Loh, C.S. (Oct 18-22, 2010). *A Real-Time Approach to Performance Assessment for Multi-User Virtual Environments.* Paper presented at E-Learn 2010 – World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education. Orlando, FL.

A Real-Time Approach to Performance Assessment for Multi-User Virtual Environments – *Ad hoc* Action Report (*a*AR) vs. After Action Report (AAR)

Christian Sebastian Loh, Ph.D.
Virtual Environment Lab
Southern Illinois University
Carbondale, Illinois 62901, USA
csloh@siu.edu

Abstract: The conventional belief of conducting a review session "after" training events should be revisited. Given the ultra high speed processing power of today's computing technologies, post hoc reviews performed after hours of training would seem extravagant. Transitioning the training into a simulated virtual environment only solves the problem partially. Since every minute spent on training still costs the organization money, it will be important to catch any mistake made as early as possible. Operationally, we need a training system that is capable of reviewing or reporting on the trainees' performance (actions performed) as it happens. An ad hoc Action Report (aAR), which allows the trainers to catch the mistakes made by the trainees in a timely manner, and to correct them before the errors become entrenched, is a better option. The "Information Trails" is an innovative performance assessment methodology targeted at multi-user virtual environments that allows for the ad hoc capturing and reporting of player actions in real time during (game) play.

INTRODUCTION

After Action Review[1] is a standardized debrief or review process developed and adopted by the U.S. military for the review of trainees' performance *after* a training event (such as using a flight simulator) is completed. An After Action Report (AAR)[2], on the other hand, is the analytical report written (often by the author of the training) after the completion of a sequence of goal-oriented actions carried out during training. Even though the After Action Review and Report may differ in formats and styles, they are both conducted *after* the training events with the same primary purpose to analyze the actions performed by trainees (during training) for performance improvement. (In this paper, AAR is used specifically to mean After Action Report, and not After Action Review.)

When AAR was first adopted by the military, the world at that time was without ultra high-speed processing capability and sophisticated tracking technology. After all, how could anyone analyze the outcome of a training event *before* it has even taken place? Hence, the notion of "*after action*" is a reasonable one. Nevertheless, given the processing speed of today's computers and advanced digital technologies, a review that was performed *after* the completion of expensive training events would seem extravagant, to say the least. Not only does it cost business organizations much money to put together training events, it costs the U.S. military even more to stage military exercises by mobilizing training personnel and co-locating heavy equipment. The current trend, therefore, is to take advantage of

[1] [Wikipedia entry] After Action Review (AAR) is a structured review or de-brief process for analyzing what happened, why it happened, and how it can be done better, by the participants and those responsible for the project or event. After-action reviews in the formal sense were originally developed by the U.S. Army and are used by all US military services and by many other non-US organizations, even though less structured de-briefs after events have existed since time immemorial.

[2] [Wikipedia entry] After Action Report (AAR) is any form of retrospective analysis on a given sequence of goal-oriented actions previously undertaken, generally by the author himself. Example: The Commentaries on the Gallic War by Julius Caesar. In this paper, AAR is used exclusively to mean After Action Report.

simulated (i.e., game-like or virtual) environments in order to reduce costs while keeping the benefits of military training.

If an organization continues to require that the reviews/reports be conducted *after* the training events (i.e., *post hoc*), then every minute spent by the trainees within the simulated environments must still carry a cost. Furthermore, any mistake performed within a simulated environment that is not immediately corrected runs a risk of being repeated (and hence, reinforced). While the MUVE provided the possibility of training massive numbers of trainees simultaneously, the onus is on the trainers to ensure that the trainees undergo sufficient training to be able to carry out their jobs. The assumption here is that the computer program (of the MUVE) must be able to replace human trainers in the mundane training tasks. But what about catching the mistakes committed by trainees?

Since it is not possible for trainers to be co-located beside trainees in a 1:1 ratio, it would be impossible to detect the mistakes made by the trainees in a timely fashion. This means that man-made errors (which may be fatal in a real life situation) could go unchallenged for the entire duration of the training (up to 20-40 hours, or more, depending on the scale of the MUVE exercise). By then, an equal amount, or more, training hours must be spent to unlearn the mistakes, making this a counter-intuitive process which could double or triple the cost of training per trainee.

A logical solution would be to perform the action review/report *as and when* the actions are carried out in real-time via *ad hoc* action reporting (*a*AR). This paper will first discuss the problems confronted in performance assessment within a simulated environment, and then demonstrate how the problems may be overcome through a working prototype which was developed by the author and his team.

The Potential: Training using Game-like/Virtual Environments

While businesses may prefer to distinguish themselves from one another by specifically naming their products to reflect the targeted audience/market/ purpose – e.g., game-like environments (entertainment), virtual worlds (socialization) and serious games (training) – a more neutral approach is to refer to all of these applications as Multi-User Virtual Environments (MUVE). Throughout this paper, MUVE will be used to denote any technology that allows its users to explore a virtual environment (of any size), while carrying out a sequence (or a range) of goal-oriented actions, for training or human performance improvement. Not every action performed in the MUVE needs to be targeted for learning, teaching, training, or instruction. Indeed, some of the MUVE may be created specifically for fun and profit; such as the World of Warcraft and Second Life.

It is no wonder that a huge volume of publications have been generated in recent years pertaining to the use of game-like environments (GLE), virtual worlds (VW) and serious games (SG) for education and learning, given the increased interests in MUVE among educators, researchers, and training organizations. This is evidenced by the tens of thousands of published journal articles on the matter indexed by the Google Scholar Search Engine. While a large proportion of these discourses were centered on the promotion or rebuttal of the technology (debating the high cost of implementation over benefits), the majority of the rest appeared to be descriptive research and phenomenological accounts of the use of games, or VW in social learning contexts.

Even though trainers and educators do recognize the need for more "data-driven assessments of learning" (Mandinach & Honey, 2008), the problem appeared to be the lack of a "killer game" that would win over the audience. Hence, very little effort has been channeled towards the development of a performance assessment methodology for use with these virtual environments. Most of the resources and money have been put towards creating the next best game or virtual world. A competitive marketing approach to produce the next best game/VW may benefit game developers in the short term, but the lack of performance assessment tools and strategies will eventually become an obstacle to widespread adoption and stunt the growth of the market.

Besides a development and marketing issue, there is another looming problem within the research community. Because many education researchers have turned away from conducting large-scale random-assignment experiments (Means, Haertel, & Moses, 2003), they must be content with fewer and fewer tools and methodologies that will yield empirical data and generalizable research findings. The lack of a working knowledge with game engines and virtual environment development tools also means they will have little influence in affecting the market trend. Left to its own devices, the growing knowledge gap caused by the lack of assessment methodologies will contribute towards the bottleneck in the adoption process. If MUVE is to experience

wider acceptance in the learning and training industry, there need to be more innovations in the area of assessment and evaluation tools that are compatible with virtual environment technologies.

THE PROBLEM: VIRTUAL ENVIRONMENTS

An important difference between video games for entertainment and 'serious games' intended for learning and instruction is that of assessment (Chen & Michael, 2005). As there is no safe way to put a probe into the mind of a learner (regardless of the learning environment) to directly sample the amount of learning that occurs, educators must rely on external indicators for performance assessment and evaluation. Within a physical face-to-face environment, a trainer can observe the learners' classroom behaviors (e.g., yawning, or on-task discussion) and document them as evidence of participation and as assessment of the learning that occurred (Harrington, Meisels, McMahon, Dichtelmiller, & Jablon, 1997). Unfortunately, as there is no direct means to observe or document learners' behaviors in MUVE, educators have thus far been prevented from using these well-tested classroom techniques to assess the learning of their students.

Game Logs

Digital game publishers who produce commercial off-the-shelf (COTS) games for training and instruction are well aware of the need for the inclusion of an assessment component in their products. Many real world examples can be found as finalist entries of the annual Serious Games Showcase and Challenge competition (http://www.sgschallege.com). Depending on the target market of these games, 'assessment components' may include: a series of timestamps for important game events, the number of 'kills' achieved (especially in first person shooter games), the amount of time taken to complete certain tasks, the total number of cases solved, or the percentages of the number of missions accomplished.

Conceivably, such information can then be retrieved either during or after the game play and used for self-evaluation by the player themselves, or be kept as evidence of the training in-progress. Because no information has been made available concerning how, or if, these data are actually being used for evaluation, no one knows if these game statistics have any true value or not. It should not be surprising that there has yet to be any consensus concerning the content or format of the game statistics collected for this purpose.

Pretest-Posttest Design

The overly simplistic nature of the game logs prompted some educators to adopt a pretest-posttest experimental design to assess the learning with MUVE. In these cases, they would typically administer a pretest before allowing the students to use MUVE (i.e., as an intervention), and then follow-up with a posttest to measure the amount of learning that took place (e.g., Kebritchi, 2008). While evidence of learning may indeed be measured using this method, the MUVE remained an impenetrable 'black box.'

The 'black-box' nature of the pretest-posttest design means that trainers will not know for sure what chain of events, or sequence of actions performed in the MUVE truly contributed to learning. External factors could have entered the system and affected the data, and the researchers would be none the wiser. Meaning, while the experiment works, no one knew for sure why and how it worked; meaning, the replication of the success became a chanced event.

Furthermore, because pretest-posttest data is externally reported, it is possible to manipulate the data collected. For example, students can manipulate the system to make a particularly interesting game appears more effective than it really is by performing 'poorly' in the pretest. From the perspectives of teachers and trainers, the reliance on posttest data could prove unsettling because by the time the effectiveness of a learning module could be determined (*post hoc*), it might be too late for remediation. Although this issue is not immediately apparent in 'clinical' research cases, it is a real problem because so many COTS games require 20 to 40 hours (or longer) for completion!

DATA COLLECTION & INFORMATION TRAILS

In order to overcome the 'black box' effect of the pretest-posttest design, it would be necessary to sample the data multiple times *during* the game play, but not *afterwards* – that is, with an *ad hoc* instead of a *post hoc* sampling technique. In this way, trainers and teachers would be informed earlier about the progress of the learners and able to prescribe remediation in a timely manner. If it is at all possible, the player data should be collected *internally* using the software engine, instead of *externally* through human input, to avoid introducing human error and self-reporting – that is, with *in situ* instead of *ex situ* recording.

Based on his earlier works involving online user tracking (Loh, 2006)(Loh, 2007), Loh proposed a new conceptual assessment design framework for serious games in 2007, which made use of *in situ* data collection techniques for *ad hoc* assessment. This was later incorporated into the "Information Trails" framework, and was tested for performance assessment using online MUVE. Conceptually, "Information Trails" was comprised of a series of strategically placed and agent-traceable objectives (events) within any information ecology (such as MUVE). Much like traffic cameras, 'event recorders' are placed at strategic locations (nodes) to capture user-actions during key events. Once captured, the users' actions may then be analyzed at any time using any appropriate method to reconstruct the decision-making process of the providers.

Since a decision is the product of a person's knowledge schema, the effectiveness of a user's actions – speed, accuracy and strategy – within the information ecology can then be expressed as a function of the users' understanding of the learning problems (what they know), as well as their problem solving skills (what they can do).

From Decisions to Actions and Behaviors

People's actions and behaviors are ultimately determined by their decision making processes, no matter the environment (virtual or physical). In a game scenario, if a particular path leads to certain death due to confrontation with a high-level boss, players must decide if they will find alternative routes, or be killed by the boss. Should they avoid the confrontation, they would gain the option to strengthen their characters and return for the challenge at a later time. The action (to turn away from a path) and the behavior (to avoid the 'road block') exhibited by players are products of their logical thinking and decision making processes. By documenting the players' actions and behaviors methodically during game play sessions, trainers and teachers will have the data they need for the assessment of learning.

From a designer's point of view, the association of assessment with actionable learning objectives is nothing new. The unique feature of Information Trails that sets it apart from the other assessment attempts was the use of *in situ* data collection during game play for *ad hoc* reporting. Because the data collection was done discreetly (both internally and automatically) by the game engine, testing and human data input would become unnecessary. In fact, once the players' data are captured by the database, the trainers would be able to identify which actionable learning objectives had been met, and how trainees met these objectives within the learning contexts of the game environments. The assessment framework of Information Trails not only took into consideration the decision-making process of the players, but also the actions and behaviors that arose from those thinking processes.

From Theory to Practice

In order to turn the Information Trails assessment framework from concept to a working prototype, a suitable development platform has to be identified. The COTS game known as Neverwinter Nights (NWN) was selected because it came with a game development kit that allowed for easy modification. Moreover, a third-party "Event Listener", called Neverwinter Nights eXtender (NWNX), was available to act as a connector between the NWN game engine and the external database server. Figure 1 shows the various relationships among various components: game engine, event listener, external database server, actionable learning and game objectives, and the NWN Tracer reporting system for data visualization. [Readers interested in more details about the Information Trails architecture are referred to other published articles by Loh and his colleagues (Loh, Anantachai, Byun, & Lenox, 2007; Loh & Byun, 2009).]

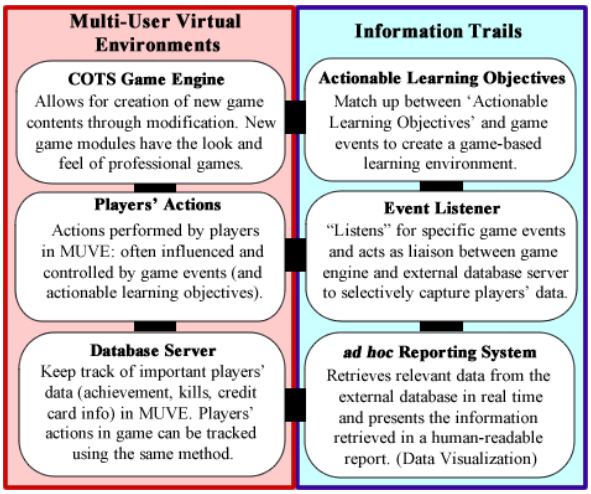

Figure 1: Game Environment and Information Trails Architecture

From its inception in 2007, the Information Trails has progressed from a standalone text-based prototype, to an integrated assessment system with Neverwinter Nights 1, and then with Neverwinter Nights 2. The NWN Tracer has likewise progressed from a JAVA application (version 1) to a Flex-based Internet

Application (version 2). Although the Information Trails and NWN Tracer system was created to track players' actions and behaviors, it did not (by design) indiscriminately capture every bit of available information. Currently, it only checks for learning-objective related events within the MUVE, and important key game events, such as: Entry/exit logging, Level ups, Items gained/lost, Items equipped/unequipped, Module & area entered, Movement (path traversed in the game world), and Conversation (not available in NWN1).

Information Trails:	Prototype 1 (2007-2008)	Prototype 2 (2009-present)
Game (year released):	Neverwinter Nights 1 (2002)	Neverwinter Nights 2 (2006)
Game Development Kit:	Aurora Toolkit	Electron Toolkit
Event Listener:	NWNX 2	NWNX 4
Tracer Report:	NWN Tracer 1	NWN2 Tracer
Development Platform:	JAVA application	Flex (Rich Internet Application)

NWN TRACER REPORT

Information Trails is an assessment design framework that advocates a match-up between goal-oriented events with key game events. Players who have completed the special game events would have also achieved the actionable learning objectives. Because it would be difficult for educators and trainers to extract the vast amount of game data for assessment directly, some amount of data cleaning and visualization is necessary to produce a humanly usable report.

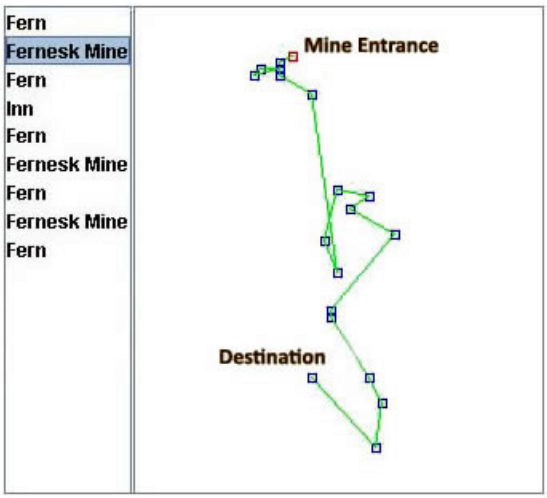

Figure 2: Box-and-line drawing showing the path traversed by a player.

Figure 2 shows the path traversed by a player in an area known as the Fernesk Mine. In this case (NWN Tracer 1), the path traversed was drawn to scale based on internal coordinates provided by the NWN game engine. The Mine Entrance is red in color, indicating it to be the "Starting Point" of the area. The rest of the blue boxes represent a sequence of snapshots for the player's position (action markers) within the MUVE, taken at six-second intervals. The green line simply connected the 'dots' of the action markers, giving a general sense of how the player traversed the Mine from Entrance to Destination. Because the sampling of data occurred at 6-second intervals, they are said to be a "lossy" (instead of "loseless') representation.

The JAVA application in NWN1 Tracer could only show the path traversed against a white background. The minimalistic box-and-line drawing failed to reveal any obstacle found in the mine that may have contributed to the path being "chosen" in this manner. This issue has been rectified in a later version of the Tracer, in which the path traversed has been overlaid upon an overhead view of the game area. The visual cues provided by the full-color bird's-eye view of the area were important to training managers because they could finally match trainee's actions to the geographical layout of the MUVE.

Figure 3. Moving the cursor over the Action Markers (white boxes) will reveal more information about the player's actions.

Because the NWN Tracers are software applications, they allowed the trainee's movement markers to be animated (time-lapsed). Furthermore, a "mouse-over"

of the action markers will reveal additional player information. In Figure 3, a trainer was able to identify the exact time (2007-01-22, 03:02:05) Junta Khan acquired the Master Key (item) to unlock the gate blocking the Maze Exit. Interactive features such as these are not possible in a printed (paper-based) AAR, at all.

Limitations

Because the NWN game engine was developed without the benefit of "Information Trails," some of the available game functions were considered incompatible or too limited for the purpose of game-based learning. For example, game developers of NWN1 wrote just two functions, *item_gained* and *item_lost*, to represent all possibilities involving the adding or removal of items from a player's inventory. There was no way to differentiate how these items were added or removed from the player's inventory within NWN1, even though players could gain items through any of the following means:

- obtained from a treasure chest
- bought from a merchant
- stolen from a non-player character (NPC)
- looted from a fallen enemy
- made by combining items (crafting) in the player's inventory
- created by a special spell
- given by an NPC, or another player in a persistent world

From the point of view of the game developer, however, all seven actions listed above were mere semantic differences and could be represented with just one function, *item_gained*. Readers should understand that the economy of game development was obviously very different for game assessment. Semantic differences such as these constituted important information to trainers and had immense value for *ad hoc* reporting, while they represent (pointless) additional work for the developers. This is reason enough to integrate Information Trails into the game engine, or at the very least during the game design phase, and not be retrofitted to the game after it has been released.

CONCLUSIONS

Training and instruction using MUVE are truly innovative approaches in training and education. However, the assessment tools for MUVE must be equally innovative! Designing games for learning is quite different from designing videogames because the former required the instructional designer to take into consideration the element of assessment. Linda G. Roberts, Director of Education Technology to the U.S. Department of Education, once said, "I believed that researchers could improve the design and collection of data. Just as new technology created new opportunities for learning, it created ways to invent new tools for research and evaluation, particularly ways to track and monitor what, how, and when learning occurred" (2003, p. viii). New assessment methodology must keep pace with the advances of technology for MUVE, in order to provide educators with the assessment data needed to garner support from stakeholders for these innovative instructional approaches.

When integrated with a game engine, the Information Trails allowed *ad hoc* action reporting that revealed what the trainees really "do" in MUVE. Instead of waiting for the entire training exercise to be complete before debriefing the trainees, the *a*AR provided trainers with a means to understand the trainees' decision making process as and when it occurred. The advantages provided by a software-based *ad hoc* Action Report include interactivity, real time reporting, and customizability. The *a*AR will not only cut down training time, but also allow for immediate feedback to take place between the trainers and the trainees. Mistakes committed during training could now be rectified immediately before they ran the risk of becoming entrenched. The implications for an *a*AR is impressive and should be targeted for further development by any organization interested in training using MUVE. As an innovative assessment methodology, "Information Trails" may be incorporated into a commercial game engine to give rise to an "improved" version of the game engine, allowing for *ad hoc* action reporting (*a*AR). It is foreseen that real-time *a*AR would serve to benefit corporations and military organizations by reducing the time and cost of training in multi-user virtual environments.

Acknowledgement: This research is made possible in part through the Defense University Research Instrumentation Program (DURIP) grant from the U.S. Army Research Office, and support from the Southern Illinois University Carbondale.

References

Chen, S., & Michael, D. (2005). Proof of learning: Assessment in serious games. *Gamasutra*. Retrieved from http://www.gamasutra.com/features/20051019/chen_01.shtml

Harrington, H. L., Meisels, S. J., McMahon, P., Dichtelmiller, M. L., & Jablon, J. R. (1997).

Observing, Documenting and Assessing Learning. Ann Arbor, MI: Rebus, Inc.

Kebritchi, M. (2008). *Effects of a computer game on mathematics achievement and class motivation: An experimental study*. Unpublished Doctoral Dissertation, University of Central Florida, Orlando, FL.

Loh, C. S. (2006). *Tracking an avatar: Designing data collection into online games.* Paper presented at the annual conference of the Association for Educational Communications and Technology (AECT 2006), Dallas, TX.

Loh, C. S. (2007). Designing online games assessment as "Information Trails". In D. Gibson, C. Aldrich & M. Prensky (Eds.), *Games and Simulation in Online Learning: Research and Development Frameworks* (pp. 323-348). Hershey, PA: Idea Group, Inc.

Loh, C. S., Anantachai, A., Byun, J., & Lenox, J. (2007). *Assessing what players learned in serious games: in situ data collection, information trails, and quantitative analysis.* Paper presented at the Computer Games: AI, Animation, Mobile, Educational & Serious Games (CGAMES 2007), Louisville, KY.

Loh, C. S., & Byun, J. H. (2009). Modding Neverwinter Nights into serious games. In D. Gibson & Y. K. Baek (Eds.), *Digital Simulations for Improving Education: Learning Through Artificial Teaching Environments* (pp. 408-426). Hershey, PA: Information Science Reference.

Mandinach, E. B., & Honey, M. (2008). *Data-Driven School Improvement: Linking Data and Learning*. New York, NY: Teachers College Press.

Means, B., Haertel, G. D., & Moses, L. (2003). Evaluating the effects of learning technologies. In G. D. Haertel & B. Means (Eds.), *Evaluating Educational Technology: Effective Research Designs For Improving Learning* (pp. 1-13). New York, NY: Teachers College Press.

Roberts, L. G. (2003). Foreword. In G. D. Haertel & B. Means (Eds.), *Evaluating Educational Technology: Effective Research Designs for Improving Learning* (pp. vii-x). New York, NY: Teachers College Press.