Distributed Learning Interoperability with Embedded Training and Intelligent Tutors

Randy Jensen, Coskun Tasoluk
Stottler Henke Associates, Inc.
San Mateo, CA
Jensen@stottlerhenke.com,
Tasoluk@stottlerhenke.com

Don Holmes, John Campbell
Imedia.it Inc.
Houston, TX
Don.holmes@imedait.com,
John.campbell@imediait.com

Dean Marvin
Joint Advanced Distributed Learning Co-Lab
Orlando, FL
Dean.marvin2@us.army.mil

Eugenio Diaz
UCF Institute for Simulation and Training
Orlando, FL
Ediaz@ist.ucf.edu

ABSTRACT

Training benefits have been widely documented for Intelligent Tutoring System (ITS) approaches integrated with virtual trainers, especially for embedded settings where human instructors are less available. Yet these forms of training are generally not available in distributed learning curricula with models such as SCORM (Sharable Content Object Reference Model), due to the limitations of browser based delivery methods. However, emerging concepts in the distributed learning community provide mechanisms that can be used to construct interoperability with such trainers. Notionally, a SCORM course could incorporate training events on a variety of available platforms, such as an embedded training system or an immersive virtual system at a training facility, which can exercise equivalent learning objectives. This paper describes findings from a prototype effort for the Joint Advanced Distributed Learning Co-Lab, to construct a mechanism for SCORM interoperability with an existing embedded training testbed, the Command and Control Vehicle (C2V) ITS at Army RDECOM. The C2V testbed exercises unmanned vehicle control concepts and skills, in a physical configuration mirroring the embedded training setting onboard an actual C2V. As a standalone trainer, this testbed cannot be integrated with a browser based learning environment in the traditional manner. Therefore the interoperable architecture includes a mechanism to configure and launch training events based on the instructional sequencing inputs from a SCORM course, while aiming to provide a simple transition for the learner. The training system performs automated assessment during simulated exercises, with a collection of evaluation mechanisms tied to specific learning objectives. Performance results are therefore compiled automatically and internally in the C2V ITS, in a format that can then be relayed to populate SCORM learner profiles at exercise conclusion. This paper summarizes the interoperable design, followed by a discussion of the road ahead for similar extensions to support additional forms of training events.

ABOUT THE AUTHORS

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

Coskun Tasoluk is a lead software engineer at Stottler Henke Associates, Inc. He has designed and implemented several Intelligent Tutoring Systems for embedded training systems, such as the Command and Control Vehicle (C2V) testbed, Embedded Simulation System (ESS), and the Virtual Warrior dismounted player unit. His primary interests are web based intelligent learning systems, information extraction, data mining and behavior modeling He holds an M.S in electrical engineering and an M.S in computer science from Drexel University.

Don Holmes is the CEO of Imedia.it, Inc. He has been working in e-learning for over 11 years. During that time he has been responsible for the effort at Imedia.it to adopt the ADL SCORM initiative. He is a member of the ADL Working Group and the SISO SCORM-Simulator Interface Study Group. He has also been involved in project management including a joint effort with Stottler Henke to develop the Intelligence in Combating Terrorism ITS which was the first SCORM 2004 conformant course delivered to the Army. He holds an M.B.A. from University of Dallas.

John Campbell is the Technical Manager at Imedia.it, Inc. He has over 11 years experience designing and developing software for a variety of industries. For the past three and one-half years, he has developed simulation-based training for the US Army. He has developed an "over the web" Java simulation of the Army's Common Ground Station (CGS) as well as numerous map based simulations for the Intelligence Analyst course. He has spoken at ADL conferences and events and has participated on technical SCORM 2004 panels. He is also an active member of the SCORM 2004 Technical Working Group. He holds a B.A. in computer science and a B.A. in Spanish Linguistics from Rice University.

Dean Marvin is a retired Marine Lieutenant Colonel and works as a Senior Military Analyst at the Joint Advanced Distributed Learning Co-Lab in Orlando, FL. He helps lead the Research and Development efforts at the Co-Lab. Dean has a BA from Florida Southern College and an MAE from Chapman University.

Eugenio Diaz is a Research Associate at the University of Central Florida's Institute for Simulation and Training (UCF IST), in the Department of Applied Research and Technology.

Distributed Learning Interoperability with Embedded Training and Intelligent Tutors

Randy Jensen, Coskun Tasoluk
Stottler Henke Associates, Inc.
San Mateo, CA
Jensen@stottlerhenke.com, Tasoluk@stottlerhenke.com

Don Holmes, John Campbell
Imedia.it Inc.
Houston, TX
Don.holmes@imedait.com, John.campbell@imediait.com

Dean Marvin
Joint Advanced Distributed Learning Co-Lab
Orlando, FL
Dean.marvin2@us.army.mil

Eugenio Diaz
UCF Institute for Simulation and Training
Orlando, FL
Ediaz@ist.ucf.edu

INTRODUCTION

Training benefits have been widely documented for Intelligent Tutoring System (ITS) approaches integrated with virtual trainers, especially for embedded settings where human instructors are less available. Such systems provide effective experiential learning with a mixture of practice and automated performance assessment. Yet these forms of training are generally not available in distributed learning curricula with models such as SCORM (Sharable Content Object Reference Model), due to the limitations of browser based delivery methods. However, emerging concepts in the distributed learning community provide mechanisms that can be used to construct interoperability with such trainers. Notionally, a SCORM course could incorporate training events on a variety of available platforms, such as an embedded training system or an immersive virtual system at a training facility, which can exercise equivalent learning objectives.

This paper describes findings from a prototype effort for the Joint Advanced Distributed Learning (JADL) Co-Lab, to construct a mechanism for SCORM interoperability with an existing embedded training testbed. The testbed used for this research is the Command and Control Vehicle (C2V) ITS at the Army Research Development and Engineering Command (RDECOM) Simulation and Training Technology Center (STTC). The C2V ITS exercises unmanned vehicle control concepts and skills, in a physical configuration mirroring the embedded training setting onboard an actual C2V. As a standalone trainer, this testbed cannot be integrated with a browser based learning environment in the traditional manner. Therefore the interoperable architecture includes a mechanism to configure and launch training events based on the instructional sequencing inputs from a SCORM course. The intended consequence is to provide a simple transition for the learner. The training system performs automated assessment during simulated exercises, with a collection of evaluation mechanisms tied to specific learning objectives. Performance results are therefore compiled automatically and internally in the C2V ITS, in a format that can then be relayed to populate SCORM learner profiles at exercise conclusion. This paper summarizes the interoperable design, followed by a discussion of the road ahead for similar extensions to support additional forms of training events.

BACKGROUND

The SCORM Run-Time Environment (RTE) standard dictates that learning content be delivered in a browser based environment, which presents a major impediment for a wide range of potential training environments and applications that would otherwise be desirable for inclusion in a SCORM course. There have been several efforts exploring methods to circumvent or eliminate this barrier, with some direct influence on the direction of this research. In particular, this research involved a collaboration with another prototype effort, conducted by BBN Technologies for the JADL Co-Lab, which provided an initial implementation of a mechanism for integrating didactic and experiential learning (Travers et al, 2007). BBN implemented a working version of a construct called the Distributed Training Event Coordination Service (DTECS), which will be elaborated upon below. The existing DTECS implementation supported an interoperable use case where a traditional browser-based didactic (i.e., SCORM) course could include simulation based training events, conducted on the same machine as the browser course.

As many embedded trainers and complex virtual trainers require entirely separate hardware, the DTECS component would need to be extended to accommodate such a use case. Following a collaborative design and requirements dialog, we proceeded with two parallel developments. BBN implemented the modifications to the DTECS component itself, while our effort focused on the development of a new component on the training system side that would be responsible for communicating with the DTECS, and managing the execution of training events for compatibility with the DTECS and SCORM course structure.

In addition to the concurrent BBN effort, there are several other related efforts that align with technical objectives in this research. Basically, a means is needed to combine didactic content and simulations on a single learning platform and to provide a means to navigate between them. If the simulation is browser based, then SCORM 2004 Simple Sequencing and Navigation is adequate. There are issues when these simulations are not browser based regardless if they are pc based or free standing units. If the content is web based, the RTE can select Sharable Content Objects (SCOs) based on sequencing parameters and report completion status for attempts on SCOs and the related learning objectives to a Learning Management System (LMS). If the content is not web based, the LMS can still select the next activity but another means is needed to launch it. Additionally, that mechanism must adequately perform those operations and should not only interface with the simulation but related sub-systems, such as ITS. Also, the methodology must adequately deal with workflows defined by strong instructional design.

Biddle et al. (2006) showed the integration of didactic courses with simulations. The student's performance in the simulation was captured and used to direct the student to didactic content using SCORM sequencing and navigation. The standalone simulation was launched from a SCORM course using ECMA Script and Java applets.

Weil et al. (2007) outlined differences between didactic instruction and simulations. They examined those differences in several areas; clarity of training objectives, granularity of training, performance tracking, definition of training objectives and implications of linearity. They found that combining the learning types in training provides different preferred instructional approaches and more comprehensive performance tracking. Additionally, having a single framework manage both pieces would make the curriculum design process easier.

Galaxy Scientific Corporation (now SRA International) sought to explicitly map the constructs of a typical ITS to the data structure in SCORM 2004 1st Edition (Anthony and Ashworth, 2006). They addressed the differences in terminology between SCORM 2004 and that used by cognitive scientists and engineers who typically develop ITS, concluding that though the features and capabilities were not neatly packaged in SCORM 2004, they nevertheless did exist.

Stottler Henke and Imedia.it developed an adaptive course to train counter-terrorism intelligence analysis, in a system called the ICT ITS (Ramachandran et al, 2006). The overall instructional scheme for the course involved first an assessment exercise scenario, the delivery of support lessons, followed by introductory exercise scenarios, and finally more advanced scenarios. The introductory scenarios utilized ITS as a learning enhancement tool and the SCORM 2004 features were used in the mapping from the ITS notion of principles to the SCO embodiment of learning content. Each support lesson was treated as a SCO, and each scenario variation was structured as a SCO. Each scenario SCO contained specific metadata necessary to link the simulation to the data files that defined the particular scenario. The implementation with this domain effectively made the training simulation accessible for distributed learning.

INTEROPERABILITY DESIGN

For the design of the interoperable architecture to be implemented in the prototype, first we laid out general high level design factors, with the goal of maintaining the potential for applicability to a *category* of training simulations as opposed to a specific implementation. This led to the next step of detailing the design itself in collaboration with BBN due to the reuse and extension of technology they had recently developed. For this paper, we will focus on the portion of the interoperable design that we developed.

Design Factors

Emerging ADL constructs contemplate a variety of training use cases or configurations beyond browser based learning. However, for our research effort we narrowed our focus to a specific use case category, with the following characteristics:

- **Standalone** - The training system resides on a different machine or entirely different hardware unit from that on which the learner's browser is running, and is being guided through didactic courseware by the RTE.

- **Networked** - Although the training system and the learner's browser are on separate machines, they have network connectivity to each other.

- **Stateless** – The training system will not be expected to support external state tracking for the purposes of bookmarking.

The **Standalone** feature represents one of the key differentiators between the goals of this effort and previous efforts. For example, BBN's existing DTECS implementation effectively handles use cases involving PC-based simulations or trainers that can be installed on the same hardware as the learner's browser. With something like an embedded training system, a potential use case could involve a crew aboard a vehicle instrumented with embedded training capabilities. A period of downtime could be used as a training opportunity, perhaps using a tablet or other lightweight computing platform to log into Learning Management Systems, and proceed with courseware until reaching a point that calls for a training event using the embedded trainer onboard the vehicle. At this point, they take stations within the vehicle, perform an exercise, the assessment results are sent back automatically, and they proceed with the course. Perhaps remediation is given if needed, perhaps a remedial exercise is run to demonstrate mastery in a follow-up run, or perhaps they simply continue with subsequent course topics as the sequencing rules dictate.

The characteristic of network connectivity might normally be taken as a given, especially with most modern systems. However, there are many existing simulation environments that do not easily meet this requirement. Firewall and security issues, legacy matters preventing modifications to existing implementations, and other similar factors may make connectivity difficult or impossible. For our effort we made the simplifying assumption that network connectivity is available. There are other options readily explored (e.g., configuration files and score files hand-delivered on portable storage devices) in such cases.

The **Stateless** feature is less of a requirement for the simulation, and pertains more to the overall training use case. Especially with complex virtual training environments, there are use cases where it is natural to support an ability to bookmark the current state in the middle of an exercise, and return to the exercise later. Ideally such exercises would also be interoperable with SCORM course structures in the future, with the launch capability being able to differentiate between starting a new exercise and resuming an existing one. However, we restricted the scope of our effort to focus on conditions where the expectation is that each exercise is either run to completion or disregarded. The consequence of focusing on a use case involving exercises without bookmarking capabilities is that the interaction between the RTE and the training system is simple in the sense that the primary requirement is to support the steps of initial scenario selection, configuration and launch, and exercise completion.

It was also an intentional design factor to focus the prototype effort on a simulation with existing ITS capabilities. There are also simulation based trainers that are standalone, networked, and stateless, and that have the potential to provide experiential learning that can be incorporated into didactic courseware in a meaningful way, without necessarily having ITS capabilities. However, as Weil et al (2007) observe, while many training simulations provide useful experiences, they frequently do not contain well structured training objectives with assessment routines and output. But for simulations that are integrated with ITS technology or similar structured training methods, typically there are constructs such as a principle hierarchy used to organize the knowledge or skills that will be taught. This parallel with learning objectives and SCOs makes for a clearer path for an interoperable mapping from the training simulation to a SCORM course, and this therefore motivated our focus on ITSs. Also, the instructional benefits of ITSs have been well-documented (Anderson et al, 1985; Jensen et al, 2005; Nichols et al, 1992, Shute & Psotka, 1996)

Architecture

As discussed earlier, the interoperable architecture makes use of a modified version of the Distributed Training Event Coordination Service (DTECS) developed by BBN. For reference, we will first give a brief overview of the existing functionality, then describe the innovations from this research.

Pre-Existing Interoperability Method

In order for training events on a simulation to be compatible with a SCORM course, they must have some form of structured representation of learning objectives, which can be mapped back to those in the course. The existing DTECS implementation provides a method for a simulation based trainer to register itself, using a construct called the Local Training

Package (LTP). Among other things, the LTP identifies the training objectives that can be exercised with the simulation. The Lightweight Scenario Format (LSF) establishes the mapping from SCORM content to learning objectives available from registered simulations via the LTP information they provide. In practice, an LSF serves a function similar to a SCO in a SCORM course, and is referenced similarly by the sequencing rules in the SCORM RTE. In the existing DTECS implementation, the sequence for launching a simulation based training event went as follows (Travers et al, 2007):

1. The operational flow starts with regular learning activities in a SCORM course. At some point the sequencing rules encounter an LSF corresponding to an experiential training event which exercises learning objectives related to the student's progress in the course.

2. The SCORM RTE searches for an available DTECS which can configure and deliver a simulation training event matching the objectives in the LSF. The DTECS searches LTP representations for training events that will satisfy the designated objectives. The training event is prepared by directing the user to a lobby web page. This can occur in parallel with multiple users if a team training event is anticipated.

3. The DTECS sends a Launcher to the user's browser with information for launching the simulation (presumed to be a local executable on the browser machine), and a set of Web Services interface instructions. Upon a start event initiated by the user in the DTECS lobby webpage, the Launcher triggers and the exercise commences.

4. At the conclusion of the exercise, the training system contacts the DTECS using instructions provided in the Launcher, to deliver training results.

5. The DTECS presents scores to the trainee in the webpage interface, allowing the trainee to determine whether to relay them to the RTE or to conduct another exercise.

The primary modification needed from this existing sequence was in steps 3 and 4. Since the simulation is standalone and is not local to the browser machine, a more complicated process is necessary at these steps. Furthermore, modifications have consequences for the other steps, especially because of the team training capabilities for which the DTECS was designed.

A Proxy Simulation Approach

The study of possible solutions started with an initial experiment implementing a "proxy simulation". This involved creating a new compact executable which could be available to run locally on the same machine as the trainee's browser. This primarily had a network communications function, so when it was launched, it would relay a signal over the network to the actual target training system. On the training system side, no pre-existing support for remote launch capabilities can be assumed to exist, so a new component called the Local Training System Controller (LTSC) was defined. The LTSC is responsible for both incoming and outgoing communications relays, receiving the initial configuration information passed from the proxy simulation, and sending scores back.

This resulted in a modified step 3 and 4 in the sequence above, as follows:

3a. The DTECS sends a Launcher to the user's browser with information for launching a proxy simulation, which will communicate with the remote simulation and relay a set of Web Services interface instructions. Upon a start event initiated by the user in the DTECS lobby webpage, the launch sequence triggers, and the exercise commences.

4a. At the conclusion of the exercise, the training system contacts the DTECS using instructions relayed by the Launcher and then the proxy simulation, to deliver training results.

This experiment with the proxy simulation established the first operable prototype. Our next goal was to review how this could be simplified or generalized. Several notes regarding complications or possible improvements were observed:

- The LTSC is implemented with a Web Services interface with which to provide results directly to the DTECS at the conclusion of an exercise. This suggests that the DTECS could directly communicate with the LTSC for the standalone trainer, rather than sending a relayed start event through other components.

- There is some redundancy in having both a Launcher (Java .jar file) and a proxy simulation executable on the browser machine, where one launches the other, which then relays the instruction to the actual simulation. Alternatives are to define a new kind of Launcher which can perform proxy simulation functions, or to eliminate both and move the launch function to the DTECS.

- The existing DTECS architecture was designed to accommodate team training and also perform initial authentication on users participating in simulation events, while also considering firewall and security issues. A re-purposing to perform some of the Launcher function within the DTECS would therefore be non-trivial.

- Because the launch process requires routing through the browser, the browser must stay on through the entire training event. This may be undesirable for some training use cases, where it would be ideal to send score information back to the server, and have this available upon the next login to the course, possibly in a completely different setting.

Generalized Interoperability Approach
Considering these and other factors, the design for the interoperable architecture to be implemented for the full prototype was developed to simplify the Launcher and the proxy simulation components, and in fact allow for direct launch of the simulation via DTECS communication with the LTSC. The resulting architecture is shown in Figure 1.

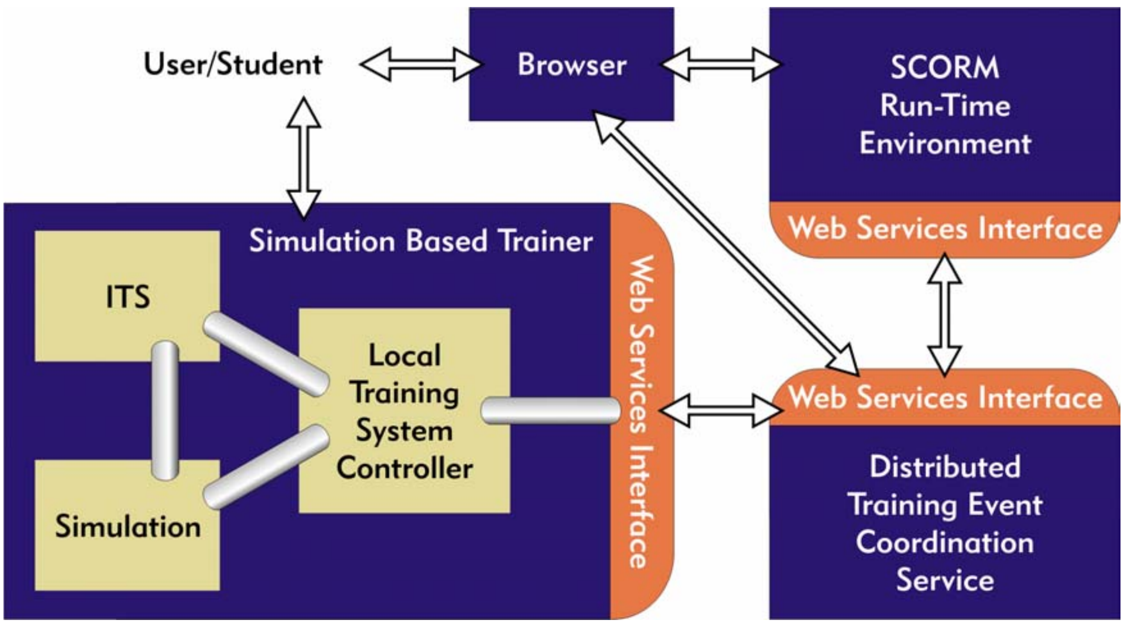

Figure 1. Interoperable Architecture

Under this architecture, much of the same operational sequence still applies, although steps 3 and 4 change once again:

3b. The DTECS contacts the LTSC Web Services Interface directly using instructions available in the LTP. It provides configuration information for the exercise and a set of Web Services interface instructions. Upon a start event initiated by the user in the DTECS lobby webpage, the LTSC triggers the launch process on the standalone trainer and the exercise commences.

4b. At the conclusion of the exercise, the LTSC contacts the DTECS Web Services Interface using instructions it initially received before the exercise, to deliver training results.

To accommodate this scheme, BBN implemented modifications to the DTECS, and our effort focused on building the full functionality of the LTSC and its interaction with the Simulation Based Trainer. The following section gives further details on these elements we developed and integrated in the prototype.

PROTOTYPE DESCRIPTION

In order to demonstrate the interoperable architecture, we integrated the components described above with an existing training testbed, the C2V ITS (Command and Control Vehicle Intelligent Tutoring System). The C2V ITS provides an ability to exercise unmanned vehicle control concepts and skills in a variety of scenarios, and in a physical configuration mirroring the embedded training setting onboard an actual C2V. It is a facility-based trainer, and for this effort we chose it to be representative of a category of training systems including "caves," or other immersive trainers, in terms of the applicability of the resulting design.

The C2V ITS consists of the C2V crewstation hardware itself, which includes a six screen Soldier interface with "on-board" views for the C2V vehicle, as well as screens for controlling unmanned vehicles and viewing sensor imagery. The testbed software includes the simulation, image generators, and control tools to drive the exercise. The ITS is integrated with this testbed, monitoring Soldier actions and providing real-time feedback on errors or deficiencies, while also logging this information for after action review (AAR). In a previous effort, (Jensen et al, 2005), the integrated ITS was used in experiments conducted with human test subjects, where it was found to deliver effective learning comparable to human-led AAR. Figure 2 below shows a feedback message being delivered on the unmanned vehicle imagery screen during an exercise. This kind of feedback event is logged, and ultimately contributes to the assessment scores that can be mapped back to learning objectives in an external course.

Figure 2. C2V ITS Training Event

Referencing the interoperable architecture diagram shown in Figure 1 earlier, the entire C2V ITS is considered the Simulation Based Trainer in the context of this effort. The standalone configuration contains all hardware and software for the Simulation (specifically the C2V crewstation) as well as the integrated ITS. Additionally, the Local Training System Controller was implemented and integrated with this component of the architecture. Because the LTSC by nature resides with the training simulation, we intentionally designed it to keep a low computing profile, so as not to compete with simulation critical processes.

Local Training System Controller (LTSC)

The LTSC performs numerous functions, both in general design, and in the implementation developed for this application.

Simulation Control

The LTSC provides a simple two-button user interface for starting and concluding exercises. The C2V ITS, like many simulations, supports free play execution in scenarios. So there are no native explicit events representing the completion of an exercise, either by time factors or by objectives accomplished in the virtual environment. Assessment occurs in real-time during the exercise and is tabulated continuously until the exercise is concluded via the LTSC exit button.

The user is also required to provide authentication before the start of an exercise, partly for normal security reasons and partly to ensure that the right exercise is being prepared and delivered, in accordance with the originating request from the DTECS. Especially given the standalone nature of the Simulation Based Trainer, in theory there could be more than one trainee with exercises queued up.

All other processes, such as starting simulation processes, loading scenario data, and tabulating and forwarding exercise results all happen automatically without being apparent to the trainee.

Training Session Store

The Training Session Store provides an interface to the user's training session information, which contains information needed to identify the user and to connect back to the DTECS. We initially implemented a simple approach using an XML file with this information, and then in the full implementation replaced this with a more flexible Java Database Connectivity (JDBC) store, where we can save user data into a compact MySQL relational database. One potential future benefit of the relational database is to allow flexibility for the Simulation and ITS to have access to various kinds of information about the trainees' progress, for example how many training sessions they conducted and so forth.

Objective Mapping

One of the most important functions in the LTSC is performed by the Objective Mapping module. This loads the mapping that establishes the relationship between ITS evaluation principles and learning objectives as identified for external purposes by the LTP. This can be a one to many mapping, and also includes weight information for how collections of principles are weighted with respect to each other for contributions to the ultimate score to be returned for a learning objective.

Suppose for example, that a SCORM course learning objective, “Controlling UAVs” involves 3 basic concepts for unmanned aerial vehicle (UAV) control using the C2V crewstation. These concepts may include knowing how to take control of the UAV, knowing to issue a HOVER command before a FLY command, and knowing how to assign a flight route. In the ITS, there are individual evaluations associated with each of these principles, to detect conditions where the operator makes an error. The Objective Mapping module contains the representation where relative weights for these three evaluations are referenced. A principle can theoretically be passed or failed zero or more times in an exercise, so there is an initial calculation to assess the distribution of results for a single ITS principle over the course of an exercise, and then this result is tabulated and weighted with other relevant principles to yield the output value for the learning objective.

Client Services

The LTSC also communicates directly with the DTECS using the Web Services Interface. Before an exercise, the DTECS contacts the LTSC web services to provide configuration information and the instructions for contacting the DTECS at the conclusion of the exercise. At that time, the LTSC contacts the DTECS web services to deliver scores. In the LTSC Web Services Interface, we conformed to BBN’s existing implementation of the DTECS web services, following W3C web services standards.

Integrated Prototype

A fully operational prototype was constructed and integrated with the existing C2V crewstation testbed at STTC in Orlando, FL. For the purposes of this effort, we constructed a simple SCORM course with two blocks of instruction for beginner and advanced students. The beginner course block involves basic operational principles in using the C2V crewstation such as how to conduct movements and send reports, while the advanced course involves some tactical principles such as cooperative engagement. There are approximately 38 evaluation principles in the C2V ITS, which were mapped to 12 learning objectives in the SCORM course. Figure 3 shows a lesson for UAV control from the beginner portion of the course.

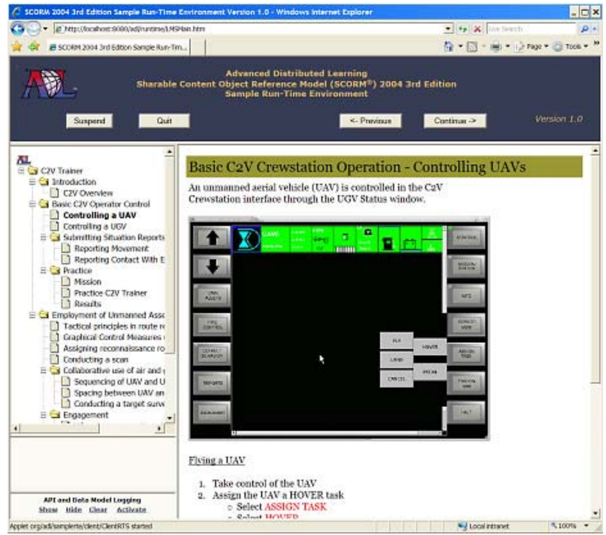

Figure 3. Browser Based Lesson

The course allows a student to go through the notional instructional sequence that this use case attempts to exemplify, for example:

1. A beginner learner goes through the Basic C2V Crewstation Operations lessons in the browser based course.

2. The course sequencing leads to a training event to be conducted on the C2V ITS, which will serve as the assessment of the trainee’s mastery of the beginner portion learning objectives. The trainee initiates the exercise from the browser, with the steps involving the communications with the DTECS and the start trigger from the DTECS webpage.

3. The trainee walks over to the C2V crewstation, where the LTSC user interface is ready for launch and authentication. Doing so, the trainee conducts an exercise, concludes it by triggering an exit in the LTSC, and returns to the browser course while scores are automatically sent to the DTECS.

4. Back on the browser machine, the trainee can review scores in the web interface to the DTECS, and accept them or reject the exercise. If these results are accepted and sent to the SCORM RTE, then the course sequencing rules determine the next step. Next steps could be either remedial lessons from the beginner course block, essentially a return to step 1. or an acknowledgement of

mastery, and instruction to proceed to the advanced course block.

CONCLUSIONS AND THE WAY FORWARD

The implementation of the prototype provided an opportunity for lessons learned not only on the technical challenges involved, but also for directions for future research.

Areas of Potential Further Work

Areas we noted for potential future research included both those directly related to the LTSC and other components of this architecture, as well as broader topics for consideration in future standards.

Generalize on the Objective Mapping Method

Within the LTSC component, the Objective Mapping module currently contains a simple representation for weighting ITS principles and mapping them to learning objectives structured for consistency with a SCORM course. This specifically could be made a general component that would be applicable for most ITSs using an explicit principle hierarchy. It is a challenging process for the instructional designer to determine a proper mapping, especially when one-to-many or potentially many-to-many relationships can apply. A set of guidelines would likely be valuable to help make instructional designers aware of what tweaks to the mappings and weights can mean. Similarly, guidelines may be helpful for determining how to handle conditions where evaluations apply multiple times in a scenario

Another area to explore may be to substitute an approach for objective mapping that doesn't necessarily use hard coded values for weights and relationships, but potentially dynamic formulae that can take other contextual information, including information from the learner profile, information about the relationship to larger training objectives, and so on.

Local Training Package Modularity

The current implementation of the Local Training Package handles a straightforward mapping from LSF learning objectives to training system configurations. With complex training systems, and especially those involving ITS functionality on interrelated collections of principles or learning objectives, the mapping may be non-trivial. For example, consider a training system for which there are a variety of scenarios offering practice and evaluation of performance on different combinations of 10 discrete principles (or learning objectives). Suppose that the sequencing in a course would benefit from being able to activate training events that involve a specific subset – 4 of the 10 principles. In the training system, these 4 principles may be suitably exercised in different configurations such as:

- Scenario A exercises 9 principles. Evaluations for 5 of these can be turned off, so that only performance with respect to the target principles is evaluated.

- Scenario B exercises 5 principles, including the same 4 and one principle not in Scenario A. Once again, the extra principle evaluation can be turned off.

- Scenario C exercises 7 principles. Although Scenario C evaluations can't be disabled, the scenario can be tailored such that hinting and also the behaviors and events within the scenario reduce the impact of the "extra" principles involved in the scenario, in terms of distracting from the target principles.

- Scenario D exercises the specific 4 principles, but with built-in hinting that is written at a level which may be most effective only for learners who have been identified as beginners with the related concepts.

Of course other variations are also possible. The logic for preferring one of the options above over the others varies with a number of factors including learner-specific information. A potential area of further investigation involves how to generalize the LTP method of registering a training simulation and representing the space of possible exercises and how they may accomplish different objectives.

Allowing the Browser Session to be Released

In the current architecture the browser session must be active during the whole training event, and the scoring results are only sent to the SCORM RTE after the user reviews them in the browser environment. There may be use cases where it would be desirable to allow the browser session to be released instead, and have the scores temporarily stored somewhere until the user's next login to the SCORM RTE and course.

Bookmarking Training Events

Similar to the previous note, the current architecture does not support conditions where the trainee may want to bookmark a training event (assuming that the simulation supports it) and return later, or perhaps another day. This capability, at least on the side of the interoperable architecture, would add flexibility in accommodating different training needs. In our research effort we noted that a portion of this functionality could be accomplished within the Training Session Store component of the LTSC, which notionally tracks current and previous session data for an individual trainee.

Related Recommendations for SCORM

Currently SCORM 2004 does not allow for significant persistence of past performance data. Data gathered and stored for a SCO, in locations such as *cmi.interactions*, is neither accessible nor guaranteed to persist beyond the initial run of the SCO. One of the items on the table for SCORM 2.0 is a more robust data model that will allow more complex results to be captured, persisted, and retrieved by future activities. Access to this data will allow for new styles of adaptive learning and allow for improved and possibly integrated remediation going forward.

Conclusions

This prototype effort ultimately aims to present a mechanism to bring the world of Distributed Learning closer to Intelligent Tutoring and Embedded Training. In parallel with studies of the technical means to do so, there have also been related efforts to analyze the circumstances where it makes sense to take the opportunity to intermingle didactic and experiential training in this way (Weil et al, 2007). Further effort in this area will also be helpful, to identify the boundaries and guidelines.

For example, if an experiential training event takes a significant amount of time, does it really make sense to use that as a mechanism for assessing mastery of a set of discrete learning objectives of the nature of those in a SCORM course? The standard method of remediation and repeat assessment may be prohibitive in such a case. Perhaps there are other alternatives that can be explored. For example, if a training system provides native automated AAR functions, which may theoretically accomplish some of the remediation functions, this might be a reasonable component of an instructional method that integrates this system with didactic training.

In the meantime, the prototype effort described here is expected to have wide applicability for a broad range of training systems. There are increasing efforts to provide Embedded Training, which often go hand in hand with Intelligent Tutoring due to the nature of the training environment. The capability to link the use of these systems with courseware and persistent learner information in Distributed Learning architectures is one in which training programs have expressed specific interest.

ACKNOWLEDGEMENTS

The research effort described in this paper was sponsored by the Joint Advanced Distributed Learning Co-Laboratory. We are also grateful to BBN Technologies for their enthusiastic collaboration on this effort.

REFERENCES

Anderson, J. R., Boyle, C., and Reiser, B, (1985), Intelligent Tutoring Systems, Science, 228, 456-462.

Anthony, M. K., and Ashworth, A. R. S. (2006). "Mapping Intelligent Tutoring System Constructs to SCORM 2004 Data Structures", *Proceedings of the Industry / Interservice, Training, Simulation & Education Conference (I/ITSEC 2006)*.

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, vol. 2006*. (See also: http://www.jointadlcolab.org/downloads/research/2005/perf_based_adv/Boeing_FinalReport_A005_rev1. pdf)

Jensen, R., Sanders, M., Marshall, H., & Tasoluk, C. (2005). "FCS Intelligent Structured Training - Experimental Results and Future Applications", *Proceedings of the Industry / Interservice, Training, Simulation & Education Conference (I/ITSEC 2005)*.

Nichols, P., Pokorny, R., Jones, G., Gott, S. P., and Alley, W. E., (1992), Evaluation of an avionics troubleshooting tutoring system, Technical Report, Human Resources Directorate, Brooks AFB, TX.

Ramachandran, S., Remolina, E., & Barksdale C. (2006). "Scenario-based Multi-Level Learning for Counterterrorism Intelligence Analysis", *Proceedings of the Industry / Interservice, Training, Simulation & Education Conference (I/ITSEC 2006)*.

Travers, V., Roberts, B., Guin, C., Tomlinson, R., Marks, J. (2007). "A Web Service Architecture for

Integrating Didactic and Experiential Learning", *Proceedings of the Industry / Interservice, Training, Simulation & Education Conference (I/ITSEC 2007)*.

Shute, V.J. & Psotka, J. (1996). Intelligent tutoring systems: Past, present, and future. In D. Jonassen (Ed.), *Handbook of research on educational communication and technology*. New York: Simon & Schuster Macmillan.

Weil, S., Mizrahi, G., Marvin, D., & Travers, V. (2007). "Integrating Didactic and Experiential Training: Round Pegs in Square Holes?", *Proceedings of the Industry / Interservice, Training, Simulation & Education Conference (I/ITSEC 2007)*.