, there is no good way to understand the automated construct's viewpoint of the situation without understanding how the behaviors work. Entities form a view of the world through their sensor data which causes each entity to have a view of the world which might be quite different than the view from the trainee. A third issue is that some of the current methods fail to engage the trainee in the diagnosis of their own performance.

Lack of Understandable Artificial Intelligence

> "Typically in live training, the trainees are allowed to ask questions of the opposing forces (OPFOR) to fill in the gaps of learning not provided by mission statistics and a list of accomplished and failed objectives. However in many simulations the opposition forces are automated constructs in a simulation rather than actual people. In this situation, the learning gained through the dialogue with the OPFOR is lost" (Johnson & Gonzalez, 2008, p. 7).

Currently, because "black-box" behaviors are used to control automated constructs, a trainee cannot ask an OPFOR unit why it performed the actions it did at a given time. In current solutions, it is even difficult to know exactly what action a unit was performing at a given point in the simulation. Without being able to present this information to the trainee, current methods hinder the trainee's ability to understand their enemy's behavior. Without understanding the behaviors of other entities, a trainee will be limited in their ability to diagnose, recall, and generalize their performance.

Individual Views of the World

Another deficiency in current solutions is in presenting automated entities' perceived view of the world. Many AAR tools are incapable of "presenting the scene as a pilot, tank driver, or other operator *perceived* it" (Wiese, et al., 2008, p. 5). This deficiency in current AAR tools makes it difficult for a trainee to see why an ally reacted to a situation the way they did, or why an OPFOR unit was able to avoid detection, or any other question a trainee might have about the actions taken by other entities. Being able to see how another unit saw the world at a specific time allows the trainee to make better estimations about how other units will behave or react to actions performed by the trainee. Thus a trainee's ability to diagnose, generalize, and assess their performance is limited due to this shortcoming in current methods.

Engage Trainee in Diagnosis of Performance

Most current AAR tools fail to engage the trainee in the diagnosis and assessment of their own performance. It was found that, "…instructors often failed to engage trainees in diagnostic (or any) discussions during AAR, and instead dominated these sessions with monologues concerning their own observations" (Wiese, et al., 2008, p. 8). This exclusion of the trainee from the AAR process defeats the main purpose of an AAR. Without the self discovery that is supposed to be encouraged in an AAR, the trainee is less likely to benefit or learn from the training simulation.

BACKGROUND

The approach for improving AARs that we present in this paper hinges upon exposing the autonomous behaviors in a training scenario to the participants of the training event. However, merely exposing the inner workings of an autonomous entity is not enough to benefit the trainer and trainees. In order to expose these behaviors to the participants in such a manner as to facilitate learning on the part of the trainee, these behaviors must not only be viewable during the AAR, but must be understandable to those trainees who are examining them. In order to make these behaviors more transparent and comprehensible to the trainees, we employ a behavior modeling approach that focuses on capturing knowledge directly from subject matter experts from the training domain. This modeling approach enables non-programmers to represent autonomous behavior models in a visual, and human-readable form that uses terminology familiar to those operating in the domain (in this case, the trainer and trainees).

Behavior Modeling Approach

In order to capture and model behaviors for autonomous entities in a training simulation, we have employed a modeling approach which represents knowledge captured from Subject Matter Experts (SMEs) in a language called Task-Method Knowledge (TMK). In contrast to representations based upon finite state machines, or dynamically applied rules used by some cognitive architectures (Newell, 1990; Anderson, 1982), these TMK models provide a mechanism by which a set of process models is created and stored from the outset. Process models are represented as a task decomposition hierarchy made up of three types of elements; Tasks, Methods, and Procedures. Figure 1 provides an example hierarchy represented in TMK. Tasks are depicted visually as rectangular orange nodes which represent a particular job to be completed. These tasks then decompose into child elements, depicted as cyan ovals (Methods) or rectangular yellow nodes (Procedures). These two element types represent alternative options for accomplishing the parent task, in its entirety. These "task-method decomposition hierarchies (strategies) represent patterns of behavior that can be applied to specific kinds of problems" (Potts et al., 2010, p. 5).

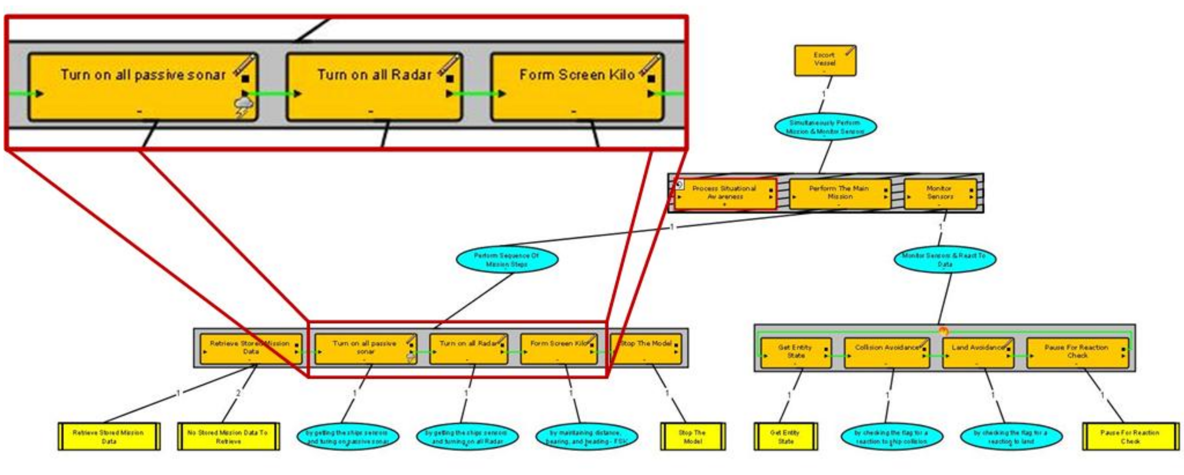

Figure 1. Example Behavior Model

Behavior Composition

We expand the usefulness of these process models by creating libraries of modular sub-hierarchies called Basic Level Actions (BLAs). Each BLA represents an independent operation built using subject matter expert knowledge. In practice, these BLAs take an average of 5 man-days to develop, test, and deploy making this process significantly faster than the current development process of behavior models in simulation environments (Potts, et al., 2010).

By creating a library of these BLAs, we provide users with a set of building blocks that any user can put together to build highly complex missions for autonomous entities in a simulation environment. The SME is able to leverage this library of BLAs through a behavior development environment called a “Console.” This Console provides a wizard-like interface, called an “advisor” which allows users to quickly compose new entity behaviors by combining BLAs into larger strategies in order to accomplish a higher level mission objective. This approach allows SMEs to develop novel behaviors for use within simulation without assistance from software engineers, enabling far greater customization of entities’ behaviors in order to meet specific training needs.

Benefits of Behavior Modeling Approach

In the context of creating automated intelligent behaviors in a simulation environment, our process of behavior modeling provides the following advantages that can be transferred to the AAR process as well (Potts, et al., 2010):

- Rapid development of new behaviors for customized training scenarios
- Expert domain knowledge captured within modeling tools to assist building more intelligent behaviors
- Transparency of behaviors for subject matter experts during and after development
- Transparency of behaviors at runtime, exposing decision making process of entities to operators

AFTER ACTION REVIEW CONSOLE

Figure 2 displays a prototype user interface for the After Action Review Console (AARC) that leverages the advantages of the behavior modeling approach and makes use of the visual knowledge representation.

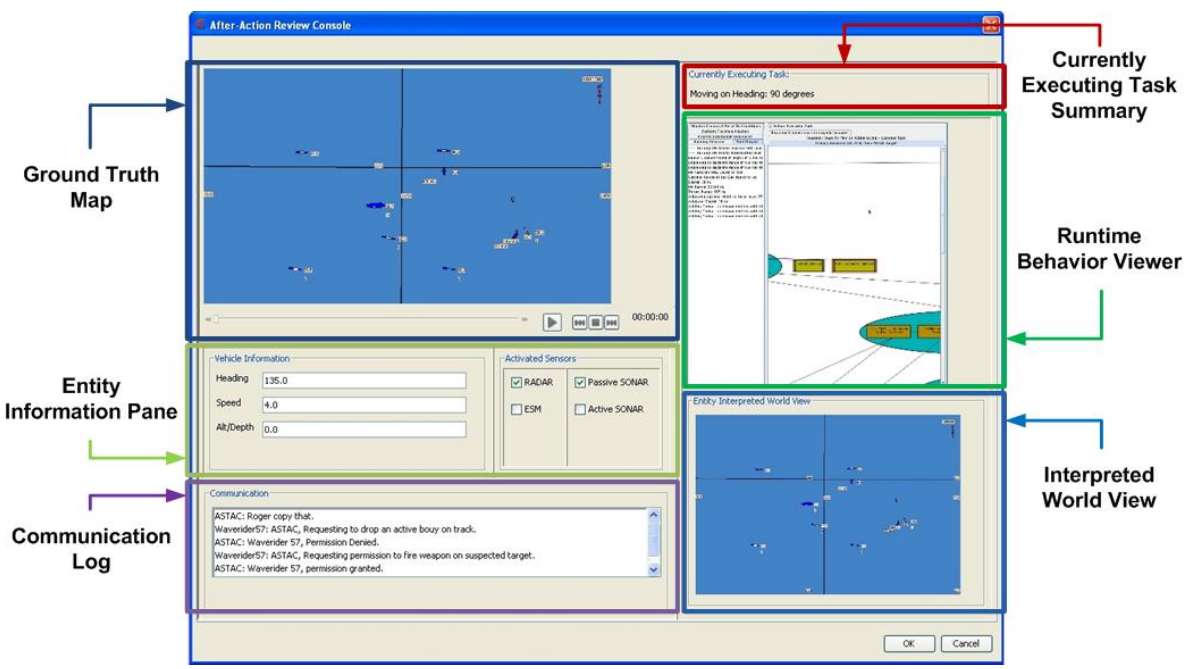

Figure 2. After Action Review Console (AARC) User Interface

The ground truth map widget displays the serial playback of the simulation, from the ground truth perspective. Thus, all units are displayed at their exact location at a selected time ***t***. The ground truth widget also contains the playback controls for the AAR tool. These playback controls are akin to a standard playback device with controls like play, pause, fast forward, rewind, and optimally allow users to select an arbitrary time to go to.

The Entity Information Pane displays basic information about a selected entity at the selected time. Information displayed in this pane pertains to, but is not limited to, location, speed, heading, altitude/depth, activated sensor equipment, damage reporting, supply inventory, fuel/battery level, etc.

The communication log displays all communications between the selected entity and other units. Due to the fact that communication between entities can heavily influence an entity's decision making process, these communications must be available for the trainee to observe. The communication log shows all communication sent and received by an entity until the currently selected time.

The Currently Executing Task Summary (CETS) displays a short summary about what the selected entity is doing at the selected time ***t***. The CETS allows users to quickly assess the current mission phase of a selected entity.

The job of explaining why an entity is performing its current actions is left for the Runtime Behavior Viewer (RBV) widget to explain. The RBV widget shows the behavior running in real time. The trainee can view the SME knowledge in the behavior model and see what decisions the entity is making and why. Not only can they see the decisions made but they can see the decisions that were not made. They can also see the values of the model parameters that led to these decisions.

By collapsing or expanding the TMK elements in the view the trainee can limit the amount of detail they need for their understanding. They can also traverse the TMK hierarchy to understand decisions that led up to the current decision and how this will impact future decisions.

The final widget is the Interpreted World View (IWV) which represents what an entity thinks the world around them looks like at a selected time. This feature assists in explaining why entities controlled by intelligent behaviors act in an unexpected manner. Unexpected behavior from an entity controlled by an intelligent behavior may have a vastly different IWV than the actual ground truth map.

A trainee using the proposed console would have the ability to recall and diagnose their performance. By being able to serially playback the entire training event and have the information provided by the AAR tool, a trainee would have a much better chance at being able to accurately recall the actions and reasons why they behaved a certain way in a specific situation. The large amount of information presented by the console would allow the trainee to diagnose their performance and investigate which actions taken by them caused the events of the scenario to unfold as they did. Thus the prototype described above, through the information it reveals and ability to playback this information, should provide a greater level of assistance in diagnosis and recall of trainee performance than other current methods.

USE CASE

The following use case example illustrates how the AARC, and more specifically its visual representation of entity knowledge, can help a trainee diagnose, recall, understand, generalize and assess during an AAR. A trainee is performing an anti-submarine warfare mission where they are tasked with finding and prosecuting an OPFOR submarine using a surface ship. The trainee runs the simulation, but fails to find the submarine in the allotted time. During the AAR process the trainee opens the AARC which loads the data from the recent training exercise. The trainee thinks that they should have detected the submarine about forty minutes into the training event, so the trainee slides the timeline to forty minutes into the simulation. The trainee checks the ground truth map and sees that the submarine was far outside the original area of probability and slowly moving farther away from the trainee's vessel. Seeing this, the trainee may wonder why the submarine is slowly moving away from trainee's vessel before they even entered the area of probability. To investigate the trainee will select the OPFOR submarine by clicking on the submarine's icon in the ground truth map.

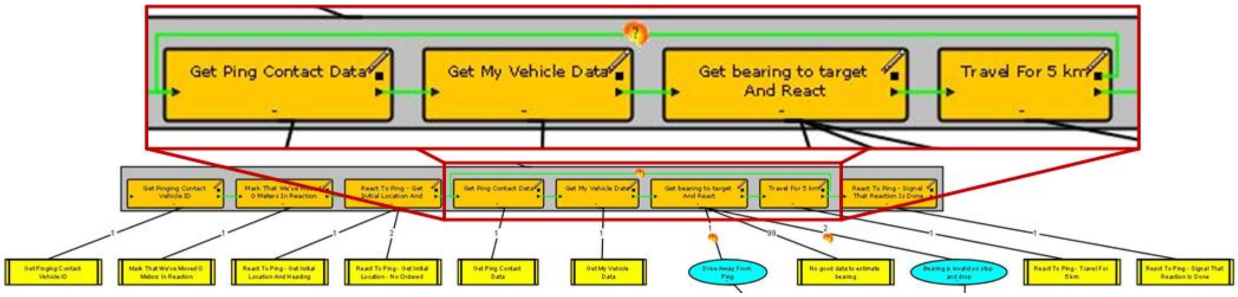

Figure 3. Submarine Behavior Model

Being that the submarine was tasked with an automated intelligent behavior, the CETS, RBV, and IWV widgets will be populated with information about the decision making process of the submarine at this time. The trainee will check the CETS and discover that the submarine is currently "Reacting to Ping by Withdrawing." To understand what this means, the trainee would look at the RBV and check the automated entity's current position in the behavior model. Looking at the behavior model (as shown in **Figure 3**), the trainee would see that the submarine is currently performing the following set of tasks: *Get Ping Contact Data*, *Get My Vehicle Data*, *Get bearing to target and react*, and *Travel For 5 km* which loops back to *Get Ping Contact Data***.** The trainee now knows that this submarine was still retreating from his vessel's active sonar at this time. The trainee can see that the submarine had a very good location on his vessel based on comparing the submarine's IWV display with that of the ground truth map. Thus the trainee now knows exactly what the OPFOR submarine was doing at the time they expected to find the submarine. The next question the trainee will want to answer is when and what caused the submarine to start withdrawing from their vessel's active sonar. With the submarine still selected in the ground truth map the trainee will reverse the playback of the scenario until they find the point where the submarine started withdrawing from active sonar. Here the trainee will check the CETS and see that the submarine was "Starting to React to Ping". The IWV of the OPFOR submarine will show the trainee that the submarine had a bearing and range to his vessel based on an active ping it detected with its active intercept sensor. The trainee checks the submarine's RBV at this time and sees that the submarine is performing a set of tasks to try and determine how to react to the ping based on the distance to the detected ping and its current mission type (see **Figure 4**).

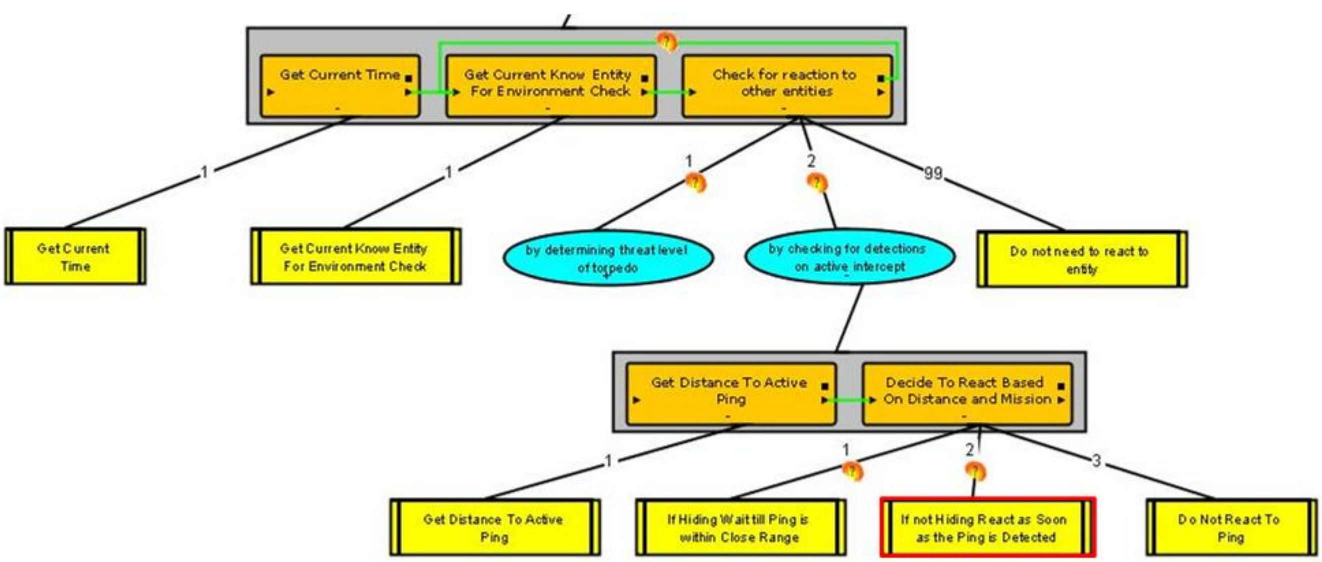

Figure 4. OPFOR Submarine's Behavior Model Deciding to React to Detected Active Sonar

The trainee can observe that the submarine decided to run away from the ping because it was not hiding and the submarine's behavior model has it react as soon as the ping is detected if it is not hiding. With all of this information in mind the trainee can make the deduction that they failed to detect the submarine because they turned on their vessel's active sonar too early. The trainee gave away their position to the submarine and gave the submarine enough time to successfully withdraw from the area of probability before their vessel could get within range of detecting the submarine. Through this diagnosis of why the sub eluded detection the trainee has effectively diagnosed the negative behavior they performed, recalled the actions they performed, can draw a generalization of waiting to be closer to an area of probability before turning on active sonar, and can do a simple assessment of how well they performed the training event.

ANTICIPATED BENEFITS

The representation of automated intelligent behaviors and the use of this representation in the AARC has the potential to improve an AAR in 3 areas:

- Understanding the behavior of the intelligent constructs
- Showing any individual's view of the world
- Engaging the trainee in the diagnosis of their own performance

Understandable Behavior

Integration of automated intelligent behavior representations into the AARC provides the ability to present details about the OPFOR's behavior to the trainee in a manner which allows the trainee to clearly see what the entity is doing at any time in the simulation. Not only will the trainee be able to see what the entity is doing, but by examining the behavior hierarchy model and their view of the world, they will also be able to understand why.

By providing the trainee with what an entity was doing and why, the trainee is able to better diagnose whether his actions against this enemy were correct. This information also allows the trainee to generalize the scenario at this selected time to apply to future situations.

Individual Views of the World

Since the automated intelligent behaviors are transparent and easy to understand, the AARC is able to present the automated entities' perspective of the world. Users are able to quickly change the perspective of the simulation from that of the trainee to any other entity in the simulation for a given time ***t***. Through the interpreted world view and entity information pane widgets, an instructor and/or trainee are able to understand another entity's status and perception of the world at large.

This ability to understand another entity's status and perception at any selected time should greatly help a trainee recall, diagnose, and generalize their performance. Thus, by improving this aspect of the AAR tool, a trainee can satisfy the diagnosis, recall, and generalization goals for an optimal AAR.

Engage Trainee in Diagnosis of Performance

The AARC approach engages the trainee in their own review. By presenting the playback of the simulation in an interactive user interface, a trainee is able to examine the data for themselves instead of having the instructor read their notes and observations in a one sided lecture-style review. By presenting the trainee with the entities' current action, intelligent behavior, interpreted world view, and status, a trainee can diagnose problems and successes in their performance. A trainee's interaction with the AAR tool should serve to engage them in the diagnosis of their performance.

TECHNICAL OVERVIEW OF AARC

In order for the user interface described above to provide the information described, there must be a sufficient level of data collected during the execution of the simulation-based training event. There must also be means by which the AAR tool can access this data and present the information in a logical way. Figure 5 highlights the process by which the data is logged and retrieved by the AARC.

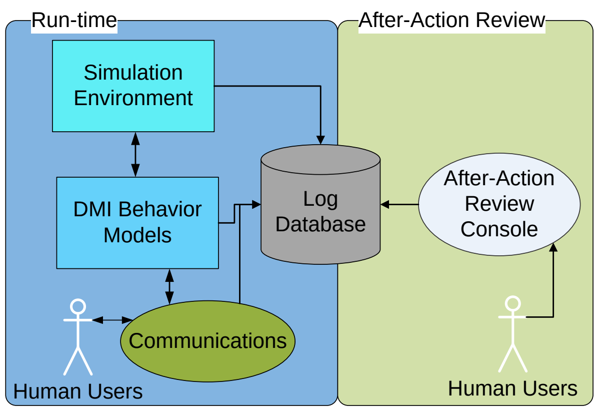

Figure 5. AARC System Architecture

Simulation Run-time

In a simulation-based training event that makes use of automated intelligent behavior models and communication with human users it is important to understand how communications, behavior models, and the simulation environment interact with each other.

The interaction of automated intelligent behaviors and the simulation environment is achieved through the use of a thin interface layer. This interface layer provides three core types of functions between the automated intelligent behaviors and the simulation environment:

- Action Functions
- Query Functions
- Data Structure Specification

Using these three functions automated intelligent behaviors are able to control entities in a simulation environment, retrieve information from the simulation environment, and pass data back and forth between the simulation and the behavior models (Potts, Griffith, Sharp, & Allison, 2010).

Another important interaction is communication between automated intelligent behavior models and human users. Speech recognition software is used to convert human speech or commands into structures that can be understood by the automated intelligent behaviors. Communications can also be sent from automated intelligent entities back to human users using text-to-speech applications. All of these communications between entities can have a large effect on the way entities behave; therefore it is important that these communications be logged into a database for use in the AAR process.

During the execution of the training event, information pertaining to entity state, communications sent and received, current execution path of automated intelligent behaviors and current world view need to be logged to a database for the AAR tool to access during the AAR process. The three areas where this information is available during the execution of the training event are: the simulation environment, automated intelligent behavior models, and in the communications between human users and automated intelligent behavior models.

Simulation Environment

The simulation environment contains all of the entity state information that needs to be accessed by the prototype described above in order to create a ground truth map and to display a selected entity's state. Thus the simulation environment needs to record information such as location, heading, speed, altitude/depth, activated sensors, force, and unit type into a database at regular intervals.

Behavior Models

The automated intelligent behavior models contain an entity's current world view and the current execution path of its assigned automated intelligent behavior. Every automated behavior model is responsible for logging its execution path as the model executes. This information is used by the AAR tool to display the decisions made by the automated intelligent entity in the RBV. The behavior model is also responsible for logging its current world view, which includes the following:

- Description of what task the automated intelligent entity is currently performing.
- All entities detected and known by this automated entity.

This information is used by the AARC to describe the automated intelligent entity's current task and generate an interpreted view of the world for the IWV.

Communications

Communications that are sent between entities and/or human users also need to be stored in a database for use by the AAR tool. A list of communications sent and received needs to be specific for each entity in the simulation.

After Action Review

The console will playback the simulation in the following way. While the simulation plays back, each widget updates their displayed information by using the logged data, current playback time, and the entity selected in the Ground Truth Map. For example, the Ground Truth Map widget redraws itself based on the entity state data stored in the database at the current playback time. In the case of the Interpreted World View, this widget redraws the position of the selected entity, and if the selected entity is an automated intelligent entity, the IWV widget draws all of this entity's detected known entities as of the current playback time.

CONCLUSION

In an optimal AAR, the trainee is engaged in the process of diagnosing their performance, recalling their performance, understanding expert behavior in similar situations, generalizing aspects of their performance to apply to future similar situations, and assessing their overall performance and competency (Wiese, et al., 2008 p. 4).

1. Diagnosis of Trainee Performance
2. Recall of Trainee Performance in Training
3. Understanding of Expert Behavior
4. Generalization to Future Situations
5. Assessment and Display of Competence

Since no current methods satisfy these five goals completely or involve the trainee as much as is encouraged, the proposed method attempts to improve the satisfaction of these five goals. Through the use of automated intelligent behaviors and the AARC, a simulation's artificial intelligence is more understandable, a trainee is able to view the simulation from different perspectives, and the trainee is engaged in the AAR process. By making these improvements, the five goals of an optimal AAR are met to a higher level of satisfaction.

REFERENCES

Anderson, J.R. (1982). Acquisition of Cognitive Skill, *Psychological* Review, *89*, 369-406.

Department of the Army, (1993). *Training circular 25-20, a leader's guide to after-action reviews* Retrieved September 8, 2010 from http://www.au.af.mil/au/awc/awcgate/army/tc_25-20/tc25-20.pdf

Freeman, J., Salter, W., & Hoch, S. (2004). The users and functions of debriefing in simulation-based team training. *Proceedings of the Conference of the human factors and ergonomic society* (pp. 2577-2581). Santa Monica, CA: HFES.

Johnson, C., & Gonzalez, A.J. (2008). Automated after action review: state-of-the-art review and trends. *Journal of Defense Modeling and Simulation: Applications, Methodology, Technology.*, *5*(2), 108-121.

Lewis, F. (2010). *President's message*. Retrieved August 3, 2010 from http://www.iitsec.org/about/Pages/PresidentsMessage.aspx

Newell, A. (1990). *Unified Theories of Cognition*, Harvard University Press.

Potts, J.R., Griffith, T., Sharp, J.J., & Allison, D. (2010). Subject matter expert-driven behavior modeling within simulation. *Proceedings of the Interservice/industry training, simulation, and education conference*

Salter, W., Hoch, S., & Freeman, J. (2005). Human factors challenges in after-action reviews in distributed simulation-based training. *Proceedings of the Human factors and ergonomics society 49th annual meeting* Santa Monica, CA: HFES.

Wiese, E., Freeman, J., Salter, W., Stelzer, E., & Jackson, C. (2008). Distributed after action review for simulation-based training. In D.A. Vincenzi, J.A. Wise, M. Mouloua, P.A. Hancock (Eds.), *Human Factors in Simulation and Training* Boca Raton, FL: CRC Press.

Discovery Machine, Inc. Patents Awarded:

US Patent 7,257,455 B1 – System and Method for Collecting and Representing Knowledge

US Patent 7,757,220 B2 – Computer Interchange of Knowledge Hierarchies

US Patent 7,958,073 – Software and Methods for Task Method Hierarchies

Discovery Machine, Inc. Patents Pending:

US Application 11/781,671 – Reflective Processing of Computer Hierarchies