Integrating Team Experiential Learning into SCORM-Conformant Training

Craig Haimson, John Feeney, David Kramer, Mark Weston
Aptima
Washington, DC/Woburn, MA
haimson, jfeeney, dkramer, mweston@aptima.com

Mike Rustici
Rustici Software
Franklin, TN
mike@scorm.com

Mary Cherng, Ben Chess
Forterra Systems
San Mateo, CA
mcherng, bchess@forterrrainc.com

ABSTRACT

Distributed training environments such as multiplayer games decrease reliance on operational equipment and resources and reduce the need for co-located teammates and instructors. However, there is a need to coordinate and synchronize online scenario-based training with other forms of distributed learning such as self-paced didactic training via the web. SCORM provides a partial solution to this problem by enabling the interoperation of technologies that facilitate self-paced distributed learning. A SCORM-conformant learning management system (LMS) can serve any SCORM-conformant training package to any standard web browser and track the performance of individual learners as they progress through this material. Moreover, an LMS can tailor the delivery of learning content in accordance with Sequencing and Navigation rules that dictate the order in which different content packages should be presented. However, SCORM only supports didactic training. It does not provide a process for configuring experiential training and assessment platforms to simulate scenarios and compute performance measures that specifically address the learning requirements of individuals and teams. Moreover, SCORM is exclusively focused on individual learners and does not provide a method for (a) representing the learning requirements of a group of individuals or (b) adjudicating between the (potentially conflicting) needs of multiple learners. In a Joint ADL Co-Lab project in which we investigated methods for enhancing SCORM-conformant technology to enable coordination of team MOUT training in a multiplayer game. We addressed two main challenges: (1) defining representations that can be used to configure simulations and performance measurement technologies for team training and (2) developing technology and methods that enable synchronization of training across teams of individuals, based on the individual learning requirements that an LMS communicates for each trainee. In this paper, we describe our research and development effort, discuss lessons learned, and suggest directions for future research.

ABOUT THE AUTHORS

Craig Haimson is the Director of Aptima's Advanced Training Solutions Division. Dr. Haimson specializes in the development of technologies that support training and decision making. His primary interests are in cognitive modeling, simulation-based training, critical thinking, and formal knowledge representation. Dr. Haimson applies his expertise in cognitive science to the design of novel systems that provide scenario-based training, performance assessment, and decision support for domains such as military intelligence, information operations, and infantry operations. His work includes the use of expert system models and multiplayer online gaming environments to train teamwork skills, as well as the application of standard information representations such as SCORM to integrate distributed learning technologies. Dr. Haimson received a Ph.D. in Cognitive Psychology from Carnegie Mellon University and a B.A. in Psychology from Harvard University. He is a member of the Human Factors and Ergonomics Society, the Institute of Electrical and Electronics Engineers, and the Project Management Institute.

John Feeney is a Cognitive Scientist at Aptima, Inc. with interests in the cognitive aspects of computing. Including: establishing cognitive models, and applications of artificial intelligence technologies and advanced analytics. Prior to Aptima, Dr. Feeney held numerous positions at National Defense University supporting Joint Professional Military Education (JPME). Among his projects at NDU, Dr. Feeney was responsible for MENTOR a large Leadership Development system that incorporated a web-based multi-instrument delivery; automated scoring and expert system based individualized feedback. Previous to his work at NDU, Dr. Feeney was a Senior Software Engineer for Science Applications International Corporation (SAIC). This work included: expert systems, neural network classification systems, analytically based detection systems, modeling/simulation, and man-machine interface subsystems for the US Navy and US Air Force. Dr. Feeney received a Ph.D. in Applied Experimental Psychology from The Catholic University of America, a M.S. in Software Engineering from National University, and a B.A. in Computer Science from the State University of New York at Oswego.

David Kramer is a Software Engineer at Aptima, Inc. Mr. Kramer has interests in process automation, open source technologies, and improving the software development process using Agile practices and the appropriate selection and application of tools and techniques. Mr. Kramer has two decades of software design and development experience in a wide array of areas, including financial, inventory control, ETL, point of sale, document management, and embedded systems. His specialty is in Linux/UNIX. He is an officer in the Boston Linux and UNIX Group, in Agile Bazaar, the local chapter of Agile Alliance, a group that promotes Agile software development, and Agile Rules, a group promoting Agile software development in embedded systems. Mr. Kramer holds a B.A. from Hofstra University with two majors, Management, and Business Computer Information Systems. He is a member of the Association of Computing Machinery and the IEEE Computer Society.

Mark Weston is a Software Engineer at Aptima, Inc. His current work involves Java Expert System (Jess) development, as well as GUI development using the .NET Windows Forms libraries and C#. Mr. Weston is the lead software engineer for the T-CAST project, on which he is implementing an expert system that generates near-real-time assessments of simulation data generated during training exercises conducted within Forterra Systems' OLIVE MMOG. He has prior experience developing applications using Java Swing and other libraries. Mr. Weston graduated from Union College with a B.S. in Computer Science. He is a member of Phi Beta Kappa and Eta Sigma Phi.

Mike Rustici is the Principal of Rustici Software. Mr. Rustici is responsible for the daily operations of a rapidly growing software consulting business. He achieved industry-wide recognition as a leading provider of SCORM consulting services and successfully helped dozens of clients convert products to conform to the SCORM and AICC specifications for e-learning interoperability. Mr. Rustici also defined and developed best of breed tools for Learning Management System and training content standardization, including the SCORM Engine and SCORM Driver. As an active and respected member of the ADL community, Mr. Rustici Software is at the forefront of the e-learning industry. The ADL recently asked him to be a guest contributor to the ADL SCORM Developer portal, the first outside contributor sought by ADL. Prior to founding Rustici Software, Mr. Rustici worked as Product Development Lead at PureSafety, where he managed a team of LMS developers and led the adoption and integration of new technologies and methods, including SCORM. Mr. Rustici also worked as a software engineer at Astor Online and Columbia/HCA. Mr. Rustici received a BS in computer science from Vanderbilt University. He is a Microsoft Certified Solution Developer (MCSD), and he holds a certification in the Science and Art of Effective Web Design from Human Factors International. Mr. Rustici is also a member of the ADL Technical Working Group and the IEEE Learning Technology Standards Committee (LTSC).

Mary Cherng is a program manager at Forterra Systems, Inc. San Mateo, CA, with 9 years of experience working in the defense industry. Prior to Forterra Systems, she was a Program Manager for General Dynamics C4 Systems and held lead technical roles for several large U.S. Army programs. She holds a Bachelors degree in computer engineering from the University of Michigan and a Masters degree in electrical engineering from Arizona State University.

Ben Chess has been engineering many components of Forterra's OLIVE engine since 2004. Prior to joining Forterra, Ben developed handheld educational software for K12 students at GoKnow in Ann Arbor, MI. He holds a Bachelors degree in computer science from the University of Michigan.

Integrating Team Experiential Learning into SCORM-Conformant Training

Craig Haimson, John Feeney, David Kramer, Mark Weston
Aptima
Washington, DC/Woburn, MA
haimson, jfeeney, dkramer, mweston@aptima.com

Mike Rustici
Rustici Software
Franklin, TN
mike@scorm.com

Mary Cherng, Ben Chess
Forterra Systems
San Mateo, CA
mcherng, bchess@forterrrainc.com

INTRODUCTION

Many operational tasks are inherently team-based, and effective team training is at the heart of successful performance in military operations. Team training generally entails multiple participants working as teams in simulated exercises. Teammates provide mutually beneficial team learning experiences for each other through their responses to operationally realistic scenario events that exercise key knowledge and skills. Team training can be highly effective; however, the financial and logistical costs of team training can be considerable. Therefore, distance learning (DL) technology that can render team training readily effective, affordable, and available has tremendous potential for increasing military readiness. In this paper, we discuss a Joint ADL Co-Lab-funded project in which we analyzed team DL requirements and prototyped a model design for systems that could coordinate and deliver blended team DL.

Team DL Methods and Requirements

Due to the critical importance of team training for the military, the Department of Defense (DoD) has funded the research and development of numerous network-based technologies that allow teams of individuals to train together whether or not they are physically co-located. Simulations such as multiplayer games (MPGs) offer cheap, lightweight simulations that capture many key features of live training. For example, MPGs can simulate the performance of military operations on urbanized terrain (MOUT) in three dimensional graphical environments resembling the real-world urban theater. Trainees can practice recognizing and responding to key environmental cues that drive decision making during MOUT activities such as cordon and search or checkpoint operations. Such practice can build mission-critical cognitive skills that support the application of psychomotor skills that trainees must exercise through simulations with greater physical fidelity.

Distributed virtual training platforms decrease reliance on operational equipment and resources. They reduce the need for co-located teammates and instructors, eliminating some of the financial and logistical burdens associated with live training. They allow for significant control over the learning environment, which enables more precise targeting of learning conditions to learning objectives. These platforms facilitate the integration of auxiliary technologies that can support and/or enhance the learning process (e.g., synthetic entities that stand in for non-core participants in training exercises, tools for collecting behavioral data and computing performance measures, etc.).

Distributed team training platforms will be most effective in combination with more traditional, self-paced instruction offered through web-based training (WBT). So-called "blended" learning solutions that combine individual didactic and group experiential learning have the potential to address a wide range of educational requirements: Students can acquire baseline knowledge via WBT and practice applying this knowledge in operationally realistic contexts via online team training exercises. Blended learning has been shown to increase learning as well as improve student interaction and satisfaction (Rossett, Douglis, & Frazee, 2003).

To realize the benefits of blended learning for team training, it is essential that combined training solutions tightly coordinate experiential and didactic components. Developers must design blended training packages around a common set of learning objectives. Learning management systems must track students' progress against the learning objectives. Instructional content must be consistent and mutually reinforcing across modes of delivery (i.e., individual didactic WBT must prepare students to participate in team experiential learning exercises). Moreover, training delivery mechanisms must be capable of adapting both forms of training in synchrony to accommodate the evolving needs of all individuals in the group. This will eliminate redundant training when learners have demonstrated mastery and remediate deficits when learners have already demonstrated specific weaknesses.

These fairly specialized training management requirements must be reconciled with the military's objectives of ensuring interoperability and reusability of e-learning technologies and content. Given the overlap in core learning objectives across the services, there is enormous potential for sharing/repurposing learning content; however, when training is not interoperable, there is a significant risk of duplicating development efforts. To maximize the impact of training investments and reduce development time, online training should be reusable, interoperable across computer platforms, and capable of being recombined into new training packages to suit the needs of different groups of consumers.

Uses and Limitations of SCORM

The DoD Instruction 1322.26 (2006) encourages interoperability and reuse by mandating the adoption of the Shareable Content Object Reference Model (SCORM) 2004 (http://www.adlnet.gov/). While it is intended to facilitate interoperability and reuse, SCORM also provide a number of features that can contribute to team blended learning. A SCORM-conformant learning management system (LMS) can serve any SCORM-conformant training package to any standard web browser and track the performance of individual learners as they progress through their material. It can maintain a record of student performance as reflected by the values of specific measures associated with specific learning objectives. An LMS can tailor the delivery of learning content in accordance with SCORM Sequencing and Navigation (SN) rules that dictate the order in which different content packages (Shareable Content Objects, or SCOs) should be presented. By consulting SN rules and assessing a learner's progress toward mastery of learning objectives (as determined by associated measures of performance and preset threshold values), an LMS can determine the next SCO to be delivered to a learner.

SCORM does not provide a convenient process for configuring experiential training and assessment platforms to simulate scenarios and compute performance measures that specifically address the learning requirements of individuals and teams. SCORM was developed to enable interoperability of didactic WBT. An LMS typically serves training content by launching it within a user's web browser. However, in many cases, experiential learning technologies cannot be accessed and utilized through standard web-based methods. For example, MPGs require dedicated servers that manage the presentation of graphics-intensive simulations and mediate rapid, complex, interactive responses of multiple distributed online users.

SCORM was designed to address training management for individual learners. It does not provide a standardized method for representing the learning requirements of a group of individuals, nor does it specify procedures for adjudicating between the (potentially conflicting) needs of multiple learners. Teams are composed of individual learners with different strengths and weaknesses in terms of requisite knowledge, skills, and abilities (KSAs). Teammates may share a set of common learning objectives, but each individual team member will require differing amounts of training to satisfy those objectives (trainees will have differing levels of prior exposure to trained concepts, differing ability to perform training tasks successfully, differing levels of competency required for execution of individual job responsibilities, etc.). For example, Figure 1 shows learner profiles for two hypothetical students with differing levels of proficiency on mission critical KSAs. To address these training needs, Student 1 might require additional practice in combat maneuvers, while Student 2 might require additional practice as a radio operator. To address instances such as these, it is critical that team learning experiences are tailored to the aggregate educational needs of all the individuals participating in the training.

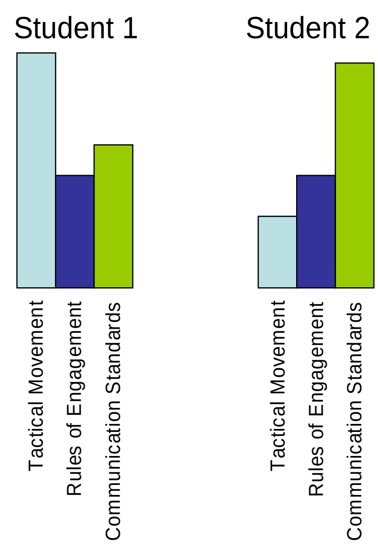

Figure 1. Two hypothetical student profiles.

In summary, SCORM satisfies many of the DoD's requirements for interoperable and reusable e-learning. However, it does not by itself support configuration of experiential learning platforms or accommodate the needs of multiple trainees.

Project Objectives

We investigated methods for enhancing SCORM-conformant technology to enable coordination of team MOUT training in a multiplayer game. We addressed two main challenges: (1) defining representations that can be used to configure simulations and performance measurement technologies for team training and (2) developing technology and methods that enable synchronization of training packages across teams of individuals, based on the individual learning requirements that an LMS communicates for each trainee. In this paper, we describe our research and development effort, discuss lessons learned, and suggest directions for future research.

PEDAGOGICAL FRAMEWORK

Integrating online experiential and didactic learning requires linking very different types of content to common learning objectives. In both types of learning, training objectives are generally expressed in terms of the acquisition of knowledge and skills. Moreover, both seek to provide training and assessment using networked technologies that minimize or eliminate the involvement of human instructors. However, didactic and experiential training support mastery of learning objectives in very different ways.

Comparison of Didactic vs Experiential Training

Web-based didactic training content may include textual, graphical/video, and audio assets that portray information in a typically “lecture-styled” format. There are few opportunities for the learner to interact with the material during the presentation. In contrast, the conditions, objects, and interactions simulated in an MPG or other platform constitute the primary training content in experiential training – they allow for specific tasks to be executed in specific ways. In combination with the back story and instructions for different players, these components comprise an essential building block of experiential learning: the scenario.

Didactic learning generally employs assessment tests as a mechanism for determining whether or not students have achieved designated learning objectives. Learners provide answers to test questions that assess fact recognition or recall. Tests may also require students to apply knowledge and skills to derive answers to problem sets; however, in almost all cases, there is a clear separation between teaching and testing phases of operation. In contrast, experiential learning assessments are entirely based on quality of simulated performance. Learners satisfy learning objectives when they attain appropriate levels of performance on measures of requisite knowledge and skills. Scenario events constitute the “test questions,” to which trainees respond by producing simulated actions. The actions accomplish a simulated task with an intended operational objective that is generally executed in response to scenario events and/or instructions.

There are different formats for presenting didactic test questions and assessing responses. Many didactic test questions require students to select from among a list of possible answers. This format simplifies assessment: Each question has a direct association with key training objectives, and each potential response has an unambiguous interpretation (i.e., whether or not it is correct, and why). As a result, automated scoring is straightforward and easy to implement. Didactic tests may include free text questions, as well, although automated evaluation of natural language responses can be complicated, inaccurate, and incomplete. In experiential learning, specific patterns of simulated action can be identified as measures of correct vs incorrect individual- and team-level performance on the simulated task, which can ultimately be traced back to trainees’ proficiency with respect to key knowledge and skills (as assessed in comparison with a pre-specified standard) these patterns of responses have correlates in the simulation activity data that an MPG records on an ongoing basis. Simulation data also provide indicators of the events that prompted the response and conditions under which those events occurred, which can significantly impact the interpretation of a trainee’s responses. The relationship between objectives, events, conditions, measures, and assessments is shown below in Figure 2.

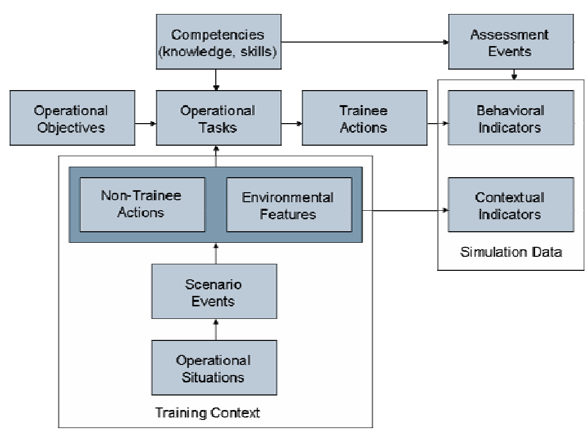

Figure 2. Framework for experiential learning.

Blended Learning Framework

With didactic and experiential content and assessments appropriately bundled, it is possible to define a blended curriculum that links didactic and experiential learning. Didactic content presents facts that can be applied during experiential exercises, as well as instructions for performing key skills. For example, didactic content might provide instruction in basic MOUT doctrine such as methods for room clearing, and experiential content might configure a MPG training scenario in which students practice clearing a virtual room. Objectives/measures for a house search might include:

- Stacking
    - Maintain appropriate distance to wall
    - Maintain appropriate inter-team distance
    - Avoid windows
    - Quickly execute entry
    - Never enter an uncleared room without stacking
- Appropriate Weapons Handling and Fire
    - Avoid flagging/firing on Blue/unarmed Red
- Establish and Maintain Room Coverage
    - Clear all parts of any uncleared room
    - Provide coverage on all doors to uncleared areas

Students might take didactic tests before progressing to experiential lessons; depending on students' demonstrated mastery of key learning objectives, they could receive additional didactic training that remediates key deficits or begin experiential training that is tailored to their current abilities. Similarly, based on demonstrated performance during experiential training, students might receive additional post-exercise didactic training that refreshes key lessons or explores topics in greater depth. The exact strategy for blending experiential and didactic training and assessment should be at the discretion of the instructor and/or training developer; in practice, the availability of relevant content may drive these kinds of curriculum decisions.

The introduction of team-level experiential learning adds elements to this framework. In general, we assume that teams of individuals will participate in experiential training, while didactic WBT remains self-paced, individual-level study (although team-level, synchronous web seminars or "webinars" are a viable didactic learning alternative). Each individual can continue to receive tailored WBT in response to demonstrated performance deficits/leaning requirements. However, it will be necessary to optimize experiential training across the team of participating individuals. Training could include both individual and team level performance objectives and measures, both of which could influence the configuration of subsequent training episodes.

There are a number of different policies that could govern this group training assignment process. One set of policies could govern the termination of a particular training package. Students could repeat the same training scenario until group performance met some established criterion, such as:

- Repeat until *ALL* Individual and *ALL* Group Training Objectives are met.
- Repeat Until *ALL* Individual Training Objectives are met.
- Repeat Until *ALL* Group Training Objectives are met.
- Repeat Scenario *x* number of times or until *ALL* Group and *ALL* Individual Training Objectives are met.
- Repeat Scenario *x* number of times regardless of performance.
- Repeat Scenario for *x* amount of time (clock time) regardless of performance.

We explored the implications of choosing different policies using Monte Carlo simulation. We assumed teams of five individuals performing training with one to four individual-level training objectives and one to four team-level training objectives. We further varied the probability of a given individual achieving individual- vs team-level objectives (65%, 70%, 75%, 80%, 85%, 90%, 95%). We simulated 1000 Monte Carlo runs for each combination of factors, calculating the number of simulated scenario attempts required to terminate training in accordance with a given policy. This exercise illustrated the importance of carefully considering trainee skill levels across the team when selecting training assignment policies. As Figure 3 shows, more restrictive policies that require consistently high performance across the team could create situations in which a team requires an unreasonably high number of scenarios to achieve criterion performance. In general, training assignments should balance the needs of learners with varying levels of proficiency, ensuring that all players will, on average, obtain useful learning experiences during team training exercises.

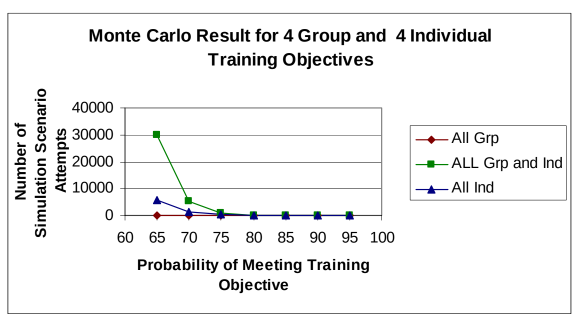

Figure 3. Sample simulation results.

PROTOTYPE SYSTEM DESIGN

We developed a prototype system design around the blended learning framework described above. Our prototype addresses issues in configuring and sequencing individual-level WBT and team-level experiential training. Our design does not address some more practical issues associated with scheduling and coordinating experiential learning exercises, as these concerns were outside the scope of our effort. Instead we chose to focus on methods by which we could extend a SCORM-conformant training design to accommodate blended learning packages for individual and teams.

Our goal was to leverage SCORM to whatever extent possible, adding middleware components to provide additional functionality wherever necessary. SCORM does not provide a standardized method for representing the learning requirements of a group of individuals, nor does it specify procedures for adjudicating between the (potentially conflicting) needs of multiple learners. However, it does dictate a set of minimal LMS behaviors and provide a standardized content representation formalism, as well as a robust methodology for sequencing SCOs based on a learner's demonstrated proficiency across a set of learning objectives. We enriched these capabilities with additional components that enabled SCORM-driven adaptation of team experiential learning.

SCORM-Simulation Integration

An LMS typically launches a SCO within a client web browser, using a Javascript API to support communication between SCO and LMS. Most SCORM content packages include an Extensible Markup Language (XML) manifest file containing SN rules that determine the next SCO that an LMS should launch next. The SCO informs the LMS when the student has completed it and communicates the results of any assessments computed while the SCO was active.

Traditional SCORM implementations do not lend themselves to interactions with training simulations. However, while SCORM does not provide methods that explicitly direct the configuration of experiential training technologies, neither does it prevent or limit such extensions, as some recent prototypes have demonstrated (e.g., Biddle, Perrin, Dargue, Pike, Marvin, & Lunsford, 2006). A SCO is free to communicate with other entities outside of the web browser and LMS; this could include a dedicated didactic learning test bank or another DL component (note that Javascript prevents cross-domain scripting, which complicates communication with components outside of the user's network domain, however). Moreover, while SCOs typically contain HyperText Markup Language (HTML) web content, they can also contain non-browser-renderable data that could be used to setup experiential learning sessions.

In our prototype design, we embedded data required to parameterize a simulation for a particular training scenario within a SCO. These could include the specific avatar/role to be assigned to an individual on the team (different avatars/roles could lend themselves to different tasks that exercise different learning objectives), as well as other configuration information (e.g., avatar equipment or terrain composition). Scenario setup could also entail configuration of experiential learning performance measures to be applied during the scenario. These could take the form of rules that a performance measurement technology (PMT) uses to identify and evaluate patterns of simulator data generated during the scenario; different rules could compute different measures or compute different assessments of the same measurement (e.g., by changing the criterion value against which the measurement is compared).

In our prototype, we use an XML Schema to direct scenario setup and performance measurement configuration. The MPG parameterizes avatars and terrain in accordance with these XML instructions before players login to the environment. Concurrently, a dedicated PMT expert system preloads a set of rules and initialization facts. Rules can represent patterns that reflect discrete examples of correct or incorrect actions, or they can capture performance trends that may be evaluated through comparison with one or more standards (e.g., expected vs observed task completion time).

As described, SCORM does allow for communication between a SCO and additional training system components such as MPGs and PMTs. Javascript does prevent the SCO from directly configuring components that exist in a different networking domain (which is likely when multiple distributed players are accessing the same server), but this limitation may be overcome by various methods, such as introducing additional middleware technologies that serve as bridges between SCOs and external applications. Our prototype system uses a middleware application that we call the SCORM-team interface component, or STIC. The STIC configures an MPG and PMT in accordance with XML descriptions of scenarios and associated measurements. However, its most important function is to overcome SCORM's inherent focus on the individual-level.

Configuring Team Training with the STIC

The ADL developed SCORM to represent and control individual-level, self-paced WBT. Multiple trainees may contact an LMS to receive training, but the LMS will launch an independent, self-contained SCO within each individual's web browser. The LMS will maintain separate communication streams with each trainee's SCO and tailor its responses to the needs of each individual trainee without regard to those of another. While these constraints are entirely appropriate for self-paced individual WBT, they do not allow for coordinated team-level training. An LMS can determine the best experiential training to be delivered for each individual trainee, but it cannot reconcile conflicts between individual learning requirements to determine the optimal training package to deliver an entire team.

In our prototype design, the STIC compensates for this limitation. The STIC serves as a clearing house for communications sent by each individual trainee's SCO. For each individual learner, the LMS launches a SCO that communicates the individual learner's unsatisfied learning objectives to the STIC. The STIC combines individual recommendations and selects a course of training that achieves the best fit to the needs of various individuals in accordance with a specific team training assignment policy. The STIC then communicates an SN request to the LMS through the SCO, which prompts the LMS to launch a specific experiential learning SCO. This, in turn, configures and launches MPG and PMT applications via the intervention of the STIC. Note that the STIC can consider team-level learning objectives, as well: As long as an external PMT can compute team-level assessments and communicate these to the SCO, the SCO could communicate these back to an LMS, which would store them within each individual's performance record (i.e., each individual record would include an identical copy of the team's performance history).

Adapting Team Training with SCORM SN

Our design uses the STIC to adjudicate conflicts in each individual's learning requirements to determine a team training package that satisfies the aggregate learning requirements of the team (as defined by STIC training assignment policies). However, our design relies on SCORM SN to determine individual learning requirements. The manifest file describes the conditions under which the LMS should deliver different experiential learning packages (i.e., different instructions for configuring the MPG and PMT) given the individual's current progress toward satisfying learning objectives.

The manifest file also describes how didactic self-study WBT SCOs may be interleaved with team experiential learning SCOs. Didactic SCOs could teach knowledge and skills required for experiential training exercises. Moreover, they could provide remediating content that addresses specific deficits that trainees demonstrate during experiential training (e.g., poor performance on a house search scenario might trigger the presentation of WBT that reviews basic room clearing tactics). Each individual trainee might come away from a team experiential training session with a different set of performance deficits requiring a different assortment of didactic WBT SCOs. By combining experiential and didactic content within the same content package and using a single SCORM manifest file to control sequencing and navigation between individual- and team-level content, we sought to create a single, fully-integrated blended learning design that leverages the full power of the SCORM standard. We believe such blended designs have the potential to maximize the use synergies between individual vs team and didactic vs experiential learning paradigms.

Our ultimate training design addresses a final requirement that we identified for using SCORM SN to determine candidate experiential learning SCOs for each individual. Consider what would happen were an LMS simply to sequence to a given candidate experiential learning SCO, communicate this SCO to the STIC, and allow the STIC to determine the best experiential learning SCO amongst all candidates and launch the winner within the MPG and PMT. The LMS would be left in a position of having recommended a particular SCO without ever knowing whether the student actually received that training (or, more precisely, it would be left assuming that the student did receive that training even if the STIC overruled the recommendation). In order for the LMS to maintain an accurate record of the training that students received, it is necessary for the STIC to communicate its SCO selection back to the LMS.

Our design recognizes a distinction between the experiential learning SCOs, which actually configure experiential training platforms, and the coordination SCOs, which facilitate a dialogue between the STIC and the LMS regarding training to be delivered. Didactic SCOs sequence the coordination SCOs that represent the LMS's suggested experiential learning package. These communicate with the STIC, which determines the actual experiential SCO to be delivered to the team. The coordination SCOs communicate this information to the LMS, which then sequences to the appropriate experiential learning SCOs, which will then work through the STIC to launch and configure the appropriate team experiential training scenario. Upon completion of the scenario, the PMT will communicate individual-level and team-level performance assessments back to the STIC, which will convey them to the appropriate individuals' SCOs and, in turn, to the appropriate individual's LMS performance record

PROTOTYPE IMPLEMENTATION

Figures 4 illustrates our implementation of this prototype design. Figure 4 shows the prototype components, including SCORM LMS (the Joint ADL Co-Lab's Sample Run Time Environment), the STIC, the user's browser, SCOs launched by the LMS, the MPG (Forterra Systems' OnLine Interactive Virtual

Environment, or OLIVE – see Figure 5 for an example), and the PMT (the Team Coaching Assistant for Simulation-Based Training, or T-CAST, developed by Aptima and BBN Technologies). Note that the OLIVE client resides on the user's machine and that the OLIVE server requires an additional component, called the Basic Application Launch Layer (BALL), to configure the OLIVE server in response to XML scenario descriptions sent by the STIC.

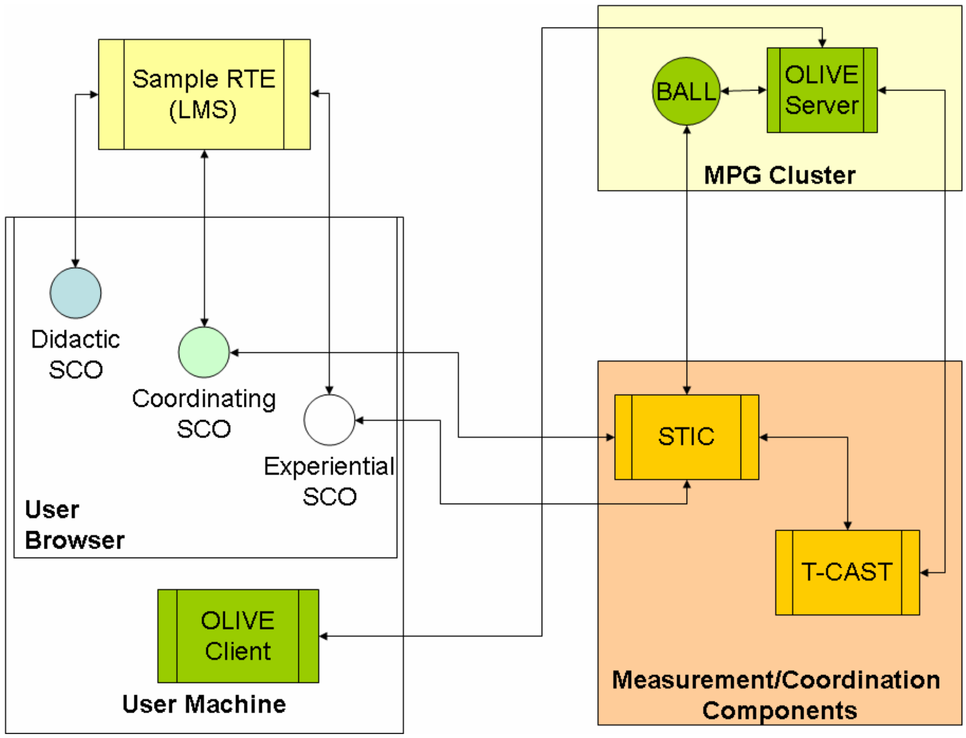

Figure 4. Prototype system components.

Forterra Systems developed OLIVE with funding from the US Army's Research Development and Engineering Command Simulation Training Technology Center (RDECOM-STTC) and additional venture backing. OLIVE is a massively multi-user persistent 3-D virtual world platform that allows users on standard PCs, widely distributed over the Internet, to interact in real time at the entity level. A notional one square kilometer urban setting geo-referenced to Baghdad has been modeled for use in MOUT infantry training (Figure 5).

Figure 5. Search scenario simulated in OLIVE

As part of a previous effort funded by Army RDECOM STTC, we developed a set of scenarios to be simulated within OLIVE and built the T-CAST PMT to provide automated support for OLIVE's after action review (AAR) module (Haimson & Lovell, 2006). We implemented T-CAST using Jess, the Java Expert System developed by Sandia Labs (http://herzberg.ca.sandia.gov/). T-CAST consists of a primary expert system containing a working memory that actively maintains representations of "rules" and "facts," as well as an inference engine that maps facts to rules that reference them as conditions (if a rule references a given set of facts, and the working memory contains those facts, the inference engine determines that the rule is "true"). Rules represent patterns of data associated with to-be-recognized events, and the event recognition technology utilizes matching techniques to map these patterns facts representing actual data collected by the gaming/simulation engine (e.g., OLIVE) during a scenario run. T-CAST performs this analysis in near-real-time and communicates the results to the OLIVE trainer module while a scenario is in progress, as well as to the AAR module and STIC at the conclusion of the scenario. After students complete the AAR, the STIC communicates T-CAST results to each student's SCO, which conveys the data to the appropriate LMS records.

For the current effort, we developed the STIC, BALL, and a blended learning SCO that includes experiential learning, didactic, and coordination SCOs. The experiential learning SCOs contain XML specifications for configuring the MPG and PMT to support different scenarios, while the didactic SCOs contain basic WBT content compiled from Army doctrine describing room clearing tactics. The OLIVE server contains more detailed instructions for rendering appropriate terrain in accordance with the scenario description communicated by the STIC. In addition, the T-CAST server contains a cache of available rules and facts that the system loads in accordance with the scenario description file.

While fairly limited in scope, this demonstration illustrates key elements of our blended learning design. Experiential and didactic training are integrated within a single SCORM-conformant training package that includes both traditional didactic WBT content, as well as MPG/PMT configuration data. The demonstration uses SCORM SN to compute experiential learning recommendations for each individual user, which the STIC adjudicates between to determine the ultimate experiential training to be delivered next to the team.

Although we developed our demonstration using OLIVE and T-CAST, the basic design could accommodate other MPGs and PMTs that can be configured on the fly using simple commands that set key simulation parameters and/or load scenario-appropriate content. In our design, the PMT communicates SCORM-appropriate assessments to the STIC, which then conveys the data to the SCOs/LMS. It would be possible to use a PMT that does not provide SCORM-appropriate assessments, but it would then be necessary to develop STIC procedures that translate raw data into the SCORM runtime environment's computer managed instruction (CMI) format.

CONCLUSIONS

Currently, there is no single e-learning solution that coordinates group blended learning while adhering to interoperability standards. Most technologies do not adapt learning experiences to individuals in a group, and those that do provide this service perform it in a non-standard manner that prevents interoperability with other training systems and learning content. This technology gap significantly limits the utility of team distributed learning. The inability to adapt training for teams of individuals in a standardized way limits the application of distributed learning as a viable paradigm for team training, impeding the DoD's ability to realize the full potential of the internet to coordinate and deliver service and interservice training.

We have described an instructional system design that enables the delivery of SCORM-conformant blended learning packages for teams of individuals, using shared learning objectives to link different types of training content and delivery mechanisms. While our system demonstration uses several components that we previously developed and integrated outside of this effort, we nevertheless believe that it illustrates a general method by which these types of components can be combined using middleware components that compensate for SCORM's inherent focus on individuals.

We note that the design does require training developers to create customized MPG simulations and PMT rulesets that are then invoked via experiential learning SCOs. However, many simulations and PMTs do include content authoring tools and the capability for adaptive configuration. We found it fairly straightforward to modify our own systems for this effort. Moreover, the capability to distribute training to a wide audience compensates for some of the added expenditures of time and resources required to setup these technologies.

In addition, we note that the blended learning content packages are somewhat difficult to design. Instructional designers must carefully construct activity trees that link didactic and experiential SCOs in appropriate ways. Moreover, because we use SCORM as an essential mechanism for adapting training, we are limited to configuring experiential content prior to the start of a scenario (i.e., our system cannot modify a simulation at runtime even though we calculate performance measures on an ongoing basis).

Our goal in this effort was to leverage SCORM to the greatest extent possible in our design. The SCORM instructional design has its limitations, but we believe it illustrates the potential of an e-learning standard for describing and delivering fully integrated blended learning packages. We found that we could use the STIC to bridge the gap between individual- and team-level training. However, we see the STIC as a temporary solution that should ultimately be addressed by changing the standard to accommodate team-level coordination of learning experiences. Future prototyping efforts should identify additional requirements to be addressed in new versions of SCORM.

ACKNOWLEDGEMENTS

This work is supported with funding from the Joint ADL Co-Lab. We thank the Co-Lab and RDECOM STTC for their support of this effort. The ideas expressed herein are those of the authors, and are not necessarily endorsed by the funding agency. This material is approved for public release; distribution is unlimited.

REFERENCES

Biddle, E., Perrin, B., Dargue, B., Pike, W.Y., Marvin, D., & Lunsford, J. (2006). Performance- based advancement using SCORM 2004. *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference* (Orlando, FL; Dec. 4-7).

DoD Instruction 1322.26, "Development, Management, and Delivery of Distributed Learning", June 16, 2006.

Haimson, C. and Lovell, S. (2006). Pattern recognition for cognitive performance modeling. Paper presented at the American Association for Artificial Intelligence (AAAI) 2006 Fall Symposium on Capturing and Using Patterns for Evidence Detection, Washington, DC.

Rossett, A; Douglis, F. And Frazee, R. (2003), Strategies for Building Blended Learning, 2003 (http://www.learningcircuits.org/2003/jul2003/rossett.htm)