Enabling Automated AAR Development by Abstracting Data Collection from Analysis

David Yu Chen, Randy Jensen, Oscar Bascara
Stottler Henke Associates, Inc.
San Mateo, CA
chen@stottlerhenke.com, jensen@stottlerhenke.com, bascara@stottlerhenke.com

Nancy Harmon
USMC PMTRASYS
Orlando, FL
nancy.j.harmon@usmc.mil

ABSTRACT

Simulation based training provides not only the benefits of immersion and interactivity during exercises, but also the prospect of automated after action review. As trainees interact with the system and with each other through various interfaces, the resulting body of data can be used to automatically draw instructional conclusions that go well beyond traditional measures of effectiveness. However, complex team training architectures often incorporate or support an entire suite of tools and interfaces with diverse protocols and data conventions. This presents a technical challenge for the development of decision-oriented automated after action review, which can be solved with an abstracted data collection and representation scheme that is compatible with all potential supported interfaces.

This paper describes an agile approach for handling analysis data, developed for the Marine Corps' Combined Arms Command and Control Trainer Upgrade System (CACCTUS). The goals of scalability and modularity target a range of data sources for this application, including the OneSAF Objective System, integrated C4I tools and human-in-the-loop interfaces, and virtual radios on which spoken transmissions are processed with speech recognition tools. Fundamentally the data analyses in a training system depend most on knowledge about the kinds of available data, and less on the collection mechanism itself, which can therefore be abstracted. A consequence is that the data analysis algorithms can be implemented in parallel with the various data collection methods for each integrated tool. Also, for any new requirement to integrate with an additional interface that was previously unsupported, the only implementation requirement is in the data collection code that writes to the repository, with little or no change on the analysis side. This paper provides design details and lessons learned from the CACCTUS effort, and summarizes the more general methodology for abstracting data collection from data analysis in training systems.

ABOUT THE AUTHORS

David Yu Chen is a lead software engineer at Stottler Henke Associates, Inc., currently working on the after action review component of the Marine Corps CACCTUS program. He holds an M.S. in Computer Science from Stanford University.

Randy Jensen is a group manager at Stottler Henke Associates, Inc., working in training systems since 1993. He has developed numerous Intelligent Tutoring Systems for Stottler Henke, as well as authoring tools, simulation controls, after action review tools, and assessment logic routines. He is currently leading projects to develop automated after action review for Marine Corps combined arms training, a framework for ITS interoperability with distributed learning architectures for the Joint ADL Co-Lab, and an authoring tool for virtual training demonstrations for the Army. He holds a B.S. with honors in symbolic systems from Stanford University.

Oscar Bascara is a software engineer at Stottler Henke Associates, Inc. His interests include simulation behavior modeling and user interface design. He holds an M.Eng. in Electrical Engineering from Cornell University and an M.A. in Mathematics from the University of California at Berkeley.

Nancy Harmon is the Marine Corps Systems Command, Program Manager Training Systems, Project Officer for the CACCTUS project and the originator of the After Action Intelligent Review System (AAIRS) concept. She has over 21 years of experience in the area of training systems design and development for the Department of Defense.

Enabling Automated AAR Development by Abstracting Data Collection from Analysis

David Yu Chen, Randy Jensen, Oscar Bascara
Stottler Henke Associates, Inc.
San Mateo, CA
chen@stottlerhenke.com, jensen@stottlerhenke.com, bascara@stottlerhenke.com

Nancy Harmon
USMC PMTRASYS
Orlando, FL
nancy.j.harmon@usmc.mil

INTRODUCTION

Simulation based training provides not only the benefits of immersion and interactivity during exercises, but also the prospect of automated After Action Review (AAR). As trainees interact with the system and with each other through various interfaces, the resulting body of data can be used to automatically draw instructional conclusions that go well beyond traditional measures of effectiveness. Automated detection of key training points allows instructors to choose among them for emphasis in the after action review, and also focus on catching non-standard training points that may also merit attention.

However, complex team training system architectures often incorporate or support an entire suite of tools and interfaces with diverse protocols and data conventions. This presents a technical challenge for the development of decision-oriented automated after action review, which can be solved with an abstracted data collection and representation scheme that is compatible with all potential supported interfaces.

Fundamentally, the data analysis in a training system depends mostly on the knowledge of the performance measures and the kinds of available data to support them, and less on the collection mechanism itself. A consequence is that the data analysis algorithms can therefore be abstracted and thus be implemented in parallel with the various data collection methods for each integrated tool. Also, for any new requirement to integrate with an additional interface that was previously unsupported, the only implementation requirement is in the data collection code that writes to the repository. This allows for little or no change on the analysis side.

This paper describes a general, agile methodology for handling training system data, both in terms of structuring the components that collect or analyze data, and performing system design and development. The methodology was developed for the Marine Corps' Combined Arms Command and Control Trainer Upgrade System (CACCTUS). In addition to examining the methodology, this paper discusses lessons learned and applicability to other training systems.

BACKGROUND

Simulation based training provides an environment in which to practice decision making skills in a training exercise, often controlled through instructional interfaces. A simulation system typically models the state and actions of entities or units under the control of the exercising force (EXFOR), instructors, or semi-automated model behaviors in the simulation itself. However, simulations are often only a part of a training system toolset. Depending on the desired or best method of training certain types of objectives, a simulation system may not be able to provide all the data required for an after action review. Simulations such as OOS (OneSAF Objective System), OTB (OneSAF Testbed Baseline), and JSAF (Joint Semi-Automated Forces) serve as powerful technological tools, yet for each there are examples of data not available for an after action review, which could be useful for training purposes.

Data Collection for Statistics

After action review systems often calculate statistical measures of performance. Many simulations calculate and report these measures natively. Other statistics are typically gathered from simulation states emitted using the DIS or HLA standards. DIS (Distributed Interactive Simulation) provides a format for sharing simulation states with formats defined for entity state, weapon effects, etc. (IEEE, 1995). HLA (High Level Architecture) specifies an architecture allowing the sharing of system states as defined by simulation-specific simulation object models (IEEE, 2000).

Statistics typically collected are exemplified by the Dismounted Infantry Virtual After Action Review System (DIVAARS). This tool provides the following statistics in its after action review: rounds fired by time, total rounds fired, kills inflicted by time, killer range, total kills inflicted, and movement rates (Clark & Lampton, 2004). DIVAARS processes logged DIS data to generate its statistics.

The Future After Action Review System (FAARS) provides a sophisticated facility to acquire virtually any statistic from the JSAF simulation (Graebener et al, 2003). For example, FAARS provides a sensor/target scoreboard both during the exercise and in after action. FAARS uses Raytheon VTC's hlaResults® COTS product to store HLA communications from JSAF into a database for processing.

Data Collection for Playback

After action review systems also commonly provide a playback function to allow an instructor to replay an exercise or event to show the ground truth at critical times in an exercise. In fact, this is the primary method for illustrating training points in many systems. Typically, the playback is accomplished via logging DIS or HLA simulation data and playing it back in stealth viewer tools with DVD-like controls and time sliders for the training audience. Playback also often involves data from other components of the training system, particularly communications.

DIVAARS provides playback functionality in a 3D virtual environment (Clark & Lampton, 2004) using DIS logs and includes synchronized audio communications.

The Aviation Combined Arms Tactical Trainer – Aviation Reconfigurable Manned Simulator (AVCATT-A) provides playback of simulation states and events, voice communications, operator/controller generated reports, and sensor readings (Knight et al, 2001). AVCATT-A records DIS updates from OTB and digital video of the sensor readings of the simulator. The operator/controller combines these recordings with his own reports for the after action review.

Data Collection for Identification of Training Points

Several after action review tools provide mechanisms for identifying training points with a decision making component. These systems require significant additional information beyond simulation data in order to determine EXFOR situational awareness and decision making.

The Virtual Soldier Skills Assessment (ViSSA) system provides support for assessing Soldier situational awareness (Lampton et al, 2005). ViSSA provides a scripting capability to augment an exercise with situational awareness measurements based on data from the simulation and subjective measures from the exercising force. For example, when the system identifies a condition where a trainee's avatar enters a room, it may prompt an observer or instructor to ask the appropriate trainee if he sees any enemies. Depending on the response, the event may be marked as significant and marked for playback during the after action review. Data from the simulation is acquired from DIS; data from the trainee is entered by an operator.

The After Action Intelligent Review System (AAIRS) system developed for CACCTUS automatically generates training points related to battlespace geometry conflict conditions occurring in simulated combined arms operations (Jensen et al, 2005). These conflict conditions are based upon a parametric rule-set that can be adjusted in accordance with the training site standard operating procedures, rules of engagement, or specific training goals. By collecting mission information directly from the system's human-in-the-loop interfaces, as well as DIS states from the simulation, AAIRS detects not only hazardous situations that occurred in simulation execution, but also hazardous situations that might have taken place as a result of EXFOR decisions, irrespective of simulation outcomes. An example of a training point based on simulation state data is the condition where an aircraft entity in the simulation flies through the Minimum Safe Distance (MSD) danger area associated with an artillery shell detonation. A decision-oriented counterpart to this example is the condition where an aircraft flies through the MSD between actual detonations, thereby avoiding the conflict, but only by chance rather than a result of proper EXFOR decision making or interventions. Both forms of training points are identified in AAIRS.

Data Collection for Causal Analysis

After action review tools which attempt to provide causal analysis functions also require significant additional information beyond traditional simulation state data in order to determine causality.

Explainable Artificial Intelligence (XAI) provides a dialog-based mechanism that allows users to investigate decision making by SAF-controlled entities (Gomboc et al, 2005). For example, after an exercise, a trainee may inquire directly with a SAF-controlled enemy he encountered during the exercise, to learn what task the entity was performing and understand why the enemy reacted the way it did. Proof of concept implementations have been developed with Full Spectrum Command and OOS. For these simulations, SAF behaviors, scenarios, logs, and some human-supplied data for an exercise were entered into an XAI database.

The AAIRS system discussed earlier automatically generates a summary of potential causal factors to augment training points generated from an exercise (Jensen et al, 2005). AAIRS uses a combination of natural language processing on communications data and review of human-in-the-loop (HITL) inputs in order to trace the links between EXFOR decision making and exercise events. For example, in a conflict involving an aircraft and an artillery MSD, the AAIRS causal analysis searches EXFOR inputs for the artillery mission approval communications from the Fire Support Coordination Center, the aircraft clearance communications from the forward air controller, and any flight altitude restrictions entered for the aircraft mission via HITL interfaces.

Developing after action review systems becomes increasingly more difficult as the objectives for debrief content scale from statistics and playback to the identification of decision-driven training points and causal analysis. When the goal is to provide training on decision making in addition to outcomes, and in fact to draw connections between the two, it becomes necessary to collect and analyze data from all possible input sources, starting from the interfaces that EXFOR use to directly interact with each other and with the training system, and ending with the event data provided by the simulation. This task is particularly challenging when a program targets a large range of data sources, such as integrated C4I tools, HITL tools, virtual radios, and the simulation. And the complexity is compounded when trying to target a large, multi-echelon training audience. The following sections describe the abstracted approach developed in AAIRS to meet these objectives.

SYSTEM DATA ARCHITECTURE

AAIRS is constructed around the training and technical requirements of the Marine Corps' CACCTUS program. CACCTUS will support a live, virtual, and constructive training environment that facilitates the command and control interactions normally conducted in the tactical environment in real-time combined arms operations. Combined arms exercises provide training and rehearsal for coordinating multiple supporting arms with maneuver. Training exercises may involve 100 or more participants at various stations in a single facility, or even more in distributed exercises across sites. There is an emphasis on providing experiential training by requiring the EXFOR to perform responsibilities during training events which mirror those during operational actions. Training events require communication and coordination skills in the employment of tactics, techniques, and procedures in support of specific scenario goals. Exercising units may make use of equipment in facilities, or augment it with their own C4I devices.

Figure 1 below shows how the data collection capabilities are structured within the overall CACCTUS architecture to support after action review with separate components for data collection and data analysis.

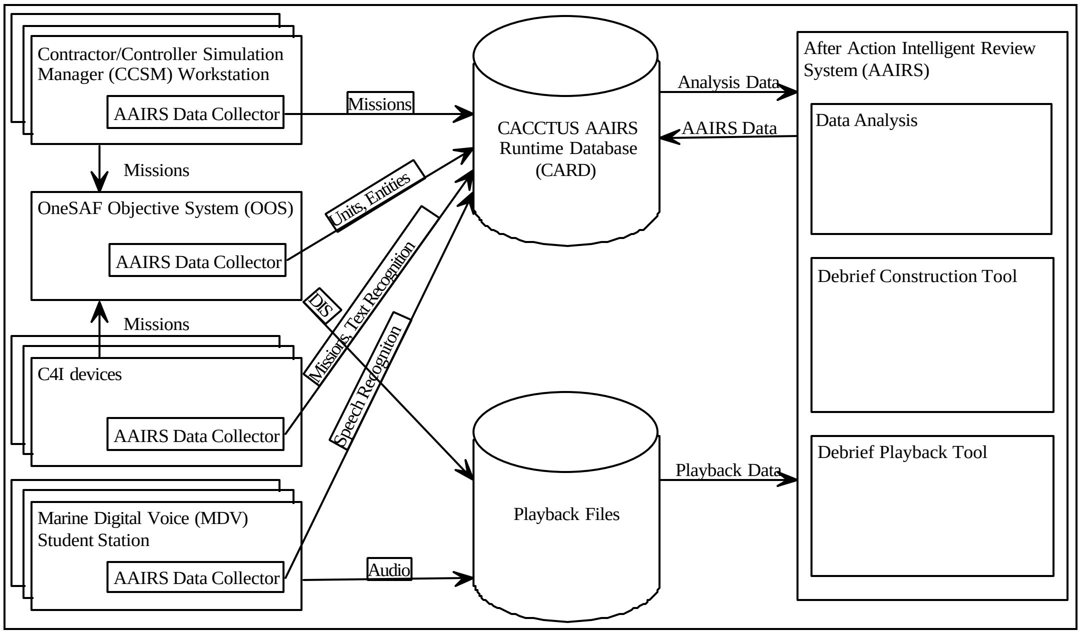

Figure 1. Data Analysis Architecture. Data collection is shown on the left and data analysis is shown on the right

The overall data analysis architecture is generally applicable for training systems containing data repositories, a simulation system, communications systems, human-in-the-loop interfaces, and an after action review component.

Data Repositories

Data from the training system is stored into two separate data repositories for playback and analysis data. Playback data consists of a recording of all transmitted simulation and communications data. This data is used only for providing playback during the after action review and stored directly onto the file system in a format optimized for live capture and later sequential playback with cueing. Analysis data contains all data necessary for training point identification and causal analysis during an exercise. Analysis data includes concise representations of simulation and communication data (e.g. the direction and speed for a moving unit versus periodic position updates for individual entities). After an exercise, analysis data is also available for review and presentation. The CACCTUS AAIRS Runtime Database (CARD) stores all analysis data in a relational database optimized for queries.

Simulation System

The OneSAF Objective System provides an entity-level simulation for CACCTUS. The simulation shares the location and timing of entities, fires, and detonations using the DIS and HLA standards. This data is used for both playback and analysis. In addition, the simulation is instrumented to provide data not necessarily provided by DIS and HLA such as unit and entity names and hierarchical organization.

Human-in-the-loop Interfaces

Commonly, HITL interfaces provide a mechanism to inject human decision making into the simulation. These range from tools that provide a graphical user interface into the simulation system, to interface software embedded with hardware systems. These interfaces must be instrumented to provide analysis data in order to capture the original intent of the training audience. Raw inputs from EXFOR may be captured by the HITL interface, but not necessarily preserved by the simulation with the data it uses to execute events. For example, the requested angle of fire for an indirect fire may be entered into a HITL interface as a standard input for a fire mission, but may not be used by the simulation if its models only represent a gun-target line and not actual trajectories of shells.

In the CACCTUS architecture, the requirement is to support both types of HITL interfaces. A component named the Contractor/Controller Simulation Manager (CCSM) provides a graphical user interface for entering EXFOR-defined missions into the system. Individual workstations within the CCSM control indirect fire, close air support, and ground maneuvers. The CCSM workstations are instrumented with a data collector to provide mission information and controls (execute mission, cancel mission, etc.) as entered by operators on behalf of the EXFOR, and deliver it to the CARD database. In addition CACCTUS also supports the use of integrated C4I devices such as AFATDS (Advanced Field Artillery Tactical Data System) to provide input into the system.

Communications Systems

The system data architecture also incorporates voice and text communications systems to capture decision making for causal analysis. The EXFOR communicate over virtual radios, which are provided on student stations with local speech recognition capabilities. This allows for spoken content to be extracted from virtual radio transmissions, with the results stored into the analysis database while the original audio is also recorded for playback. C4I devices with text messaging provide a similar window into EXFOR communications to be collected for after action.

After Action Review Component

The after action review component processes the data collected for analysis and may generate additional analysis data into the database. Notably, AAIRS creates database entries for automatically generated training points, causal factors, condensed route information, and playback display artifacts into the analysis database. In addition, the after action review component contains mechanisms to customize and perform playback.

The data analysis architecture provides a framework sufficient for supporting an after action review that reports on decision making. However, the diversity of the data sources and the complexity of the data further require an effective methodology for handling analysis data in a manner that isn't tightly bound by the data collection requirements of individual components.

DATA COLLECTION AND ANALYSIS METHODOLOGY

The development of an automated after action review system requires analysis data from all components of a system in order to automatically generate training points and causal factors, as discussed in the previous section. This development typically exhibits the following challenges:

- The system contains a large number of components from both simulation and communications sub-systems. Each component requires a separate interface.

- Significant data is required from each component to convey the actions and inputs of the exercising force, not simply the execution by the simulation system. This data includes interactions with user interfaces that are typically not shared with other components.

- Some components may be under development simultaneous with the development of the after action review system.

Meeting these challenges requires a highly agile methodology for data definition.

Agile programming focuses on rapid software development (Beck et al, 2001). Typical agile practices include frequent releases of software; the close collaboration between requirement-makers and developers; simple, elegant design; and the reevaluation of requirements with every release. Agile programming thrives with complex development environments and changing requirements.

The abstraction of data collection from data analysis provides the key to a highly agile data definition methodology. Ultimately, the functional requirements for both the training system and after action review are based on a common source: the tactics, techniques, and procedures for the training audience. This common source allows the data collection and data analysis to be decoupled and developed rapidly in parallel.

This section compares an agile, abstracted development methodology with a standard spiral development methodology, using an example of data collection and analysis to detect training points involving artillery Minimum Safe Distances (MSD).

Standard Spiral Development Methodology

Under the standard spiral methodology, a strict sequence of steps takes place from determining requirements to testing the analysis.
Figure 2 shows the sequence in a standard spiral development pattern.

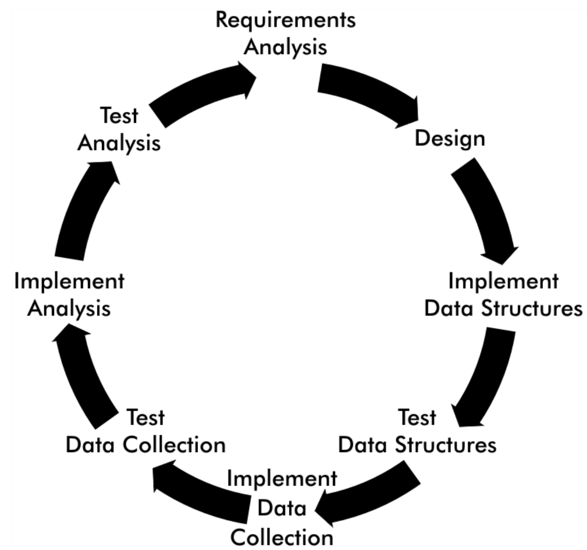

Figure 2. Spiral Development Methodology

Requirements Analysis: First, use cases and a software requirements specification are developed for the new functionality. For example, use cases would include the detection of simulation entities within an artillery danger area hemisphere when a detonation occurs in the simulation. The requirements would include acquiring artillery fire mission data to support the modeling of artillery danger areas.

Design: Following the requirements analysis, a schema for the data, the mechanism for data collection, and the mechanism for the analysis would be created. For example, the structure for an artillery fire mission, the mechanism for acquiring the mission data from the simulation or HITL interface, and the analysis logic would all be defined.

Implement and Test Data Structures: After designing the new functionality, the new structure for the analysis data would be defined and tested. Often, the data structure must also be implemented within an intermediate database.

Implement and Test Data Collection: With the appropriate data structures created, the data collection from the training system to the intermediate database / after action review can be implemented and tested.

Implement and Test Analysis: Finally, the step of reading and analyzing the data can be implemented and tested with actual collected data.

Agile Abstracted Methodology

Under the methodology developed in AAIRS for the CACCTUS program, the steps of the spiral development are restructured with separate paths for data collection and data analysis.
Figure 3 below shows the sequence as modified for the agile methodology.

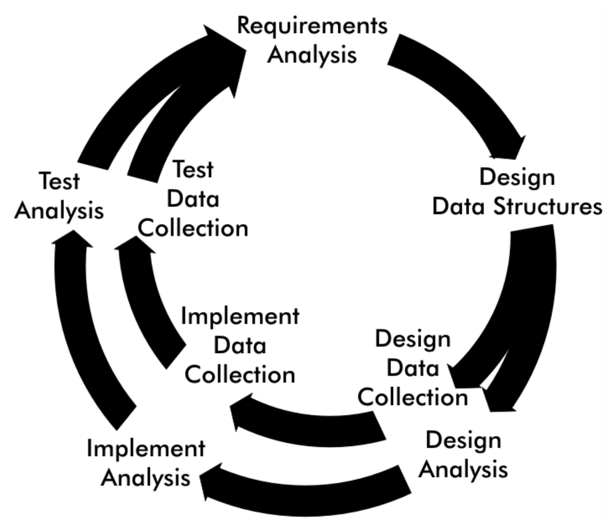

Figure 3. Agile Methodology

Requirements Analysis: The first step of requirements analysis remains largely the same as with standard spiral development. However, requirements may be revisited at any time during development of the data collection and analysis for incorporation in the next iteration of development.

Design Data Structures: Following the requirements analysis, the first step is to design the data. The data is described in a spreadsheet (see Table 1 for an example of the data structures for an artillery mission). Scripts developed for AAIRS read the spreadsheet and automatically generate the database schema, data structures, and tests. Notably, data structures contain standard mechanisms to insert values into the database, to read values from the database, and to notify an analysis of updated entries. This immediate implementation and testing of data structures allows the data structures to be readily and rapidly modified for the parallel development of data collection and analysis.

Table 1. Indirect Fire Mission Specification

Field	Type	Description
id	UUID	Identifier for Attack.
unit	UUID	Identifier for Unit performing Attack.
tot	TIMESTAMP	Planned (simulation) time of detonations on target.
attackDuration	INTEGER	Duration of Attack in seconds.
warningOrder	WarningOrder Enum	Warning order of Attack. (E.g. "Immediate suppression", "Fire for effect", etc.).
methodOfControl	MethodOf ControlEnum	Method of Control of Attack. (E.g. "At My Command", "TOT", etc.).
aimLatitude	DOUBLE	Location of the aim point.
aimLongitude	DOUBLE	Location of the aim point.
aimElevation	DOUBLE	Location of the aim point elevation in meters above mean sea level.
ammoDescription	VARCHAR(127)	Description of ammunition type.
numberRounds	INTEGER	Total number of rounds.
angle	AngleEnum	Angle of Attack. Values are H(igh), L(ow), or null for default.
charge	ChargeEnum	Charge used for Attack. For Arty, values are 1G, 2G, 3G, 4G, 5G, 3W, 4W, 5W, 6W, 7W, 8, or null for default.

Design, Implement, and Test Data Collection: With standard mechanisms to insert data into the database, the design and implementation of data collection focuses on retrieving data from the training system. After implementation, the data collection is tested both in isolation as well as with regression testing of existing analyses.

Design, Implement, and Test Analysis: A key dividend from this methodology is that the development of analysis logic can be carried out in parallel or even preceding the development of the data collection layer for any specific components. With standard mechanisms to retrieve data, the design and implementation focuses on the training requirements, and therefore the *information* that must be analyzed for the after action review, as opposed to *data*. After implementation, the analysis logic is tested both in isolation as well as with regression testing of existing and future data collection.

By separating data collection and data analysis, this methodology provides a strategy for incorporating the data from the many elements of a training system. For example, data collection from only the simulation can provide status for units and basic mission data (e.g. target location, mission start time, and mission end time). Adding data collection from HITL interfaces fills in additional mission information originating from the training audience but not available natively from the simulation (e.g. angle and charge). Because the analysis logic is developed independently from the data collection, the analysis can be easily adapted to handle new or changing components.

LESSONS LEARNED

The agile methodology provides flexibility during the development process. With the separation of data collection from data analysis, it is possible for development on the after action review component to continue while fundamental design decisions in the other components of the system architecture are still being addressed. In particular, conflict detection logic for indirect fire minimum safe distance and trajectory battlespace geometries developed first with simulated missions before data collection was available. Later, the mechanism to collect data about the indirect fire missions from the simulation was implemented. Later still, the mechanism to collect additional related data from HITL tools was constructed. This allowed the identification and request for data missing from the simulation, such as the angle to fire indirect fire, and also provided a means of ensuring the robustness of both the data collection and data analyses. At each stage, the analysis required only minimal changes, such as a review to ensure that firing units were identified identically.

Similarly, this methodology provides a strategy for incrementally incorporating the many parts of the training system. AAIRS started with a prototype that processed DIS from the simulation and speech recognition results from the virtual radio communications. Next, AAIRS developed data collection for indirect fire from the simulation and more robust speech recognition. Afterwards, data collection transferred to the HITL workstation interface with indirect fire, ground maneuver, and close air support. Currently, data collection for C4I devices is planned for AAIRS with missions similar to those created by the HITL workstation interfaces and messaging similar to content created through the virtual radios. Minimal changes are expected to the data structures and analysis logic.

The flexibility of the methodology extends to updating analyses to account for unanticipated or updated capabilities in the training system. Often, this involves generalizing the data collection and analysis. For example, indirect fire missions may involve a sequence of adjusted target positions in consecutive fires. This level of complexity was not captured in the initial representational scheme for indirect fire, which was oriented toward a premise of one mission, one target location. So this notional mission structure required an adaptation to the indirect fire data structures, to be organized in a manner that treats each adjustment as a separate mission with a single target location. Under this scheme, additional data collection provides information used by the analysis logic to relate the individual adjustments back to the primary fire mission for debrief purposes in after action. As a result, this approach for handling both data collection and data analysis for indirect fire missions can accommodate a wider variety of source data. With the data collection layer responsible for decomposing a sequence of adjustments into the data structures used by the analysis logic, it remains transparent to the analysis code whether or not the original source data structures handled individual adjustments separately.

The key to the methodology presented in this paper is the separation of data collection and data analysis and the flexibility to support changes in either. This quality suggests the data analysis methodology extends to supporting many training systems and analyses for after action review. For a given training domain, the initial steps involve a high level review of the kinds of analysis that are needed, and what *information* is required to perform the associated analysis logic. Although the data structures are also initially designed to accommodate the known nature of actual data from system components, this is only one factor in the representational scheme, where the decision making and analysis requirements weigh heavier. Once initial data structures are implemented, the analysis logic can be developed and tested in parallel with the specific implementations of the data collection layer for any training system component that will contribute information in exercises. An iterative refinement of data collection mechanisms, data structures, and analysis logic is inevitable, but greatly facilitated when a layer of abstraction between these enables parallel and flexible development.

Continuing work on AAIRS for the CACCTUS program will require the addition of data collection support for further HITL tools, specifically making use of the abstracted approach already in place. An initial prototype of the after action review toolset has been delivered to a Marine Corps training site, and will serve as a basis for feedback from users and the training command.

ACKNOWLEDGEMENTS

The CACCTUS effort described in this paper is sponsored by the USMC PMTRASYS through a contract with Cole Engineering Services, Inc., and with additional collaboration from the NAVAIR Orlando Concept Development and Integration Lab.

REFERENCES

Beck, K. et al. (2001). Manifesto for Agile Software Development. Retrieved from http://www.agilemanifesto.org/

Clark, B.R., & Lampton, D.R. (2004). Dismounted Infantry Virtual After Action Review System (DIVAARS) Users Manual. Retrieved from http://www.vr.ist.ucf.edu/divaars/

Gomboc, D., Solomon, S., Core, M.G., Lane, H.C., & van Lent, M. (2005). Design Recommendations to Support Automated Explanation and Tutoring. In Proc. of the Fourteenth Conference on Behavior Representation in Modeling and Simulation.

Graebener, R.J., Rafuse, G., Miller, R., & Yao, K. (2003), The Road to Successful Joint Experimentation Starts at the Data Collection Trail, In Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC) 2003.

IEEE. (1995). IEEE Standard for Distributed Interactive Simulation Communication Services and Profiles. Retrieved from http://ieeexplore.ieee.org/servlet/opac?punumber=3693

IEEE. (2000). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Framework and Rules. Retrieved from http://ieeexplore.ieee.org/servlet/opac?punumber=7179

Jensen, R., Nolan, M., & Chen, D.Y. (2005). Automatic Causal Explanation Analysis in Combined Arms Training AAR. In Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC) 2005.

Knight, S., Reese, W.C., Durham, W.H., & George, G.R. (2001). Innovative Training Technologies In AVCATT-A. In Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC) 2001.

Lampton, D., Endsley, M., Gately, M., Cohn, J., Freeman, J., & Martin, G. (2005). Measuring Situation Awareness for Dismounted Infantry Squads. In Proceedings of I/ITSEC 2005.