A Federating Protocol for Distributed After Action Review

Virginia Travers, William Ferguson
BBN Technologies
Cambridge, Massachusetts
travers@bbn.com, wferguso@bbn.com

Timothy Langevin
MÄK Technologies
Cambridge, Massachusetts
langevin@mak.com

ABSTRACT

How can learners in a distributed virtual exercise participate in a combined, virtual after action review (AAR)? The current solution is to provide each learner with an instance of the same "tool". This tool might be as simple as a phone hooked to a conference call or as elaborate as distributed groupware that allows sharing of audio and visual channels. The problem with these "uniform" solutions (where each learner uses the same tool) is that they do not allow each learner to have access to an AAR system that is tailored to his or her own point of view and tasking. We suggest the alternative of offering a federating protocol for AAR systems that allows users to participate in a distributed AAR while using visualization and other tools that are tailored to their own role in the experience. Using an open federating protocol (which we developed under the DARWARS program), very different systems can be coordinated during AAR to provide a shared focus of attention and other needed "orchestration" (such as the ability to refer everyone to the same entity or point in time). This federating AAR is especially useful when very different, existing simulation systems are connected to enable large, distributed training events.

ABOUT THE AUTHORS

Virginia Travers joined BBN Technologies in April 1975. She is a senior scientist, currently working as the technical lead for the DARWARS project. She is the Principal Investigator of the JADL IPA Project, a project for the Joint Advanced Distributed Learning Co-Laboratory focused on integrating support for simulation-based training with the SCORM Runtime Environment. Ms. Travers has worked on a variety of projects for BBN, both as a technical contributor and project manager, including the COVE project (a simulated training environment for ship handling), and the DARPA sponsored ALP and Cougaar projects for developing distributed agent-based technologies. Ms. Travers holds a Bachelor of Science in Computer Engineering from Case Western Reserve University.

William Ferguson joined BBN Technologies in 1991. He has expertise in artificial intelligence, cognitive science, computer-based training, and commercial game technology. At BBN he applies this knowledge as a designer, business developer, evangelist, and technology visionary, most recently on the DARWARS project. He is the Principal Investigator on the Behavior Authoring in Commercial Games Project that endeavors to make behavior creation technology developed in the commercial game world available to behavior modelers in military and academic environments. He is Co-Principal Investigator for the Cultural Modeling Testbed which is a joint DMSO/ARL/AFRL project to build a software game based testbed for performing experimental research on culture and personality, and their effects on team behavior. Mr. Ferguson holds a Master's Degree in Computer Science from Yale University.

Tim Langevin joined MÄK Technologies in July 1993. As one of MÄK's most experienced senior engineers, he has worked on numerous projects involving distributed networked simulation technology. Currently, he is the lead engineer for MÄK's new Simulation Manager product. Most recently, he led the HLA federation integration work associated with DARPA's DARWARS architecture development effort. He has also been assigned as lead engineer for the development of MÄK's Computer Generated Forces (CGF) product, VR-Forces. From July 1986 until April 1993, Mr. Langevin worked for Bolt, Beranek, and Newman, Inc. While at BBN, he was the project leader of BBN's Standard Network Interface Package (SNIP) DIS toolkit and also worked on a variety of SIMNET projects. Mr. Langevin received a Bachelor of Science in Mathematics from Virginia Commonwealth University.

A Federating Protocol for Distributed After Action Review

Virginia Travers, William Ferguson
BBN Technologies
Cambridge, Massachusetts
travers@bbn.com, wferguso@bbn.com

Timothy Langevin
MÄK Technologies
Cambridge, Massachusetts
langevin@mak.com

INTRODUCTION

How can learners who have completed a distributed, virtual exercise employing different training tools participate in a combined, virtual after action review (AAR)? The current solution is for each learner to use the same AAR tool. This AAR tool might be as simple as a phone hooked to a conference call or as elaborate as distributed groupware that allows sharing of audio and visual channels. The problem with these "one-size-fits-all" solutions (where each learner uses the same tool) is that they provide functionality at the lowest common denominator, rather than supporting each user with an AAR system that is tailored to his or her own point of view and tasking. Additionally, this approach does not take advantage of any special AAR capabilities built into the training tools.

As an alternative to what we call the "homogeneous AAR", we are developing a federating protocol for AAR systems that allows users to participate in a distributed AAR while using visualization and other tools that are tailored to their own role in the experience. We call this approach the "heterogeneous AAR". Using an open federating protocol, very different systems can be coordinated during an AAR to provide a shared focus of attention and other needed "orchestration," such as the ability to direct everyone's attention to the same entity or point in time. Use of a federating protocol is an economical approach as it allows existing simulation systems to be connected without requiring re-implementation of their basic functionality.

We defined a set of XML messages that specify the interactions that must be shared: pointing or referring to entities or events; moving the point of view in space and time; highlighting specific types of entities; as well as starting, stopping, and synchronizing replay.

MOTIVATION

In order to motivate this discussion, it's useful to consider a training exercise involving different simulations. Suppose that a MAGTF (Marine Air-Ground Task Force) commander is coordinating a NEO (non-combat evacuation operation), controlling ship-based aircraft, while a convoy leader is coordinating a convoy tasked with delivering the non-combatants to a helicopter-landing site. During the simulation exercise, the MAGTF commander views a map indicating locations of all his assets, as well as the location of the convoy. At the same time, the Convoy leader sees a detailed first-person view of a town as the convoy drives to the rendezvous site.

One approach to the AAR would be to provide a single homogeneous tool, separate from the training tools used in the exercise. An example of such a special-purpose AAR tool is a map viewer that could display the area at different resolutions. When discussing co-ordination with the MAGTF, the display would encompass the town and rendezvous point, as well as the aircraft offshore supporting the rescue helicopter. When discussing an enemy attack on the convoy, the display would show a detailed, road-level view of the town. However, neither of these displays would provide the same information that the commanders had during the actual simulation. Instead, the MAGTF commander would view a slightly different, and hence, possibly confusing map of the area, while the convoy commander would be viewing a map of the town, rather than a first person view of the buildings and enemy snipers.

As an alternative, consider an AAR that is coordinated, yet allows each participant to use the same tool that they used during the training exercise. This would allow the MAGTF commander to view the MAGTF display, and the Convoy commander to view the convoy-simulation display. The instructor leading the AAR, through the use of a shared AAR protocol, directs the MAGTF tool and the convoy simulation to display their original views at the same point in simulation time. For example, the instructor might direct both tools to display the situation at the point when the enemy attacked the convoy. While the convoy leader would see the position of his convoy and the enemy combatants in the town, the MAGTF leader would see the position of his assets at that time on his original map. These two AAR approaches are depicted in Figure 1, with the MAGTF Commander using the MAGTF XXI tool developed by MÄK

Technologies, and the Convoy Commander using the Ambush tool developed by BBN Technologies.

Performing an AAR with the tools used in the simulation exercise takes advantage of the user's familiarity with the tools, while providing all the detail of the original simulation. In the exercise, participants use tools tailored to their roles in order to maximize the training benefits; for this same reason participants should use these tools in the AAR.

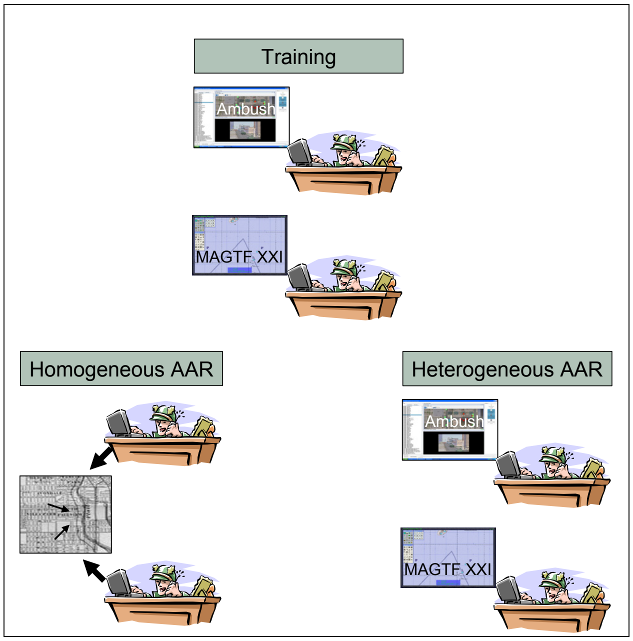

Figure 1. The Heterogeneous AAR can preserve the detail and functionality of the training itself.

The down side of this approach is the additional development required in each tool in order to support co-ordination among all the other tools. Our goal is to minimize the additional development effort needed to support heterogeneous AAR in order to achieve greater buy-in from simulation developers for this approach. There are several ways in which our approach has attempted to minimize co-ordination among heterogeneous tools:

- First, we recognize that *audio communication is an essential part of an AAR*, especially a distributed AAR. Rather than incorporate support for distributed audio communication in the AAR protocol, we suggest that audio communication be supported out-of-band via phone conferencing or VOIP (Voice over Internet Protocol) communications.
- Second, we have designed the AAR protocol to *use the existing XML standards*. The XML messages can be communicated using any method desired.
- Third, we have designed the AAR protocol with *maximum flexibility*. For example, an AAR protocol message may indicate: "display the specified geographical location at the specified point in time"; how or what is displayed is left to the individual tool. The convoy simulation might display the convoy on the streets of town, while the MAGTF tool that generally displays a higher-level view, might simply mark the convoy's location on a map. This "loose specification" allows each simulation developer to employ the information and functionality at the level of detail available in their tool, while isolating them from the specific implementations of the other heterogeneous tools.

PEDAGOGICAL ISSUES

There are many factors that might make AARs more or less effective for trainees. We focus here on those factors that are affected by the two approaches outlined above: all trainees sharing the same system during AAR (the homogeneous approach) vs. each trainee using the tool that they used in the original simulation (the heterogeneous approach). In the following discussion we will explore how AARs contribute to learning and provide advice as to when the heterogeneous approach is best and when a homogeneous approach should be preferred. A precondition for achieving training value by employing the federating protocol proposed in this paper is that the training systems being used in the experiential training event are capable of providing good AAR support tailored to the experience that each system provides.

After action review provides a critical part of the training value in experiential training. It does this through several mechanisms:

- It allows learners to re-immerse themselves in the experience and sort out what they knew when, what they were thinking, and why they made the choices that they made.
- It allows learners to see the big picture: the overall objective and course of execution and their effect on it. They can see the impact of their choices on other team members and therefore, see how actions that might have seemed correct (or superfluous) from their point of view could be harmful (or important) in the greater context.

- It allows learners to review their situation and their decisions from various vantage points (such as from the adversary's view or from an ideal view).
- It allows learners to share their experiences and learn from each other's observations, successes, and failures.

Which of these benefits are best served by a federated, heterogeneous AAR vs. a homogeneous AAR? We have not gathered empirical evidence to support strong conclusions here but we make the following hypotheses based on our observations and experience with AAR:

- Re-immersion is more easily achieved if the student again sees and hears exactly what he or she was seeing and hearing during the experience. This is strongly supported by the heterogeneous AAR enabled by our protocol.
- Getting the big picture and understanding the impact of one's actions on others may best be accomplished by seeing others' points of view or by sharing a global viewpoint. Therefore, homogenous AAR (perhaps supported by a groupware solution that allows learners to see each other's screens) best supports this goal.
- Tailored AAR tools often provide specialized points of view that will be of pedagogical value to the learner. Heterogeneous AAR excels here since it allows each user to use his or her own tools.
- Finally, group interaction may benefit from everyone seeing the same thing at the same time. This can be supported by both approaches; homogeneous AAR can provide a big picture point of view, while heterogeneous AAR combined with groupware that allows learners to share screens can allow everyone to share each learner's experience.

We argue that heterogeneous AAR is an important approach if learners are using systems that can provide tailored AAR support. There are obviously benefits to both homogeneous and heterogeneous AARs, and in the best of all worlds, both approaches would be supported. Using a groupware system that allows shared screens or windows should satisfy the requirement for homogeneous episodes during an AAR even when heterogeneous AAR systems are in use. Groupware can also allow learners to share their individual views with the group.

THE AAR PROTOCOL

In order to create a protocol that allows different AAR systems to coordinate their activities and share actions such as focusing on a specific point in space and time, we need to define a set of abstractions that capture the actions, objects, and data needed for such orchestration. We have defined this information at a high-enough level of abstraction that it is usable by very different AAR tools. In particular, we have defined an "Event," which identifies a particular moment in space and time along with a set of "interesting" objects from the simulation. The Event includes text strings: human readable annotations that systems can display to the instructors and trainees in order to provide additional information.

During the simulation, the different simulation tools must collect and store information in the form of Events (defined below). Events are specified in an XML format. Prior to the AAR, these Event specifications must be transmitted from each simulation to the instructor's workstation. The mechanism for this transport is not specified; at its simplest, it could be transferring an XML file defining the Events.

During the AAR, information flow is from an instructor, leading an AAR, to the trainees, who participated in the original simulations. In the simplest version, there is no floor control mechanism (in other words, the instructor always has floor control). Commands and data are sent from the instructor to the trainees. There is no communication from the trainees to the instructor. Commands and data are encoded in XML.

Each tool participating in the AAR must have the ability to play back events that it generated in the original simulation, and it must be able to accept and display events that other tools generated in the original simulation. Events can be displayed in any appropriate tool-specific manner, e.g., highlighting a location on a map, displaying a frame of video, etc. It is expected that a tool will be able to provide a much richer display and more detailed information for events that it generated itself, as opposed to events from another tool. At a minimum, it is expected that each tool will be able to display text annotations for events generated by other tools. In addition, each tool participating in the AAR must be able to play back the original simulation from its point of view, with the ability to start, stop and pause the playback.

The instructor can use either one of the tools originally used in the simulation, or a separate, generic AAR tool. The details of the instructor's tool are not specified, but it must have the following functionality:

- The instructor must have the ability to control time (from the original simulation); in particular, the instructor must be able to specify start and stop times, and to jump to a specified time. This capability is often implemented as a time slider, though the particular implementation is not specified.
- The instructor must be able to display Events (defined below). The format of this display is not specified, but at a minimum it should include text descriptions composed from the Events, and the ability to select Events.

The following commands can be sent from the instructor to the trainees:

- *Events:* One or more events as described below; an Event implies a Set operation (i.e., go to the time specified in the event).
- *Start:* Start event playback at a specified time (where the start of the original simulation is time zero).
- *Stop:* Stop playback.
- *Sync:* A lightweight version of Set, specifying a time.
- *Set:* Go to the specified time, and maintain your run status (i.e., either running or stopped).

Events are sent first, followed by Start, then Stop, which is followed by the next Events command. The Events that are played back with any Start command are the Events most recently sent. Playback terminates when either all of the Events are played back or a Stop command is received.

Sync commands are "time changes in progress". For example, Sync commands are generated when an instructor moves a time slider. Sync commands are "lightweight"; they may be sent fairly often (e.g. once per second), but they can be ignored.

Set commands indicate a jump in time. For example, a Set command is generated when an instructor releases a time slider. Set commands are "heavyweight"; a Set command cannot be ignored.

An event includes the following information:

- *ID:* Simulation name: event number. These are globally unique IDs. For HLA, the simulation name is the federate ID. This is a string; non-HLA systems can provide their own unique string identifier. These IDs can be referenced by other events.
- *Time:* Wall clock time; in milliseconds, relative to the start of the original simulation.
- *Event Type:* Text, initially one of Simulation, Measurement, Annotation (additional event types can be easily supported if necessary). Measurement and Annotation events are not yet defined
- *Description:* Text, an open-ended explanation of the event in the simulation; this should characterize the action in the simulation; for example, walking, shooting, flying; this text is displayed for the instructor, to help the instructor select relevant events.
- *List of Subject Locations:* For each subject:
  - *Entity ID:* A unique object identifier. For HLA simulations, this is the HLA object identifier. This is a string; non-HLA systems can provide their own unique string identifier.
  - *Entity name (optional):* If this is provided and if the simulation can interpret this, it is intended to indicate the object in the simulation on which to focus, i.e., a truck, person, plane, etc.
  - *Location:* Latitude, longitude, altitude, annotation; latitude and longitude are in decimal degrees; elevations are ASL (above sea level).
  - *Annotation:* Open ended text identifying this entity, i.e., truck, plane, person; this text is displayed for the instructor.
- *List of Object Locations*; for each object: (Entity information is optional; an object may simply be a location. "Object" here is used in the sense of Subject-Verb-Object, for example, an event might be "Leader Shoots Sniper" or "Helicopter Lands at Rendezvous Point".)
  - *Entity Id:* A Unique object identifier. For HLA simulations, this is the HLA object identifier. This is a string; non-HLA systems can provide their own unique string identifier.
  - *Entity name (optional):* If this is provided and if the simulation can interpret this, and is intended to indicate the object in the simulation on which to focus, i.e., a truck, person, plane, etc.

- *Location:* Latitude, longitude, altitude, annotation; latitude and longitude are in decimal degrees; elevations are ASL (above sea level).
- *Annotation:* Open-ended text identifying this entity, i.e., truck, plane, person; this text is displayed for the instructor.

- *Point of View:* location of viewer , specified by pan, yaw, pitch, roll and zoom.
- *Simulation-specific Information:* Open ended text identifying this event in this simulation; for example, this can be used to specify an identifier that is matched to additional data maintained by the simulation.

The inclusion of entity information is optional. If an entity is specified, then AAR tools should focus on that entity in their playback. Annotations are text that will be displayed in each tool. This information will help the instructor select AAR events of interest. The simulation-specific information is included to allow each training system to produce a richer display for the training events that it originally produced. For example, the AAR events might include an event generated by the MAGTF tool and an event generated by the convoy tool. We expect that the convoy tool will be able to provide a detailed display for the event that it originally produced, but only a "skeletal" display (i.e., an X on a map, a mark on a timeline, a text annotation) for the event originally produced by the MAGTF tool.

The entity within an object is optional; an object may simply be a location, as for example, when ordinance is dropped on a location. Locations are XML elements that include latitude, longitude, altitude, and an optional annotation. Note that annotations can appear both within a location and within an entity. See the example below.

The protocol attempts to optimize performance in a distributed environment. Optimization is done in two ways: first, the amount of information that must be exchanged is minimized. For example, to specify a location of interest, the instructor station must send the latitude, longitude, and elevation; it is assumed that each trainee's workstation will either have appropriate maps pre-loaded, or will be able to obtain maps from a local (as opposed to wide-area) source in real time. Second, the number of exchanges between the instructor and trainee workstations is minimized. During playback in the AAR, it is not critical that all the workstations remain in time sync. Messages are sent periodically to help maintain synchronization, but the workstations are not bound to synchronize to these times. Playback can be free running, operating independently at each workstation, with synchronization only done with "set" commands, i.e., when an instructor specifically directs the AAR to an interesting and significant point in time.

HLA EXAMPLE

The Battlefield Tactical Trainer (BTT) AAR Tool, developed by MÄK Technologies, focuses on distributed, collaborative AAR for heterogeneous trainers using HLA and the RPR FOM. Using the BTT AAR, students and instructors can work together to create the best possible understanding of what transpired in a training exercise. The AAR tool makes it easy to point out events, areas, and views of significant interest amongst participants. Both instructor and students are able to review key events; replay part (or all) of the exercise; review annotations; see other users' views of the exercise; and direct other users to their own current view. Instructors can add and save annotations for both AAR preparation and subsequent AAR archival.

When the BTT AAR tool is launched, it automatically generates a list of events using logging information collected during the training exercise. These events currently include weapon discharges, detonations, and entities destroyed. The event format used is the same as the XML format described below. These XML events are stored as annotations to the original log; the instructor can also create additional annotation events. While the events are used for display and navigation, the original logger recording can still be used to play back the recorded exercise from any point in time (and at any speed).

An XML-based protocol is used to provide control and user information among BTT AAR clients. The XML commands are transmitted using HLA / RPR Comment Interactions. The protocol itself is a superset of the XML protocol outlined below, and includes the following messages:

- *start:* Corresponds to the XML start message; indicates that the logger on the receiving client should begin playback at the time indicated in the message.
- *stop:* Corresponds to the XML stop message; indicates that the logger playback on the receiving client should stop (this is treated as a pause so the entity display is not lost).

- *set:* Set one of the following parameters:
  - *time:* Corresponds to the XML *set* message; indicates that the logger playback time on the receiving client should be changed to the time indicated (logger may be playing or paused).
  - *timescale:* Indicates that the logger playback timescale on the receiving client should be adjusted as indicated (negative values can be used to play in reverse).
  - *extent:* Indicates the map display on the receiving client should be set to the indicated extents (these are provided as northeast and southwest corners of the new display area, in Geocentric Coordinates).
  - *highlight:* Indicates that the receiving client should highlight one or more locations and / or entities.
- *select-event* – Indicates that the receiving client should show the specified event as selected (note that the event is identified by its event ID).
- *player:* Tells a client about the presence of another client, and the student or instructor name associated with it (sent as a heartbeat so that client exits can be detected).
- *offer:* Indicates that the sending client is offering to take control of the receiving client (note that start, stop, and set commands are ignored unless the receiving client is under the sending client's control).
- *accept:* Indicates that the sending client has accepted control from the client to which this message is sent (sent in response to an offer message).
- *reject:* Indicates that the sending client has rejected control from the client to which this message is sent (sent in response to an offer message).
- *release:* Indicates that the receiving client should no longer be controlled by the sending client.
- *break:* Indicates that the sending client chooses to no longer be controlled by the receiving client.

Optionally, the BTT AAR Tool can establish a TCP connection with a non-HLA-based instructor's AAR tool. Through this connection, it can accept XML events and commands as described below. Events received from an external AAR tool are added to the BTT AAR Tool's event list.. When external events are received, the BTT AAR tool automatically sets its internal logger to the time of that event and sets its display to focus on the entities or locations specified in that event.

XML EXAMPLE

Figure 2 shows an excerpt of the XML information exchanged in support of the AAR demonstration that we conducted at I/ITSEC 2005. The XML describes a single Event involving two entities, which are trucks in a convoy operation.

THE I/ITSEC DEMO

This AAR protocol was implemented in several simulation tools and demonstrated at I/ITSEC 2005. The tools included:

- *The Battlefield Tactical Trainer AAR Tool* (described above, a tool for visualizing Joint Task Force operations from MÄK Technologies).
- *DARWARS Ambush* (a training tool developed by BBN Technologies for simulating convoy operations).
- *STRATA* (a training tool developed by Chi Corporation for simulating calls for Close Air Support).
- A training tool for simulating naval operations (from Sonalysts).

The scenario involved the evacuation of non-combatants from a combat zone. A convoy, collecting non-combatants from a town, coordinated with ship-based aircraft to effect their evacuation. When the convoy was attacked, it coordinated with aircraft to provide air support for the operation. In the AAR, an instructor used a "generic" AAR tool that implements the AAR protocol allowing him to start, stop and pause playback at the trainees' workstations, and to send significant events to the trainees' workstations where their individual simulation tools displayed the information. The following is a synopsis of the Events from the joint AAR demonstration.

- Convoy enroute to the hotel comes under fire.
- Convoy reports erroneous location.
- Air cover called in to wrong location.
- Air cover cannot find targets.
- The overall commander communicates with ground, ship, helicopter, and aircraft commanders via radio to resolve error.
- Successful close air support operation.

```
<?xml version="1.0" encoding="UTF-8"?>
<aar>
<event>
  <id>convoy:3</id>
  <time>53</time>
  <type>simulation</type>
  <simulation event type>action</simulation event type>
  <description>lead truck arrives at rendezvous point</description>
  <subject>
     <entity>
         <id>Convoy:1</id>
         <name>truck</truck>
         <latitude>42.74</latitude>
         <longitude>73.44</longitude>
         <elevation>10</elevation>
         <annotation>the lead truck in the convoy</annotation>
     </entity>
  </subject>
  <object>
     <entity>
         <id>Convoy:2</id>
         <name>truck</truck>
         <latitude>42.77</latitude>
         <longitude>73.44</longitude>
         <elevation>10</elevation>
         <annotation>the last truck in the convoy</annotation>
     </entity>
 </object>
 <pointofview>
  <pan>270.5</pan>
  <yaw>10</yaw>
  <pitch>20</pitch>
  <roll>30</roll>
  <zoom>1.5</zoom>
 </pointofview>
 <trainingsysteminfo>ID=43</trainingsysteminfo>
</event>
</aar>
```


Figure 2. XML example, showing an AAR excerpt.

These events were generated by the various simulation tools during the original training or were added by an instructor who annotated the information in the process of preparing the AAR. In the AAR, each tool represented the event using that tool's own capabilities, i.e., displaying text, highlighting a location on a map, playing a frame from the simulation, etc. For the demonstration, AAR participants were co-located in a conference room on the exhibit floor; in a real-life AAR, participants in different locations would communicate via phone or VOIP to discuss events as they were selected and reviewed by the instructor.

FUTURE

There are several possible extensions to the AAR protocol. The protocol should be extended to support floor control, effectively allowing each trainee to take on the instructor's role in directing the AAR. Oftentimes an individual can contribute a unique and important perspective during the AAR, based on their experience in the original training event. Functionality required for shared floor control includes: registering a request to take the floor, designating the new floor controller, relinquishing the floor, and forcibly removing the floor controller. Note that there are two capabilities here: having the floor and controlling who has the floor. It may be desirable to allow any trainee to have the floor, while only the instructor controls who has the floor.

Another possible extension is to define classes of events (as opposed to single events) or regions of interest (as opposed to pinpoint locations), and ranges of time in the protocol in order to support a richer AAR. For example, the instructor may want to point out all events related to the lead convoy vehicle over a period of time, or all events that occur within a quarter mile of the rendezvous point in the half hour preceding the rendezvous time.

MANAGEMENT OF AAR GROUPS

Another area for investigation involves how to conduct a relevant and compelling AAR with disparate groups. Simply playing back what happened over time may leave some groups bored ("we weren't involved in the action until much later; this has nothing to do with us"). In a joint operation, one approach would be to conduct an "overview" AAR for all participants, followed by multiple, simultaneous "small group" AARs focused on each group's activities (e.g., the convoy, the rescue helicopter), and finishing with an AAR for all participants in the joint activities (e.g., the convoy calls for air support). This dynamically changes the underlying communications topology, changing between an instructor communicating with all trainees, and multiple instructors communicating with disjointed groups of trainees. Although in theory this approach could provide a more interesting and time-efficient AAR, there is much work to be done to provide appropriate tools and procedures for instructors and trainees to run such an AAR smoothly in a distributed environment.

WHERE DO WE GO FROM HERE

Our future plans include:

- Gathering community input on methods for conducting AARs for simulations that employ heterogeneous tools.
- Further prototype development of the AAR protocol.
- Test deployment in support of an AAR in a "real life" exercise.
- Tracking and contributing to standardization efforts related to AARs.

CONCLUSION

The best distributed AARs might come from employing a heterogeneous set of after action review tools coordinated through an orchestration protocol such as the one proposed here along with a groupware solution that allows learners to see each other's or the instructor's screen at those times in the AAR when a common point of view is desirable. Additional work in real life settings is required to verify the benefits of this approach.

REFERENCES

Garvin, D. (2000) *Learning in Action, A Guide to Putting the Learning Organization to Work*, Boston: Harvard Business School Press.

Department of the Army, (1993). *A Leader's Guide to After Action Reviews*, Training Circular 25-20, 30 Sept.