Learner Assessment Data Models for standardizing assessment across Live, Virtual and Constructive Domains

Brent Smith
ECS
Orlando, FL
brents@ecsorl.com

Chris DuBuc
ECS
Orlando, FL
chrisd@ecsorl.com

Dean Marvin
Joint ADL Co-Lab
Orlando, FL
dean.marvin2@us.army.mil

ABSTRACT

Feedback and guidance is an essential component of any learning environment so that errors in performance can be pointed out and corrected. A crucial part of this process is the ability to identify gaps in skills or knowledge and assess the underlying causes. Currently, there is little consistency in how a student is evaluated across the different training environments they will encounter in their career, whether in the school, in the field, at home or in simulated training exercises.

The Joint ADL Co-Laboratory, in cooperation with the US Navy, the US Army Research, Development and Engineering Command's Simulation Training Technology Center, and the US Army's Program Executive Office for Simulation, Training and Instrumentation, is funding the development of the Learner Assessment Data Model and Authoring Tools (LADMAT) project. This project is focused on development of an assessment data model and associated authoring tools that are capable of capturing complex assessment data across multiple learning and training systems. Specifically, the project will integrate assessment capabilities into a live simulation through the One Tactical Engagement Simulation System and with multiple virtual simulations using the Gamebryo game engine and the Delta3D game engine to ensure that training objectives are being met by the learner.

One key component of this research effort is to test and evaluate this technology as it is integrated into these dynamic learning environments. This paper will discuss in detail the complexities of assessing performance, the underlying technologies used in this project to simplify the assessment process and how they can be used to help standardize the manner in which a student is assessed through their career. It will also identify and discuss supporting technologies, standards, specifications, foreseeable challenges, best practices and lessons learned through the development and implementation of this project.

ABOUT THE AUTHORS

Mr. Brent Smith is the Vice President and Chief Technology Officer for Engineering & Computer Simulations, Inc. While at ECS, he has performed extensive research in the areas of collaborative distributed learning architectures, distributed simulations, and the use of commercial gaming technologies as educational tools for the US military. His professional focus is on the research and development of technologies to enable new learning methods and tools to improve training effectiveness. Mr. Smith is a frequent speaker on the topic of advanced learning technologies and has spoken at many conferences even testifying before Congress. He has also published numerous articles and papers on the topic as well. He is a member of the Association of the US Army (AUSA), MORS, the National Defense Industry Association (NDIA), and the Interservice / Industry Training, Simulation and Educations Conference (I/ITSEC) Subcommittee for Education. Currently, he is also secretary and technical writer for a collaborative effort between IEEE Learning Technologies Standards Committee and the Simulation Interoperability Standards Organization which was formed to identify potential standards for the key interface points between simulations and SCORM managed learning environments.

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, Chris 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. Chris was also V.P of Florida operations for Sage Software, where he oversaw

development of a wide rand of web-based software solutions for the telecommunications and mining/exploration industries. Chris has a BS in Management systems with a minor in Industrial Psychology from Rensselaer Polytechnic Institute (RPI).

Mr. 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.

Learner Assessment Data Models for standardizing assessment across Live, Virtual and Constructive Domains

Brent Smith
ECS
Orlando, FL
brents@ecsorl.com

Chris Dubuc
ECS
Orlando, FL
chrisd@ecsorl.com

Dean Marvin
Joint ADL Co-Lab
Orlando, FL
dean.marvin2@us.army.mil

INTRODUCTION

Training and education is a comprehensive process that does not simply consist of the transmission of content. Next generation instructional technologies will have the ability to track and assess individuals and teams at varying levels of proficiency while providing feedback and guidance so that errors in performance can be pointed out and corrected. The challenge doesn't end there, training developers employ many different forms of training for a diverse group of learners to produce personnel that can plan, move, and communicate through a wide range of situations.

Currently, there is little consistency in how students are evaluated across the different training environments they will encounter in their career, whether in the classroom, in the field, at home or in simulated training exercises. In the past, there has not been a consistent approach to how different learning environments manage assessment. The problem is that assessment strategies and associated development tools have been non-standard and closely coupled to the learning environment that they support. In other words, an assessment system developed for one learning domain can rarely be used in other domains.

The Joint Advanced Distributed Learning Co-Laboratory, in cooperation with the U.S. Navy, is investigating these issues and is investing in new technologies to help revolutionize the exchange of information between disparate learning systems and content. Their prototype program supports training transformation by overseeing the development and delivery of R&D efforts across the Services with the goal of advancing technologies to provide enhanced learning experiences to our Service members.

The Learner Assessment Data Model and Authoring Tools (LADMAT) program is one of the 2007-2008 prototypes currently being funded. A primary objective of the LADMAT program is to develop prototype tools, technologies and data models that are capable of capturing learner assessment data from multiple training environments in order to track learner progress towards a defined set of objectives.

CURRENT STATE OF THE ART

A competency-based approach to assessment is considered by many as the bridge between traditional measures of student achievement and the future[1]. However, there are numerous challenges associated with developing and assessing competency-based learning initiatives. The definition of a competency for the purpose of this paper is the combination of skills, abilities and knowledge needed to perform a specific task. Figure 1 below shows that for each individual student, skills and knowledge are acquired through learning experiences. Different combinations of skills and knowledge define the competencies that an individual may possess. Finally, different combinations of competencies are required to carry out different sets of tasks. Note that with this approach, similar competencies within different contexts may require different bundles of skills and knowledge.

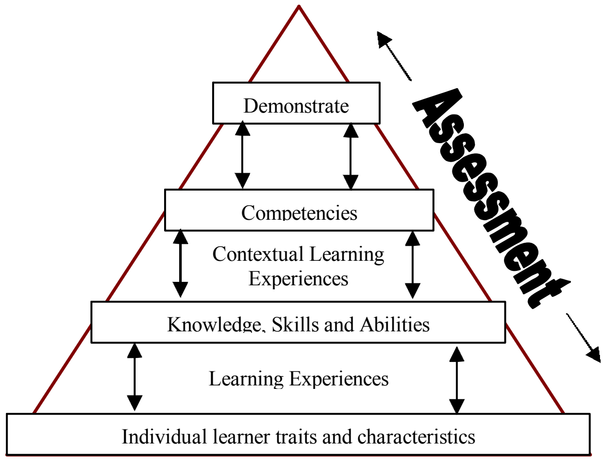

Demonstrate
Competencies
Contextual Learning Experiences
Knowledge, Skills and Abilities
Learning Experiences
Individual learner traits and characteristics
Assessment

Figure 1 - Source: Department of Education

Current technology makes it possible to analyze the sequences of actions learners take as they work through a problem and compare these sequences of actions against models of knowledge and performance associated with different levels of expertise. Most assessments, however, provide snapshots of achievement at ticker points in time, but do not capture the progression of a learners' conceptual understanding[2].

Existing assessment mechanisms can be loosely divided into two camps based on the type of learning environment within which they are used. *Static* assessment is generally associated with teaching methods where the content for a given unit of learning is the same; whether the content is comprised of lectures, text, pictures or video, each student will experience the same material in more-or-less the same order. In these environments, assessment generally comes in the form of a written test with questions that the student answers in a linear fashion. The benefit of this form of learning assessment is that it can be highly structured and easily integrated into an overall curriculum.

In contrast, a *dynamic* learning environment generally refers to one in which the student interacts with the training material on some level, such as in games, simulations or a live training exercise. In a dynamic learning environment, the training experience may be different each time the learner is engaged with it. The means of assessment in dynamic learning environments is varied. Learning pathways can no longer be defined in terms of highly structured, linear patterns and timeframes. Rather, assessment must accommodate transitions between learning experiences and ultimately between further training within other learning environments.

In order to overcome these limitations, training developers typically develop customized methods for assessing performance. In a game or simulation, there may be some internal scoring mechanism, while in a live training situation assessment may take the form of a subjective review by an observer/controller or an after action review. In these dynamic learning environments, if a formal scoring/assessment mechanism exists at all, it is most likely tied specifically to a particular training system, and therefore is not designed to be used by alternate or external systems. Additionally, the tasks/objectives defined in the assessment system will most likely not map directly to the overall learning objectives of a larger training curriculum.

In recent years, there have been many research efforts that have looked at the issues of how to tie dynamic learning environments to other managed learning environments. The overwhelming majority of this research is focused on assessment within a single training system and uses an approach that tracks a learner's response to critical events within that system. These events are tracked within a training system for use as evidence that will be used to determine whether or not a given learning objective has been satisfied. However, there are several problems that occur when coupling the assessment logic to a particular training system.

The biggest problem is the lack of interoperability. Currently, there are no standard definitions or data formats for such concepts as "evidence," "assessment logic," and "results." Therefore, these components are often developed specific to each training system. This process of "reinventing the wheel" for each new training environment increases development times and discourages the development of mature, stable, reusable and extensible components. It also inhibits the development of tools and utilities that could make it easier for non-programmers to create and modify assessment models.

OVERVIEW OF THE LADMAT PROGRAM

The main goal of the Learner Assessment Data Model and Authoring Tools (LADMAT) project is to provide a generic, adaptable, and standardized mechanism for learner assessment[3]. The term "learner assessment," in the context of this project, refers to the ability of a system to track a learner's progress through a training environment and generate a set of performance metrics. These metrics (i.e. scores, grades), can then be read in by other systems (i.e. an LMS or AAR system) to determine if the learner has satisfactorily met the defined learning objectives. Furthermore, these metrics can then be used by other systems to provide structured feedback and/or appropriate remediation to the student.

In order to meet these requirements, technology needs to be developed that allows in-depth assessments of student performance against a defined set of training objectives in order to identify student deficiencies and provide feedback to both students and instructors. The challenge therefore is a series of questions:

1. How do we create a set of data models and protocols that can formally represent the complex interactions between the events raised within a given learning environment and the training objectives, task conditions, and standards defined within a given training management system (i.e., an LMS)? These models must be able to account for the different relationships between the learner, other learners, and the environment where the training takes place.
2. How can we provide standardized tools to create and/or modify assessment models, so that trainers and instructors can design the way a student is evaluated inside each learning system (without having to write code)?

3. Ho do we develop an assessment architecture that uses these models to track and assess student performance across multiple training environments? Data warehousing structures also need to be defined to ensure that all of the activities that take place during a training exercise are logged and available for review. From this data, patterns of routine cognitive activities can be discerned.

In order to overcome the problems arising from existing assessment architectures, the LADMAT project introduces a new architecture wherein the assessment functionality is logically separated from the learning environment and the training management system.

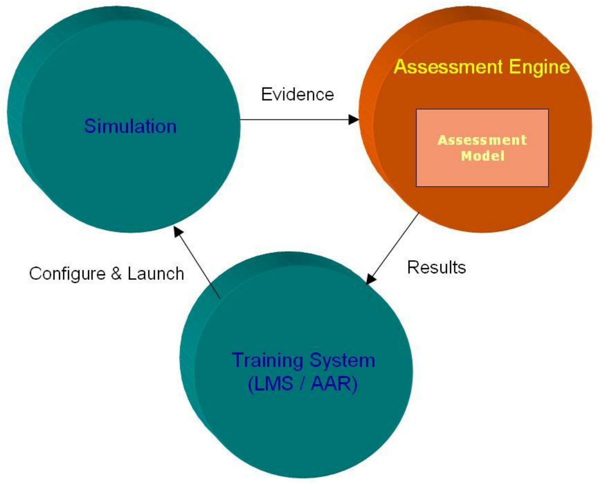


Figure 2 – LADMAT assessment engine

As shown above in figure 2, a new software component called an "assessment engine" is tasked with listening to event data (evidence) coming from the learning environment. The assessment engine processes this data according to a defined assessment model, and broadcasts the results (i.e., status, scores, grades) to a training system (i.e., LMS or AAR system). The assessment engine acts as middleware between the learning environment and the training system, and as such frees them from having to handle any assessment logic internally. This architecture also allows the assessment functionality to become more generic by allowing it to be used by any number of learning environments and/or training systems.

In order to extract the assessment functionality into its own process, yet still work within a variety of system architectures, there needs to be a standardized set of data formats and messaging protocols to enable communication between the learning environment and the assessment engine, and the assessment engine and the training system (i.e., evidence and results). These protocols serve as a "contract" between the three processes, allowing each component to run independently. This also makes it easier for an instructional component to be replaced by an entirely new instructional component if desired.

ARCHITECTURAL CONSIDERATIONS

As part of the Joint ADL prototype program, the LADMAT architecture is being developed and tested with two different virtual simulations using two different game engines. Additionally, this architecture is being tested with the One Tactical Engagement Simulation System (OneTESS), which is a live simulation being developed through the U.S. Army's Program Executive Office for Simulation, Training and Instrumentation (PEO STRI).

For each training environment, the assessment process is broken into logical components. The first step in the assessment development process is to specify the constructs of each training objective that needs to be measured within each of these dynamic learning environments. Once performance measures are determined, they can be mapped to "*events*" within each environment. Events are raised as the user interacts with the learning environment. The events raised by each learning environment are designed to be relevant markers of something important happening during the training session.

In the case of a simulation, these events may be triggered by simple user actions, such as "student pushed button." Alternatively, the events may be defined on a higher level and encapsulate more complex concepts that capture the interaction between objects and other system components such as "time" and "state" variables within the learning environment. The detail and complexity of what constitutes an event should be determined by instructional designers and simulation developers according to instructional needs and simulation architecture. A method of describing these events therefore needs to be developed so that tasks and objectives can be defined[4].

The assessment engine listens to these events, and uses them to track a learner's progress toward completion of defined tasks. An **Assessment Model** is used to combine the tasks in a hierarchal manner to form more complex tasks, which ultimately culminate in one or more objectives. As a learner completes the training objectives, the assessment engine communicates this information to the training system.

The preceding design goals dictate a tiered development approach. The tiers can be conceptually divided into the Learning Environment Tier, the Data Tier, and the Assessment Tier.

Learning Environment Tier

The **Learning Environment Tier** is the system and/or process where the actual training takes place. In the case of simulations, this tier will represent the student interacting with the simulation; in the case of a live training event; this tier will represent aspects of the physical training environment, including the students, equipment, and observer/controllers. At this level, this tier is responsible for generating simple messages related to relevant events and system state. For simulations, these messages are generated by the simulation code, but may also be generated by instrumented devices and equipment used by observers. The messages notify the assessment engine when an assessment object changes state or raises an event. They can either be sent directly to the assessment engine in real-time or be stored in an intermediary data store. Once in the assessment tier, the messages will be validated against the Data Tier.

Data Tier

The **Data Tier** is defined as the data model and schemas that describe the assessment objects, events, tasks/objectives, and logic for a given learning environment. Each training objective will be defined based on a known list of messages (events and state data) that are published by the learning environment. The assessment model will define which events it is dependant on and any rules concerning the sequencing or timing of those events. It will also define progress measurement criteria, such as complete/not-complete, percent complete, score, and so on.

Assessment Tier

The **Assessment Tier** is defined as the layer which “listens” to training events and processes them according to rules defined in the Data tier. This architecture is a logical model and does not mandate where this processing will take place or how this information is communicated to other training systems, thus it may be deployed in a number of configurations depending on the needs of the overall system. By developing this component as a separate code library, it may be implemented as a client application, a server application, a web service or even within the same physical process as a simulation. However, it is still logically separated from the other components and can be updated and maintained separately.

This component of the LADMAT project also consists of formalizing the constructs of an event publishing system that listens for any events that are associated with training objectives within the learning environment. These events are processed through the assessment model in order to track the state of each objective. The published event listings and assessment models for each training objective are tagged with versioning information and other meta-data for potential reuse.

The guiding principle here is that while a training system may track many assessment variables internally, it needs to be able to combine these variables into data values that other systems are able to understand. For example, SCORM 2004 provides standard specifications for assessment results and rollup functions to aggregate assessment results. In order for a SCORM conformant LMS to track student performance during a dynamic learning experience such as a simulation, the data communicated to the LMS must be confined to the data model that SCORM provides.

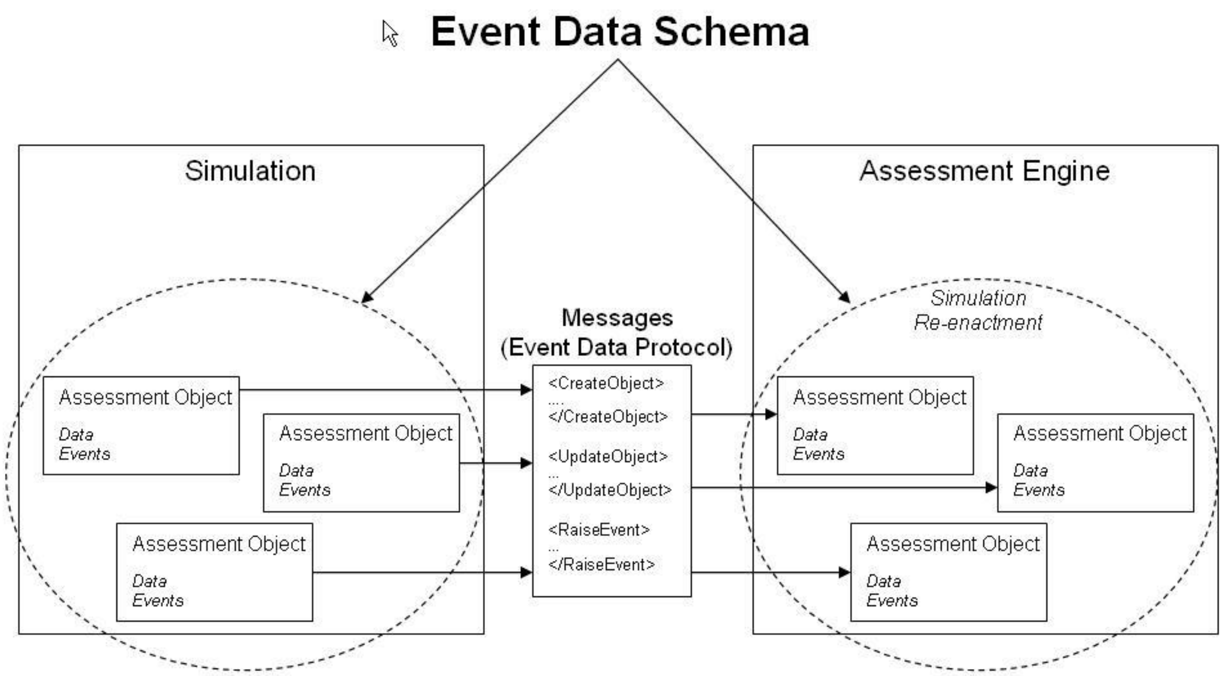
However, as the Assessment Engine processes events from the learning environment, there is potential to collect other contextual information that can be used as evidence to support competency[5]. From an instructional effectiveness point of view, the capture of this information could greatly increase the effectiveness of dynamic learning environments by allowing a more in-depth review of what a student has and hasn’t mastered across multiple learning experiences.

DATA MODELS AND PROTOCOLS

In order to enable the separation of the learning environment, the Assessment Engine, and training systems, a number of data models (schemas) and communication protocols need to be defined. These schemas and protocols help ensure that the various components can communicate with each other either directly or through an intermediary data store (such as a file or database).

Event Data Schema

The Event Data Schema represents the “contract” between a given learning environment and the assessment model (see Figure 3). Specifically, it defines the list of data structures and events that a given learning environment will be using at run-time. The architecture specifies that the environment will need to broadcast messages by creating and


Event Data Schema
Simulation
Assessment Engine
Messages (Event Data Protocol)
<CreateObject> .... </CreateObject>
<UpdateObject> ... </UpdateObject>
<RaiseEvent> ... </RaiseEvent>
Assessment Object
Data
Events
Simulation Re-enactment

maintaining *assessment objects*, which are data structures that encapsulate the state of the learning environment at a given point in time. In most cases, these assessment objects will act as proxies for actual objects in the environment, such as a player, team, vehicle, weapon, fire alarm, etc. However a learning environment developer is not bound by this methodology and can define assessment objects however it is deemed appropriate.

<u>Assessment Objects</u>

The Event Data Schema defines the structure of assessment objects and is compiled into code libraries for various languages by the Assessment Modeling Tool. It is these code libraries that are dynamically loaded into the learning environment and the Assessment Engine at run-time. The Event Data Schema will be different for any given environment, as each will contain its own unique data structures and raises its own unique events. However all Event Data Schemas are based upon a common logical model, allowing them to be manipulated and processed in a formalized manner. The Event Data Schema uses an *object-oriented* approach, meaning that it is made up of object definitions. Each object definition describes an "assessment object" that can be conceptually instantiated and updated by both the learning environment and the Assessment Engine. In this way the Assessment Engine will essentially be able to "re-enact" a given training experience.

<u>Event Data Protocol</u>

The Event Data Protocol describes the logical format of the *messages* that are created and sent by the learning environment to the Assessment Engine. These messages are used to log the important events that occur during any dynamic learning experience, and can be stored so that this experience can be "re-enacted" at any point in the future for assessment purposes. The Event Data Protocol is related to the Event Data Schema, in that the assessment object types defined in the schema represent the valid "source material" for the messages in the protocol.

The Event Data Protocol in the current LADMAT project uses a custom XML format for messaging purposes. This however is not mandatory, as the Event Data Protocol described herein is meant to be a logical format, with the actual physical message format being created and extracted by specialized translator components discussed later in this paper. Depending on the volume of data, different environments may work optimally with different messaging formats (i.e. SOAP messages, High Level Architecture (HLA), binary formats, etc.).

The Event Data Protocol is essentially the specification for event messages that when put into a list describe the significant events of a learning experience. There are six general types of event messages:

- *SimulationStart*
- *ObjectCreated*
- *ObjectUpdated*
- *ObjectDisposed*
- *CustomEvent*
- *SimulationStop*

According to the architecture of the Event Data Schema, all events originate from assessment objects. Therefore, every event message in the Event Data Protocol will contain a reference to the type and ID of the object that raised the event. The general requirements of the LADMAT project specify that event data (evidence) may be sent directly from the learning environment to the Assessment Engine in real-time. Alternatively, the event data may also be sent from the environment to an intermediary data store (i.e. file or database), to be read by the Assessment Engine at a future point in time. Accordingly, the event messages as defined by the Event Data Protocol will stand alone as individual chunks of data that can be sent over the wire as the events occur in real-time, or assembled into a single log file to be post-processed by the Assessment Engine.

Assessment Model and Schema

It is important to make a distinction between the terms "Assessment Model Schema" and "assessment model." The Assessment Model Schema is the conceptual structure and data format used to create assessment models. An assessment model is a concrete implementation of this schema created for a specific learning environment.

The Assessment Model Schema describes how a given assessment model can encapsulate the logic and performance measurements that are used to assess the actions of a learner during a given learning experience. This schema defines the format for outlining a task hierarchy made up of tasks, sub-tasks, task conditions, and other related information within an assessment model. It also provides a framework for the mechanisms and logical constructs that are used within assessment models to evaluate the evidence of a dynamic learning experience (the Event Data Protocol messages), and roll-up that information into a set of assessment results.

An assessment model is a logical structure that can take on different physical forms depending upon the context in which it is being used. Because assessment models are defined outside of the learning environment, they can be created and modified by course designers without having to change and re-compile any code. While being created and/or updated, an assessment model persists in the form of an XML file. It is this format that the Assessment Modeling Tool uses when saving and opening assessment model projects. At run-time, however, the assessment model takes the form of a compiled code library (i.e., a DLL or Java class/jar file). These code libraries are created by the Assessment Modeler Tool for use within different programming languages depending on what language was used in the development of the learning environment. An assessment model is comprised of:

- **Tasks** - The primary construct of the Assessment Model Schema is a "Task." A Task defines a particular unit of accomplishment within a simulation. Tasks are the basic building blocks of an assessment model, and information pertaining to tasks is what is compiled in order to produce the assessment results for a learner during a given learning experience. Top-level tasks can be mapped to specific learning objectives within an LMS or AAR system, and therefore provide the interface between the world of dynamic learning environments and the world of managed learning environments.
- **Task Conditions -** Structurally, tasks are made up of a list of "task conditions." Task conditions are the "rules" that determine whether a task has been completed and, optionally, how a task is scored. There are various types of task conditions and a task can contain any number of them. Additionally, task condition types can be used side-by-side within the same task. Task conditions interact with their parent tasks through an event-based mechanism, in that as the status of a task condition changes it raises events that are heard and evaluated by the parent task. Also, the task condition architecture is designed such that designers and/or developers can create new types of task conditions.

Tasks also implement an interface that determines when they have been "activated" and "deactivated." This is so that assessment model logic can track when a task has been started and stopped. A task's conditions will only be evaluated while the task is active. In the current implementation of the Assessment Model Schema, this is accomplished by assigning each task an "activate event" and a "deactivate event." In the current LADMAT project, the task activation interface limits each task to being activated only once. However, the activation interface has been designed so that the ability to activate/deactivate tasks repeatedly during the course of a single training attempt may be implemented in the future.

Results Data Protocol

The Results Data Protocol describes the logical format of the messages that are created and sent by the

Assessment Engine to other systems (such as an LMS). These messages are used to log the assessment results that are generated by an assessment model for a given learning experience, so that they can be uploaded into a managed learning environment. The messages in the Results Data Protocol are in XML format, and can be sent directly to an Assessment Results Component in real-time, or they can be stored in a file or database (data store) for upload into a managed learning environment at a later date.

There are two basic kinds of messages described by the Results Data Protocol, *TaskUpdate* messages and *Summary* messages. *TaskUpdate* messages are sent by the Assessment Engine each time an assessment task's status and/or score changes. This allows a reader of these messages to track when and in what order the various tasks were completed (or failed). *Summary* messages are sent by the Assessment Engine only once and simply list each task defined in the assessment model to provide its final status and score. If a reader is only interested in the final results, then they can use the *Summary* messages to extract the needed information.

SOFTWARE COMPONENTS

The LADMAT project also entails the development of various software components that are used to stream training event data to the assessment engine, evaluate that data according a set of rules, and output a stream of performance data. These include:

"BaseAssessmentTypes" Component

The "BaseAssessmentTypes" component is a code library that defines the various base classes used by other software components. The purpose of this library is to provide a common set of interfaces throughout the assessment system. Therefore, it is referenced by all other software components, including the Assessment Modeling Tool, Simulation Assessment Component, Assessment Engine, and Message Translators.

Assessment Modeling Tool

The Assessment Modeling Tool is the software responsible for creating Event Data Schemas for each specific learning environment and for creating Assessment Models based on those schemas. The Assessment Modeling Tool currently consists of a command-line executable and a software component, written in C#, that act as a "code generator," creating and compiling code libraries that encapsulate assessment objects (Event Data Schemas) and assessment logic (Assessment Models). The code libraries that the tool creates can be compiled for different languages (C#, Java, C++) so that they can be used on different platforms. For example, a C++ data schema could be compiled and integrated into a simulation in order to output event messages to a Java servlet-based assessment engine that uses an assessment model compiled in Java. Phase II of the LADMAT Project will focus on the development of GUI based tools to perform this functionality

The Learning Environment's Assessment Component

The purpose of the learning environment's assessment component is to provide a simple mechanism for each simulation to output event messages in the proper format for communication to the Assessment Engine. It is up to the developer to integrate this component into each learning environment. However the developer is not required to use the components and/or mechanisms described here and may provide their own components/mechanisms for creating and broadcasting these messages.

The learning environment's assessment component is compiled into a dynamic library written in C++ that can be loaded into the simulation process. C++ was selected because this is the language used in development of the learning environments being integrated into the LADMAT project. The assessment component contains an API that exposes a few simple methods that simulations can call to initialize the component and to create and send event messages.

Event Message Translators

The purpose of Event Message Translator libraries is to translate information used internally by the learning environment's Assessment Component and the Assessment Engine to send and receive event messages into a specific physical format.

The reason for this architecture is to allow additional message translators to be built that can handle many different physical formats. In the LADMAT project, an XML Message Translator was built to send and receive event messages in an XML format, according to the Event Data Protocol. However, XML is a verbose format that may prove inefficient for learning environments that output a copious amount of event messages. Therefore, the architecture was designed so that other message translators can be built that work with alternative message formats, such as HLA.

As mentioned previously, an event message translator library is loaded dynamically into the learning environment and the Assessment Engine. If used in the learning environment, the translator will send messages; alternatively, if used in the assessment engine, the translator will retrieve messages. The methods provided by the translator library will generally provide the capability to send messages both to and from a file and to and from a TCP/IP socket.

In general, a developer of a specific learning environment and/or assessment engine client will not call these methods directly. Instead, the methods are the required API for the library to be used as a "plug-in" for the learning environment's assessment component and/or the assessment engine component. If the simulation and assessment engine are running on different platforms (i.e., C++ and Java), an implementation of the translator will need to provided for both platforms.

The Assessment Engine

The Assessment Engine is the software component that captures event messages from a learning experience, analyzes the information per a given assessment model, and writes out the assessment results. Currently, the Assessment Engine takes the form of a code library (C++/C# .dll or Java .class file) so that it can be integrated into different learning environments in various ways (desktop application, Java applet, web service, etc.). It should be noted that in a strict sense the Assessment Engine component does not perform any analysis itself; rather it acts as a "broker" that manages the interaction between the event messages, the message translator, an assessment model, and a results processor. The LADMAT project is currently supporting multiple results processors to include:

- **Standard Results Processor** will be used to output the assessment results according to the format of the aforementioned Results Data Protocol. Any client application or software component that is designed to read in messages in this format may use this to gain access to the assessment results. This processor is also able to send the result messages directly to another software component in real-time via a TCP/IP socket connection, or alternatively it will be able to save the messages to a file to be read in by another component/application at a later date.
- **SCORM Results Processor -** In order to track a simulation within an LMS, the tasks defined in the assessment model need to be mapped to specific SCORM 'Objectives' defined in a Shareable Content Object (SCO). This processor provides a method that the client application will call once for each Objective. When the method is called, it will check to see if the Task has been mapped to a SCORM objective; if so, the processor will translate the Task's status and score into valid SCORM values and create an XML message to be sent to a client component.
- **Competency Results Processor -** The Reusable Definition of Competency or Educational Objective (RDCEO) specification, as defined by the IMS Global Learning Consortium, defines an information model for describing, referencing, and exchanging definitions of competencies. According to this specification, a "competency" generally refers to skills, knowledge, tasks, and/or learning outcomes that are used in the context of distributed learning systems. These competencies are often broken down into *Tasks*, *Conditions*, and *Standards*. In this context, *Tasks* refer to the behavior exhibited by a student, *Conditions* describe the circumstances under which the student performed the task, and *Standards* describes to what level of proficiency the student performed the task. These constructs roughly correlate to the Tasks, task conditions, and Task Scores as defined by an assessment model. Since the Assessment Engine reports on when assessment object's change state, when tasks are completed, and when task scores are updated, it becomes feasible to create a results processor that maps these assessment events to the Tasks, Conditions, and Standards of a competency (as defined by the RDCEO).

STANDARDS AND SPECIFICATIONS

In the past, there has not been a consistent approach to how dynamic learning environments, such as serious games and simulations, are developed; none of the major courseware initiatives, such as re-use, interoperability and sharing have been addressed[6]. As this technology becomes more advanced and costs are reduced, we theorize that the training community will embrace games and simulations as another media type that may be used to convey instruction.

While many prototype data models, schemas and software components have been developed for this project, it is important to note that in the future, they will need to interoperate with a multitude of other learning technologies that have not been fully developed or implemented at a large scale. New technologies, such as people objects or learner profiles, will also play a large role in helping to track a learner's progress across a wide range of training solutions.

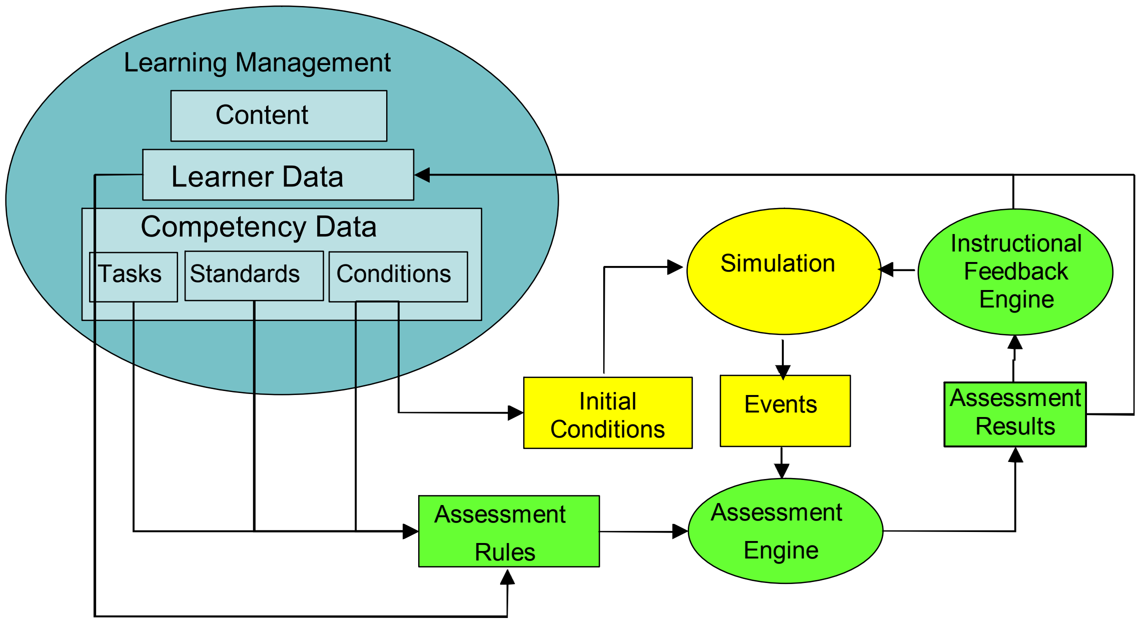


Figure 4: Overview of LMS and Simulation Interfaces

Recognizing the importance of these requirements, two IEEE standards committees have formed a collaborative study group to investigate the potential of formalizing a standard set of technical specifications to allow simulations and/or games to be launched and managed through SCORM-conformant content and Learning Management Systems. The LADMAT project has attempted to articulate the many best practices discussed in this study group[7]

Figure 4 shows key elements and interfaces that have emerged from the study group discussions. This diagram is color coded. Yellow is used to represent a dynamic learning environment that takes initial conditions and produces events during a learning experience.

In the SCORM context, the Learning Management System (shown as the blue oval) is responsible for managing three forms of data (shown in medium blue):

- Learner data, including learner identification and learner assessment records.
- Competency information, shown here in terms of tasks, conditions, and standards, but possibly including other forms, such as learning objectives.
- Content, the traditional SCOs, typically HTML-based data.

The assessment engine, in green, takes streams of events generated by the simulation and generates assessment results by comparing these events to rule-based models of assessment. While a simulation may track many assessment variables internally, it needs to be able to combine these variables into data values that an LMS is able to understand. This appears to be a common consideration across the study group and has been identified as an area that warrants further investigation for the development of potential standards. As shown in figure 4, there is a mechanism to listen to events or messages from a simulation, process them, translate them into data that an LMS can understand and communicate them to the SCORM data model either through the LMS or via web services.

The LADMAT project has helped highlight the area of emerging standards for dynamic learning environments such as serious games and simulations. Working groups such as the IEEE SCORM-Sim Interoperability study group are working to identify potential standards to more directly support this type of integration. This project will help further this effort.

Together, these standards, technologies and tools have the potential to make future training simulations even more accessible, adaptable, affordable, interoperable and reusable. This objective is consistent with Department of Defense's desires to make training more dynamic and engaging, more accessible, more deployable and less expensive. Although the required technologies exist and have been successfully demonstrated, the challenge is to formalize them together in an architecture that can be integrated and used across the spectrum of training strategies and mediums.

LESSONS LEARNED AND NEXT STEPS

Over the course of the development process, a number of lessons were learned:

- Most interactive training scenarios will require incredibly complicated assessment logic in order to be useful. The challenge is in creating a system that is capable of handling specific and intricate logic specific to one learning experience, yet flexible enough to be applied to any number of other contexts, scenarios and environments

- Complicated assessment logic increases the need for authoring capabilities that are simple to use and easy to understand. The challenge here is to provide a tool that is flexible enough to allow the creation of complicated assessment logic while still being simple enough to allow the designer to see the "big picture" through a "drag and drop" user interface
- Typically, a simulation developer will wish to minimize any additional complication introduced by assessment logic. Taking the assessment logic out of the learning environment greatly reduces the complexity required for the simulation developer. However, this component's interface needs to be flexible enough to handle any number and type of events and data values that need to be tracked. The challenge here is to find the correct balance between being non-intrusive to the simulation developers, yet providing flexibility to the instructional designers/training developers.

There are two funded projects that will extend and expand the focus of the LADMAT project. Both projects address the aforementioned challenges:

- The Joint ADL Co-Lab Prototype Program – Phase II - *Develop a more advanced GUI for simulation packaging and assessment modeling*. The focus of the phase I LADMAT project was on the data models, messaging protocols, and the underlying infrastructure to enable assessment in dynamic learning environments. Phase II will create authoring tools to build and package these models for use with various learning environments and managed training systems.
- NAVAIR is expanding the LADMAT architecture to enable the assessment results to be fed back into the simulation or systems, such as an After Action Review. This feedback could be used to modify initialization parameters for future simulation scenarios or sent back to trigger events in the current simulation in real-time (allowing the simulation to dynamically adapt to a student's progress).
- The NAVAIR project will also develop additional messaging protocols (HLA, OWL, binary) and/or translators. As development continues, however, the intention is to develop new formats and to create translators to existing formats that will allow the assessment components to communicate more efficiently within the environment they are deployed. Formats under specific consideration are XML Schema, OWL (Web Ontology Language), and HLA (High Level Architecture). Other formats may also be considered as research and testing continue.

REFERENCES

1. US Department of Education, National Postsecondary Education Cooperative, working group on competency-based initiatives in postsecondary education "Defining and Assessing Learning: Exploring Competency-Based Initiatives", 2001, publication 2002159,.
2. Federation of American Scientists, The Learning Federation Project "Component Roadmap: Learner modeling and Assessment R&D for Technology-Enabled Learning Systems", 2003
3. "LADMAT Software Design Document", Joint ADL Co-Laboratory, 2007,
4. Conkey, Smith, DuBuc & Smith (2006), Integrating Simulations into Sharable Content Object Reference Model Learning Environments, Paper presented at the Interservice/Industry Training, Simulation and Education Conference, Orlando FL, 2006
5. Dargue, Smith & Franks (2007), Interfacing Simulations with Training Content, European Simulation Interoperability Workshop, Rome Italy, 2007
6. Learning Systems Architecture Laboratory, "The Technical Evolution of SCORM", Carnegie Mellon University
7. Dargue, Morse, Smith & Franks (2006), Interfacing Simulations with Training Content", NATO MSG Symposium on Transforming Training and Experimentation through Modelling and Simulation, Rome Italy, 2006