OmniScribe – Enhancing AAR in an LVC Environment

Alden Peterson, Stephen Gilbert, Eliot Winer
Iowa State University – Virtual Realities Applications Center
Ames, Iowa
<alden, gilbert, ewiner>@iastate.edu

Jeffrey Welch
Dignitas Technologies, LLC
Orlando FL
<jwelch>@dignitastechnologies.com

Julio de la Cruz, Hector Gonzalez
US Army RDECOM STTC
Orlando, FL
<julio.delacruz, hector.gonzalez24>@us.army.mil

ABSTRACT

Innovations in live, virtual and constructive (LVC) environments geared for US military joint force training allow a more effective utilization of space and time for training exercises across the globe. As this use becomes more prominent, the need for a suitable after action review (AAR) tool to incorporate an ever-increasing number of data sources is fast becoming a requirement. To perform an AAR fully, data from a variety of input sources must be saved, synchronized, and analyzed. It is important to equip military trainers with an effective tool to facilitate this need for comprehensive data in AARs to maximize the effectiveness of LVC training environments.

Iowa State University is developing an open source software tool for the U.S. Army to address shortcomings of existing AAR tools. Utilizing an innovative modular domain-independent API, users can combine inputs from multiple sources such as simulation data, physiological sensor information, discrete events, and video feeds into a single application. The aggregated information can then be replayed during an AAR session allowing simulation event information to be supplemented with sources not traditionally incorporated in AAR and providing a framework to greatly enhance AAR.

This paper describes such a system (OmniScribe) at its current stage of development, describing its API for the integration of disparate inputs within a single tool and illustrating using a working prototype. It will discuss the current state of the architectural framework, designed to allow users the ability to add additional playback functionality by developing unique modules, and the prototype. Additionally, the paper will briefly discuss the implications a foundation of disparate data stream integration within LVC training will have on future real-time data mining, decision visualization, and enabling deep behavioral analysis of trainee performance.

ABOUT THE AUTHORS

Alden Peterson works for the Virtual Realities Application Center at Iowa State University with Dr. Eliot Winer and Dr. Stephen Gilbert, pursuing a Masters in Human Computer Interaction (HCI) and Mechanical Engineering. His primary research interests are usability, virtual reality, and simulation.

Stephen B. Gilbert, Ph.D. is Associate Director of the Iowa State University Virtual Reality Applications Center and of the graduate program in Human Computer Interaction. With a Ph.D. in brain and cognitive sciences from MIT, Gilbert is assistant professor in industrial engineering and psychology. He brings rich experience from the for-profit education and instructional design industry to research on intelligent tutoring systems and training in virtual environments. He was CoPI on a project for AFOSR on the design of interfaces for the teleoperation of multiple UAVs and is currently PI on an Army RDECOM STTC project for creating mixed reality environments for better LVC training.

Eliot H. Winer, Ph.D. is Associate Director of the Virtual Reality Applications Center and an Associate Professor in Mechanical Engineering. His research interests include using computer graphics and immersive virtual reality to facilitate decision making in medical treatment, complex system design, and military training.

Jeffrey L. Welch is an 11 year veteran of software development within the modeling and simulation industry. He has worked directly on virtual and constructive simulation systems with emphasis on scenario generation, dynamic environments and complex system integration. His project involvements include direct support for the Brigade Combat Team Modernization (BCTM), Synthetic Environment (SE) Core, One Semi-Automated Forces (OneSAF), CACCTUS and Joint Simulation System (JSIMS) programs. He holds an M.S. and B.S. in Computer Science from The University of Central Florida in Orlando, FL.

Julio de la Cruz is Chief Engineer for Synthetic Natural Environment and Simulation Technologies. Mr. de la Cruz has a Bachelor of Science degree in Electrical Engineering from the City College of New York and a Master of Science degree in Industrial Engineering from Texas A&M University. Mr. de la Cruz has over eighteen years experience in government and private sector. Mr. de la Cruz previous positions include lead researcher for Army Technology Objectives (ATOs) focused in the core simulation areas of Synthetic Natural Environment and Computer Generated Forces. As the Chief Engineer and Manager for Synthetic Natural Environment at the Simulation Technology and Training Center, Research Development and Engineering command (RDECOM-STTC) he oversees research and development of current Database Generation Systems/tools technology and supports the Command to address the defined Army needs in Modeling and Simulation.

Hector Gonzalez is a Science and Technology Manager at the ARMY Research Laboratory, Simulation and Training Technology Center in Orlando supporting research and development efforts in Synthetic Natural Environments for training applications. Hector has 27 years of experience working NAVY Submarine training devices and ARMY Research projects. He holds a BS in Electrical Engineering from University of Puerto Rico.

OmniScribe – Enhancing AAR in an LVC Environment

Alden Peterson, Stephen Gilbert, Eliot Winer
Iowa State University – Virtual Realities Applications Center
Ames, Iowa
<alden, gilbert, ewiner>@iastate.edu

Jeffrey Welch
Dignitas Technologies, LLC
Orlando FL
<jwelch>@dignitastechnologies.com

Julio de la Cruz, Hector Gonzalez
US Army RDECOM STTC
Orlando, FL
<julio.delacruz, hector.gonzalez24>@us.army.mil

BACKGROUND

What is AAR?

An after-action review (AAR) system is an important tool in improving the effectiveness of military training operations by enabling performance feedback. Military training often entails exercises followed by a time dedicated for AAR. Allen and Smith (1994) point out that AAR feedback is highly time sensitive, where time delays between training event and feedback diminish the usefulness of collected information. Immediate feedback allows for corrective action. Effective AAR allows instructors or those leading the exercise to provide valuable constructive feedback to those involved. AAR systems benefit trainees by visualizing their performance and providing awareness of their actions so that they can direct their own behavioral change. After action review seeks to address four primary stakeholder questions (Department of the Army, 1993):

1. What was expected to happen?
2. What actually occurred?
3. What went well, and why?
4. What can be improved, and how?

The first objective is to identify and understand the intended outcome for the exercise. This may result in a variety of major or minor goals and objectives, depending on the complexity of the exercise. It is difficult to review objective achievement and performance unless the desired goals and objectives are clearly identified.

More relevant to this paper is the objective listed second above. To conduct an effective AAR session, those leading it must have access to enough relevant information to adequately and accurately determine what actually happened during the exercise. The larger and more complicated exercises possible with LVC training present an increasingly difficult challenge to meeting this goal. It is critical as well for the large amount of disparate kinds of data generated within an LVC exercise to be time synchronized for accurate analysis of temporal relationships. For AAR to succeed, playback must allow data to be reviewed simultaneously without being overwhelming (Allen & Smith, 1994), to allow trainers and trainees to visualize relationships between many entities, their spatial, temporal, and communication actions, and the resulting positive or negative outcomes.

Due to the nature of LVC military exercises, often with multiple critical events occurring in rapid succession and/or simultaneously (in both physical and virtual domains), the ability to compile and time synchronize as much relevant information as possible is needed to allow instructors and trainees access to the information required to arrive at a consensus on what events did happen during the exercise. The large volume of information from differing sources makes this difficult, e.g., high frequency analog data from physiological sensors, video logs from MOUT site cameras, simulation events from a game engine, 3D tracking of soldier movements, or blackbox-style logs from cockpit simulators, among other sources.

Likewise, increased exercise complexity results in significantly more difficult post-exercise analysis. An ideal AAR tool would facilitate this analysis by assisting instructors in identifying key events, performing real-time data mining/analysis, and in general allowing the instructor the ability to more effectively conduct an AAR session. Automatic bookmarking of important events would allow instructors to determine what went well and what can be improved more efficiently, allowing them to focus on key exercise events.

Importance of AAR

Jones and Mastiago (2006) found virtual training exercises to be quite valuable for actual combat effectiveness. Allen and Smith (1994) state, "structured immediate feedback gives the training audience an opportunity to learn as they proceed." Johnson and Gonzalez (2008) stated that timely and accurate feedback is essential in effective training. To maximize the benefits of the training exercises, mistakes have to be identified and good decisions have to be recognized immediately as well. A good AAR process allows the unit to benefit from both a higher ranking officer's evaluation and a group discussion of the scenario immediately after the task has been performed. "No commander, no matter how skilled, will see as much as the individual soldier and leaders who actually conduct training… The AAR is the keystone of the evaluation process" (Department of the Army, 1993).

AAR's are an effective and well-established process dating back to the mid 1970s. They are now the standard method for providing performance feedback from a collective rating exercise (Morrison & Meliza, 1999) and have been cited as a best practice in performance evaluation (Salter & Klein, 2007).

Stakeholder Needs

Prior to development, the team researched the core requirements for an effective AAR system. The utility AAR systems extend beyond military applications. Similar methods to evaluate results from data collected during experiments within simulations or virtual environments such as interpersonal communication training (Raij & Lok, 2008), behavioral research, and medical system training (Quarles, Lampotang, Fischer, Fishwick, Lok, 2008), provide opportunities for AAR systems to be leveraged. Many of the tools used by these researchers and instructors were specifically constructed to facilitate their needs for the specific experimental scenarios. In general, these needs had significant overlap, but resulted in a large variety of similar AAR/data review tools being created.

Combined with active collaboration with industry and military personnel, the team surveyed a core group of ten stakeholders to gain perspective on the most critical/useful features they needed in an AAR system. This information would supplement existing literature to guide the development of OmniScribe.

Stakeholders were asked:

- What types of data do you typically collect during an experiment or simulation?
- What, if any, types of physiological data do you collect?
- Please describe how you currently evaluate collected data from a typical simulation or experiment, e.g. software packages you use and an overall process. Does your current process present any particular challenges?
- What is the typical duration you are collecting data over, i.e. if working with a participant how long might it take them to complete the task(s)?
- During that duration in which you're collecting data, do you:
  - Collect all the time (continuously)
  - Collect at preset periodic intervals
  - Collect at times triggered by the participant
- Approximately how frequently are you collecting data samples, e.g. 75 Hz, three times per day, etc.?
- If you had a system to synchronize multiple data streams, and playback for later analysis, how would you use it? How would it change your workflow?
- Additional thoughts or comments about what you would like in a software package to help you review both what participants did and their data?

This research was used to drive development and requirements for an open-source tool to better meet the needs of AAR in an LVC environment.

Methodology and Motivation

Having collected survey results the team began development of specific objectives to meet many of the wide variety of needs discovered throughout the survey. Our findings indicated five primary needs for an effective AAR tool. These needs are the ability to:

1. Record and play-back common data sources
2. Time-synchronize all data sources
3. Annotation of data
4. Add unique and new data sources
5. Offer an API for real-time data-mining and analysis

Much of the feedback indicated difficulty in incorporating secondary information to augment AAR. For example, Virtual Battlespace 2 (or VBS2) provides built in functionality for AAR (Bohemia Interactive, 2012); however, if participants desired to supplement this built-in AAR with a video camera or any other non-VBS2 source of information related to the exercise, an entirely new framework would be required – either completely separate from the VBS2 application or as a modular extension to it. These data sources would then need to somehow be time synchronized to the VBS2 scenario to allow proper playback. Additionally, flexibility in adding additional sources was a key driving force within the stakeholder survey.

A concurrent need with incorporating multiple data sources into training AAR is the ability to allow annotation or notes and real-time feedback to individual trainees. Observing and analyzing trainees during training, with the ability to provide individualized feedback real-time was an additional need described by stakeholders.

These difficulties reported by the stakeholders through the survey the team conducted were found to be quite similar to literature regarding limitations of current AAR.

Johnson and Gonzalez (2008) found, "future research into automating AAR should concentrate on providing a means to tailor an expert performance to each individual event." (p. 12)

In addition to the need for individualized performance feedback, they also found while existing "…tools provide an objective viewpoint of the training exercise, they do not provide training feedback or assessment on their own" (p. 4). A difficulty in automatic, individualized feedback to trainees is central to the Johnson and Gonzalez 2008 paper discussing AAR.

The Leader's Guide to After-Action Reviews (U.S. Army, 2011) states that observer-controllers need to keep "accurate records of what they see and hear, and record events, actions, and observations by time sequence to prevent loss of valuable information and feedback" (p. 9). To facilitate these records, it is often necessarily to supplement easily available information in an LVC training environment with information not traditionally collected in an AAR.

OMNISCRIBE

Motivation

The inability to efficiently incorporate data within LVC training from a variety of disparate sources drives the development of an open-source tool to accommodate these needs. In order to enhance AAR within these training exercises, a tool is needed to be able to record, display, and playback data for purposes of AAR.

Additionally, the wide variety of data sources found in LVC training allows the unique opportunity for live feedback. Synthesizing these data sources and displaying real-time data during training exercises will allow trainers the possibility to provide directed and real-time feedback when appropriate based on a large variety of data inputs.

Architecture

A primary concern in the development of an open-source tool which can meet these needs for AAR is the ability to incorporate a modular design. In order to facilitate easy addition of additional, new, currently unavailable data-streams to existing software, the core functionality of an application must be modularly separated as much as possible from enhancing additions.

To do this, development with two primary types of functionality in mind has been considered by the team (which are incorporated into several layers of software classes). First, functionality commonly required by multiple different datastreams is moved, whenever possible, to the core program architecture, so all datastream modules can access it. The second type is datastream specific functionality, or processes unique to each datastream. The relationship between the core architecture and modules is shown in Figure 1 and will be discussed further in the following sections.

OmniScribe Core

To facilitate AAR and future data mining multiple key functionalities are required – by all data desired to be used in the AAR. First, all data entering the recording system must be time stamped, to allow synchronous and accurate playback and visualization. A framework for analysis of time information is fundamental and critical if any tool is to address difficulties in AAR described previously. This framework extends to the development of the user interface, with functionality allowing visualization of information (both real-time as well as recorded).

By utilizing a core-level time synchronization framework, OmniScribe addresses one of the key difficulties in conducting AAR in LVC environments – namely the ability to provide the synchronization of all data within a single application. Therefore, users wishing to add additional data can take advantage of and reuse this core functionality.

Additionally, users can utilize existing modules within the user interface to simplify their own development efforts. When appropriate, the core user interface architecture incorporates generic elements, allowing users desiring to add additional functionality the ability to utilize previously developed functionality rather than re-implementing and redesigning large amounts of core architectural components.

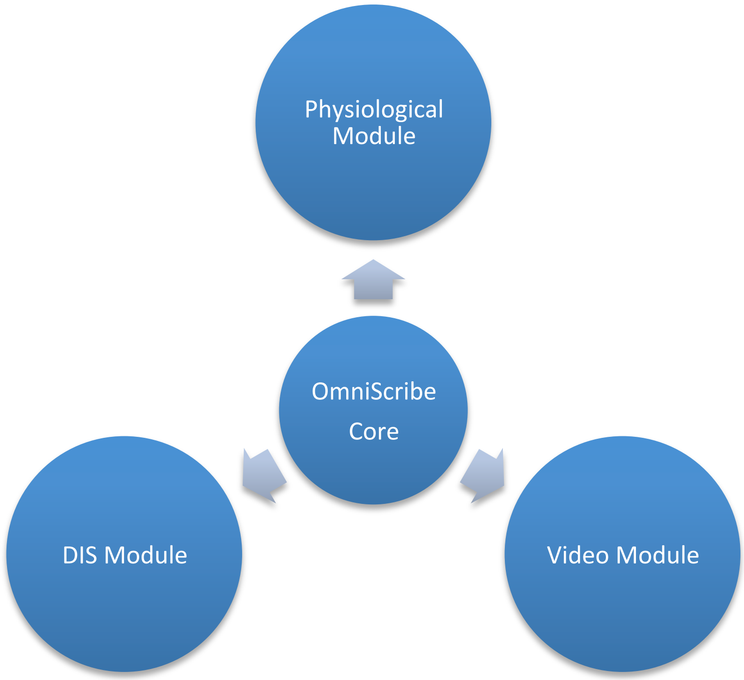

Figure 1. Relationship between architectural core and modules. Datastream specific functionality is isolated from core functionality and contained within a module, separate from other datastream code.

OmniScribe Modules

In order to capture, record, display, and playback data (where appropriate), developers will naturally have to develop software functionality for enabling these abilities.

Maintaining a modular architecture allows the majority of this development work to be contained within modules, with each module consisting of any code specific to its particular data-stream. Using existing modules as examples, adding a new data-stream to OmniScribe requires development of classes containing specific functionality for:

- Data recording
- Data visualization
- Data playback (if applicable)
- Data mining (if applicable)

These modules can then be added to the software to interface with the core architecture and allow open-source developers to leverage existing core functionality, under a modified BSD license (Open Source Initiative, 2008).

Current Development Status

The most recent version of OmniScribe currently has been developed to include a refined core-level architecture with reusable components and several datastream modules. Each of the modules has been developed specifically for the purpose of AAR and will be discussed in more detail in the following sections.

DIS Module

In order to more effectively facilitate AAR within the Iowa State University MIRAGE training environment (Newendorp, Noon, Holub, Gilbert, & De La Cruz, J., 2011), more advanced analysis of the DIS datastream was required. While some DIS analysis is available from commercial packages such as the VBS2 system used by the MIRAGE, these systems do not include support for deep, complex analysis of the events occurring during the scenario.

Having spoken with stakeholders for AAR within the MIRAGE system as well as other military applications, a primary need unique to LVC environments was the ability to distinguish events affecting live, virtual, or constructive entities. During large training exercises, there may be many events happening in rapid succession, but for purposes of AAR, the trainer's focus might be more refined. For example, the training objectives may be focused primarily to those directly involved in the training exercise and not every constructive entity. Additionally, trainers may desire to see specific types of events – gunfire, detonations, entity deaths, etc – and wish to visualize those events independently.

Utilizing a filtering process, the graphical user interface (GUI) allows the ability to filter the incoming DIS data based on a variety of criteria. Incoming DIS information is analyzed and can be assigned identification tags based on user-defined criteria. The user interface can then be customized to display combinations of these tags in individual plots. This allows trainers to focus information available to them to exactly what they desire to see – perhaps all gunfire events associated with each of the trainees in the exercise or any entity deaths – and visualize these independently.

This could be used in practice to allow trainers the ability to focus simulation event information to specific combinations of interest. For example, there might be a particular virtual entity in an exercise the trainer would like to focus AAR on, and the trainer conducting AAR can choose to visualize simulation events (perhaps gunfire) only involving the particular entity of interest. This allows the trainer to quickly and easily determine events important to the particular trainee.

Additionally, DIS module provides the ability to filter based on multiple categories and criteria – simulation PDUs can be tagged with appropriate categories (such as "live" or "virtual" or "death" or "gunfire") and the GUI can display combinations of these categories. For example, a trainer may be interested in all constructive deaths and live gunfire events – the GUI can be configured to display these combinations to allow easy visual determination as to when events of interest occurred (rather than seeing all simulation events along the same timeline). An example of this filtering and layering is shown in Figure 2 below.

The DIS communication events received during a scenario can also be rebroadcast during AAR activities to allow visualizing of simulation events within the LVC scenario on the systems used to generate the recordings. This allows AAR to be greatly enhanced by incorporating other recorded data into AAR through the same interface, while preserving the LVC scenario. Note this playback functionality is not unique to DIS, but rather can any datastream where appropriate.

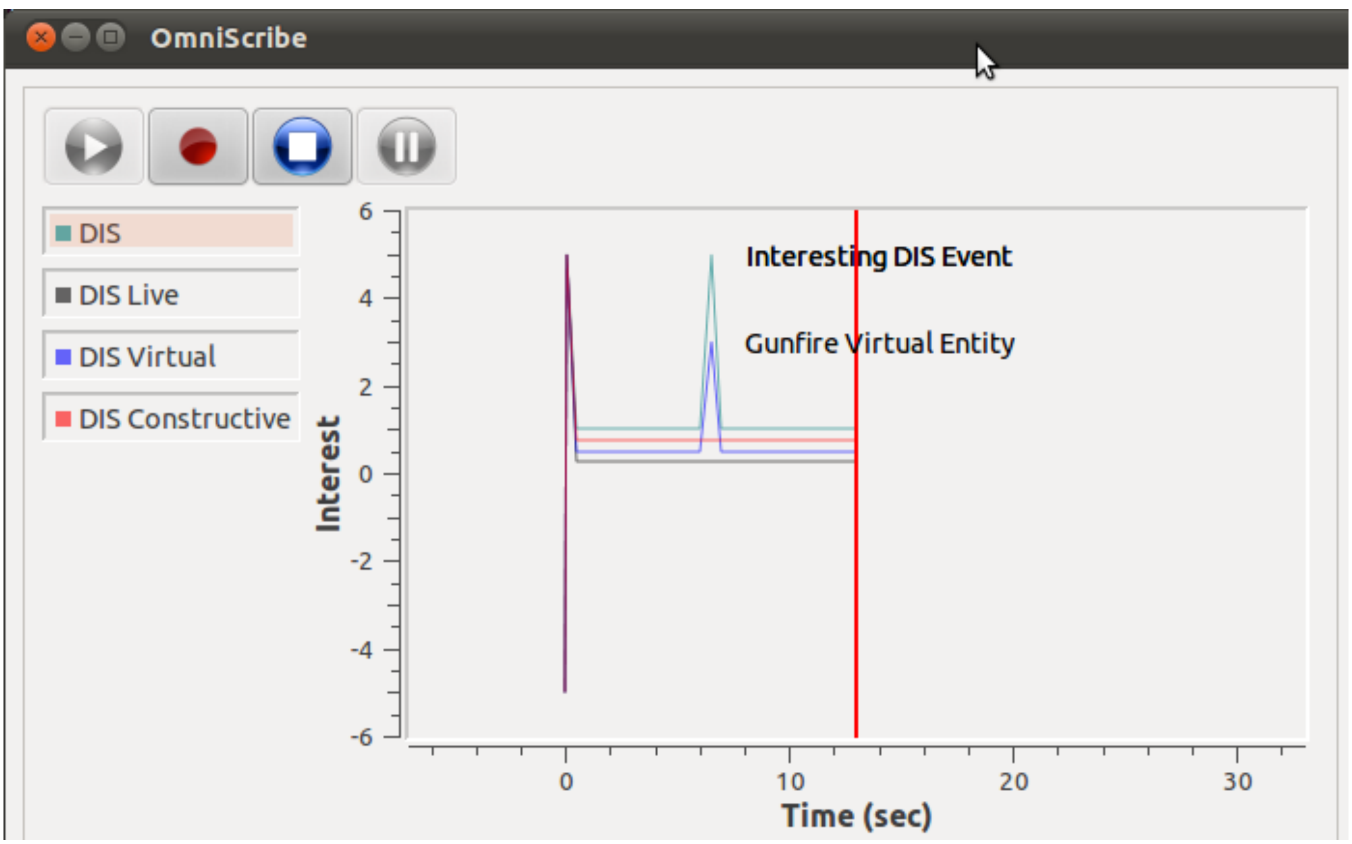

Figure 2 – Example of DIS Filtering Functionality. A virtual entity has caused a gunfire event to occur and the event is flagged as such in the GUI, shown as an interesting event for "DIS" as well as "DIS Virtual."

Video Module

To augment information received from DIS within an LVC scenario, and again based on feedback primarily from stakeholders, the ability to record and playback video was identified as an important development effort. The importance of video to learning has been documented by Ives, Straub, and Shelly, who state with respect to athletic training, "contrary to what might be expected, video has created opportunities for enhanced communication, building trust and relationships…" (p. 240) as well as others (Dowrick, 1999; Ives, et al., 2002).

As discussed before, keeping "accurate records of what they see and hear" is an important element of AAR (U.S Army, 2011 p. 9). Video is a critical element in LVC mixed-reality training environments where trainees may not only be involved virtually but also physically. In these environments it is possible that a trainee's movements and directional gaze or other physically oriented elements of the training scenario will not be captured via the software controlling the training exercise, causing a key element of AAR to be lost.

For this reason, development focused on integration of video capabilities as the first outside data source to augment DIS data. Two primary types of video were focused on by the development team – locally connected webcams and Internet-delivered video streams.

Currently, the ability to augment DIS scenario information with as many webcams as the computer can support allows users to connect and record webcameras, utilizing the open source OpenCV libraries (OpenCV, 2012). It has been tested using combinations of up to five web/IP cameras.

The video feeds can be selected via a selection dialogue process, and once chosen by the user to be added to the scenario, are recorded during the scenario and can be played back during AAR. They can also be visualized during the scenario. This allows trainers more in-depth insight into real-time events during training and an opportunity to provide immediate and real-time feedback as well based on observed trainee actions.

To facilitate AAR a "keyframe" visualization element is added during playback. Along the master timeline, images are sampled at intervals from the recorded video and displayed (see Figure 3). Combined with the ability to jump to time via the slider, the user can identify visually events recorded from the video sources and more easily immediately proceed to the desired location for AAR.

Supporting the goals to synthesize a large variety of sources and allow customization resulted in a module developed to be agnostic of input source as possible. Internal processing within the module for video is therefore independent of video source; to add additional video datastreams other than IP/webcam simply requires translation of the input video to an internal video frame object and allows the new source to re-use all existing video functionality.

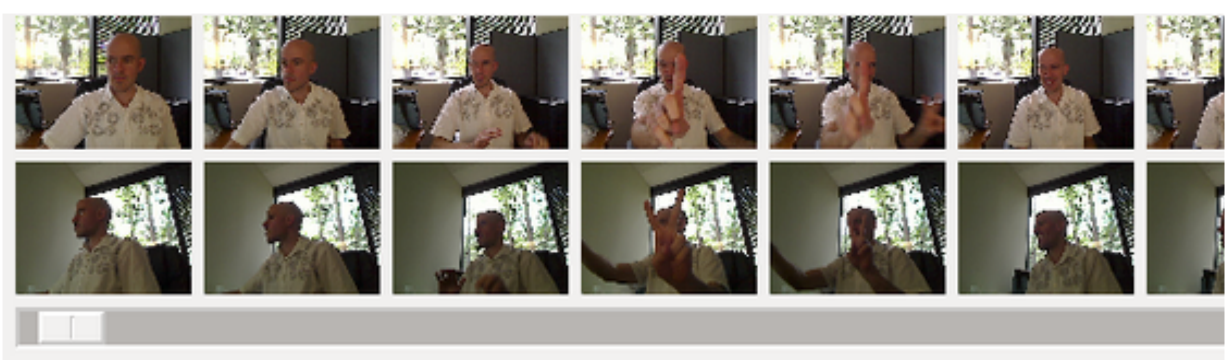

Figure 3. OmniScribe video key frame example showing sampling and key frames during playback.

Physiological Module

The third key component developed into a module is the ability to add physiological information (in the format of time/value pairs) to supplement data recorded during the training exercise. Physiological data can come in many different forms, such as:

- Heart-rate
- Blood pressure
- EEG
- Galvanic skin response
- Temperature

During LVC training scenarios, it is often desirable to include the ability to record some or all of these for use in AAR. Determination of trainee physiological stress levels can be compared against their performance during exercises for example, or the effect of various exercise events can be compared with the physiological response of trainees. Physiological data may also be useful in determining the vigilance and sense of presence in the training environment (Nasoz et al., 2004; Bouchard et al., 2008, Meehan et al., 2002).

Unfortunately, this sort of data is difficult to integrate into AAR using existing tools. Normally requiring an additional software package to record, view/display, and time-stamp, the efficient inclusion into AAR becomes difficult and time consuming. The time synchronization of physiological data is also crucial to effectively utilizing any recorded data in AAR – for example, to attempt to determine the response to a training gunfire event may require a very precise time alignment between the AAR within the simulation tool to add value to the AAR session.

The third key module must therefore be capable of interfacing with various physiological sensors. Similar to the previously discussed video module, a primary development goal was abstracting the internal-to-OmniScribe elements as much as possible, to make the module capable of receiving data from many different sensors with as minimal requirements on the developer for implementation as possible.

The physiological module implementation receives numeric data from a source and processes, saves, and displays (graphically) with the associated timestamp. To connect additional physiological sensors, the developer needs only to write code to receive the sensor information (in whatever form it may be transmitted in) and then "translate" this to the generic, internal-to-OmniScribe physiological data element.

The aggregate physiological data can be displayed on a single plot (see Figure 4). Because of how many physiological data streams may be present, the GUI has the ability to turn visualization of each on/off, allowing trainers easier access to desired information when needed.

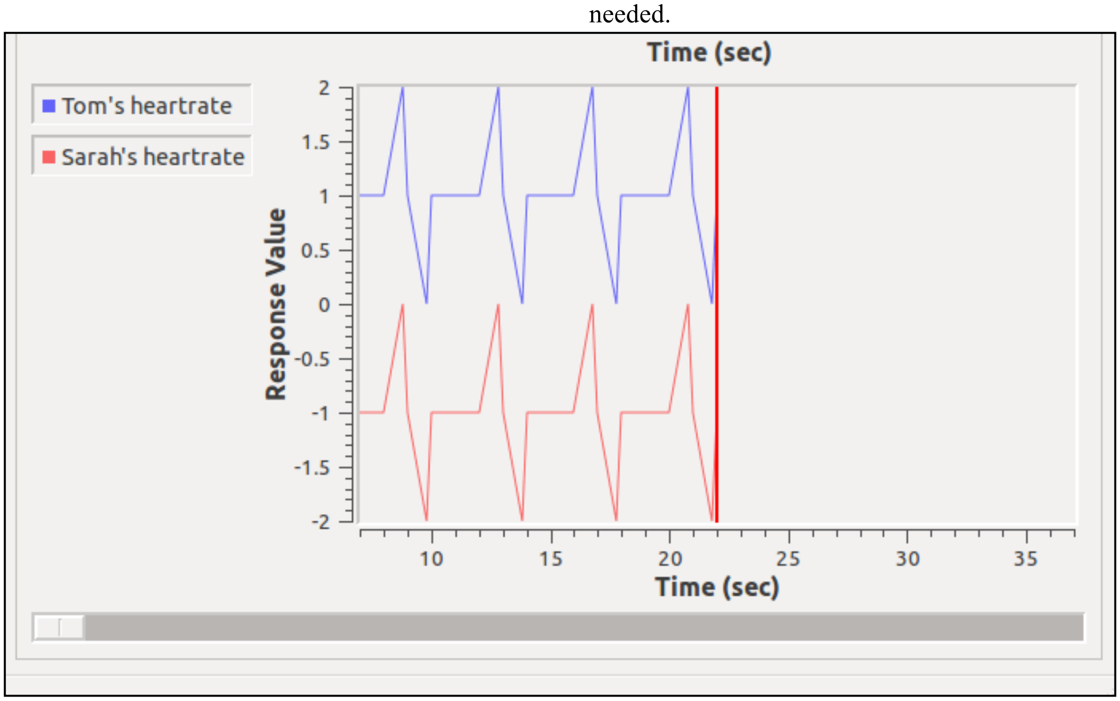

Figure 4. Example showing multiple physiological sensor streams. Note that users can click on the label to turn visualization on or off for each independent physiological datastream

Applications

The combination of DIS, video, and physiological information allows for enhanced AAR. Consider the following scenario possible with current functionality.

Sergeant Thompson is going to monitor four trainees – two controlling virtual entities on laptops and the others participating physically in the LVC MIRAGE environment at Iowa State University. A visually tracked system integrated into a VBS2 environment supporting multiple displays will allow him to evaluate performance, but he was going to have a lot of data to manage. With multiple webcams showing various areas of the physical environment and several physiological sensors for each of the trainees operating virtual entities, he configures the AAR utility to allow individual displays of all gunfire, individual gunfire feeds for each trainee, and another showing any constructive entity deaths. This allows him to quickly and easily see when each trainee fires his weapon. After starting the exercise, Thomson quickly sees one of the trainees consistently reacts slower than his counterparts – each time gunfire is exchanged, the same trainee reacts slower (this is seen from the filtered DIS graph showing all gunfire events) – this trainee always lags behind. Thompson also sees from webcam video feed this trainee is consistently looking in places different than where he needs to look. Using the radio system, Thompson advises the trainee to pay more attention to specific areas of the training environment where he had previously missed key events and throughout the rest of the exercise, the trainee's reaction time increase.

Providing software capable of integrating disparate datastreams within a single application allows for significantly deeper analysis during real-time training exercises and AAR. By combining a multitude of data into a single application, trainers can have the ability to visualize and direct training content immediately to needs of the trainees.

CONCLUSION

Summary

The need for a tool capable of better enabling AAR in LVC training environments has driven development of an open-source tool capable of enhancing AAR. By providing a framework for incorporating disparate datastreams into a single AAR utility the ability to both perform AAR as well as provide targeted feedback during training is enhanced.

By providing a user-needs driven architectural framework, and being designed in consideration of future open-source and modular development efforts, the ability for future expansion is also increased as developers can isolate and develop only functionality directly relevant to their material (Wheeler, 2007).

At the current time, this framework has a developed core architecture as well as modules supporting DIS, video, and physiological datastreams within AAR. This allows users to incorporate these data sources into training exercises. The functionality (as well as data-mining capabilities, where applicable) of each module has been discussed in previous sections.

Future Work

While a significant work has been done towards the long term objectives of an open-source AAR tool for LVC environments, there are three key steps in the future development.

The first is continuing to develop and refine modules and allow for even more disparate data-streams to be supported within OmniScribe. Throughout this process, refinements and enhancements to the core architecture will also likely be required to streamline the module development process, as well as continue to abstract functionality where appropriate.

The second key step will be inclusion of the larger software development community through open-source development efforts. A critical component to the long-term utility of OmniScribe is this inclusion and community effort to facilitate module development. As time progresses, developers and users would be able to incorporate the development work from others into their own applications (for example, if someone required a specific data source previously not included, and developed the module, this would then become available for others to use with minimal development efforts of their own). It is an important next step to finalize documentation for module development to support this goal.

Finally, it is important to develop and implement more advanced real-time data mining algorithms to supplement AAR and real-time feedback for trainers. Johnson and Gonzalez (2008) have found that "Improving performance diagnostics and presenting expert performance are where intelligent systems research could have the most significant impact on automating the AAR process" (p.12). OmniScribe currently does limited data mining by allowing DIS information to be filtered and more efficiently utilized. However, expansion of data mining capabilities and integration into video (such as object recognition) or physiological data (such as automatic identification of when trainees are stressed/tired/etc) would serve to further enhance AAR and allow OmniScribe to be a powerful tool for military trainers.

Cohesive and integrated data mining algorithms are possible due to data being combined into a single application. By doing this, a framework for considerably more intricate and cohesive AAR is presented. Integration of data mining algorithms would allow this to be further explored, advancing the capabilities for both AAR as well as real-time training enhancements.

It is important to note that throughout the development process, user studies and usability testing will be an important part of ensuring OmniScribe can be advanced, remain a useful tool for trainers, and continue facilitating AAR in a variety of environments.

ACKNOWLEDGEMENTS

The writers of this paper would like to thank the Army RDECOM STTC and the members of the team participating on the Iowa State University LVC research project.

REFERENCES

Allen G. & Smith R. (1994) After Action Review in Military Training Simulations. Proceedings of the 1994 Winter Simulation Conference, 845-849.

Bohemia Interactive Australia Pty Ltd. (2011) VBS2 VTK. http://products.bisimulations.com/products/vbs2/

Bouchard et al. Anxiety increases the feeling of presence in virtual reality. Presence: Teleoperators and Virtual Environments (2008) vol. 17 (4)

Department of the Army, Washington DC. Training Circular 25-20: A Leader's Guide to After-Action Reviews. September, 1993. http://www.au.af.mil/au/awc/awcgate/army/tc_25-20/tc25-20.pdf

Dowrick, P.W. (1999). A review of self modeling and related interventions. *Applied & Preventive Psychology, 8*, 23-39.

Ives, J.S., Straub, W.F., & Shelley, G.A. (2002). Enhancing athletic performance using digital video in consulting. *Journal of Applied Sports Psychology, 14,* 237-245.

Johnson, C. & Gonzalez, A. (2008). Automated After Action Review: State-of-the-Art Review and Trends. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology. 5, 108.

Jones PN, Mastaglio PW. Evaluating the Contributions of Virtual Simulations to Combat Effectiveness. March, 2006. http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA448151&Location=U2&doc=GetTRDoc.pdf

Meehan et al. Physiological measures of presence in stressful virtual environments. SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques (2002)

Morrison, J. E. & Meliza, L. L. (1999) Foundations of the After Action Review Process. U.S. Army Research Institute for the Behavioral and Social Sciences. Special Report 42.

Nasoz et al. Emotion recognition from physiological signals using wireless sensors for presence technologies. Cognition, Technology and Work (2004) vol. 6 (1)

Newendorp, B., Noon, C., Holub, J., Winer, E., Gilbert, S., De La Cruz, J. (In press 2011) Configuring Virtual Reality Displays in a Mixed-Reality Environment for LVC Training. *Proceedings of the World Conference on Innovative VR 2011*.

Open Source Initiative. The BSD 3-Clause License. (2008) From http://www.opensource.org/licenses/BSD-3-Clause

Open Source Computer Vision (OpenCV), 2012, http://opencv.willowgarage.com/wiki/

Quarles J., Lampotang S., Fischer I., Fishwick P., and Lok B. (2008) Collocated AAR: Augmenting After Action Review with Mixed Reality. Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on Mixed and Augmented Reality 2008, pp. 107 - 116

Raij A. & Lok B. (2008). IPSVIz: An after-action review tool for human-virtual human experiences. Proceedings of The IEEE Virtual Reality Conference (2008), 91-98.

Salter, M. S. & Klein, G.E. (2007) After Action Reviews: Current Observations and Recommendations. U.S. Army Research Institute for the Behavioral and Social Sciences. Research Report 1867.

U.S. Army Combined Arms Center –Training. (2011). Leader's Guide to After-Action Reviews (AAR). Retrieved from: www.jackson.army.mil/sites/leaderdevelopment/docs/710

Wheeler D. A. (April 16, 2007) Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers! From http://www.dwheeler.com/oss_fs_why.html