Game-based Learning Assessment

Curtis Conkey[1], Chris DuBuc[2], Kelsey Henderson[1], & Katrina Ricci[1]
[1]Naval Air Warfare Center Training Systems Division
Orlando, FL
[2]Engineering & Computer Simulations, Inc.
Orlando FL
curtis.conkey@navy.mil, chrisd@ecsorl.com, kelsey.henderson@navy.mil,
katrina.ricci@navy.mil

ABSTRACT

Games and simulations have great potential to support adaptive learning by placing learners in a "real-world" environment and allowing them to learn in context. While these types of simulations offer the opportunity to immerse a student within a context based scenario, student tracking, assessment, and feedback play a key role in the learning process and in learning management. From a learning perspective, feedback provides critical information to a student for insight regarding knowledge deficiencies and strengths. From a learning management perspective, knowledge of a learner's progress can provide opportunities for more precise remediation or advancement to a more challenging level of training. However, historically most game-based training applications have not had the capability to interface with SCORM-conformant Learning Management Systems (LMSs) to track and record trainee progress. Moreover, there is a lack of ability to assess performance and provide learners immediate feedback in a game-based environment. The goal of the effort presented here is to provide a tool capability for interfacing a simulation's characteristics with an LMS for control and tracking. A SCORM conformant game-based assessment capability will be presented. Potential strategies to enhance the state-of-the-art of scenario-based, self-paced learning will be explored and recommendations for future study will be discussed.

ABOUT THE AUTHORS

Mr. Curtis Conkey is the Principle Investigator and Team Lead for the Learning Technologies Lab for the Naval Air Warfare Center Training Systems Division (NAWCTSD). The lab's primary charter is to experiment with the appropriate application of emerging technologies to US Navy training curriculum. Focus areas have included Open Source Training Solutions, low-cost training based on commercial gaming technologies, Learning Management Systems and Integrated Learning Environments for game based training environments. He has prior experience in conventional simulation systems having worked with Onboard Team Training applications for the Next Generation Virginia class submarine. Prior to his Navy career, Mr. Conkey completed 20 years with AT&T / Lucent Bell Laboratories where he managed the creation of next generation networking solutions for high profile customers. Mr. Conkey has a Bachelors Degree in Electronics Engineering, a Masters Degree in Computer Science (emphasis Computer Graphics) and is currently a Doctoral Student in Modeling and Simulation at the University of Central Florida.

Mr. Chris DuBuc is a Senior Software Engineer and Project Manager for Advanced Learning Technologies with Engineering & Computer Simulations. He is currently directing work efforts that involve using the Delta3D game engine within the Navy's Integrated Learning Environment. He has extensive knowledge of the evolving standards and operational requirements involved with using games and simulations within a managed learning environment. Prior to joining ECS in 2005, Mr. DuBuc ran his own consulting company, Buke Inc., where he created custom, web-based software solutions for both PC environments and Pocket PC environments specifically focusing on web-services and database connectivity. Mr. DuBuc was also V.P of Florida operations for Sage Software, where he oversaw development of a wide range of web-based software solutions for the telecommunications and mining/exploration industries. Mr. DuBuc has a BS in Management systems with a minor in Industrial Psychology from Rensselaer Polytechnic Institute (RPI).

Dr. Kelsey L. Henderson is the Lead Instructional System Designer for Cross Warfare Programs at NAWCTSD. He has twelve years of training technology experience, having held positions as an instructional technologist, project manager, instructor, and consultant. He holds a PhD in Education, specializing in Instructional Technology, from the University of Central Florida. His research interests include learning objects, performance support tools, and assessment strategies.

Dr. Katrina Ricci is head of the Training and Human Performance Research and Development Branch at NAWCTSD. Dr. Ricci has over 20 years of experience in basic and applied research focusing on learning and retention, maintenance training and aiding, and user interface design.

Game-based Learning Assessment

Curtis Conkey[1], Chris Dubuc[2], Kelsey Henderson[1], & Katrina Ricci[1]
[1]Naval Air Warfare Center Training Systems Division
Orlando, FL
[2]Engineering & Computer Simulations, Inc.
Orlando FL
curtis.conkey@navy.mil, chrisd@ecsorl.com, kelsey.henderson@navy.mil,
katrina.ricci@navy.mil

INTRODUCTION

The ever increasing availability and affordability of commercial off-the-shelf gaming environments continues to enhance their desirability as training solutions. These 3D Interactive (3DI) gaming and simulation environments have great potential to support adaptive learning by placing the learner in a "real-world" environment and allowing them to learn in context. The added capability for launching scenario-based environments from a learning management system (LMS) furthers that appeal (Conkey, Smith, DuBuc, & Smith, 2006). However, student tracking, assessment, and feedback play a key role in the learning process and in learning management and must be addressed in any complete e-learning solution. From a learning perspective, feedback provides critical information to a student that is valuable for insight into knowledge deficiencies and strengths. As Garris, Ahlers, and Driskell (2002) note, feedback is the link between what is represented in the simulation and the real world and, more importantly, the feedback process mediates the instructional component in order to transform the gaming environment into a learning environment.

From a learning management perspective, knowledge of a learner's progress can provide information for more precise remediation or more timely advancement to a more challenging level of training. However, historically most game-based training applications have not had the capability to interface with SCORM-conformant Learning Management Systems (LMSs) in order to track and record trainee progress. Moreover, there is a lack of ability to assess performance and provide learners immediate feedback in a game-based environment. Additionally, the modification of a scenario environment, as well as other very critical aspects of a training event such as:. feedback, debrief, and remediation is dependent upon a computer programmer, not an instructional systems specialist. This severely hampers flexibility in defining performance parameters, specifying targeted skills and knowledge, and providing trainees' knowledge of results.

The purpose of the effort presented here is to provide an Instructional Systems Designer (ISD) with capabilities for interfacing a simulation's parameters with an LMS for purposes of control and tracking of performance data. A government owned SCORM conformant game-based assessment capability will be presented. Potential strategies for enhancing the state-of-the-art of scenario-based, self-paced learning will be explored and recommendations for future study will be discussed.

THE ENVIRONMENT

Previous work demonstrated the ability to provide interface mechanisms between 3D Interactive (3DI or often simply referred to as games and simulations) training materials and an LMS (Conkey et. al., 2006). Specifically, SCORM 2004 requirements were met and 3DI training successfully communicated with an LMS in real time. The mechanisms needed to track events internal to the 3DI trainer and to match those events with externally defined training objectives were also developed. Finally, the "plumbing" for distributing and launching a simulation from within a SCORM environment and communicating performance metrics from the simulation to the LMS was developed. This work constituted the Phase I effort referred to as Learner Assessment Data Models and Authoring Tools (LADMAT).

Once it was shown that the 3DI training could communicate with LMSs, the follow-on effort focused on developing tool sets to enable integrating both existing and new 3DI simulations with an LMS. In Phase II, the development effort shifted to creating tools to enable non-programmers to use this technology. Thus, the focus moved from software developer tools to tools for the instructional designers needing to integrate 3DI simulations into their curriculum.

Ultimately, the target environment for this technology is the desk of the instructional designer, because it is the designer who has the expertise in creating practical, relevant, and effective training curricula. However, an instructional designer is not necessarily an expert in writing code and/or manipulating simulation files. Therefore, the technology needs to allow the designer to focus on developing content and learning strategies by abstracting away as much of the technical details as practically possible. Thus, command-line tools were replaced with a Graphical User Interface(GUI)-based application. Instead of manually creating and editing XML files, a “design surface” is provided where a designer can create models of assessment, feedback, and remediation via a more familiar “drag-and-drop” interface. This approach attempts to diminish the need for custom scripting/coding via the creation of reusable simulation and web-based components.

In addition to making it technically easier to integrate simulations into SCORM-based environments, the Phase II development effort also addressed the practical implications of such integration including the generally more complicated assessment, feedback, and remediation requirements inherent in dynamic simulations. In previous efforts, the assessment logic addresses simple procedural, task-based models (that matched the static structure of traditional e-learning content). However in Phase II, a new “rule-based” mechanism was implemented to more closely fulfill the assessment needs in a dynamic environment and a new set of feedback, review, and remediation capabilities were implemented to compliment this mechanism. The intended purpose of all this work is to unlock the full potential of simulations as a training tool, and to enable instructional designers to do what they do best – create compelling and effective training experiences. The Phase II effort was referred to as Simulations for Integrated Learning Environments (SIMILE).

REQUIREMENTS

The move to a more user friendly, designer-focused environment for the SIMILE project imposed new requirements upon the Phase II development effort. Several major areas needed development: (i) simulation launch mechanisms, (ii) integration of a simulation into a general training curriculum, (iii) data collection and management schemes, (iv) support for multiple execution and presentation environments, (v) generic assessment systems applicable across a variety of simulations and (vi) user friendly tools for SCORM packaging of the interactive content.

First, the simulation launching mechanism created in the LADMAT project needed modification. While not strictly related to the work of an instructional designer, the implementation of the launching mechanism needed to be overhauled so that it could more seamlessly interface with a typical designer toolset. For example, in order to support custom launch scenarios, the launching mechanism needs to pass designer specified launch parameters to the simulation upon startup, to allow designers to uniquely customize a simulation to a given lesson or remediation strategy. Additionally, reusable web-based components (Java applets, JavaScript, PHP script, etc.) were needed to support the launch process and communication between the simulation and the LMS server.

Second, the more tightly integrated simulation within the context of larger training curricula provides additional opportunities for new and unique learning strategies and more in-depth performance measurement. Therefore, upgrading the assessment modeling capability from that of a “task-based” system to that of “rule-based” system was a major requirement. In practical terms, this required that the assessment system be able to track not only “what happened” (e.g. trainee launched grenade), but also “the state of the environment when it happened” (e.g.. trainee launched grenade in an environment containing 2 enemies and 4 civilians within 100 meters of each other). The ability to define success and measure performance with this amount of granularity is a training necessity and a requirement for fully utilizing the training potential of simulations.

Third, a corollary to enhancing the ability to track a student’s performance within a simulation is how best to use the copious data generated during this process to greatest effect. Simulations and related assessment systems can collect much more information than a SCORM-based LMS can store or process. Therefore, SIMILE included collecting data requirements for providing detailed feedback to students, and after-action-review (AAR), and for enabling automated remediation within the SCORM course (This included the ability to re-launch the simulation with a specifically chosen scenario).

Fourth, support for multiple programming and execution environments must be provided. The primarily code-based developments of 3DI projects differ significantly from the web and scripting world of e-learning environments. A means to bridge these two development worlds is needed. For instance, simulation development teams typically work in C++ to create the simulation components, in C# to create the GUI applications, in Java to create the launching and communication mechanisms, and in various web/scripting languages (HTML, ECMAScript, PHP, etc.) to create the run-time and SCORM sub-systems. Thus the problems presented by integrating simulations into SCORM-based learning curricula require a multi-environment approach. The inherent complexity of this work is probably the main reason why the use of simulations is not currently more widespread within existing SCORM courseware. The attempt at tackling this challenge and building tools to bridge this gap may arguably represent the biggest value provided by this project.

The assessment, feedback, and remediation systems have to be practical and generic in order to work with future, unknown simulations. There is an inherent conflict between developing a technology that is powerful enough to solve a particular problem, while remaining generic enough to solve many other problems as well. For example, one of the use cases used for designing Phase II features was combat medic training, specifically, the TC3 Combat Medic simulation. Even within this narrowly constrained simulation there were multiple assessment scenarios: single casualty scenarios, multiple casualty scenarios (triage), and various combat situations. Therefore, an assessment system needed to track the status of care given to a particular patient, the priority given to various patients, and the state of the treatment environment (e.g. hostile, friendly). Successful performance for each of these tasks (e.g., tracking correct procedures, the efficient use of time, or correct decision making) is calculated differently. Building a system that can address each of these problems effectively, while being extensible to a variety of complex simulations is a challenge.

Finally and perhaps most importantly, was a requirement for a GUI to allow a non-programmer to package the web content, simulation, assessment model(s), student feedback, and remediation strategies into a SCORM content package suitable for upload into an LMS. This application would provide a single place where an instructional designer could:

- Define the simulation structure (i.e. Event/Data schema)
- Define launch scenarios for the simulation, based upon sets of valid launch parameters
- Identify the code and data files that make up the simulation
- Create one or more assessment models based upon the simulation's Event/Data structure
- Enter unique feedback text related to rules and tasks
- Define different remediation strategies for various outcomes (as defined by the assessment model)
- Identify the structure of the SCORM organization tree by defining SCO's (Shareable Content Objects) and identifying the content files that the SCO's contain, and
- Gather all of the above and build into a valid SCORM content package.

In summary, the goal was to provide a tool that provides an ISD with a user friendly interface for integrating a 3DI simulation or game into a training curriculum. This tool had to integrate the complexities of simulation development with effective e-learning principles such as feedback, assessment and remediation. Equally important it had to allow an ISD to tag specific training objectives in the 3DI simulation for tracking and manipulation by an LMS.

SOLUTION APPROACHES

In order to satisfy the requirements and to best meet the challenges presented, the project team divided the tasks into two sub-systems – the Design-Time system and the Run-Time system. The Design-Time system consisted of creating/upgrading the components needed by ISD's for designing the simulation-based SCORM course, such as the GUI application and the underlying tools and data models. The Run-Time system consisted of creating/upgrading of the components used for administering the course to the student. For the Design-Time system, the project developers created the SIMILE Workbench application, which is an MS Windows-based GUI application written in C#. This application is the basis for all configurations of simulation-based SCORM content packages. It is designed mainly for use by ISD's, with possible input from simulation developers as well. Whereas in previous phases of the project all configuration was done by manually creating and editing XML files and feeding them into command-line tools, this new application allows a user to organize and manage these data files and tools in one place. The application is Windows-based, so the user no longer needs to type in correctly formatted XML. However, the data is still saved in XML, so external automated processes or more technical users can still manipulate the data at this level, when needed.

SIMILE Workbench Application

The SIMILE Workbench application contains three main sections – Simulation, Assessment, and Deployment. Within the Simulation section, the designer (or simulation developer) defines the structure of the simulation, as seen by the Assessment system. Thus the developer defines the simulation's data objects (see Figure 1) to be tracked, the simulation events that they generate (i.e. Event/Data schema), and the valid simulation launch parameters/scenarios.

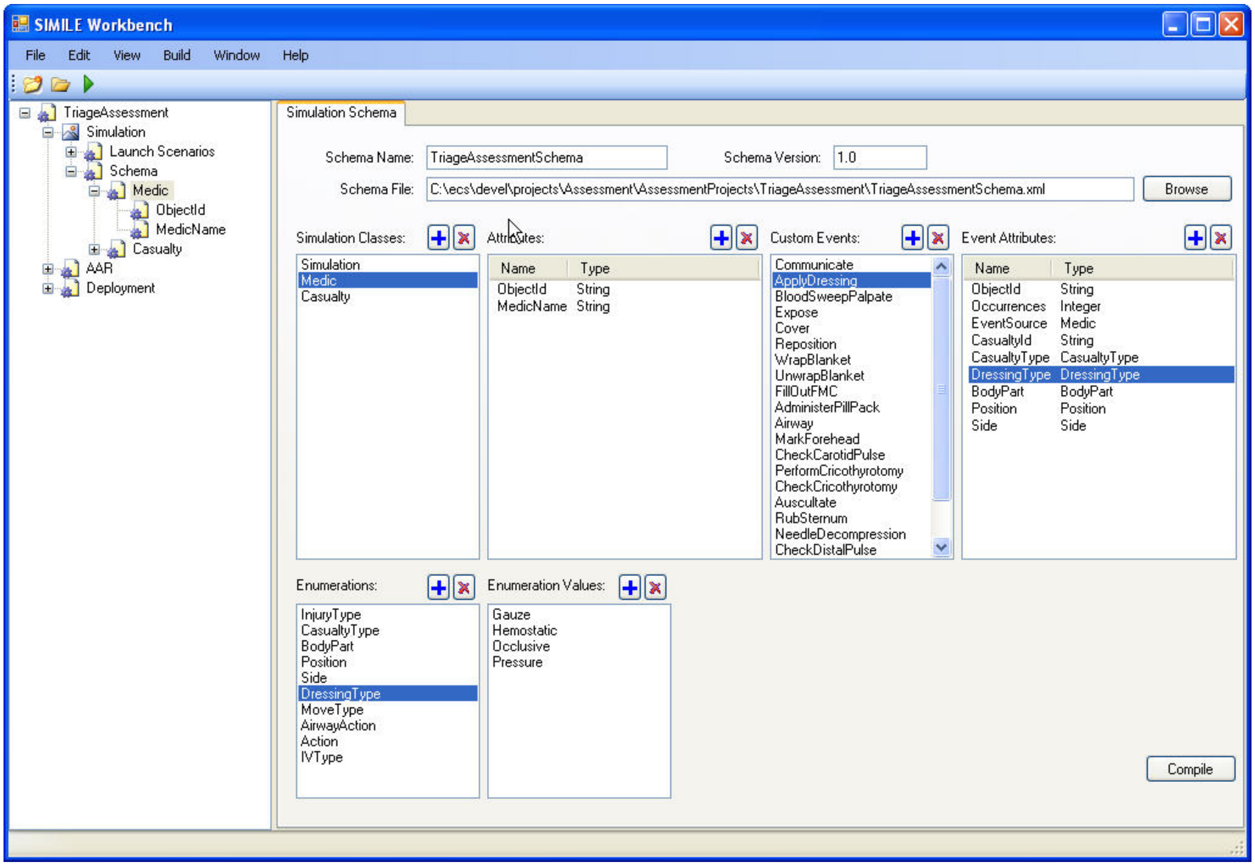

Figure 1 - Simulation Schema Creation

In the Assessment section, the designer graphically creates the conditions, rules, and tasks that make up an assessment model for the simulation (see Figure 2). Conditions roll up into rules, rules roll up into tasks, and tasks roll up into parent tasks, and so on in defining the model. Rules and tasks can be combined into sets, utilizing 'Union' and 'Intersection' (AND/OR) logic to model complicated, branching assessment scenarios. Tasks can be required to be performed in a specific order, set to be activated/deactivated as specific times, and scored numerically or 'pass/fail'.

Each assessment model can also contain a Remediation section which defines a number of remediation strategies. A remediation strategy identifies a specific possible simulation outcome based upon which rules and tasks were satisfied or not. These remediation strategies point to specific SCO's, allowing the LMS to automatically direct a student to a new lesson, a previous lesson, or to force the student to run the simulation again (perhaps with a modified scenario), based upon that student's performance.

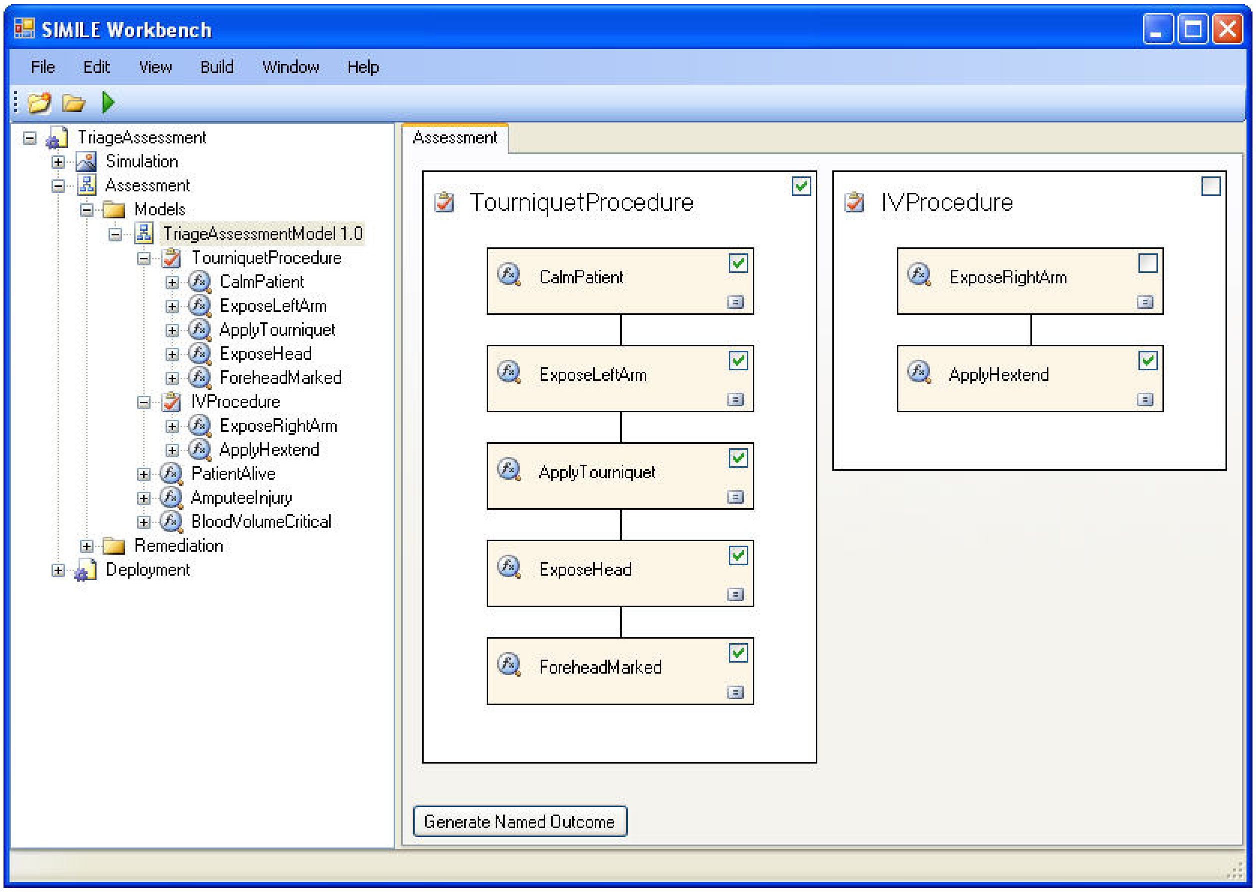

Figure 2 - Assessment Model

These assessment and remediation strategies can be incorporated into trainee feedback schemes that inform and direct the trainee (see Figure 3).
In the Deployment section of the SIMILE Workbench Application, the designer defines and organizes the various lessons (SCO's) that make up the SCORM content. The designer chooses from standard static content SCOs, simulation launch SCOs, feedback SCOs, and After-Action-Review SCOs. For each SCO the designer can specify the files (web pages, images, etc.) that are required.

After completing scenario configuration, the designer presses the "Build" button in order to start the SIMILE Workbench application's SCORM content package creation process. This process compiles the assessment model code libraries (both C++ and Java versions), creates all needed configuration files (XML, etc.), copies and compresses all simulation, assessment, web component, and other files into a single *.ZIP file, and creates a basic manifest (imsmanifest.xml) file to be included in the package for SCORM compatibility.

Run-Time Components

Once the content package is created by the SIMILE Workbench application, it is ready to be uploaded into an LMS for use by students as a part of a running e-learning system. In order to facilitate standardization, and to make it easier for simulation and web developers to manage the various run-time systems, a number of simulation and web components were developed from scratch and/or enhanced from previous iterations.

The two main Run-Time components that manage the bridge from simulation to LMS are the SIMILELaunch application (Java) and the SIMILEMessageApplet (Java Applet). The SIMILELaunch application uses Java Web Start technology to control installing and launching a simulation from a web page. SIMILELaunch has been substantially upgraded during this phase of development, and contains some significant architectural changes. The most important of these is that the SimileLaunch application now launches

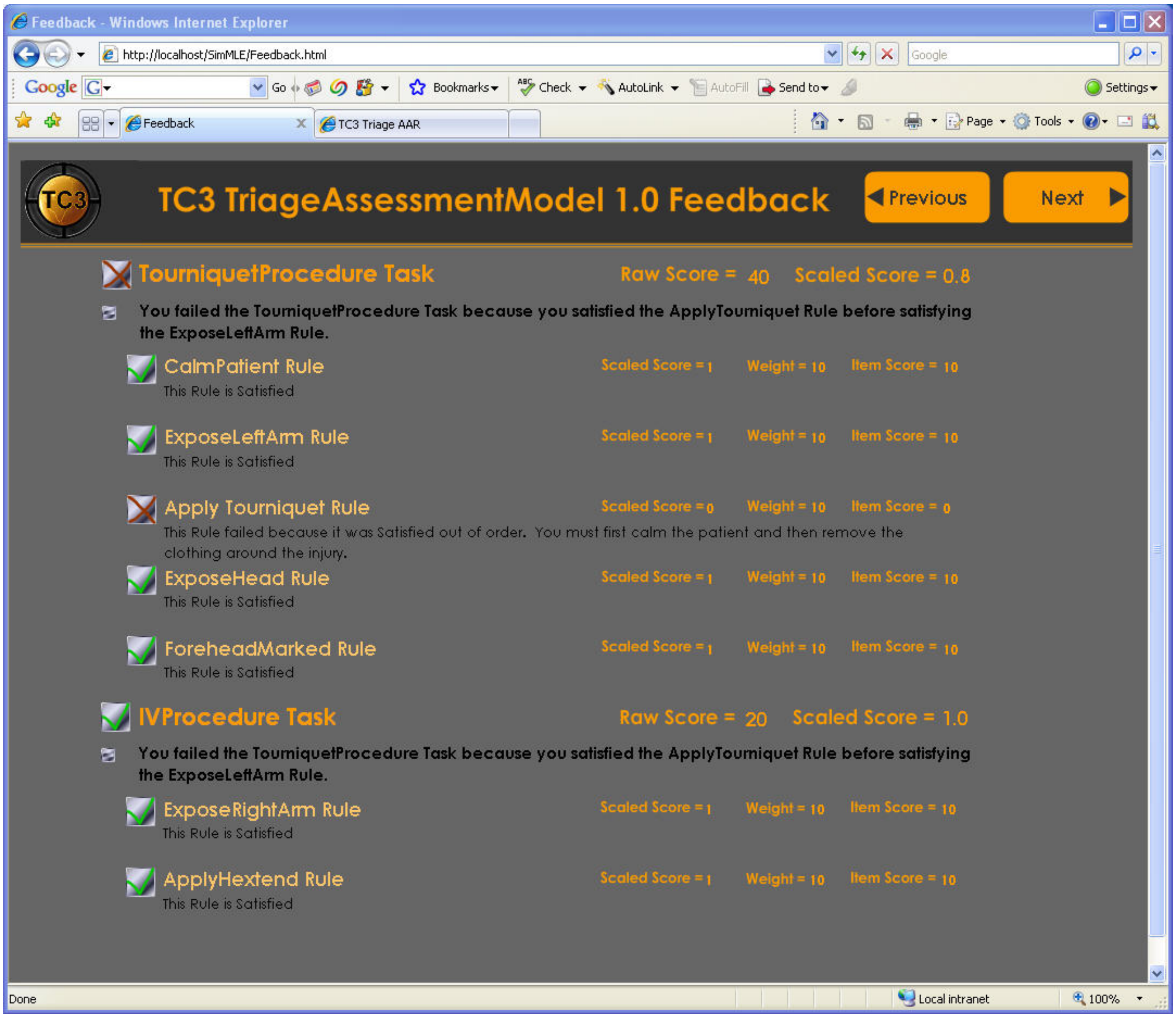

Figure 3 Assessment Model Feedback

standard executable files (*.exe), instead of wrapping the simulation as a compiled library (*.dll). This is significant because a simulation no longer needs to be compiled differently from its stand-alone version, and can be launched directly from the local hard drive rather than requiring launching from the web page.

The SIMILEMessageApplet is a Java applet that is run within the context of the simulation launch web page. Its job is to listen over a TCP/IP socket to messages coming from a running simulation, and to translate and forward them on to the web page scripts. This applet has been upgraded from previous versions in order to stay current with the latest data format and protocol changes; however its basic purpose and functionality has not changed from previous iterations.

In addition to the SIMILELaunch and SIMILEMessageApplet components, the project team has also created a number of other reusable web components that are designed to make setting up a SIMILE-based training course easier. These components include HTML forms for launching a simulation, ECMAScript (JavaScript) libraries that process and translate SIMILE messages, PHP scripts for server-side processing of data, and Flash-based components for displaying feedback and AAR data. The motivation for all of these components is to provide working examples for ISDs and web designers to use in creating their own SIMILE-based training courses. These components can be used as-is, or can be broken out, examined, and modified as needed.

CONCLUSIONS AND RECOMMENDATIONS

The US Navy's ILE funded a project to produce a tool kit suitable for use by ISD teams who are integrating 3DI training into curriculum. The technologies represented in SIMILE represent an effort to bridge the assessment and feedback gap in e-learning environments that attempt to use 3DI modules. This effort is consistent with SCORM working group efforts at evaluating the ability to interface simulations with learning content in preparation for SCORM2.0 (Darque, Morse, Smith, & Frank, 2006).

This project represent a potential paradigm shift for ISDs in the design and creation of e-learning curriculum. E-learning content can move from Interactive Multimedia Instruction (IMI) level 1-2 materials to IMI level 3-4 material (Department of Defense (DoD), 2001). Instead of passive presentations, trainees can have complex interactions with curriculum where branching paths, complex procedures, and psychomotor performance are the norm and not the exception. Assessments can be active instead of passive multiple choice questions where trainees actively demonstrate obtained cognitive skills and knowledge in virtual settings with the ability to repeat and practice skills. Integrated gaming technologies can bring engagement, competition and fantasy into LMS delivered content.

The capability available within SIMILE pushes the bounds of traditional courseware available within an LMS environment. Current LMS-based training consists predominantly of material geared toward the development of declarative knowledge. Glaser, Lesgold, and Gott (1991) describe learning as the acquisition of structures and the integration of conceptual and procedural knowledge. Clearly, the ability to provide more skills-based training enhances the opportunities available to trainees to build a declarative knowledge base, a procedural skill set, and the structuring of that knowledge within the context of a more realistic work environment provided by the 3DI tools. The integration of interactive capabilities into SCORM compatible environments increases availability and access to multimedia rich training, enhancing acquisition and retention of critical knowledge. This truly bridges the gaps and allows our warfighters the ability to *Learn, Train, Win!*

ACKNOWLEDGEMENTS

The authors would like to thank the US Navy's SEAWARRIOR program for funding this development effort, and for the vision of increased training quality through interactive multimedia and active assessment. They would also like to acknowledge the collaborative effort with the Orlando Advanced Distance Learning CoLab that co-funded this effort.

REFERENCES

Conkey, C., Smith, B., DuBuc, C., & Smith, P. A. (2006). *Integrating Simulations into Sharable Content Object Reference Model Learning Environments*. Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

Darque, B., Morse, K., Smith, B., & Frank, G. (2006). *Interfacing Simulations with Training Content*. Paper presented at the NATO Modeling and Simulation Group Conference: Transforming Training and Experimentation through Modeling and. Simulation.

Department of Defense (DoD). (2001). *MIL-HDBK-29612-3A - DEVELOPMENT OF INTERACTIVE MULTIMEDIA INSTRUCTION (IMI)*. Retrieved June 15, 2008. from http://nawctsd.navair.navy.mil/Resources/Library/Acqguide/296123a.pdf.

Garris, R., Ahlers, R., & Driskell, J. E. (2002). Games, Motivation, and Learning: A Research and Practice Model. [Journal Article]. *Simulation and Gaming, 33*, 441-467

Glaser, R., Lesgold, A. & Gott, S. (1991). Implications of cognitive psychology for measuring job performance. In K. Wigdor & B. Green (Eds.), Performance Assessment for the Workplace (Vol. II.). Washington, DC: National Academy Press.